

SPEC-MAIN

| | |
|---------------|---------------------|
| Last changed: | 21.11.2025 13:58:11 |
| Version: | 3.0.0-rc.6 |
| Creator: | VDV ETS |

Table of Contents

| | | |
|--------|---|----|
| 1 | Purpose | 25 |
| 2 | Scope | 26 |
| 3 | Introduction | 26 |
| 3.1 | History and Transition | 27 |
| 3.2 | Specification Overview | 29 |
| 3.2.1 | Model Specification Overview | 29 |
| 3.2.2 | Model Specification Overview DE | 30 |
| 3.2.3 | Further Specification Overview | 33 |
| 3.2.4 | Further Specification Overview DE | 34 |
| 3.3 | System Overview | 36 |
| 4 | User Manual | 36 |
| 4.1 | Interaction of Basic Processes (Layer 1) | 36 |
| 4.2 | Basic Processes (Layer 2) | 38 |
| 4.3 | Workflows and Basic Processes (Layer 1 and 2) | 39 |
| 4.3.1 | Layer 1 - BPMN Choreography | 39 |
| 4.3.2 | Basic Process 1 | 39 |
| 4.3.3 | Basic Process 2 | 40 |
| 4.3.4 | Out-of-Scope Process | 40 |
| 4.3.5 | Process of another Specification | 40 |
| 4.3.6 | Check Failed | 40 |
| 4.3.7 | End Event | 40 |
| 4.3.8 | Start Event | 40 |
| 4.3.9 | Short Description for Layer 1 Diagram | 40 |
| 4.3.10 | Alternate Flow? | 41 |
| 4.3.11 | Check | 41 |
| 4.3.12 | Re-join | 41 |
| 4.3.13 | Layer 2 - BPMN Collaboration | 42 |
| 4.3.14 | Description: Layer 2 - BPMN Collaboration Example | 43 |
| 4.3.15 | Description: Layer 2 - BPMN Collaboration Example with Terminal | 44 |
| 4.3.16 | Customer | 44 |
| 4.3.17 | Initiating Participant | 44 |
| 4.3.18 | Initiating Participant with Terminal | 45 |
| 4.3.19 | Reactive Participant | 45 |
| 5 | Role Model | 46 |
| 5.1 | Role Model etiCORE | 46 |
| 5.1.1 | Role Model etiCORE | 46 |
| 5.1.2 | Role Model etiCORE Ordered Action Management | 47 |

| | | |
|--------|--|----|
| 5.1.3 | Customer | 48 |
| 5.1.4 | Customer Contract Partner | 50 |
| 5.1.5 | Service Operator | 53 |
| 5.1.6 | Product Owner | 54 |
| 5.1.7 | Scheme Manager | 57 |
| 5.2 | Transition ISO 24014 to etiCORE | 60 |
| 5.2.1 | Transition ISO 24014 to etiCORE | 60 |
| 5.3 | Role Model ISO 24014-1 | 61 |
| 5.3.1 | Role Model ISO 24014-1 | 61 |
| 5.3.2 | Application Owner | 62 |
| 5.3.3 | Product Owner | 62 |
| 5.3.4 | Product Retailer | 62 |
| 5.3.5 | Application Retailer | 63 |
| 5.3.6 | Service Operator | 63 |
| 5.3.7 | Customer Service | 63 |
| 5.3.8 | Customer | 63 |
| 5.3.9 | Passenger | 63 |
| 5.3.10 | Registrar | 63 |
| 5.3.11 | Security Manager | 63 |
| 5.3.12 | Collection and Forwarding | 64 |
| 6 | Deployment Variants | 64 |
| 6.1 | Variant 1 - ePayment | 64 |
| 6.2 | Variant 2 Electronic Tickets and Payment Methods | 64 |
| 6.2.1 | Variant 2a - Electronic Ticket with Subscription | 64 |
| 6.2.2 | Variant 2b-1 - Electronic Ticket and ABPM | 65 |
| 6.2.3 | Variant 2b-2 - Electronic Ticket with SVPM | 65 |
| 6.3 | Variant 3 - IN-OUT Systems | 65 |
| 6.4 | Variant for D-Ticket | 65 |
| 6.4.1 | Variant for D-Ticket full | 65 |
| 6.4.2 | Variant for D-Ticket as Electronic Ticket only | 65 |
| 6.4.3 | Variant for D-Ticket as Static Entitlement only | 65 |
| 6.5 | Variant for Action Execution | 65 |
| 7 | Component Functionality Bundles | 66 |
| 7.1 | Basic Functionality Bundles | 66 |
| 7.1.1 | Terminals | 66 |
| 7.1.2 | Back-Office Systems | 67 |
| 7.2 | Electronic Ticket | 68 |
| 7.2.1 | Electronic Ticket Bundle CCP-System | 68 |
| 7.2.2 | Electronic Ticket Bundle CCP-Terminal | 68 |
| 7.2.3 | Electronic Ticket Bundle SO-Terminal | 68 |
| 7.2.4 | Electronic Ticket Bundle SO-System | 69 |
| 7.2.5 | Electronic Ticket Bundle PO-System | 69 |

| | | |
|-------|--|----|
| 7.3 | Account-Based Payment | 69 |
| 7.3.1 | Account-Based Payment Bundle CCP-Terminal | 69 |
| 7.3.2 | Account-Based Payment Bundle CCP-System | 69 |
| 7.3.3 | Account-Based Payment Bundle SO-System | 69 |
| 7.3.4 | Account-Based Payment Bundle SO-Terminal | 69 |
| 7.3.5 | Account-Based Payment Bundle PO-System | 69 |
| 7.4 | Stored-Value Payment | 69 |
| 7.4.1 | Stored-Value Payment Bundle CCP-System | 69 |
| 7.4.2 | Stored-Value Payment Bundle CCP-Terminal | 69 |
| 7.4.3 | Stored-Value Payment Bundle SO-System | 70 |
| 7.4.4 | Stored-Value Payment Bundle SO-Terminal | 70 |
| 7.4.5 | Stored-Value Payment Bundle PO-System | 70 |
| 7.5 | Sale Electronic Ticket via Account-Based Payment | 70 |
| 7.5.1 | Sale Electronic Ticket via Account-Based Payment Bundle CCP-Terminal | 70 |
| 7.5.2 | Sale Electronic Ticket via Account-Based Payment Bundle CCP-System | 70 |
| 7.5.3 | Sale Electronic Ticket via Account-Based Payment Bundle SO-Terminal | 70 |
| 7.5.4 | Sale Electronic Ticket via Account-Based Payment Bundle SO-System | 70 |
| 7.5.5 | Sale Electronic Ticket via Account-Based Payment Bundle PO-System | 70 |
| 7.6 | Sale Electronic Ticket via Stored-Value Payment | 71 |
| 7.6.1 | Sale Electronic Ticket via Stored-Value Payment Bundle CCP-Terminal | 71 |
| 7.6.2 | Sale Electronic Ticket via Stored-Value Payment Bundle CCP-System | 71 |
| 7.6.3 | Sale Electronic Ticket via Stored-Value Payment Bundle SO-Terminal | 71 |
| 7.6.4 | Sale Electronic Ticket via Stored-Value Payment Bundle SO-System | 71 |
| 7.6.5 | Sale Electronic Ticket via Stored-Value Payment Bundle PO-System | 71 |
| 7.7 | IN-OUT-Systems | 71 |
| 7.7.1 | IN-OUT Bundle CCP-Terminal | 71 |
| 7.7.2 | IN-OUT Bundle CCP-System | 71 |
| 7.7.3 | IN-OUT Bundle SO-Terminal | 72 |
| 7.7.4 | IN-OUT Bundle SO-System | 72 |
| 7.7.5 | IN-OUT Bundle PO-System | 72 |
| 7.8 | Ordered Action Management | 72 |
| 7.8.1 | Ordered Action Management Bundle Executing CCP-Terminal | 72 |
| 7.8.2 | Ordered Action Management Bundle Executing-CCP-System | 72 |
| 7.8.3 | Ordered Action Management Bundle Ordering-CCP-System | 72 |
| 7.8.4 | Ordered Action Management Bundle SO-Terminal | 72 |
| 7.8.5 | Ordered Action Management Bundle SO-System | 72 |
| 7.8.6 | Ordered Action Management Bundle PO-System | 73 |
| 7.9 | Static Entitlements | 73 |
| 7.9.1 | Static Entitlements Bundle CCP-Terminal | 73 |
| 7.9.2 | Static Entitlements Bundle CCP-System | 73 |
| 7.9.3 | Static Entitlements Bundle SO-Terminal | 73 |
| 7.9.4 | Static Entitlements Bundle SO-System | 73 |
| 7.9.5 | Static Entitlements Bundle PO-System | 73 |

| | | |
|--------|---|-----|
| 7.10 | Miscellaneous | 73 |
| 7.10.1 | Miscellaneous Bundle CCP-Terminal | 73 |
| 7.10.2 | Miscellaneous Bundle CCP-System | 73 |
| 7.10.3 | Miscellaneous Bundle SO-Terminal | 73 |
| 7.10.4 | Miscellaneous Bundle SO-System | 73 |
| 7.10.5 | Miscellaneous Bundle PO-System | 74 |
| 8 | Entitlement Categories | 74 |
| 8.1 | Entitlement Structure | 74 |
| 8.2 | Electronic Ticket | 74 |
| 8.3 | Payment Method | 74 |
| 8.4 | Account-based Payment Method | 75 |
| 8.5 | Stored Value Payment Method | 75 |
| 8.6 | Interoperable Account-based Payment Method | 75 |
| 8.7 | Interoperable Stored Value Payment Method | 75 |
| 8.8 | Non-interoperable Account-based Payment Method | 75 |
| 8.9 | Non-interoperable Stored Value Payment Method | 75 |
| 9 | Layer 1 and 2 - Workflows and Basic Processes in BPMN | 76 |
| 9.1 | Layer 1 - Workflows as BPMN Choreography | 77 |
| 9.1.1 | Sale | 84 |
| 9.1.2 | Issue user medium with application | 84 |
| 9.1.3 | Sell electronic ticket on existing user medium | 86 |
| 9.1.4 | Sell static entitlement | 87 |
| 9.1.5 | Issue entitlement | 89 |
| 9.1.6 | Inspection | 90 |
| 9.1.7 | Inspection and related basic processes | 90 |
| 9.1.8 | Hotlisting and blocking | 92 |
| 9.1.9 | Hotlist and block application | 92 |
| 9.1.10 | Hotlist and block entitlement | 93 |
| 9.1.11 | Hotlist a SAM | 94 |
| 9.1.12 | Hotlist an organisation | 96 |
| 9.1.13 | Hotlist an authentication key | 98 |
| 9.1.14 | Update hotlist inventories with external verification | 99 |
| 9.1.15 | Revoke hotlisting | 101 |
| 9.1.16 | Application hotlisting revocation | 101 |
| 9.1.17 | Entitlement hotlisting revocation | 102 |
| 9.1.18 | SAM hotlisting removal | 104 |
| 9.1.19 | Organisation hotlisting removal | 105 |
| 9.1.20 | Authentication key hotlisting removal | 106 |
| 9.1.21 | Customer service | 108 |
| 9.1.22 | Defective user medium | 108 |
| 9.1.23 | Lost user medium | 109 |
| 9.1.24 | Unblock application | 110 |

| | | |
|--------|--|-----|
| 9.1.25 | Unblock entitlement | 111 |
| 9.1.26 | Exchange user medium with application | 112 |
| 9.1.27 | Process new information about customer and discounts | 113 |
| 9.1.28 | Change entitlement | 114 |
| 9.1.29 | Change static entitlement | 115 |
| 9.1.30 | Take back | 116 |
| 9.1.31 | Take back application | 116 |
| 9.1.32 | Reset application | 116 |
| 9.1.33 | Take back entitlement | 117 |
| 9.1.34 | Take back static entitlement | 119 |
| 9.1.35 | Ordered action management | 121 |
| 9.1.36 | Action list configuration | 121 |
| 9.1.37 | Order entitlement issuance | 121 |
| 9.1.38 | Order entitlement termination | 122 |
| 9.1.39 | Order entitlement blocking | 123 |
| 9.1.40 | Order entitlement unblocking | 124 |
| 9.1.41 | Order entitlement replacement | 124 |
| 9.1.42 | Cancel order | 125 |
| 9.1.43 | Handle obsolete order | 126 |
| 9.1.44 | CICO | 128 |
| 9.1.45 | Record entitlement within the CICO process | 128 |
| 9.1.46 | Change user tariff parameters | 129 |
| 9.1.47 | Validation | 131 |
| 9.1.48 | Validate electronic ticket | 131 |
| 9.1.49 | Monitoring and notification | 132 |
| 9.1.50 | Extended Logging | 132 |
| 9.1.51 | Monitoring and notification | 133 |
| 9.1.52 | Queuing and notification | 134 |
| 9.1.53 | Configuration | 136 |
| 9.1.54 | Certificates and organisations | 136 |
| 9.1.55 | Service availability in CRE | 138 |
| 9.1.56 | Hotlist configuration | 138 |
| 9.1.57 | Tariff modules | 140 |
| 9.1.58 | SAM | 141 |
| 9.1.59 | Individualise and configure SAM | 141 |
| 9.1.60 | Reset SAM | 142 |
| 9.1.61 | User Medium | 144 |
| 9.1.62 | Order individualised or configured user media | 144 |
| 9.2 | Layer 2 - Basic Processes as BPMN Collaboration | 146 |
| 9.2.1 | Notification process patterns | 146 |
| 9.2.2 | Application non-owned | 147 |
| 9.2.3 | Application owned | 149 |
| 9.2.4 | Entitlement non-owned | 150 |

| | | |
|--------|--|-----|
| 9.2.5 | Entitlement owned | 151 |
| 9.2.6 | Sale | 154 |
| 9.2.7 | Issue entitlement starting in back-office system | 154 |
| 9.2.8 | Issue entitlement starting in terminal | 155 |
| 9.2.9 | Issue static entitlement | 157 |
| 9.2.10 | Recharge non-owned stored-value payment method | 158 |
| 9.2.11 | Recharge owned stored-value payment method | 160 |
| 9.2.12 | Autoload non-owned stored-value payment method | 162 |
| 9.2.13 | Autoload owned stored-value payment method | 164 |
| 9.2.14 | Personalise application | 166 |
| 9.2.15 | Debit non-owned stored-value payment method | 167 |
| 9.2.16 | Debit owned stored-value payment method | 169 |
| 9.2.17 | Debit non-owned account-based payment method | 171 |
| 9.2.18 | Debit owned account-based payment method | 173 |
| 9.2.19 | Inspection | 176 |
| 9.2.20 | Inspect user medium with application | 176 |
| 9.2.21 | Inspect user medium without application | 178 |
| 9.2.22 | Hotlisting and blocking | 181 |
| 9.2.23 | Unblocking | 236 |
| 9.2.24 | Unblock application | 236 |
| 9.2.25 | Unblock entitlement | 237 |
| 9.2.26 | Customer service | 239 |
| 9.2.27 | Process new information about customer and discounts | 239 |
| 9.2.28 | Take back | 241 |
| 9.2.29 | Take back application | 241 |
| 9.2.30 | De-personalise application | 242 |
| 9.2.31 | Take back non-owned entitlement | 243 |
| 9.2.32 | Take back owned entitlement | 245 |
| 9.2.33 | Take back owned static entitlement | 247 |
| 9.2.34 | Credit non-owned stored-value payment method | 249 |
| 9.2.35 | Credit owned stored-value payment method | 251 |
| 9.2.36 | Credit non-owned account-based payment method | 252 |
| 9.2.37 | Credit owned account-based payment method | 254 |
| 9.2.38 | Reimburse owned stored-value payment method | 255 |
| 9.2.39 | Reimburse non-owned stored-value payment method | 257 |
| 9.2.40 | Handle defective or lost user medium | 259 |
| 9.2.41 | Get entitlements of a lost user medium | 259 |
| 9.2.42 | Handle defective user medium | 260 |
| 9.2.43 | Look up application instance ID | 261 |
| 9.2.44 | Ordered action management | 263 |
| 9.2.45 | Distribute action list retrieval configuration | 263 |
| 9.2.46 | Order entitlement issuance | 263 |
| 9.2.47 | Order entitlement termination | 264 |

| | | |
|--------|--|-----|
| 9.2.48 | Order entitlement blocking | 265 |
| 9.2.49 | Order entitlement unblocking | 267 |
| 9.2.50 | Order group | 268 |
| 9.2.51 | Update action list inventory | 269 |
| 9.2.52 | Execute ordered entitlement issuance | 271 |
| 9.2.53 | Execute ordered entitlement termination | 272 |
| 9.2.54 | Execute ordered entitlement blocking | 274 |
| 9.2.55 | Execute ordered entitlement unblocking | 276 |
| 9.2.56 | Cancel order | 278 |
| 9.2.57 | Handle obsolete order | 279 |
| 9.2.58 | CICO | 281 |
| 9.2.59 | Change user tariff parameters of a non-owned entitlement | 281 |
| 9.2.60 | Change user tariff parameters of an owned entitlement | 283 |
| 9.2.61 | Charge account-based payment method | 285 |
| 9.2.62 | Record entitlement within the check-in process | 286 |
| 9.2.63 | Record entitlement within the check-out process | 288 |
| 9.2.64 | Validation | 291 |
| 9.2.65 | Validate electronic ticket | 291 |
| 9.2.66 | Monitoring and notification | 293 |
| 9.2.67 | Discarded messages | 295 |
| 9.2.68 | Notify events | 296 |
| 9.2.69 | Configurations | 298 |
| 9.2.70 | Distribute tariff modules | 302 |
| 9.2.71 | Set a service as available for a participant | 302 |
| 9.2.72 | Set a service as unavailable for a participant | 303 |
| 9.2.73 | Retrieve organisation list | 305 |
| 9.2.74 | Retrieve the CA certificate repository | 306 |
| 9.2.75 | Retrieve the CV certificate revocation list | 307 |
| 9.2.76 | SAM | 309 |
| 9.2.77 | User Medium | 320 |
| 9.2.78 | Individualise UM | 320 |
| 9.2.79 | Configure UM | 322 |
| 9.3 | Supporting Choreography Models | 325 |
| 9.3.1 | Sale | 325 |
| 9.3.2 | Debit account-based payment method | 325 |
| 9.3.3 | Debit stored-value payment method | 326 |
| 9.3.4 | Autoload stored-value payment method | 326 |
| 9.3.5 | Recharge stored-value payment method | 327 |
| 9.3.6 | Issue entitlement | 328 |
| 9.3.7 | Hotlisting and blocking | 330 |
| 9.3.8 | Block hotlisted application | 330 |
| 9.3.9 | Block hotlisted entitlement | 330 |
| 9.3.10 | Hotlist entitlement | 331 |

| | | |
|---------|---------------------------------------|-----|
| 9.3.11 | Hotlist application | 332 |
| 9.3.12 | Hotlist SAM | 333 |
| 9.3.13 | Take back | 335 |
| 9.3.14 | Take back entitlement | 335 |
| 9.3.15 | Credit account-based payment method | 335 |
| 9.3.16 | Credit stored-value payment method | 336 |
| 9.3.17 | Reimburse stored-value payment method | 337 |
| 9.4 | Participants | 339 |
| 9.5 | Customer Contract Partner | 341 |
| 9.6 | Primary Customer Contract Partner | 341 |
| 9.7 | Secondary Customer Contract Partner | 341 |
| 9.8 | Ordering Customer Contract Partner | 341 |
| 9.9 | Executing Customer Contract Partner | 341 |
| 9.10 | Service Operator | 342 |
| 9.11 | Product Owner | 342 |
| 9.12 | Scheme Manager | 342 |
| 9.13 | Hotlist Service | 342 |
| 9.14 | SO, CCP or Scheme Manager | 342 |
| 9.15 | SO, CCP, PO or Hotlist service | 342 |
| 9.16 | SO, CCP, PO or Scheme Manager | 342 |
| 9.17 | PO, SO or CCP | 342 |
| 9.18 | PO or Scheme Manager | 342 |
| 9.19 | SO or CCP | 343 |
| 9.20 | SO or sCCP | 343 |
| 9.21 | SO, sCCP or PO | 343 |
| 9.22 | Customer | 343 |
| 9.23 | Card Manufacturer | 343 |
| 9.24 | Mass Personaliser | 343 |
| 9.25 | Central Routing Engine | 343 |
| 10 | Actors | 343 |
| 10.1 | Explanatory Actors | 344 |
| 10.1.1 | Back-Office Main Module | 345 |
| 10.1.2 | Customer Contract Partner System | 345 |
| 10.1.3 | Customer Contract Partner Terminal | 346 |
| 10.1.4 | Product Owner System | 346 |
| 10.1.5 | Scheme Manager System | 346 |
| 10.1.6 | Service Operator System | 346 |
| 10.1.7 | Service Operator Terminal | 346 |
| 10.1.8 | Terminal Main Module | 346 |
| 10.1.9 | Customer | 346 |
| 10.1.10 | Customer Contract Partner | 346 |
| 10.1.11 | Product Owner | 346 |
| 10.1.12 | SAM Owner | 346 |

| | | |
|---------|---|-----|
| 10.1.13 | Scheme Manager | 347 |
| 10.1.14 | User Medium | 347 |
| 10.1.15 | Service Operator | 347 |
| 10.2 | Customer Contract Partner | 348 |
| 10.2.1 | Customer Contract Partner Terminal Main Module | 348 |
| 10.2.2 | Customer Contract Partner Terminal Order Execution Module | 348 |
| 10.2.3 | Customer Contract Partner Terminal Static Entitlement Module | 349 |
| 10.2.4 | Customer Contract Partner Back-Office Main Module | 349 |
| 10.2.5 | Primary Customer Contract Partner Back-Office Main Module | 349 |
| 10.2.6 | Secondary Customer Contract Partner Back-Office Main Module | 349 |
| 10.2.7 | Customer Contract Partner Back-Office Action Ordering Module | 349 |
| 10.2.8 | Customer Contract Partner Back-Office Order Execution Module | 350 |
| 10.2.9 | Customer Contract Partner Back-Office Static Entitlement Module | 350 |
| 10.2.10 | Primary Customer Contract Partner Back-Office Static Entitlement Module | 350 |
| 10.3 | Product Owner | 351 |
| 10.3.1 | Product Owner Back-Office Main Module | 351 |
| 10.3.2 | Product Owner Back-Office Action Management Module | 352 |
| 10.3.3 | Product Owner Back-Office Static Entitlement Module | 352 |
| 10.4 | Service Operator | 353 |
| 10.4.1 | Service Operator Terminal Main Module | 353 |
| 10.4.2 | Service Operator Terminal Static Entitlement Module | 353 |
| 10.4.3 | Service Operator Back-Office Main Module | 353 |
| 10.4.4 | Service Operator Back-Office Static Entitlement Module | 353 |
| 10.5 | Scheme Manager | 354 |
| 10.5.1 | Hotlist Service System | 354 |
| 10.5.2 | (((eTicket Security Hub | 354 |
| 10.5.3 | Media Management System | 355 |
| 10.5.4 | Media Public Key Infrastructure | 355 |
| 10.6 | User Medium | 356 |
| 10.6.1 | User Medium with application | 356 |
| 10.6.2 | User Medium without application | 356 |
| 11 | List of Use Cases | 356 |
| 11.1 | Activate SAM | 357 |
| 11.2 | Add application to hotlist | 358 |
| 11.3 | Add authentication key to hotlist | 358 |
| 11.4 | Add entitlement to hotlist | 359 |
| 11.5 | Add organisation to hotlist | 360 |
| 11.6 | Add product acceptance entry to hotlist configuration | 361 |
| 11.7 | Add SAM to hotlist | 362 |
| 11.8 | Analyse application history from contractual perspective | 363 |
| 11.9 | Analyse application history from contractual perspective | 364 |
| 11.10 | Analyse entitlement history from contractual perspective | 364 |
| 11.11 | Analyse entitlement history from product perspective | 365 |

| | | |
|-------|---|-----|
| 11.12 | Analyse order history from contractual perspective | 366 |
| 11.13 | Analyse order history from product perspective | 367 |
| 11.14 | Authorise static entitlement | 368 |
| 11.15 | Autoload stored-value payment method | 369 |
| 11.16 | Block entitlement triggered by action order | 370 |
| 11.17 | Cancel order | 371 |
| 11.18 | Cancel order | 371 |
| 11.19 | Change customer and discounts | 372 |
| 11.20 | Change entitlement | 373 |
| 11.21 | Change favourites | 374 |
| 11.22 | Change password | 375 |
| 11.23 | Change static entitlement | 376 |
| 11.24 | Change user tariff parameters | 377 |
| 11.25 | Check and add SAM to hotlist | 378 |
| 11.26 | Check entitlement notifications against issuance notification from contractual perspective | 380 |
| 11.27 | Check entitlement notifications against issuance notification from product perspective | 380 |
| 11.28 | Check for order obsolescence | 381 |
| 11.29 | Check MOTICS requirements | 382 |
| 11.30 | Check static entitlement notifications against issuance notification from contractual perspective | 383 |
| 11.31 | Check static entitlement notifications against issuance notification from product perspective | 384 |
| 11.32 | Check static entitlement notifications for plausibility | 385 |
| 11.33 | Check user medium with application as CCP | 386 |
| 11.34 | Check user medium with application as CCP | 386 |
| 11.35 | Check user medium with application as service operator | 387 |
| 11.36 | Check user medium with application as service operator | 387 |
| 11.37 | Check user medium without application | 388 |
| 11.38 | Configure user medium application | 389 |
| 11.39 | Create extended logging for an application | 390 |
| 11.40 | Create extended logging for an entitlement | 391 |
| 11.41 | Credit account-based payment method | 392 |
| 11.42 | Credit account-based payment method | 392 |
| 11.43 | Credit stored-value payment method | 393 |
| 11.44 | Credit stored-value payment method | 393 |
| 11.45 | Debit account-based payment method | 394 |
| 11.46 | Debit account-based payment method | 394 |
| 11.47 | Debit stored-value payment method | 395 |
| 11.48 | Delete customer | 396 |
| 11.49 | Delete discounts | 397 |
| 11.50 | Delete entitlement | 398 |
| 11.51 | Delete favourites | 398 |

| | | |
|-------|--|-----|
| 11.52 | Demand account-based payment method charging | 399 |
| 11.53 | Demand application hotlisting | 400 |
| 11.54 | Demand application hotlisting | 400 |
| 11.55 | Demand entitlement hotlisting | 401 |
| 11.56 | Demand entitlement hotlisting | 402 |
| 11.57 | Demand SAM hotlisting | 403 |
| 11.58 | Demand SAM hotlisting | 403 |
| 11.59 | De-personalise application | 403 |
| 11.60 | Determine SAM owner | 404 |
| 11.61 | Determine UM app instance ID for Medium ID | 405 |
| 11.62 | Determine user medium owner | 406 |
| 11.63 | Determine valid entitlements for given app instance ID | 407 |
| 11.64 | Display application data | 408 |
| 11.65 | Display customer | 409 |
| 11.66 | Display customer | 409 |
| 11.67 | Display discounts | 410 |
| 11.68 | Display entitlement | 411 |
| 11.69 | Display entitlement | 411 |
| 11.70 | Display favourites | 411 |
| 11.71 | Display static entitlement | 412 |
| 11.72 | Distribute SAM configuration | 413 |
| 11.73 | Distribute tariff module | 414 |
| 11.74 | Distribute the SAM reset script | 415 |
| 11.75 | Establish session and get entitlement directory | 416 |
| 11.76 | Establish session based on certificates | 417 |
| 11.77 | Establish session based on certificates | 417 |
| 11.78 | Exchange user medium with application | 418 |
| 11.79 | Exchange user medium with application | 418 |
| 11.80 | Execute action list entries | 419 |
| 11.81 | Export derived key | 420 |
| 11.82 | Export derived key | 420 |
| 11.83 | Generate current action list | 421 |
| 11.84 | Generate current hotlists | 422 |
| 11.85 | Get entitlement and check attestations | 422 |
| 11.86 | Get product acceptance configuration list | 423 |
| 11.87 | Get unclaimed list information | 424 |
| 11.88 | Handle account-based payment method charging from contractual perspective | 425 |
| 11.89 | Handle account-based payment method charging from contractual perspective | 426 |
| 11.90 | Handle account-based payment method credited notification from contractual perspective | 426 |
| 11.91 | Handle account-based payment method credited notification from contractual perspective | 426 |
| 11.92 | Handle account-based payment method credited notification from operational perspective | 427 |

| | | |
|--------|---|-----|
| 11.93 | Handle account-based payment method credited notification from product perspective | 428 |
| 11.94 | Handle account-based payment method credited notification from product perspective | 428 |
| 11.95 | Handle account-based payment method debited notification from contractual perspective | 429 |
| 11.96 | Handle account-based payment method debited notification from operational perspective | 430 |
| 11.97 | Handle account-based payment method debited notification from operational perspective | 431 |
| 11.98 | Handle account-based payment method debited notification from product perspective | 432 |
| 11.99 | Handle application blocked notification from contractual perspective | 432 |
| 11.100 | Handle application blocked notification from operational perspective | 433 |
| 11.101 | Handle application blocked notification from operational perspective | 433 |
| 11.102 | Handle application hotlisting demand | 434 |
| 11.103 | Handle application hotlisting demand | 434 |
| 11.104 | Handle application terminated notification from contractual perspective | 436 |
| 11.105 | Handle application terminated notification from operational perspective | 436 |
| 11.106 | Handle application terminated notification from operational perspective | 436 |
| 11.107 | Handle application unblocked notification from contractual perspective | 437 |
| 11.108 | Handle application unblocked notification from contractual perspective | 437 |
| 11.109 | Handle application unblocked notification from operational perspective | 438 |
| 11.110 | Handle application unblocked notification from operational perspective | 438 |
| 11.111 | Handle application XY notification from contractual perspective | 439 |
| 11.112 | Handle application XY notification from operational perspective | 440 |
| 11.113 | Handle application XY notification from operational perspective | 440 |
| 11.114 | Handle authentication key hotlisting demand | 441 |
| 11.115 | Handle check-in notification from contractual perspective | 442 |
| 11.116 | Handle check-in notification from operational perspective | 443 |
| 11.117 | Handle check-in notification from product perspective | 444 |
| 11.118 | Handle check-out notification from contractual perspective | 445 |
| 11.119 | Handle check-out notification from operational perspective | 446 |
| 11.120 | Handle check-out notification from product perspective | 447 |
| 11.121 | Handle defective user medium with application | 448 |
| 11.122 | Handle discarded messages information | 449 |
| 11.123 | Handle entitlement blocked notification from contractual perspective | 450 |
| 11.124 | Handle entitlement blocked notification from contractual perspective | 450 |
| 11.125 | Handle entitlement blocked notification from operational perspective | 451 |
| 11.126 | Handle entitlement blocked notification from product perspective | 452 |
| 11.127 | Handle entitlement blocked notification from product perspective | 453 |
| 11.128 | Handle entitlement blocking order | 453 |
| 11.129 | Handle entitlement blocking order | 453 |
| 11.130 | Handle entitlement hotlisting demand | 454 |

| | | |
|--------|--|-----|
| 11.131 | Handle entitlement hotlisting demand | 454 |
| 11.132 | Handle entitlement inspected notification from contractual perspective | 456 |
| 11.133 | Handle entitlement inspected notification from contractual perspective | 456 |
| 11.134 | Handle entitlement inspected notification from operational perspective | 456 |
| 11.135 | Handle entitlement inspected notification from product perspective | 457 |
| 11.136 | Handle entitlement inspected notification from product perspective | 457 |
| 11.137 | Handle entitlement issuance order | 458 |
| 11.138 | Handle entitlement issuance order | 458 |
| 11.139 | Handle entitlement issued notification from contractual perspective | 459 |
| 11.140 | Handle entitlement issued notification from operational perspective | 460 |
| 11.141 | Handle entitlement issued notification from operational perspective | 460 |
| 11.142 | Handle entitlement issued notification from product perspective | 461 |
| 11.143 | Handle entitlement terminated notification from contractual perspective | 462 |
| 11.144 | Handle entitlement terminated notification from operational perspective | 463 |
| 11.145 | Handle entitlement terminated notification from operational perspective | 464 |
| 11.146 | Handle entitlement terminated notification from product perspective | 464 |
| 11.147 | Handle entitlement termination order | 465 |
| 11.148 | Handle entitlement termination order | 465 |
| 11.149 | Handle entitlement unblocked notification from contractual perspective | 466 |
| 11.150 | Handle entitlement unblocked notification from operational perspective | 467 |
| 11.151 | Handle entitlement unblocked notification from product perspective | 468 |
| 11.152 | Handle entitlement unblocked notification from product perspective | 468 |
| 11.153 | Handle entitlement unblocking order | 469 |
| 11.154 | Handle entitlement validated notification from contractual perspective | 470 |
| 11.155 | Handle entitlement validated notification from operational perspective | 471 |
| 11.156 | Handle entitlement validated notification from product perspective | 472 |
| 11.157 | Handle entitlement XY notification from contractual perspective | 473 |
| 11.158 | Handle entitlement XY notification from operational perspective | 474 |
| 11.159 | Handle entitlement XY notification from operational perspective | 474 |
| 11.160 | Handle entitlement XY notification from product perspective | 475 |
| 11.161 | Handle events notification | 476 |
| 11.162 | Handle LDAP search request | 477 |
| 11.163 | Handle lost user medium with application | 478 |
| 11.164 | Handle order cancellation | 479 |
| 11.165 | Handle order group | 480 |
| 11.166 | Handle order group | 480 |
| 11.167 | Handle order obsolescence notification | 481 |
| 11.168 | Handle ordered entitlement blocked notification from contractual perspective | 482 |
| 11.169 | Handle ordered entitlement blocked notification from operational perspective | 483 |
| 11.170 | Handle ordered entitlement blocked notification from product perspective | 484 |
| 11.171 | Handle ordered entitlement issued notification from contractual perspective | 485 |
| 11.172 | Handle ordered entitlement issued notification from contractual perspective | 485 |
| 11.173 | Handle ordered entitlement issued notification from operational perspective | 486 |

| | | |
|--------|--|-----|
| 11.174 | Handle ordered entitlement issued notification from operational perspective | 486 |
| 11.175 | Handle ordered entitlement issued notification from product perspective | 487 |
| 11.176 | Handle ordered entitlement terminated notification from contractual perspective | 488 |
| 11.177 | Handle ordered entitlement terminated notification from contractual perspective | 488 |
| 11.178 | Handle ordered entitlement terminated notification from operational perspective | 489 |
| 11.179 | Handle ordered entitlement terminated notification from product perspective | 490 |
| 11.180 | Handle ordered entitlement terminated notification from product perspective | 490 |
| 11.181 | Handle ordered entitlement unblocked notification from contractual perspective | 491 |
| 11.182 | Handle ordered entitlement unblocked notification from operational perspective | 492 |
| 11.183 | Handle ordered entitlement unblocked notification from operational perspective | 492 |
| 11.184 | Handle ordered entitlement unblocked notification from product perspective | 493 |
| 11.185 | Handle organisation hotlisting demand | 494 |
| 11.186 | Handle request for product acceptance configuration list | 495 |
| 11.187 | Handle request to add product acceptance entry to hotlist service configuration | 496 |
| 11.188 | Handle request to add product acceptance entry to hotlist service configuration | 496 |
| 11.189 | Handle request to determine SAM owner | 497 |
| 11.190 | Handle request to determine UM app instance ID for Medium ID | 498 |
| 11.191 | Handle request to remove product acceptance entry from hotlist service configuration | 499 |
| 11.192 | Handle request to remove product acceptance from participants | 500 |
| 11.193 | Handle retrieval request for action list | 501 |
| 11.194 | Handle retrieval request for action list | 501 |
| 11.195 | Handle retrieval request for application hotlist | 502 |
| 11.196 | Handle retrieval request for application hotlist | 502 |
| 11.197 | Handle retrieval request for authentication key hotlist | 503 |
| 11.198 | Handle retrieval request for entitlement hotlist | 504 |
| 11.199 | Handle retrieval request for entitlement hotlist | 504 |
| 11.200 | Handle retrieval request for entitlement hotlist with product information | 505 |
| 11.201 | Handle retrieval request for incremental action list | 506 |
| 11.202 | Handle retrieval request for incremental application hotlist | 507 |
| 11.203 | Handle retrieval request for incremental application hotlist | 507 |
| 11.204 | Handle retrieval request for incremental entitlement hotlist | 508 |
| 11.205 | Handle retrieval request for organisation hotlist | 509 |
| 11.206 | Handle retrieval request for organisation hotlist | 509 |
| 11.207 | Handle retrieval request for SAM hotlist | 510 |
| 11.208 | Handle retrieval request for unclaimed list information | 511 |
| 11.209 | Handle revocation for application hotlisting demand | 512 |
| 11.210 | Handle revocation for authentication key hotlisting demand | 513 |
| 11.211 | Handle revocation for entitlement hotlisting demand | 514 |
| 11.212 | Handle revocation for organisation hotlisting demand | 515 |
| 11.213 | Handle revocation for SAM hotlisting demand | 516 |

| | | |
|--------|---|-----|
| 11.214 | Handle SAM hotlisting demand | 517 |
| 11.215 | Handle static entitlement inspected notification from contractual perspective | 518 |
| 11.216 | Handle static entitlement inspected notification from operational perspective | 519 |
| 11.217 | Handle static entitlement inspected notification from product perspective | 520 |
| 11.218 | Handle static entitlement inspected notification from product perspective | 520 |
| 11.219 | Handle static entitlement issued notification from contractual perspective | 521 |
| 11.220 | Handle static entitlement issued notification from operational perspective | 522 |
| 11.221 | Handle static entitlement issued notification from operational perspective | 523 |
| 11.222 | Handle static entitlement issued notification from product perspective | 523 |
| 11.223 | Handle static entitlement terminated notification from contractual perspective | 524 |
| 11.224 | Handle static entitlement terminated notification from operational perspective | 525 |
| 11.225 | Handle static entitlement terminated notification from product perspective | 526 |
| 11.226 | Handle stored-value payment method credited notification from contractual perspective | 527 |
| 11.227 | Handle stored-value payment method credited notification from contractual perspective | 527 |
| 11.228 | Handle stored-value payment method credited notification from operational perspective | 528 |
| 11.229 | Handle stored-value payment method credited notification from product perspective | 529 |
| 11.230 | Handle stored-value payment method credited notification from product perspective | 530 |
| 11.231 | Handle stored-value payment method debited notification from contractual perspective | 530 |
| 11.232 | Handle stored-value payment method debited notification from operational perspective | 531 |
| 11.233 | Handle stored-value payment method debited notification from operational perspective | 531 |
| 11.234 | Handle stored-value payment method debited notification from product perspective | 532 |
| 11.235 | Handle stored-value payment method recharged notification from contractual perspective | 533 |
| 11.236 | Handle stored-value payment method recharged notification from operational perspective | 534 |
| 11.237 | Handle stored-value payment method recharged notification from operational perspective | 535 |
| 11.238 | Handle stored-value payment method recharged notification from product perspective | 535 |
| 11.239 | Handle stored-value payment method reimbursed notification from contractual perspective | 536 |
| 11.240 | Handle stored-value payment method reimbursed notification from contractual perspective | 536 |
| 11.241 | Handle stored-value payment method reimbursed notification from operational perspective | 537 |
| 11.242 | Handle stored-value payment method reimbursed notification from product perspective | 538 |

| | | |
|--------|---|-----|
| 11.243 | Handle stored-value payment method reimbursed notification from product perspective | 538 |
| 11.244 | Handle user tariff parameters changed notification from contractual perspective | 539 |
| 11.245 | Handle user tariff parameters changed notification from operational perspective | 540 |
| 11.246 | Handle user tariff parameters changed notification from product perspective | 541 |
| 11.247 | Handle verification request for action list updated via increments | 542 |
| 11.248 | Handle verification request for action list updated via increments | 542 |
| 11.249 | Handle verification request for application hotlist updated via increments | 543 |
| 11.250 | Handle verification request for entitlement hotlist updated via increments | 544 |
| 11.251 | Handle verification request for entitlement hotlist updated via increments | 544 |
| 11.252 | Initialise password | 545 |
| 11.253 | Initialise User Medium with application for customer | 546 |
| 11.254 | Inspect user medium with application | 547 |
| 11.255 | Inspect user medium with application | 547 |
| 11.256 | Inspect user medium without application | 548 |
| 11.257 | Issue entitlement | 549 |
| 11.258 | Issue entitlement triggered by action order | 550 |
| 11.259 | Issue entitlement triggered by action order | 550 |
| 11.260 | Issue static entitlement | 551 |
| 11.261 | Log defective user medium with application | 552 |
| 11.262 | Monitor SAMs from operational perspective | 554 |
| 11.263 | Monitor SAMs from product perspective | 554 |
| 11.264 | Monitor SAMs from product perspective | 554 |
| 11.265 | Notify events | 555 |
| 11.266 | Notify static entitlement inspected | 556 |
| 11.267 | Notify static entitlement inspected | 556 |
| 11.268 | Notify static entitlement terminated | 557 |
| 11.269 | Order entitlement blocking | 558 |
| 11.270 | Order entitlement blocking | 558 |
| 11.271 | Order entitlement issuance | 559 |
| 11.272 | Order entitlement issuance | 559 |
| 11.273 | Order entitlement termination | 560 |
| 11.274 | Order entitlement unblocking | 561 |
| 11.275 | Order entitlement unblocking | 561 |
| 11.276 | Order group | 562 |
| 11.277 | Perform account-based payment method crediting and notify | 563 |
| 11.278 | Perform account-based payment method debiting and notify | 564 |
| 11.279 | Perform account-based payment method debiting and notify | 564 |
| 11.280 | Perform application blocking and notify | 565 |
| 11.281 | Perform application termination and notify | 566 |
| 11.282 | Perform application unblocking and notify | 567 |
| 11.283 | Perform application unblocking and notify | 567 |
| 11.284 | Perform application XY and notify | 568 |

| | | |
|--------|--|-----|
| 11.285 | Perform check in and notify | 569 |
| 11.286 | Perform check out and notify | 570 |
| 11.287 | Perform check out and notify | 570 |
| 11.288 | Perform entitlement blocking and notify | 571 |
| 11.289 | Perform entitlement blocking and notify | 571 |
| 11.290 | Perform entitlement inspection and notify | 572 |
| 11.291 | Perform entitlement issuance and notify | 573 |
| 11.292 | Perform entitlement issuance and notify | 573 |
| 11.293 | Perform entitlement termination and notify | 574 |
| 11.294 | Perform entitlement termination and notify | 574 |
| 11.295 | Perform entitlement unblocking and notify | 575 |
| 11.296 | Perform entitlement validation and notify | 576 |
| 11.297 | Perform entitlement validation and notify | 576 |
| 11.298 | Perform entitlement XY and notify | 577 |
| 11.299 | Perform entitlement XY and notify | 577 |
| 11.300 | Perform ordered entitlement blocking and notify | 578 |
| 11.301 | Perform ordered entitlement issuance and notify | 579 |
| 11.302 | Perform ordered entitlement issuance and notify | 579 |
| 11.303 | Perform ordered entitlement termination and notify | 580 |
| 11.304 | Perform ordered entitlement termination and notify | 580 |
| 11.305 | Perform ordered entitlement unblocking and notify | 581 |
| 11.306 | Perform stored-value payment method crediting and notify | 582 |
| 11.307 | Perform stored-value payment method debiting and notify | 583 |
| 11.308 | Perform stored-value payment method debiting and notify | 583 |
| 11.309 | Perform stored-value payment method recharging and notify | 584 |
| 11.310 | Perform stored-value payment method recharging and notify | 584 |
| 11.311 | Perform stored-value payment method reimbursing and notify | 585 |
| 11.312 | Perform stored-value payment method reimbursing and notify | 585 |
| 11.313 | Perform user tariff parameters change and notify | 586 |
| 11.314 | Perform user tariff parameters change and notify | 586 |
| 11.315 | Personalise application | 587 |
| 11.316 | Print customer receipt | 588 |
| 11.317 | Print customer receipt | 588 |
| 11.318 | Process action list retrieval configuration | 589 |
| 11.319 | Process extended logging for an application | 590 |
| 11.320 | Process extended logging for an entitlement | 591 |
| 11.321 | Process extended logging for an entitlement | 591 |
| 11.322 | Process media shipment list | 592 |
| 11.323 | Process new information about customer and discounts | 593 |
| 11.324 | Process new information about customer and discounts | 593 |
| 11.325 | Process retrieval request for organisation list | 594 |
| 11.326 | Process retrieval request for organisation list | 594 |
| 11.327 | Put action list retrieval configuration | 595 |

| | | |
|--------|--|-----|
| 11.328 | Read customer and discounts | 596 |
| 11.329 | Read information from user medium with application | 597 |
| 11.330 | Recharge stored-value payment method | 598 |
| 11.331 | Recharge stored-value payment method | 598 |
| 11.332 | Record entitlement within CICO system | 599 |
| 11.333 | Reimburse and terminate account-based payment method | 600 |
| 11.334 | Reimburse and terminate electronic ticket | 601 |
| 11.335 | Reimburse and terminate static entitlement | 602 |
| 11.336 | Reimburse and terminate static entitlement | 602 |
| 11.337 | Reimburse and terminate stored-value payment method | 603 |
| 11.338 | Reimburse and terminate stored-value payment method | 603 |
| 11.339 | Reimburse stored-value payment method | 604 |
| 11.340 | Reimburse stored-value payment method | 605 |
| 11.341 | Reissue entitlements | 605 |
| 11.342 | Remove application from hotlist | 606 |
| 11.343 | Remove authentication key from hotlist | 607 |
| 11.344 | Remove entitlement from hotlist | 608 |
| 11.345 | Remove organisation from hotlist | 609 |
| 11.346 | Remove product acceptance entry from hotlist configuration | 610 |
| 11.347 | Remove product acceptance from participants | 611 |
| 11.348 | Remove product acceptance from participants | 612 |
| 11.349 | Remove SAM from hotlist | 612 |
| 11.350 | Resolve notifications with timeout warnings | 613 |
| 11.351 | Retrieve action list | 614 |
| 11.352 | Retrieve and distribute organisation list | 615 |
| 11.353 | Retrieve and distribute organisation list | 615 |
| 11.354 | Retrieve and distribute the CA certificate repository | 616 |
| 11.355 | Retrieve and distribute the CV certificate revocation list | 617 |
| 11.356 | Retrieve application hotlist | 618 |
| 11.357 | Retrieve CV certificate over signing key | 619 |
| 11.358 | Retrieve entitlement hotlist | 620 |
| 11.359 | Retrieve entitlement hotlist | 620 |
| 11.360 | Retrieve entitlement hotlist with product information | 621 |
| 11.361 | Retrieve incremental action list | 622 |
| 11.362 | Retrieve incremental action list | 622 |
| 11.363 | Retrieve incremental application hotlist | 623 |
| 11.364 | Retrieve incremental entitlement hotlist | 624 |
| 11.365 | Retrieve valid entitlements for given app instance ID | 625 |
| 11.366 | Retrieve valid entitlements for given app instance ID | 625 |
| 11.367 | Revoke application hotlisting demand | 626 |
| 11.368 | Revoke entitlement hotlisting demand | 627 |
| 11.369 | Save electronic ticket as favourite | 628 |
| 11.370 | Securely retrieve entitlement | 629 |

| | | |
|--------|--|-----|
| 11.371 | Sell electronic ticket using account-based payment method | 629 |
| 11.372 | Sell electronic ticket using stored-value payment method | 630 |
| 11.373 | Sell static entitlement using account-based payment method | 631 |
| 11.374 | Sell static entitlement using stored-value payment method | 633 |
| 11.375 | Set service as available for a participant | 634 |
| 11.376 | Set service as unavailable for a participant | 634 |
| 11.377 | Set service as unavailable for a participant | 635 |
| 11.378 | Take back application | 635 |
| 11.379 | Take back entitlement | 636 |
| 11.380 | Take back entitlement | 636 |
| 11.381 | Take back static entitlement | 637 |
| 11.382 | Terminal startup procedure | 638 |
| 11.383 | Terminal startup procedure | 638 |
| 11.384 | Terminate entitlement triggered by action order | 639 |
| 11.385 | Terminate UM application | 640 |
| 11.386 | Trigger entitlement issuance | 641 |
| 11.387 | Unblock application | 642 |
| 11.388 | Unblock application | 642 |
| 11.389 | Unblock entitlement | 643 |
| 11.390 | Unblock entitlement triggered by action order | 644 |
| 11.391 | Unblock entitlement triggered by action order | 644 |
| 11.392 | Update action list inventory from operational perspective | 645 |
| 11.393 | Update authentication key hotlist inventory | 647 |
| 11.394 | Update CA certificate repository | 647 |
| 11.395 | Update CV certificate revocation list | 648 |
| 11.396 | Update CV certificate revocation list | 648 |
| 11.397 | Update hotlist inventory from operational perspective | 649 |
| 11.398 | Update hotlist inventory from product perspective | 651 |
| 11.399 | Update organisation hotlist inventory | 653 |
| 11.400 | Update organisation hotlist inventory | 653 |
| 11.401 | Update organisation list | 653 |
| 11.402 | Update SAM configuration | 654 |
| 11.403 | Update SAM hotlist inventory | 655 |
| 11.404 | Update SAM reset data | 656 |
| 11.405 | Update tariff module | 657 |
| 11.406 | Update terminal action list | 658 |
| 11.407 | Update terminal hotlists | 659 |
| 11.408 | Update terminal hotlists | 659 |
| 11.409 | Validate electronic ticket | 660 |
| 11.410 | Verify action list updated via increments | 661 |
| 11.411 | Verify action list updated via increments | 661 |
| 11.412 | Verify application hotlist updated via increments | 662 |
| 11.413 | Verify entitlement hotlist updated via increments | 663 |

| | | |
|---------|---|-----|
| 11.414 | Verify entitlement hotlist updated via increments | 664 |
| 11.415 | Verify password | 664 |
| 11.416 | Write customer | 665 |
| 11.417 | Write discounts | 666 |
| 11.418 | Write favourites | 666 |
| 11.419 | Write password configuration | 667 |
| 12 | Data Protection | 668 |
| 13 | List of References | 668 |
| 14 | Glossary | 669 |
| 14.1 | BPMN Terms | 669 |
| 14.1.1 | Abstract Process | 669 |
| 14.1.2 | Activity | 669 |
| 14.1.3 | Artifact | 669 |
| 14.1.4 | Association | 669 |
| 14.1.5 | Atomic Activity | 669 |
| 14.1.6 | BPM | 669 |
| 14.1.7 | BPM System | 669 |
| 14.1.8 | BPMN | 670 |
| 14.1.9 | Business Analyst | 670 |
| 14.1.10 | Business Process | 670 |
| 14.1.11 | Business Process Management | 670 |
| 14.1.12 | Choreography | 670 |
| 14.1.13 | Collaboration | 670 |
| 14.1.14 | Collapsed Sub-Process | 670 |
| 14.1.15 | Compensation Flow | 670 |
| 14.1.16 | Compound Activity | 670 |
| 14.1.17 | Controlled Flow | 671 |
| 14.1.18 | Decision | 671 |
| 14.1.19 | End Event | 671 |
| 14.1.20 | Event Context | 671 |
| 14.1.21 | Exception | 671 |
| 14.1.22 | Exception Flow | 671 |
| 14.1.23 | Expanded Sub-Process | 671 |
| 14.1.24 | Flow | 671 |
| 14.1.25 | Flow Object | 672 |
| 14.1.26 | Fork | 672 |
| 14.1.27 | Intermediate Event | 672 |
| 14.1.28 | Join | 672 |
| 14.1.29 | Lane | 672 |
| 14.1.30 | Merge | 672 |
| 14.1.31 | Message | 672 |
| 14.1.32 | Message Flow | 673 |

| | | |
|-----------|--------------------------|-----|
| 14.1.1.33 | Normal Flow | 673 |
| 14.1.1.34 | Parent Process | 673 |
| 14.1.1.35 | Participant | 673 |
| 14.1.1.36 | Pool | 673 |
| 14.1.1.37 | Private Business Process | 673 |
| 14.1.1.38 | Process | 673 |
| 14.1.1.39 | Result | 673 |
| 14.1.1.40 | Sequence Flow | 673 |
| 14.1.1.41 | Start Event | 674 |
| 14.1.1.42 | Sub-Process | 674 |
| 14.1.1.43 | Swimlane | 674 |
| 14.1.1.44 | Task | 674 |
| 14.1.1.45 | Token | 674 |
| 14.1.1.46 | Transaction | 674 |
| 14.1.1.47 | Trigger | 674 |
| 14.1.1.48 | Uncontrolled Flow | 674 |
| 14.2 | Common Terms | 676 |
| 14.2.1 | APDU | 676 |
| 14.2.2 | API | 676 |
| 14.2.3 | CSR | 676 |
| 14.2.4 | EFM | 676 |
| 14.2.5 | IN-OUT Payment Method | 676 |
| 14.2.6 | LDAP | 676 |
| 14.2.7 | MTOM | 676 |
| 14.2.8 | NFC | 676 |
| 14.2.9 | OCSP | 676 |
| 14.2.10 | PKI | 676 |
| 14.2.11 | RSA | 677 |
| 14.2.12 | SAM | 677 |
| 14.2.13 | SAMs | 677 |
| 14.2.14 | SLA | 677 |
| 14.2.15 | SM | 677 |
| 14.2.16 | TLS | 677 |
| 14.2.17 | TLV | 677 |
| 14.2.18 | UM | 677 |
| 14.2.19 | UML | 677 |
| 14.2.20 | Use case | 677 |
| 14.2.21 | WS | 677 |
| 14.2.22 | WS-I | 678 |
| 14.2.23 | WSDL | 678 |
| 14.2.24 | WSS | 678 |
| 14.3 | EN 24014 Terms | 679 |
| 14.3.1 | AO | 679 |

| | | |
|---------|------------------------|-----|
| 14.3.2 | Application Template | 679 |
| 14.3.3 | Commercial Rules | 679 |
| 14.3.4 | Customer Medium | 679 |
| 14.3.5 | IFM | 679 |
| 14.3.6 | MAD | 679 |
| 14.3.7 | Pricing Rules | 679 |
| 14.3.8 | Product Rules | 679 |
| 14.3.9 | Product Template | 679 |
| 14.3.10 | Usage Rules | 679 |
| 14.4 | EN1545 Terms | 681 |
| 14.4.1 | Hotlist | 681 |
| 14.4.2 | Hotlist Service | 681 |
| 14.5 | etiCORE Terms | 682 |
| 14.5.1 | ABPM | 682 |
| 14.5.2 | ALISE | 682 |
| 14.5.3 | Basic process | 682 |
| 14.5.4 | BO-* | 682 |
| 14.5.5 | BO-Main | 682 |
| 14.5.6 | BO-S | 682 |
| 14.5.7 | CCP | 682 |
| 14.5.8 | CCP-BO-AO | 682 |
| 14.5.9 | CCP-BO-Main | 682 |
| 14.5.10 | CCP-BO-OE | 683 |
| 14.5.11 | CCP-BO-STE | 683 |
| 14.5.12 | CCP-T-Main | 683 |
| 14.5.13 | CCP-T-OE | 683 |
| 14.5.14 | CCP-T-STE | 683 |
| 14.5.15 | Central Routing Engine | 683 |
| 14.5.16 | CICO | 683 |
| 14.5.17 | Component | 683 |
| 14.5.18 | CRE | 684 |
| 14.5.19 | EO | 684 |
| 14.5.20 | ESH | 684 |
| 14.5.21 | ESI | 684 |
| 14.5.22 | etiCORE | 684 |
| 14.5.23 | EUI | 684 |
| 14.5.24 | HLS-S | 684 |
| 14.5.25 | IN-OUT | 684 |
| 14.5.26 | ION | 684 |
| 14.5.27 | JSB | 684 |
| 14.5.28 | KA | 684 |
| 14.5.29 | Layer | 685 |
| 14.5.30 | Level-1 | 685 |

| | | |
|---------|---------------------|-----|
| 14.5.31 | Level-2 | 685 |
| 14.5.32 | Level-3 | 685 |
| 14.5.33 | MMS | 685 |
| 14.5.34 | MOTICS | 685 |
| 14.5.35 | OA | 685 |
| 14.5.36 | Ordered action | 685 |
| 14.5.37 | Organisational unit | 686 |
| 14.5.38 | pCCP | 686 |
| 14.5.39 | pCCP-BO-Main | 686 |
| 14.5.40 | pCCP-BO-STE | 686 |
| 14.5.41 | PO | 686 |
| 14.5.42 | PO-BO-AM | 686 |
| 14.5.43 | PO-BO-Main | 686 |
| 14.5.44 | PO-BO-STE | 686 |
| 14.5.45 | sCCP | 686 |
| 14.5.46 | sCCP-BO-Main | 686 |
| 14.5.47 | SCE | 686 |
| 14.5.48 | SO | 687 |
| 14.5.49 | SO-BO-Main | 687 |
| 14.5.50 | SO-BO-STE | 687 |
| 14.5.51 | SO-T-Main | 687 |
| 14.5.52 | SO-T-STE | 687 |
| 14.5.53 | Static Entitlement | 687 |
| 14.5.54 | STE | 687 |
| 14.5.55 | SVPM | 687 |
| 14.5.56 | T-Main | 687 |
| 14.5.57 | TO | 687 |
| 14.5.58 | UMO | 688 |
| 14.5.59 | User Medium | 688 |
| 14.5.60 | User Medium Owner | 688 |
| 14.5.61 | Workflow | 688 |



1 Purpose

The document aims to provide an overview of the processes and roles involved. Furthermore, the distributed use cases in the system components are shown.

The quick reference guide is intended to explain the systematics of the processes and their representation in the diagrams.

2 Scope

This part of the specification contains all the general prerequisite parts needed for understanding the use cases and processes.

In addition to a quick reference guide and the glossary, the role model, as well as the descriptions of the elementary processes and their interaction, are included.

All basic processes are listed for the different business segments. Furthermore, the specification contains all use case descriptions in alphabetical order.

Note: use cases and workflows of the SPEC-ION and SPEC-EUI-ESI-MPI are not listed in this specification.

3 Introduction

((eTicket Deutschland is the German standard for electronic fare management (EFM) in public transport.

The technical specification of this standard is called (((etiCORE and stands for future-proof electronic fare management in German public transport.

This standard provides electronic tickets as well as cashless payment methods for public transport services and describes the processes in in-out systems with automatic or semi-automatic recording of travel entitlements.

Chip cards, smartphones and other media are supported. The associated specification describes data and processes concerning security, device communication and message exchange.

An important part of the technical standard is the security infrastructure, which primarily aims at fraud prevention, forgery protection and copy protection and, thus, at securing revenue for transport companies. With (((etiCORE, this security architecture is specified with the help of today's current standards.

Scalable and efficient security algorithms are introduced to ensure future-proofing in this area. This raises the security infrastructure to the next generation, hence the abbreviation 2GSI (second-generation security infrastructure) for some of the near-hardware specifications.

(((etiCORE is based - like the previous VDV Kern Applikation - on international standards such as the role model according to ISO 24014 and the identification card systems according to EN 1545-1 and EN 1545-2.

The (((etiCORE role model adapts the role model according to ISO 24014-1 with the help of some changes to the conditions of the German public transport. For each role in the (((etiCORE role model there is a reference specification which contains all required functionality of back-office systems or terminals for the respective tasks.

For each reference specification, there is, in turn, an interface specification that defines the exchange of data from the user medium (smart card, smartphone, etc.) via the terminal to the back-office systems.

All specification parts are based on a comprehensive specification model in which it is also possible to navigate via a web browser. In addition to the above-mentioned reference specifications, this model also offers other views, e.g. workflows and basic processes. In this

way, the systems and their functions can be viewed in the overall context. This enables a quick overview and a classification of one's own tasks for a specific role.

Functionality bundles are also considered. It is possible to filter quickly via navigation or with the help of the reference specifications and their chapter structure, which functionality is required for which expansion variant.

The functionality bundles range from the basic package to possible cashless payment methods to the automatic recording of trip entitlements in IN-OUT systems.

The technical concept also defines the structure of interoperable fare products such as the Deutschland-Ticket or interoperable payment methods that enable Germany-wide participation in IN-OUT systems in addition to payment for public transport services. The features of the fare products can be limited to fare zones as well as extended to an entire area, which makes them valid as interoperable fare products.

The technical concept takes into account both contact-based processes with the user medium as well as the compulsory contactless collection processes in accordance with ISO 14443.

The technical concept also defines the security processes that take place during the exchange of data between individual partners/roles. It starts from the concept of trustworthiness: all participants of (((eTicket Deutschland must be able to trust that the data transmitted to them is genuine. This trust is established by the participants being able to directly check the authenticity of the counterpart using cryptographic methods. This verification takes place bidirectionally.

This applies to every point in the communication chain.

A user medium checks whether the terminal involved is authorised to perform actions with it. In addition to the mutual verification of the respective partners, the value objects in the form of electronic tickets or payment methods are authentic because these are exclusively applied between authentic user media and SAMs with a secured session.

The proof of issuance is secured by the signature of the user medium. With the help of this signature, a transport company checks, when submitting the proof of issuance for a value object, whether this value object is authentic and its issuance was authorised.

The further message exchange between the participating background systems in the ION (interoperable network) is secured by transport encryption (TLS) and message encryption and signature via web service security.

The different parts in the present specification model are presented in the following chapter (see [Specification Overview](#)) as an overview.

3.1 History and Transition

Since 2006, the VDV-KA (KA = Kernapplikation, VDV = Association of German Transport Companies, Verband deutscher Verkehrsunternehmen) has been the technical description of the standard for electronic fare management and, thus, for (((eTicket-Deutschland. Designed for interoperable operation, the potential of the VDV-KA has been evident since the introduction of the Deutschland-Ticket in 2023.

In 2026, this standard VDV-KA will be replaced - with a transitional period until 2031 - by the new version (((etiCORE, which includes the strengths of the VDV-KA and introduces many improvements.

The replacement at the above-mentioned date is primarily for technical reasons: the security infrastructure, which serves to secure revenue for the public transport companies, must be renewed.

On the one hand, this is because, to put it simply, the certification authorities for key certificates (root and sub-CA) will finally expire on 01 December 2031. Both, the certification authorities and the certificates issued by these certification authorities, are required in all smart cards, barcodes and security modules (SAMs). They determine their validity period, and can no longer be renewed after the above-mentioned date.

On the other hand, the RSA algorithms used so far for encryption and signature are not scalable at any size. To be able to guarantee a consistent level of security in the future, the key length, which is essential for security, would have to be doubled. However, doubling the key length would mean a significantly longer processing time.

By using a more scalable security algorithm (elliptic curves), the security infrastructure becomes future-proof. However, the new certification authorities and, thus, all keys and certificates will then work with the new security algorithm. This, in turn, requires a new and no longer backwards-compatible implementation of smart cards, barcodes and security modules (SAMs).

This circumstance has prompted VDV-ETS to revise and improve the complete standard to create a future-proof and long-term version that offers corresponding investment security for a long period of time (> 20 years).

In addition, this new version offers several other advantages such as

- Reduced complexity and scope
- Increased performance
- Reusability of SAM and user medium after expiry of the certificate period
- Medium and long-term cost reduction in distribution, control and security management

Even though this is essentially only a software update for the IT systems of the transport companies and associations, the changes are very extensive and no longer backwards compatible. Therefore, a corresponding migration is necessary.

Internationally-used standard tools for software development open up access to German public transport for more developers and companies. Therefore, (((etiCORE was written in English.

3.2 Specification Overview

The following overview shows the specification artefacts of etiCORE.

3.2.1 Model Specification Overview

| Full Title | Short Title | Description |
|--|-------------|---|
| etiCORE Specification (Model) | | Contains all the information included in the specifications below. The following specifications represent a special export from the specification model. The model itself is available as a browser version . |
| Glossary | Glossary | Contains all important terms and abbreviations. |
| Main Specification | SPEC-MAIN | Contains all general parts that are needed as a prerequisite for understanding the use cases and processes. Among other things, the role model and the process descriptions in BPMN are contained there. |
| Interoperable Network Specification | SPEC-ION | Describes the secure exchange of messages between back-office systems. Specifies general procedures, checks and requirements for the messages. Contains instructions for the web service security used and parts for the intermediate storage of messages in the central routing engine. The document contains direct links to the specification model. |
| Personalisation Unit (Interface) Specification | SPEC-PUI | Description of the interfaces and processes between a control unit (in the case of a sales terminal, the sales unit) and the execution unit (in the case of a sales terminal, the personalisation unit) of a terminal. |
| External System (Interface) Specification | SPEC-ESI | Describes the interfaces and processes for issuing electronic tickets via a sales terminal controlled by an external system. |

| | | |
|--|--------------|---|
| Mass Personaliser (Interface) Specification | SPEC-MPI | Describes the interfaces and processes for mass personalisation of user media. |
| Customer Contract Partner Reference System Specification | SPEC-CCP-RS | Describes all use cases for the customer contract partner back-office system including functionality bundles. Hybrid document with a link to the specification model. |
| Customer Contract Partner Reference Terminal Specification | SPEC-CCP-RT | Describes all use cases for the customer contract partner terminal including functionality bundles. Hybrid document with a link to the specification model. |
| Service Operator Reference System Specification | SPEC-SO-RS | Describes all use cases for the service operator back-office system including functionality bundles. Hybrid document with a link to the specification model. |
| Service Operator Reference Terminal Specification | SPEC-SO-RT | Describes all use cases for the service operator terminal including functionality bundles. Hybrid document with a link to the specification model. |
| Product Owner Reference System Specification | SPEC-PO-RS | Describes all use cases for the product owner back-office system including functionality bundles. Hybrid document with a link to the specification model. |
| Hotlist Service System Specification | SPEC-Hotlist | Describes all use cases for the central system of the hotlist service. Hybrid document with linking into the specification model. |
| Central Routing Engine Specification | SPEC-CRE | Describes all use cases for the central routing engine. Hybrid document with linking into the specification model. |
| Application and Security Management Specification | SPEC-ASM | Part of the specification of the ASM for the (((etiCORE functionality with the required use cases. Hybrid document with linking into the specification model. |
| SAM-Server Specification | SPEC-SAMS | Describes the interface and functionality of the SAM server. |

3.2.2 Model Specification Overview DE

| Vollständiger Titel | Kurztitel | Beschreibung |
|--|-----------|--|
| Spezifikationsmodell | | Beinhaltet alle Informationen, die in den nachfolgenden Spezifikationen enthalten sind. Die nachfolgenden Spezifikationen stellen einen speziellen Export aus dem Spezifikationsmodell dar. Das Modell selber liegt als Browser-Version vor. |
| Glossary | Glossary | Enthält alle wichtigen Begriffe und Abkürzungen. |
| Main Specification | SPEC-MAIN | Beinhaltet alle allgemeinen Anteile, die als Voraussetzung für das Verständnis der Anwendungsfälle und Prozesse benötigt werden. Dort sind unter anderem das Rollenmodell sowie die Prozessbeschreibungen in BPMN enthalten. |
| Interoperable Network Specification | SPEC-ION | Beschreibt den sicheren Nachrichtenaustausch zwischen Hintergrundsystemen. Spezifiziert allgemeine Abläufe, Prüfungen und Anforderungen an die Nachrichten. Enthält Hinweise für die verwendete Webservice Security und Teile für die Zwischenspeicherung von Nachrichten in der zentralen Vermittlungsstelle. Das Dokument enthält direkte Links in das Spezifikationsmodell. |
| Personalisation Unit (Interface) Specification | SPEC-PUI | Beschreibung der Schnittstellen und Abläufe zwischen einer Steuereinheit (bei Vertriebsterminal die Vertriebsseinheit) und der Ausführungseinheit (im Vertriebsterminal die Personalisierungseinheit) eines Terminals. |
| External System (Interface) Specification | SPEC-ESI | Beschreibt die Schnittstellen und Prozesse für eine durch ein Fremdsystem gesteuerte Ausgabe von elektronischen Tickets über ein Vertriebsterminal. |
| Mass Personaliser (Interface) Specification | SPEC-MPI | Beschreibt die Schnittstellen und Prozesse zur Massenpersonalisierung |

| | | |
|--|--------------|--|
| | | von Nutzermedien. |
| Customer Contract Partner Reference System Specification | SPEC-CCP-RS | Beschreibt alle Anwendungsfälle für das Kundenvertragspartner Hintergrundsystem inklusive Ausbauvarianten. Hybrid-Dokument mit Verlinkung in das Spezifikationsmodell. |
| Customer Contract Partner Reference Terminal Specification | SPEC-CCP-RT | Beschreibt alle Anwendungsfälle für das Kundenvertragspartner Terminal inklusive Ausbauvarianten. Hybrid-Dokument mit Verlinkung in das Spezifikationsmodell. |
| Service Operator Reference System Specification | SPEC-SO-RS | Beschreibt alle Anwendungsfälle für das Dienstleister Hintergrundsystem inklusive Ausbauvarianten. Hybrid-Dokument mit Verlinkung in das Spezifikationsmodell. |
| Service Operator Reference Terminal Specification | SPEC-SO-RT | Beschreibt alle Anwendungsfälle für das Dienstleister Terminal inklusive Ausbauvarianten. Hybrid-Dokument mit Verlinkung in das Spezifikationsmodell. |
| Product Owner Reference System Specification | SPEC-PO-RS | Beschreibt alle Anwendungsfälle für das Hintergrundsystem des Produktverantwortlichen inklusive Ausbauvarianten. Hybrid-Dokument mit Verlinkung in das Spezifikationsmodell. |
| Hotlist Service System Specification | SPEC-Hotlist | Beschreibt alle Anwendungsfälle für das zentrale System des Hotlist-Service. Hybrid-Dokument mit Verlinkung in das Spezifikationsmodell. |
| Central Routing Engine Specification | SPEC-CRE | Beschreibt alle Anwendungsfälle für die zentrale Vermittlungsstelle. Hybrid-Dokument mit Verlinkung in das Spezifikationsmodell. |
| Application and Security Management Specification | SPEC-ASM | Anteil der Spezifikation des ASM für die (((etiCORE Funktionalität mit den benötigten Anwendungsfällen. Hybrid-Dokument mit Verlinkung in das Spezifikationsmodell. |
| SAM-Server Specification | SPEC-SAMS | Beschreibt die Schnittstelle und Funktionalität des SAM Servers. |

3.2.3 Further Specification Overview

| Full Title | Short Title | Description |
|--|------------------|---|
| 2GSI CipherSuite Specification | SPEC-CipherSuite | Specifies the cryptographic algorithms used in (((etiCORE/2GSI as used on the user medium, SAM and in the signature of barcodes or MOTICS. |
| 2GSI M2M Certificates Specification | SPEC-M2MC | Specifies structure and content of machine-to-machine certificates, based on the Card Verifiable (CV) certificate structure according to ISO/IEC 7816-8. |
| 2GSI PKI Setup Specification | SPEC-PKI-Setup | Specifies parameters and data for the 2GSI Media PKI, which is responsible for issuing certificates for SAM and user media. |
| 2GSI SAM Individualisation Specification | SPEC-SAM-I | Specifies the individualisation process of SAMs. For SAM manufacturers only. |
| 2GSI SAM Specification | SPEC-SAM | Specification of the data model and command set of the SAM application. Note: Processes between user medium, terminal and SAM can be found in the specification model. |
| 2GSI User Medium Specification | SPEC-UM | Specification of the data model and command set of the smart card application. Note: Processes between user medium, terminal and SAM are now found in the specification model. Mechanical properties, etc. are outsourced to data sheets or capability sheets. |
| MOTICS Specification | SPEC-MOTICS | Specifies the structure and use of the MOTICS copy protection |

| | | |
|---|----------|--|
| | | container in interaction with ticket purchasing and ticket control. |
| Terminal-Customer-Interface Specification | SPEC-TCI | Specification for the uniform customer interface for multi-level interoperable electronic fare management. |

3.2.4 Further Specification Overview DE

| Vollständiger Titel | Kurztitel | Beschreibung |
|--|------------------|--|
| 2GSI CipherSuite Specification | SPEC-CipherSuite | Spezifiziert die in (((etiCORE/2GSI verwendeten kryptographischen Algorithmen, wie sie im Nutzermedium, SAM und bei der Signatur von Barcodes oder im MOTICS zum Einsatz kommen. |
| 2GSI M2M Certificates Specification | SPEC-M2MC | Spezifiziert Struktur und Inhalt von Machine-to-Machine Zertifikaten, basierend auf der Card Verifiable (CV) Zertifikatsstruktur nach ISO/IEC 7816-8. |
| 2GSI PKI Setup Specification | SPEC-PKI-Setup | Spezifiziert Parameter und Daten für die 2GSI Medien PKI, die für die Ausgabe von Zertifikaten für SAM und Nutzermedien zuständig ist. |
| 2GSI SAM Individualisation Specification | SPEC-SAM-I | Spezifiziert den Individualisierungsprozess von SAMs. Nur für SAM Hersteller. |
| 2GSI SAM Specification | SPEC-SAM | Spezifikation von Datenmodell und Befehlssatz der SAM-Applikation. Hinweis: Abläufe zwischen Nutzermedium, Terminal und SAM finden sich nun im Spezifikationsmodell. |
| 2GSI User Medium Specification | SPEC-UM | Spezifikation von Datenmodell und Befehlssatz der Chipkarten Applikation. Hinweis: Abläufe zwischen Nutzermedium, Terminal und SAM |



| | | |
|---|-------------|---|
| | | finden sich nun im Spezifikationsmodell. Mechanische Eigenschaften, etc. werden in Datenblätter bzw. Capability-Sheets ausgelagert. |
| Motics Specification | SPEC-MOTICS | Spezifiziert Aufbau und Einsatz des MOTICS Kopierschutzcontainers im Zusammenspiel mit Ticketerwerb und Ticketkontrolle |
| Terminal-Customer-Interface Specification | SPEC-TCI | Spezifikation zur einheitlichen Kundenschnittstelle für ein mehrstufiges interoperables elektronisches Fahrgeldmanagement |

3.3 System Overview

The overview shows the interaction of the components in the overall context of the etiCORE specification.

The central routing engine was not shown in order to better illustrate the logical exchange of information.

The arrows show the most important messages that are exchanged among the components. The arrow means that the addressed system provides the functionality for receiving the information or the information itself.

In reality, the components shown have a much larger range of functions, but these are not part of the etiCORE specification (e.g. booking system, customer administration, etc.). These parts are not shown.

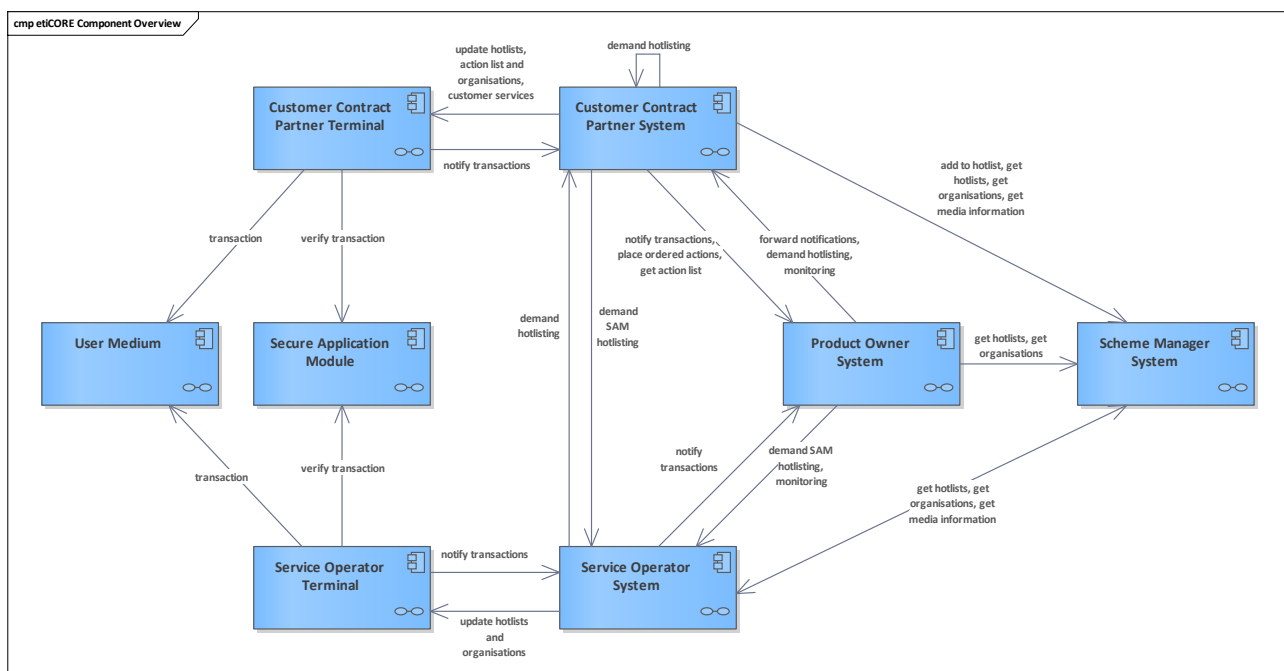


Figure 1: etiCORE Component Overview

See [System Overview](#).

4 User Manual

The following chapter contains an extract from the [User Manual](#) that helps to understand the following diagrams and concepts.

4.1 Interaction of Basic Processes (Layer 1)

Layer 1 shows the interaction of the basic processes.



A typical, more complex business process can consist of several basic processes (e.g. purchase of an electronic trip entitlement) or other basic processes that can be triggered (e.g. during entitlement inspection).

Layer 1 thus provides an overview of processes that consist of basic processes, which are represented in the form of BPMN choreography diagrams with respective descriptions.

Target groups are

- Business analysts
- Sales staff
- Clerks
- Software developers (to gain a first overview)

4.2 Basic Processes (Layer 2)

Layer 2 shows the activities within a basic process, including the various participants. Layer 2 thus shows the interaction of the participants, including the activities required in each case and the messages exchanged. Activities may also be listed that are not specified in etiCORE but are necessary in the process flow. This results in an overall picture of the process flow, which contributes to understanding.

Target groups are

- Business analysts
- Sales staff
- Clerks
- Software developers (to gain a first overview)

4.3 Workflows and Basic Processes (Layer 1 and 2)

The following chapters describe the considerations for workflows and basic processes concerning reading, navigation and understanding.

These chapters cover all BPMN-based specification parts.

The modelling rules for BPMN diagrams were based on the well-known rules of the Swiss government, which has developed a comprehensive set of rules for this purpose. See also eCH-0158 BPMN modelling conventions at <https://www.ech.ch/de/standards/60265>. (Only available in German and French).

4.3.1 Layer 1 - BPMN Choreography

Layer 1 in the model is represented as a BPMN choreography. The following describes how to read and understand diagrams and elements.

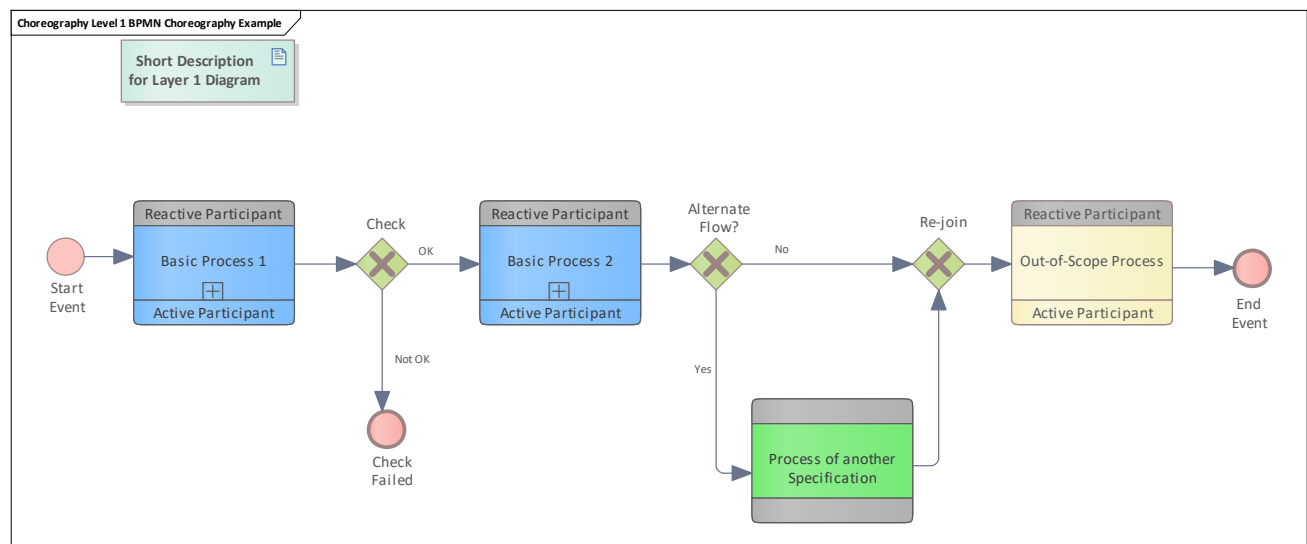


Figure 2: Level 1 BPMN Choreography Example

See [Short Description for Layer 1 Diagram](#).

4.3.2 Basic Process 1

This element is usually representative of a basic process that is specified within the etiCORE specification. This is symbolised by the blue colour of the element.

The active participant is the one that initiates the process. This participant always has the same colour as the choreography element. The reactive participant always has a grey colour and is the one that has to process data from the active participant or provides data for it.

On this top-layer 1, the process is usually only explained with a short description.

As a rule, the "plus" symbol in the choreography element means that further details are available by double-clicking it.

By clicking on it, the process can be viewed in more detail. In this case, a layer 2 with a BPMN Collaboration Diagram is usually opened automatically (HTML Version only). Please see [Basic Process 2](#) to try this.

4.3.3 Basic Process 2

This element represents another etiCORE basic process that is executed as a consequence of the first process.

This element is joined with a layer 2 diagram.

At this point, no further details are given for the user manual.

4.3.4 Out-of-Scope Process

This element stands for a process that is not further specified in etiCORE. This is symbolised by the yellow colour. It can contain details which are then not specified further but merely contain short descriptions for each case.

4.3.5 Process of another Specification

This process is specified in more detail in another specification. This is symbolised by the green colour. In our model, this is currently only the specification for tariff modules.

4.3.6 Check Failed

End event that indicates an exception. In this case, any subsequent basic processes will not be executed.

4.3.7 End Event

End event for the standard run ("positive case") of the process.

Often, a brief description is given of the state of the system after the process runs.

4.3.8 Start Event

Start event of our example process. Start events are rarely described in detail.

4.3.9 Short Description for Layer 1 Diagram

This diagram shows a flow with several elementary processes in BPMN choreography form. The standard BPMN elements (decision gateways, etc.) are available as control flow elements.

The process participants are listed in the respective choreography element. Usually, there are two participants. The active participant is usually at the bottom of the element and is in the

element colour, while the reactive participant is on the top and marked with the colour grey. Some basic processes have more than 2 participants, sometimes additionally with alternating active and reactive parts. In this case, either another choreography element is used, or only the first two participants are listed and the details are only visible in layer 2. The procedure depends on the complexity of layer 1.

The reading direction is always from left to right; the start event is always on the left, the and end event is always on the right.

Exception handling is not shown in these diagrams. However, error end states may be included. In the diagram, the base path ("positive case") is always shown as straight as possible; error cases then deviate (usually downwards).

4.3.10 Alternate Flow?

Same as in gateway [Check](#).

4.3.11 Check

Our sample process gateway. Usually an exclusive-or gateway. The name of the check or the deviation condition is shown above the gateway. The result is shown in the sequence flow after the gateway.

In BPMN, gateways of alternate flows must be rejoined with the same gateway type. The join-gateway has no checks or conditions, so usually, it has no name or further description. Additionally, if the name of the fork-gateway is sufficient for understanding the sequence flow, no further description is given for the gateway.

4.3.12 Re-join

Gateway to re-join the process. Usually without description. Must be the same type as the fork gateway. The cross means XOR (exclusive or).

4.3.13 Layer 2 - BPMN Collaboration

Layer 2 in the model is represented as a BPMN Collaboration. The following describes how to read and understand diagrams and elements.

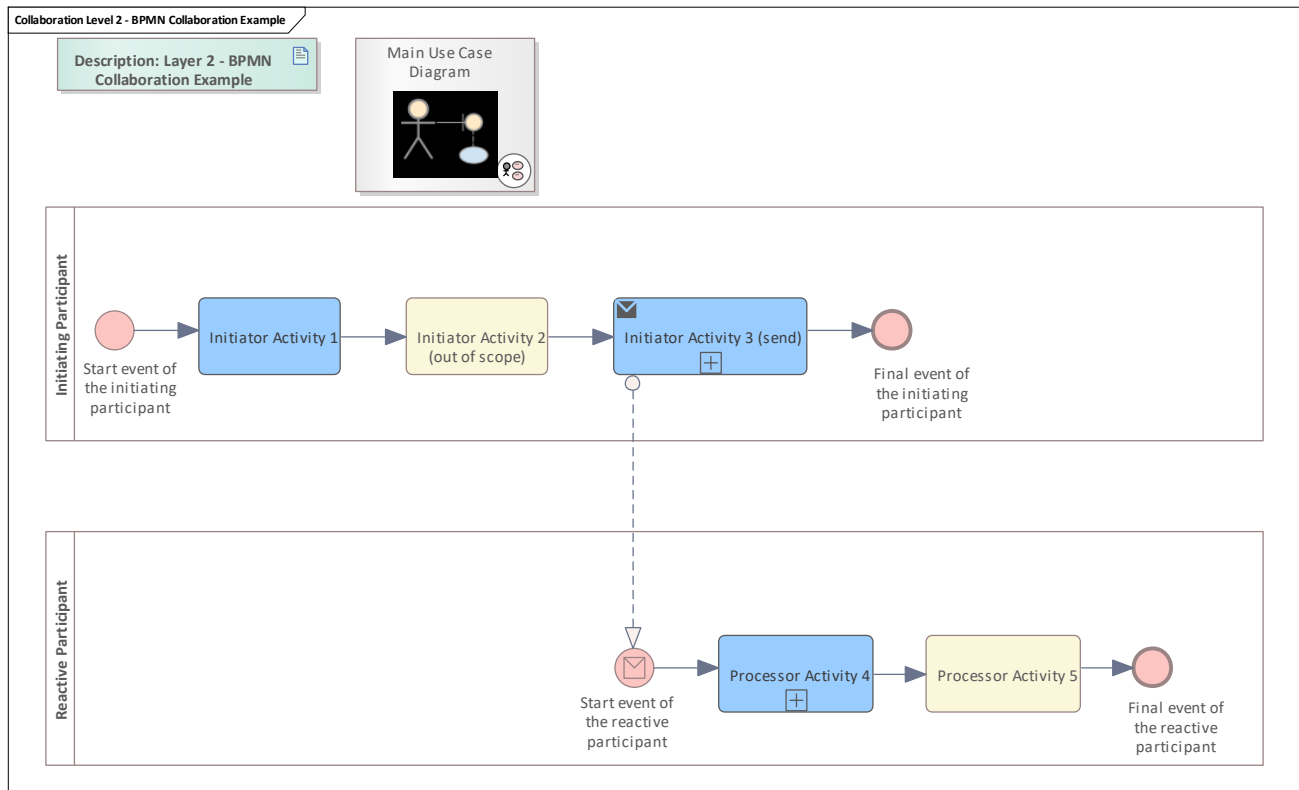


Figure 3: Level 2 - BPMN Collaboration Example

See [Description: Level 2 - BPMN Collaboration Example](#).

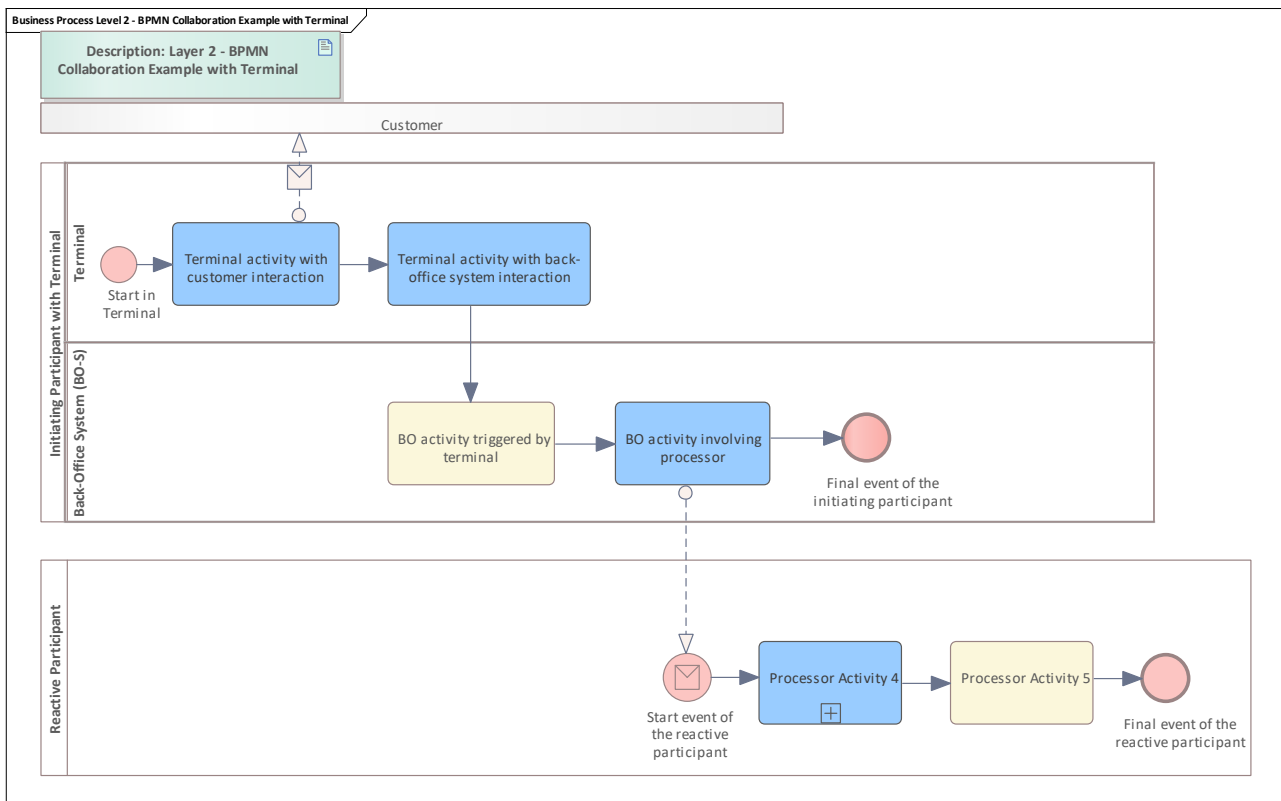


Figure 4: Level 2 - BPMN Collaboration Example with Terminal

See [Description: Level 2 - BPMN Collaboration Example with Terminal](#).

4.3.14 Description: Layer 2 - BPMN Collaboration Example

This diagram generally shows a flow within an elementary process in BPMN collaboration form. The standard BPMN elements are available as contour flow elements (gateways for decisions, etc.). The respective pools (the horizontal rectangles) represent the process participants. The initiating participant is often the top pool (here in white). If necessary, participants are also shown as a closed pool (black box) if data is exchanged. Here, the process in this pool is not relevant for the current process. In the model, this is often the case with the customer. This then represents the user medium.

The other pools are usually represented as a white box, i.e. it shows which of the activities and message exchanges take place.

The reading direction is always from left to right; the start event is always on the left, and the end event is always on the right. Exception handling is not shown in these diagrams. However, error end states may be included. In the diagram, the base path ("positive case") is always shown as straight as possible; error cases then deviate (usually downwards).

Below an activity, further activities could be hidden in the form of use cases. In this case, the activity is marked with a "plus" symbol. Clicking on the activity then opens a use case diagram showing further details (HTML version only).

The same rules apply to the colouring of the elements as in layer 1. Blue is an activity specified in etiCORE, yellow activities are only described but not specified and green activities are specified in other specifications.

If the entire basic process is specified again at the use case layer (layer 3), there is a hyperlink or navigation cell at the top left of the diagram which then opens the corresponding use case diagram (HTML version only).

4.3.15 Description: Layer 2 - BPMN Collaboration Example with Terminal

This diagram generally shows a flow within a basic process in BPMN collaboration form as described in [Description: Layer 2 - BPMN Collaboration Example](#).

Additionally, two lanes are used to emphasise that a participant's terminal is involved, as well as the back office system.

If a pool is involved which is not further described (black box, i.e. customer here), this pool must be marked as a black box and is shown as a narrow, long bar without further details.

4.3.16 Customer

BPMN Pool for customer (as black box pool).

4.3.17 Initiating Participant

This pool identifies the participant where the process initially starts.

4.3.17.1 Initiator Activity 1

etiCORE specified activity 1 of the initiating participant.

4.3.17.2 Initiator Activity 2 (out of scope)

Not in etiCORE specified activity 2 of the initiating participant.

4.3.17.3 Initiator Activity 3 (send)

etiCORE specified activity 3 of the initiating participant.

The "plus" shows that additional details are available. A click will open a layer 3 UML use case diagram. The letter symbol indicates that a message is sent within this activity.

4.3.17.4 Final event of the initiating participant

Final event of the initiating participant.

4.3.17.5 Start event of the initiating participant

Start or triggering event of the initiating participant.

4.3.18 Initiating Participant with Terminal

This pool identifies the participant where the process initially starts. Additionally, the pool is divided into a lane for the terminal and a lane for the back-office system.

4.3.18.1 Back-Office System (BO-S)

Lane for the back-office system.

4.3.18.1.1 BO activity involving processor

Activity in the back-office system which involves a further processing system.

4.3.18.1.2 BO activity triggered by terminal

Activity in the back-office system, triggered by the message coming from the terminal.

4.3.18.2 Terminal

Lane for the terminal.

4.3.18.2.1 Terminal activity with back-office system interaction

Activity which collects data to forward the transaction information to the back-office system.

4.3.18.2.2 Terminal activity with customer interaction

Activity which shows the interaction between a terminal and the customer - normally represented by the customer's user medium.

4.3.18.2.3 Start in Terminal

Start or triggering event in the terminal.

4.3.19 Reactive Participant

This pool identifies the (reactive) participant who is addressed by the initiating participant. In a chain of participants (≥ 3), a distinction between active participant and reactive participant cannot be made precisely, as the participant "in the middle" can be active and reactive as well.

4.3.19.1 Processor Activity 4

etiCORE specified activity 4 of the reactive participant.

4.3.19.2 Processor Activity 5

Not in etiCORE specified activity 5 of the reactive participant.

4.3.19.3 Start event of the reactive participant

Start or triggering event of the reactive participant.

4.3.19.4 Final event of the reactive participant

End event of the reactive participant.

5 Role Model

The role model consists of the interaction of the roles on one hand and the roles and their description on the other hand. Although each role is modelled as a partner role, these partner roles may combine roles into one natural person or system.

5.1 Role Model etiCORE

This chapter shows the extended role model used inside (((etiCORE. Most of the roles are similar and only differ in scope. Roles which correspond exactly to the standard 24014-1 are not redefined here.

The overview in [Role Model etiCORE](#) shows the BPMN-based notation of the standard (((etiCORE role model as an adaptation of the ISO 24014-1 role model. It employs partner roles and partner entities.

The overview in [Role Model etiCORE Ordered Action Management](#) shows the BPMN-based notation of the extended (((etiCORE role model for the remote ordering of actions to be done later with user media when contacting a suitable terminal.

To understand how the (((etiCORE roles match the ISO 24014-1 roles, see the [Transition ISO 24014 to etiCORE](#).

5.1.1 Role Model etiCORE

See [Role Model etiCORE](#).

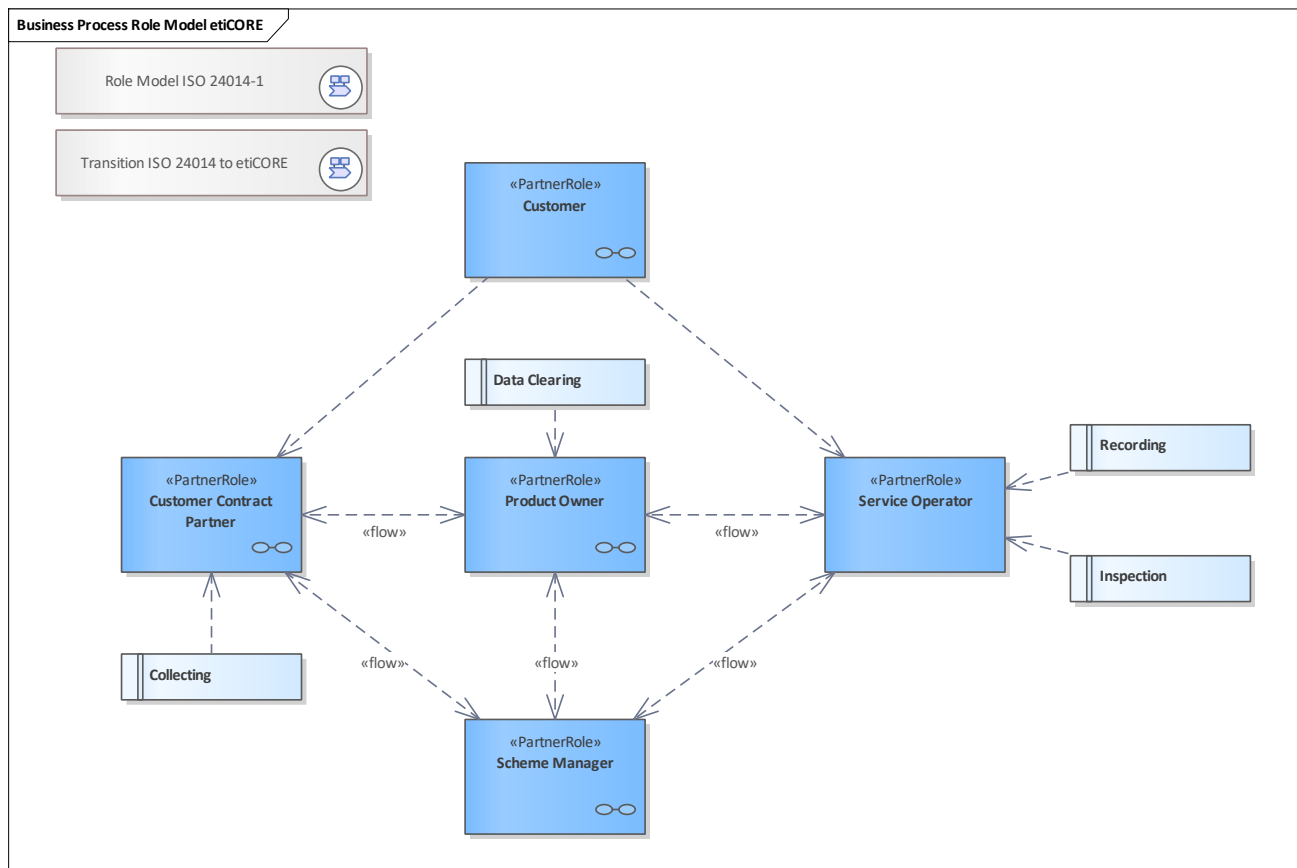


Figure 5: Role Model etiCORE

5.1.2 Role Model etiCORE Ordered Action Management

See [Role Model etiCORE](#).

In the context of Ordered Action Management, it is the customer's task to pay attention to the following special aspects:

- Since the distribution of the action list to the sales terminals requires a certain amount of time, the customer must allow for a corresponding lead time for the use of the processed entitlement.
- Action orders can be provided at distribution terminals named by the Ordering CCP to the customer. If the action order for the entitlement is not provided at all terminals, the customer must bring his user medium into contact with one of the specified terminals before the first use.

5.1.3.1 Customer

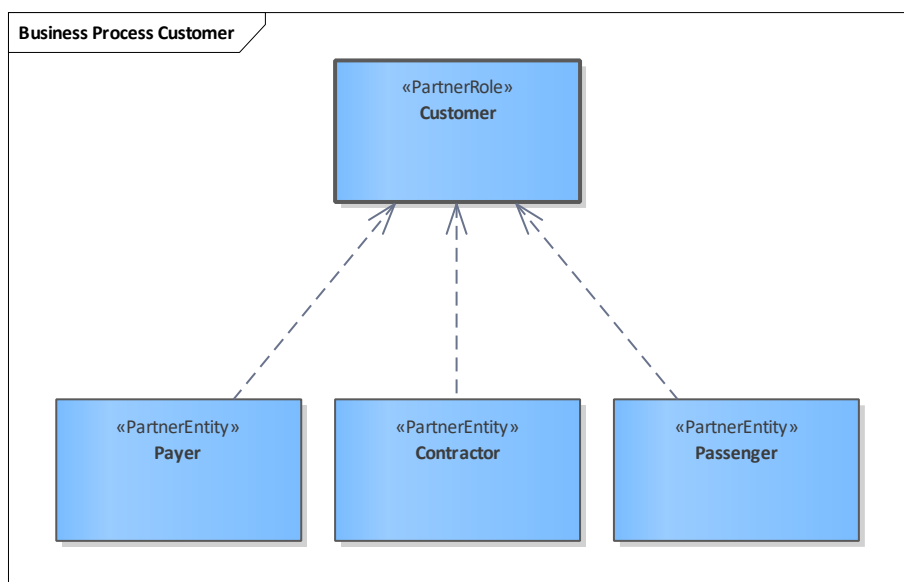


Figure 7: Customer

5.1.3.2 Contractor

The contractor is a sub-role of the [Customer](#), who concludes contracts with a [Customer Contract Partner](#) regarding the type of use.

5.1.3.3 Passenger

The [Customer](#) can be the same person as the passenger. Definition due to ISO 24014 is [Passenger](#).

5.1.3.4 Payer

The Payer is a sub-role of [Customer](#) and specifies who is responsible for paying for the use of the service.

The payment is done towards the [Customer Contract Partner](#).

5.1.4 Customer Contract Partner

The role of the Customer Contract Partner (CCP) is a customised role in (((etiCORE. It consists of several roles defined in ISO 24014:

- [Product Retailer](#)
- [Application Retailer](#)
- [Customer Service](#)
- Identity Provider
- Account Provider
- Payment Provider

The customer contract partner regulates the customer's sales, taking into account the contractual dependencies towards the [Scheme Manager](#) and the [Product Owner](#) of different mobility services.

In particular, the Customer Contract Partner is responsible for the accounting, ticketing and billing of different mobility services to the [Customer](#). Thus, he takes on the role of the [Product Retailer](#), [Application Retailer](#) (or Issuer) and Account Provider. As an Account Provider, he uses the services of his chosen Payment Providers with whom he is contractually bound.

As a [Product Retailer](#), he sells tickets or other services to the customer. As an [Application Retailer](#), he sells a whole application to the customer. In (((etiCORE, there is no sale of applications at the moment, but this might change in the future. For now, the Customer Contract Partner can be considered as an Application (Instance) Issuer.

Basically, the Customer Contract Partner serves as an authority to conclude a mobility contract with the Customer. In addition, he offers a [Customer Service](#) for matters relating to the conclusion of the contract.

The customer contract partner acquires, as regulated in a contractual relationship, the right from the [Scheme Manager](#) to participate in the EFM system and acquires the required SAM (in the required configuration and with the required keys and certificates) and necessary identifiers. Within the framework of the [Security Manager](#), he orders the keys and certificates required for the generation of entitlements and the transmission of messages via ION (the interoperability network of etiCORE).

The customer contract partner issues the public transport application to the customer on behalf of the [Application Owner](#). He is, thus, the [Primary Customer Contract Partner](#) for the application on the user medium.

The Customer Contract Partner acquires, as regulated in a contractual relationship, from [Product Owners](#) the right to sell their products as entitlements to customers and receives from the product owners the necessary product definitions (tariff modules) and templates for issuing the entitlements.

The customer contract partner issues interoperable entitlements (valid for all companies participating in (((etiCORE in the role of [Service Operators](#) and customer contract partners who accept the respective product on which the entitlement is based) to customers and collects the usage fees from the customer in return.

The Customer Contract Partner issuing an entitlement is referred to as the [Primary Customer Contract Partner](#) concerning this entitlement.

If the terms of use of an entitlement allow transactions to be carried out on further Customer Contract Partners, these other Customer Contract Partners are referred to as [Secondary Customer Contract Partners](#).

The [Primary Customer Contract Partner](#) settles the usage fees received from its customers with the other (((etiCORE system participants.

It bears the payment risk vis-à-vis the other (((etiCORE system participants.

The Customer Contract Partner provides [Customer Service](#) in all matters relating to the EFM system.

If a Payment Method with subsequent payment (post-pricing) is used, the [Primary Customer Contract Partner](#) receives all transaction data required for its customer invoices (entry vouchers, possibly linked to the fare calculated by the Product Owner) recorded by the Service Operators and via the [Product Owner](#).

The [Primary Customer Contract Partner](#) performs blocking orders for applications and entitlements which he has issued if blocking requests are pending in this regard. The [Primary Customer Contract Partner](#) decides on blocking or unblocking (releasing) of the applications and entitlements issued by him.

The [Primary Customer Contract Partner](#) decides on blocking/unblocking requests of other (((etiCORE system participants for the applications and entitlements issued by himself and issues the blocking orders or blocking release orders to the [Hotlist Service](#).

It informs the requesting instance of the decision, i.e. whether or not a blocking/unblocking has been initiated and, for this purpose, forwards corresponding block/unblock notifications to the requesting instance.

The customer contract partner sends blocking/unblocking requests to another responsible [Primary Customer Contract Partner](#) (applications and entitlements) or the [Scheme Manager](#) (SAMs and organisations), if necessary.

The customer contract partner regularly procures the current blocking lists from the [Hotlist Service](#), ensures distribution to its terminals and checks all recorded user media there against these blocking lists.

The customer contract partner blocks entitlements or the entire public transport application if these are on the hotlist and are recorded at one of its terminals. It reports the successful physical blocking to the [Product Owner](#).

The customer contract partner monitors its sales and service terminals for compliance with security regulations.

A customer contract partner that commissions others to perform tasks on its behalf may also commission these companies to operate his terminals.

A customer contract partner may make its terminals available to other companies for operation, as required.

A Customer Contract Partner can operate its terminals or those which have been made available by another company.

The sales and service terminals of customer contract partners can be operated by agencies.

The agency sells entitlements to customers and provides services on behalf of, and in the name of, customer contract partners.

An agency can distribute entitlements for several customer contract partners.

5.1.4.1 Customer Contract Partner

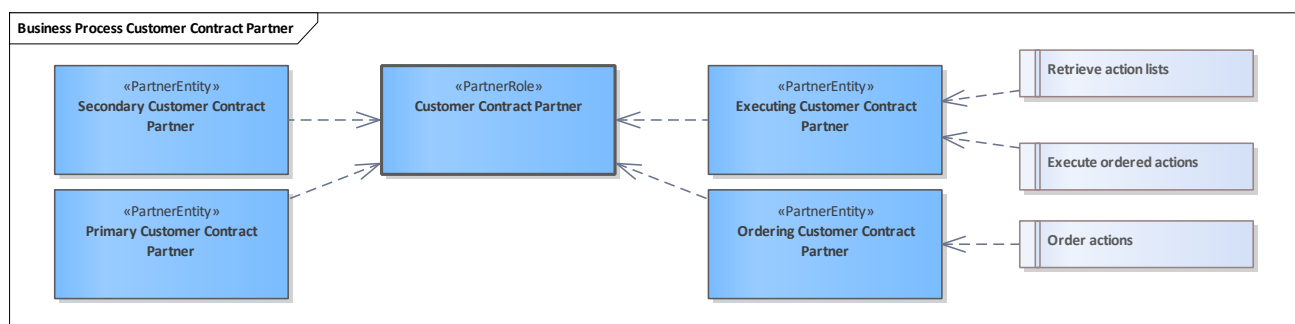


Figure 8: Customer Contract Partner

5.1.4.2 Executing Customer Contract Partner

This role is a sub-role of the [Customer Contract Partner](#). As an executing customer contract partner, terminals are operated that can perform actions with user media. These actions were previously ordered either by the [Customer](#) or by the [Ordering Customer Contract Partner](#) in the [Product Owner Action Management](#).

The executing customer contract partner receives access to these actions, which are tied to the customer's user medium (via its application instance ID). In this way, a user medium can be updated when contacting one of the terminals, e.g. receiving a new entitlement.

Note: the company that holds the sub-role of the executing customer contract partner can also be the ordering customer contract partner.

5.1.4.3 Ordering Customer Contract Partner

This role is a sub-role of the [Customer Contract Partner](#). As the ordering customer contract partner, subsequent actions with user media are ordered in the central action management of the product owner. These actions were previously initiated either by the [customer](#), e.g. via a web portal, or by the ordering customer contract partner itself.

The [Ordering Customer Contract Partner](#) informs the customer

- at which terminals its order may be executed,
- about the earliest time its order may be executed, and
- about the applicable terms and conditions.

Accepting an order from a customer constitutes the conclusion of a contract. Potential receivables are independent of the actual technical execution of the ordered action(s). The Ordering CCP provides the customer with a receipt.

The [executing customer contract partner](#) receives access to these actions, which are tied to the customer's user medium (via its application instance ID).

In this way, a user medium can be updated when contacting one of the terminals, e.g. receiving a new entitlement.

Note: the company that holds the sub-role of the ordering customer contract partner can also be the executing customer contract partner.

5.1.4.4 Primary Customer Contract Partner

The *primary* customer contract partner sells tickets or other services to the [Customer](#) and has a direct contractual relationship with the customer. The customer is known via his master data and/or payment data in the primary customer contract partner's system. Otherwise, the description is the same as for the [Customer Contract Partner](#).

5.1.4.5 Secondary Customer Contract Partner

The *secondary* customer contract partner does not have a permanent contract with the customer but serves as a product reseller. It sells tickets or other services to a [Customer](#) who is not registered with his master data or payment data in its system. From the [Product Retailer](#)'s perspective, the customer is anonymous.

Otherwise, the description is the same as for the [Customer Contract Partner](#).

5.1.4.6 Collecting

The [Customer Contract Partner](#) collects the data concerning entitlements issued from its terminals and sends the data to the [Product Owner](#) for [Data Clearing](#) purposes.

5.1.4.7 Execute ordered actions

Task of the [Executing Customer Contract Partner](#) which provides its terminals with information about actions to be executed with certain user media coming from the [Product Owner Action Management](#). These actions have to be executed, if an involved user medium contacts one of these terminals. The [Product Owner Action Management](#) is informed about action execution via dedicated notifications.

5.1.4.8 Order actions

Task of the [Ordering Customer Contract Partner](#) to order actions either forwarded from a customer or due to own needs e.g. the internal contract management. These actions are ordered in the [Product Owner Action Management](#).

5.1.4.9 Retrieve action lists

Task of the [executing customer contract partner](#). To provide its terminals with action lists (these ensure that the terminal can also perform actions offline), these must first be retrieved from the [Product Owner Action Management](#) at regular intervals and distributed to the terminals.

5.1.5 Service Operator

The role of the Service Operator (SO) in (((etiCORE matches the [Service Operator](#) as defined in ISO 24014-1.

The service operator provides transport services to a [Customer](#) who has gained a matching entitlement for using these services.

Within these mobility services, a public transport service operator becomes a Service Provider. The Service Provider delivers mobility services in different areas.

The public transport service provider acquires, regulated in a contractual relationship, the right from the [Scheme Manager](#) to participate in the EFM system.

It acquires the required SAMs in the required configuration, with the required keys and certificates.

Within the framework of security management, it orders the keys and certificates required for the transmission of messages via the (((etiCORE interoperability network (ION).

The transport contract is concluded between the transporting Service Operator and a [Passenger](#) upon entering the means of transport.

The service operator concludes contracts with [Product Owners](#) for the acceptance of products and the payment of services rendered and, with the [Customer Contract Partner](#), for the settlement of the payments determined.

The service operator captures data in its special role, [Recording](#).

In its special role, [Inspection](#), the service operator inspects entitlements due to tariff rules given by the product owner. The service operator may collect the "Penalty Fare Notice" if the inspected entitlement is deemed invalid.

Recording and Inspection attestations are sent to the Product Owner via ION.

The Service Operator monitors its collection and control terminals for compliance with safety regulations.

The service operator regularly requests the current hotlists from the [Hotlist Service](#), ensures their distribution to its terminals and checks all recorded user media against these lists. The service operator blocks entitlements or the entire public transport application if they exist as an entry on the hotlist and reports the physical blocking to the product owner. If necessary, the service operator performs blocking/unblocking requests to the responsible Customer Contract Partner (applications, entitlements and SAMs (hotlist entry only)), or the [Scheme Manager](#) (organisations via service management of VDV-ETS).

5.1.5.1 Inspection

The [Service Operator](#) in the specialised role, inspection, checks the customers for the existence of a valid entitlement when using services that use the product definitions supplied by the [Product Owner](#).

If necessary, it charges the Penalty Fare Notice (German abbreviation: EBE) or triggers the crediting of the same.

Each entitlement inspection results in a message to the product owner, confirming the inspection procedure.

5.1.5.2 Recording

The [Service Operator](#) also has the specialised role - called recording - of gathering service usage records created by the use of these services by the customers in EFM IN-OUT systems with an automated fare collection.

The service operator calculates the fare in the case of on-trip price calculation (using the product definitions supplied by the [Product Owner](#)).

The service operator delivers the recorded data after a specified pre-processing to the responsible product owner.

The Product Owner then distributes these records to the appropriate [Customer Contract Partner](#).

5.1.6 Product Owner

The role of the Product Owner (PO) in (((etiCORE is basically defined as [Product Owner](#) in ISO 24014-1.

In the German EFM Standard (((etiCORE, the product owner has the following extended definition.

Functions of ownership

The product owner acquires, as regulated in a contractual relationship, the right from the [Scheme Manager](#) to participate in the EFM systems and to register its products there. It receives the necessary identifiers and information for the administration of its tokens, rights and security modules from the [Scheme Manager](#).

The product owner also orders the keys and certificates required for the transmission of messages via the (((etiCORE interoperability network (ION) as part of security management. As part of security management, the product owner authorises [Customer Contract Partners](#) to generate entitlements and use its tokens in their SAMs. The product owner entitles customer contract partners to sell its products.

The product owner develops EFM products from its tariffs for transport services in one or more geographical areas, in which different [Service Operators](#) provide transport services. The product owner determines the necessary contractual terms between itself, the customer contract partners and the service operators.

The product owner defines these EFM products to be issued/purchased as entitlements and makes these products available to the customer contract partners in the form of product definitions (tariff modules) and templates for distribution. The products can be composed of several heterogeneous service types.

The Product Owner makes product definitions (tariff modules) for on-trip price calculation available to service operators in systems with automated fare collection (AFC) and for checking entitlements, as well as to all service operators who accept its products.

Thus, the product owner ensures the necessary contractual terms between itself, the customer contract partners, the service operators and the [Customer](#) for the sale of entitlements and for the use and settlement of services (time and location validity, group of persons, remuneration, commissions, product use and, if applicable, product distribution rules).

Functions of Reporting, Collection and Forwarding

The product owner receives information from the customer contract partner on the proceeds from the sale of entitlements.

For IN-OUT Systems with subsequent pricing (postpaid, prepaid or stored-value payment), it collects, sorts, checks and evaluates the recording data received from the service provider for the use of services and determines the price based on the product definitions.

Thus, the product owner is responsible for mapping and managing the effective costs after the execution of a journey (possibly with more than one trips).

Furthermore, the product owner forwards the recording and control data received from the service operators with the determined price to the responsible customer contract partners.

Moreover, the product owner participates in the [Hotlist Service](#) specified by the scheme manager. If necessary, it submits blocking/unblocking requests to the responsible customer contract partners (for entitlements and SAMs (only for hotlist entry)) or the scheme manager (for organisations via VDV-ETS service-management).

It implements the registration of blocking attestations of entitlements coming from service operators or customer contract partners and their forwarding to the [Primary Customer Contract Partner](#).

It matches the hotlist service with the configuration data for the organisation-specific hotlists concerning the hotlist entries of entitlements on a user medium.

Functions of Clearing

The product owner carries out the necessary checks ([Data Clearing](#)) for the transactions of the entitlements issued for its products and, if necessary, makes its data available as the basis for revenue distribution.

5.1.6.1 Product Owner

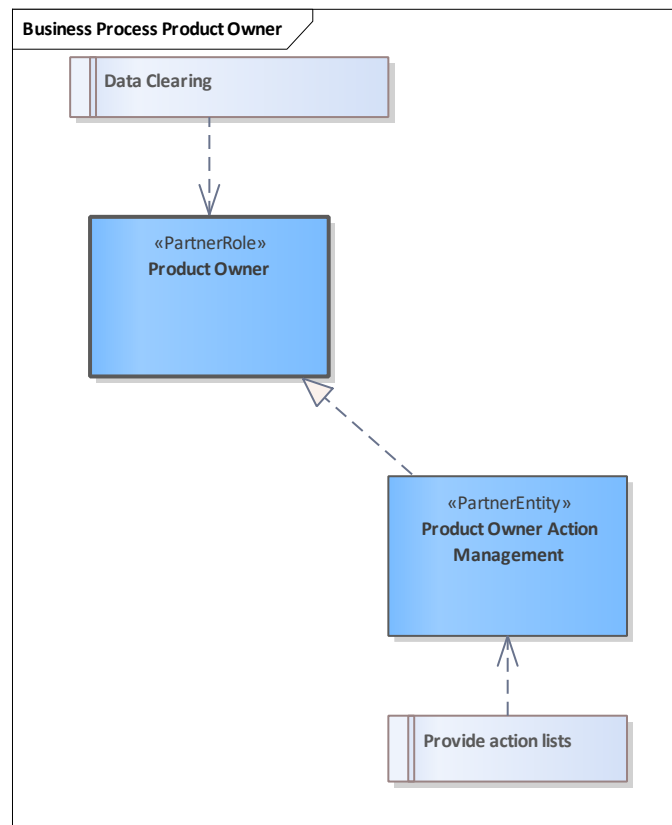


Figure 9: Product Owner

5.1.6.2 Product Owner Action Management

Product Owner Action Management is a sub-role of the [Product Owner](#).

Actions that are to be carried out later on a user medium are managed here.

These actions always refer to entitlements.

Action management is always available to a larger number of transport companies in the form of executing or ordering customer contract partners. Action management is available within a transport association or nationwide. This depends on the validity range of the entitlements that are processed with the help of the actions.

The actions are managed with regard to their life cycle. Normally, actions are ordered, distributed to terminals in the case of possible offline operation and then executed there on contact with the affected user medium.

The notification of execution then leads to the removal of the action from the data inventory.

This notification is also passed on to the [Ordering Customer Contract Partner](#).

The decision on the application of ordered action management is the responsibility of the respective PO, who may allow this option for individual EFM products.

The PO is responsible for the necessary organisational and contractual arrangements with the involved CCPs. The permissibility and the type of action orders (issuing, termination, blocking and unblocking) are to be stipulated as part of the product definition by the PO.

5.1.6.3 Data Clearing

Data clearing uses the data coming from the ISO 24014-1 defined [Collection and Forwarding](#) and is done by the [Product Owner](#) (PO).

In the context of the product owner the data clearing means:

- The validation and monitoring of all incoming messages (event records) and their approval
- The revenue sharing and settlement of all revenues coming from the public transport companies based on the validated data
- For event records based on IN-OUT payment methods, the data clearing performs a rating of these records as the basis for subsequent billing

5.1.6.4 Provide action lists

Task for the [Product Owner Action Management](#). It has to provide actions from its inventory in the form of lists to be fetched by [Executing Customer Contract Partners](#) that distribute them to their suitable terminals. For their own scheduling, the [Executing Customer Contract Partners](#) are informed about the times when these lists are updated.

5.1.7 Scheme Manager

The Scheme Manager is the highest authority of (((etiCORE. It combines the ISO 24014-1 role of [Application Owner](#) with the roles of [Registrar](#) and [Security Manager](#), which specifies the security policies.

It draws up the regulations and monitors them. It exists only once in the system and is uniquely identified.

The Scheme Manager has the following responsibilities and assignments:

- As an [Application Owner](#), it defines the application rules and grants a [Product Owner](#), [Customer Contract Partner](#) and [Service Operator](#) the right to participate in the EFM systems, provides them with the necessary identifiers and grants the right to use the related keys hosted by the [Security Manager](#). It authorises [Customer Contract Partners](#) to issue etiCORE-user medium applications to their [Customers](#). Furthermore, it is responsible for providing information on the application ID (usually identifies a user medium) in the event of queries regarding defective user media. As the application owner, it implements a lifecycle management concerning all application instances.
- Within the framework of the [Security Manager](#), it organises the keys and certificates required for the transmission of messages via the etiCORE interoperability network (ION).
- As [Registrar](#), it registers all participating companies in the participating role. This also includes the assignment of the specified identifications within data objects (e.g. operator ID) necessary for the system.
- The Scheme Manager manages the usage of all identifiers.
- The Scheme Manager certifies EFM system components and interfaces.
- The Scheme Manager monitors compliance with the safety regulations of the EFM system.
- The Scheme Manager guarantees the provision of a [Hotlist Service](#) for the generation and provision of hotlists. This hotlist service accepts hotlist orders for organisations, SAMs, keys, applications and entitlements.
- The Scheme Manager decides on hotlist requests for SAMs (if requesting the SAM owner for hotlisting had no effect) and organisations (via service management) made by other (((etiCORE system participants. It informs the requesting instance of the decision, that is, whether a hotlisting was initiated or not.
- The Scheme Manager organises the further development of the specifications and the security concept of the whole EFM system.

5.1.7.1 Scheme Manager

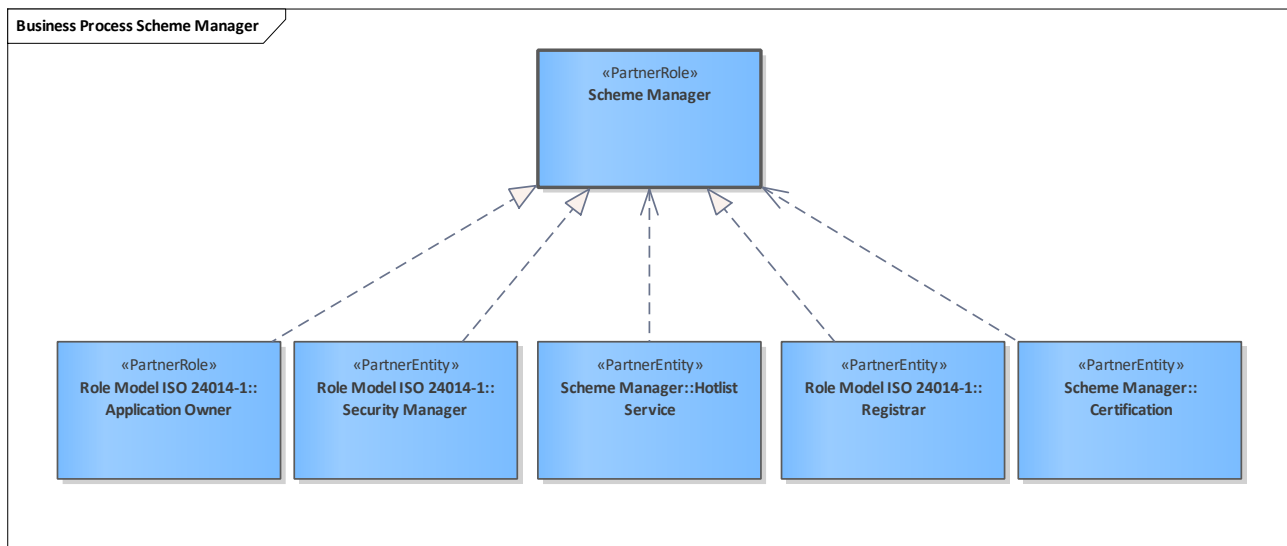


Figure 10: Scheme Manager

5.1.7.2 Certification

The certification instance of the [Scheme Manager](#) is responsible for the specification of the certification procedures required for (((etiCORE.

Furthermore, it is responsible for carrying out the certification of all system components and interfaces and for issuing certificates for the relevant system components of electronic fare management.

It can make use of an external test laboratory for this purpose.

5.1.7.3 Hotlist Service

The Hotlist Service provides lists containing hotlist entries of

- Entitlements
- Applications
- SAMs
- Authentication keys
- Organisations

Entries in the dedicated lists give information as to whether the referenced instance was stolen, compromised, invalid, etc.

The term "hotlist" originates from compromised or stolen credit and debit cards and was adapted to public transport by the norm EN1545.

For all these entities, new orders to add an entry to or remove an entry from those hotlists can be performed.

The Hotlist Service is part of the [Scheme Manager](#).

The necessity to execute the Hotlist Service is part of the security concept of (((etiCORE.

The Hotlist Service instance accepts hotlist orders coming from the scheme manager for organisations, keys (each via service-management) and SAMs (if requesting the SAM owner for hotlisting had no effect), hotlist orders from customer contract partners for application and entitlement blocking.

The Hotlist Service generates the current hotlists from and makes them available for collection by the customer contract partners, service operators and product owners.

The user-media-based hotlists (so-called user medium hotlists, divided into application and entitlement hotlists), is tailored to products used by the customer contract partner and service



operator. The Hotlist Service receives the necessary configuration information from each product owner, who defines which products are accepted by which customer contract partner or service operator.

5.2 Transition ISO 24014 to etiCORE

The overview shows the transition between the ISO 24014 role model (yellow) and the ((etiCORE role model (blue) using a BPMN-based notation with partner roles and partner entities.

Some ISO 24014 roles are combined into a customised ((etiCORE role (for example the [Scheme Manager](#) or the [Customer Contract Partner](#)).

Other parts or roles are split, for example, the [Collection and Forwarding](#), which is not assumed as one role but certain tasks for the roles

- [Service Operator](#) (Recording and Forwarding)
- [Customer Contract Partner](#) (Collecting and Forwarding)
- [Product Owner](#) (Collecting and Forwarding)
- [Application Owner](#) (Collecting and Forwarding)

For (technical) collection and forwarding, the Central Routing Engine and the Interoperability Network are employed.

Application Owner, customer and service operator roles are a 1:1 match in ISO 24014-1 and ((etiCORE.

5.2.1 Transition ISO 24014 to etiCORE

See [Transition ISO 24014 to etiCORE](#).

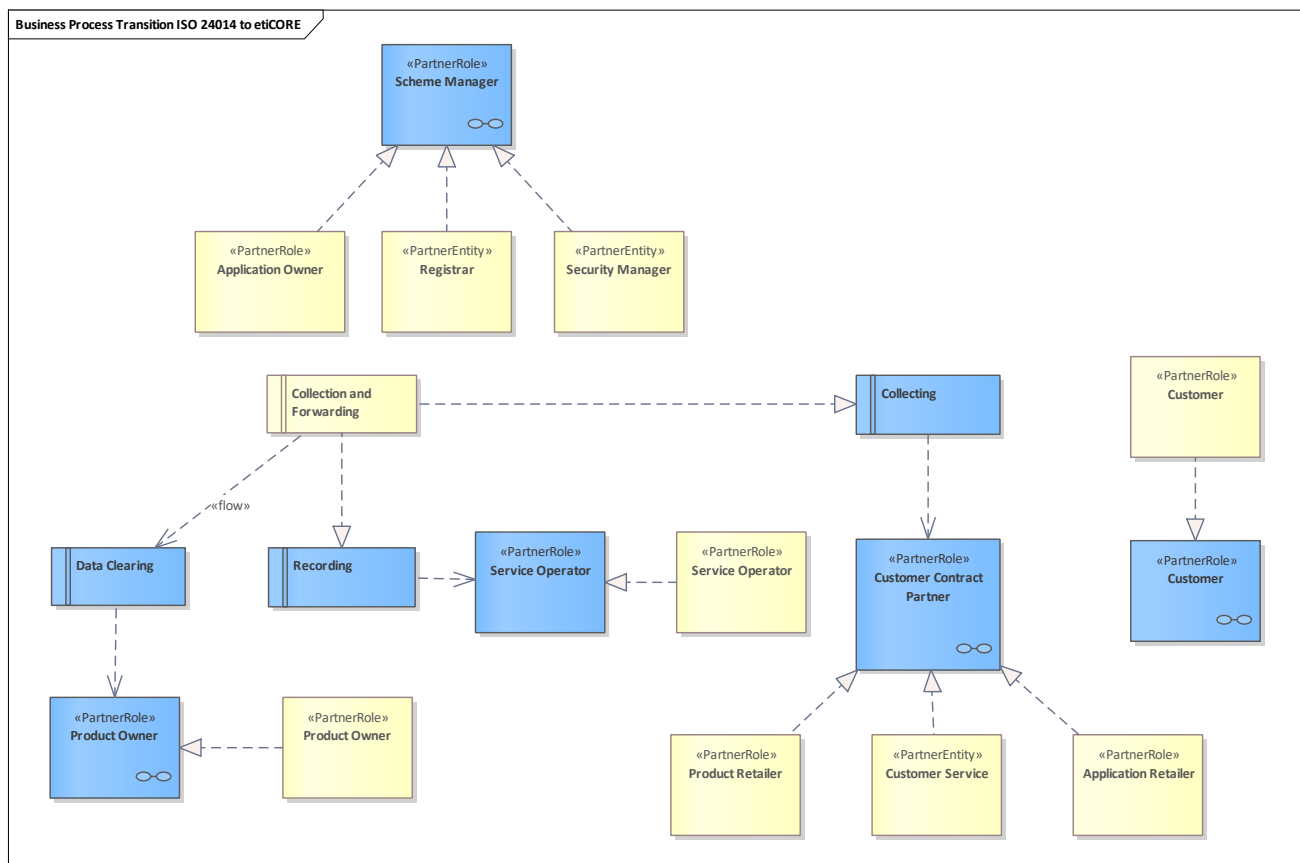


Figure 11: Transition ISO 24014 to etiCORE

5.3 Role Model ISO 24014-1

This chapter contains an introduction to the role model and role interaction defined in ISO 24014-1. See <https://www.iso.org/obp/ui/#iso:std:62354:en>.

The (non-technical) diagrams also show the dependencies of the roles and their main tasks. The different partner roles are described in a very short form here due to copyright reasons. For more information concerning the ISO-based descriptions, please refer to <https://www.iso.org/obp/ui/#iso:std:62354:en> directly.

The overview in [Role Model ISO 24014-1](#) shows a BPMN-based notation of the ISO 24014-1 role model. It employs BPMN partner roles and partner entities and will be used as a foundation diagram to show the differences in the transition to the German EFM Standard's extended role model, which is described later.

It shows the dependencies of the roles and the data flow between the roles concerning the later participants who take each specific role. For each role, a short description is given. The detailed descriptions can be found in the German EFM [Role Model etiCORE](#).

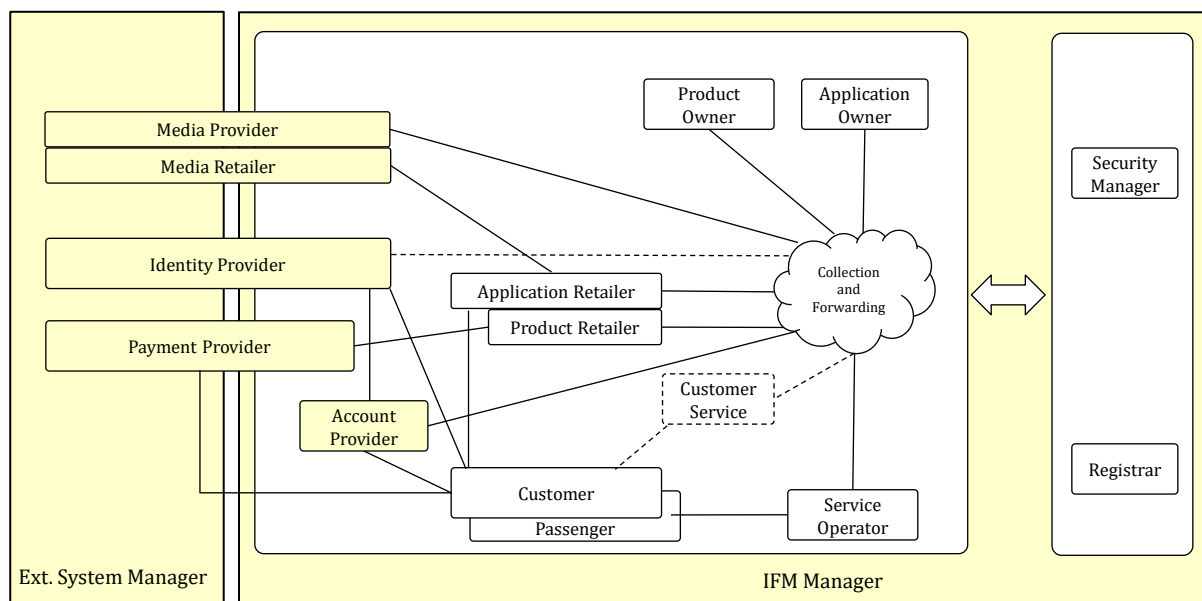


Figure 12: Role Model (latest version)

5.3.1 Role Model ISO 24014-1

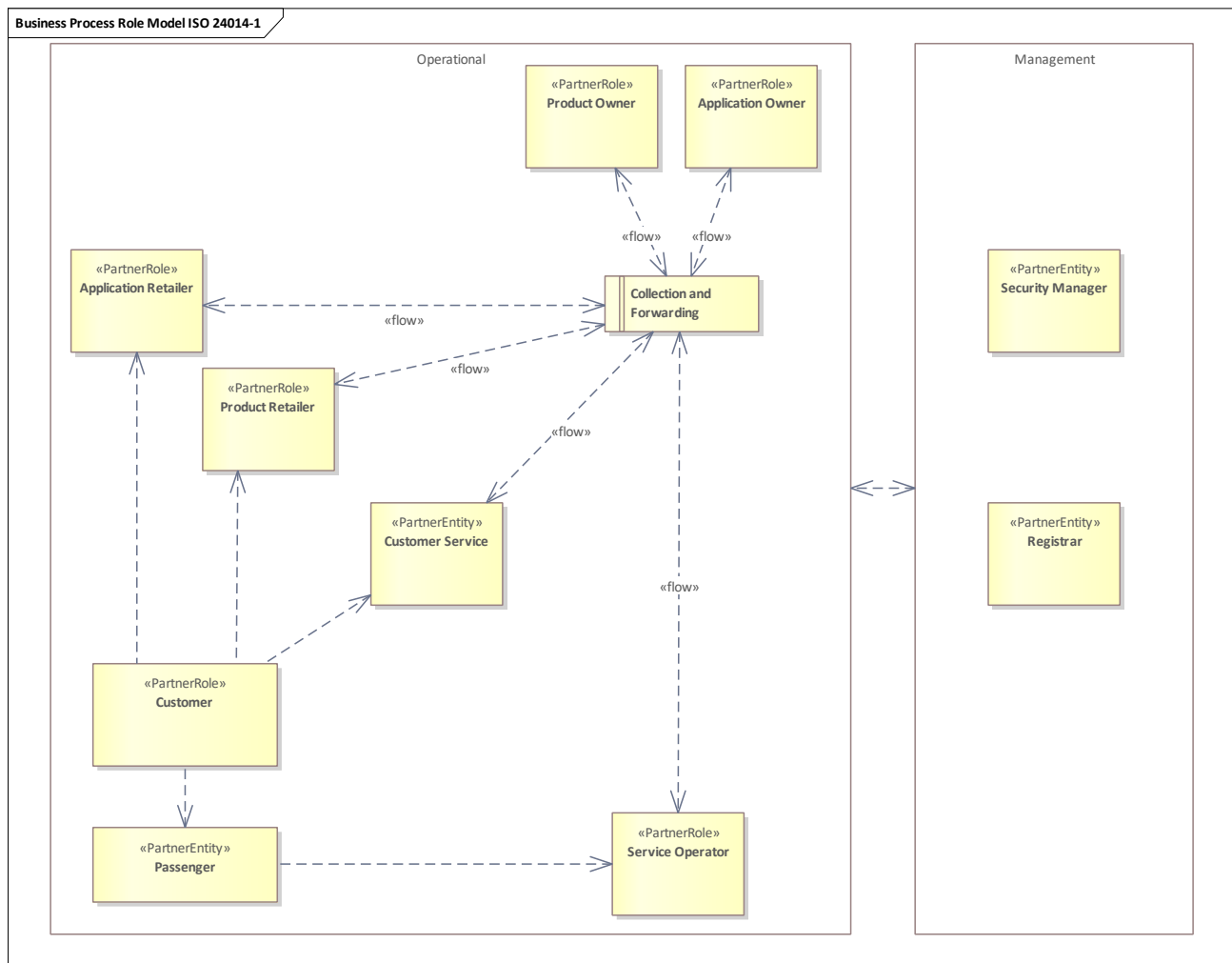


Figure 13: Role Model ISO 24014-1

5.3.2 Application Owner

The application owner (AO) holds the application contract for the use of the application. It authorises the reselling of the application by an [Application Retailer](#) to a [Customer](#).

5.3.3 Product Owner

The product owner (PO) is responsible for its products. It specifies product rules, which consist of pricing rules, usage rules, and commercial rules. It is responsible for clearing and reporting.

5.3.4 Product Retailer

The product retailer sells and withdraws products, collects and refunds value to a [Customer](#), as authorised by a [Product Owner](#). The product retailer is the only financial interface between the customer and the EFM (Electronic Fare Management) system related to products.

5.3.5 Application Retailer

The application retailer sells and withdraws applications, collects and refunds value to a customer, as authorised by an [Application Owner](#).

The application retailer is the only financial interface between the [Customer](#) and the EFM system related to applications.

5.3.6 Service Operator

The service operator provides a service to the [Customer](#) for the use of a public transport product.

5.3.7 Customer Service

Subject to commercial agreements, may provide a service line and other similar facilities including stolen and damaged customer medium replacement and subsequent product (application/entitlement) reinstallation.

5.3.8 Customer

The customer holds a customer medium (user medium) with an application and acquires products in order to use public transport services. In many cases, the customer is also a [Passenger](#). The customer may also hold a personal account and external media or applications which may be used for purposes of the EFM system.

The customer may acquire customer media, applications and products for himself or other passengers. Examples:

- Parents may act as customers and purchase products for their children.
- Companies may act as customers while the employees take the role of passengers.

5.3.9 Passenger

The passenger uses a product to obtain services provided by the [Service Operator](#).

5.3.10 Registrar

The registrar is appointed by the [Scheme Manager](#). He issues unique registration codes for organisations, components, application templates and product templates.

The registrar function also issues rules for generating unique identifiers for applications, products and messages.

5.3.11 Security Manager

The security manager is appointed by the [Scheme Manager](#). He is responsible for:

- establishing the security policy
- certification of organisations, application templates, components and product templates
- auditing of organisations, application templates/applications, components and product templates/products
- monitoring the system
- operation of the EFM system security, e.g. key management.

5.3.12 Collection and Forwarding

Collection and Forwarding is the facilitation of data interchanges of the EFM system. They contain the following functions:

Collection Functions

- Receiving application template from the [Application Owner](#).
- Receiving product template from the [Product Owner](#).
- Receiving data from the [Service Operator](#).
- Receiving data from the [Product Retailer](#).
- Receiving data from the [Application Retailer](#).
- Receiving data from the Media Retailer.
- Receiving data from other collection and forwarding functions.
- Receiving security list data from the [Security Manager](#).
- Receiving clearing reports from the [Product Owner](#).
- Consistency and completeness check of the data collected, on a technical level.
- Receiving the address list of all EFM-roles from the [Registrar](#).

Forwarding Functions

- Forwarding data to other Collection and Forwarding functions.
- Recording data.
- Forwarding data with a corrupt destination address to the security manager.
- Forwarding data to product owners for clearing and reporting.
- Forwarding clearing reports, application templates, product templates and security list data to the product retailer and service operator.
- Forwarding application templates and security list data to the application retailer and service operator.

6 Deployment Variants

The (((etiCORE standard offers a wide range of basic processes and use cases, which in turn are organised into functionality bundles.

Certain requirements have emerged in practice, which group the necessary parts of the (((etiCORE standard into deployment variants.

6.1 Variant 1 - ePayment

This deployment variant enables payment methods on a user medium with an application (e.g. a chip card). This payment functionality can be used to purchase tickets without cash.

6.2 Variant 2 Electronic Tickets and Payment Methods

{Pkg.Notes}

6.2.1 Variant 2a - Electronic Ticket with Subscription

This deployment variant allows electronic tickets to be stored on a user medium with an application (e.g. a chip card). These tickets are typically purchased as part of a subscription.

6.2.2 Variant 2b-1 - Electronic Ticket and ABPM

This deployment variant enables electronic tickets to be stored on a user medium with an application (e.g. a chip card). Additionally, account-based payment methods (ABPM) can be used to purchase these tickets, which are also stored on the same medium.

6.2.3 Variant 2b-2 - Electronic Ticket with SVPM

This deployment variant enables electronic tickets to be stored on a user medium with an application (e.g. a chip card). Additionally, stored-value payment methods (SVPM) can be used to purchase these tickets, which are also stored on the same medium.

6.3 Variant 3 - IN-OUT Systems

This deployment variant enables electronic payment methods such as ABPM or SVPM to be employed as IN-OUT-payment-methods. The trip or journey is composed and debited via check-in and check-out procedures. For SVPM, the debit is made directly; for ABPM the debit can be made after a downstream best-price calculation.

6.4 Variant for D-Ticket

{Pkg.Notes}

6.4.1 Variant for D-Ticket full

This deployment variant provides the German-wide interoperable D-Ticket, which is an enhancement of [2a - Electronic Ticket with Subscription](#). The CCP side enables the D-Ticket to be issued on a user medium with application (e.g. chip card) and as a static entitlement as well. The SO side must handle both ticket variants as well as the PO side.

6.4.2 Variant for D-Ticket as Electronic Ticket only

This deployment variant provides the German-wide interoperable D-Ticket, which is an enhancement of [2a - Electronic Ticket with Subscription](#). The CCP side enables the D-Ticket to be issued on a user medium with application (e.g. chip card) only. The SO and PO side must handle both ticket variants (electronic ticket and static entitlement).

6.4.3 Variant for D-Ticket as Static Entitlement only

This deployment variant provides the German-wide interoperable D-Ticket. The CCP side enables the D-Ticket to be issued as static entitlement only. The SO and PO side must handle both ticket variants (electronic ticket and static entitlement). This deployment variant assumes that all CCPs involved work with static entitlements only. For this reason, the functionality bundle for the action management is not an option.

6.5 Variant for Action Execution

This deployment variant provides the combination of an SO with its system components together with the action execution functionality of an executing CCP. The SO uses the standard basic functionality bundles and the functionality bundle for electronic tickets. Only the portion for the execution of ordered actions is used as CCP functionality. The PO must support the basic functionality bundle as well as the electronic tickets bundle and the action management.

7 Component Functionality Bundles

This chapter lists the functionality bundles.

7.1 Basic Functionality Bundles

Basic functionality bundles that must be implemented by the system components depending on their composition.

These functionality bundles are tailored so that no unnecessary functionality arises in higher-level deployment variants.

7.1.1 Terminals

Basic functionality bundles for terminals.

7.1.1.1 Basic Bundle Terminal - Foundation

Basic functionality bundle that covers the use cases which have to be implemented for all terminals.

7.1.1.2 Basic Bundle Terminal - UM with Application

Functionality bundle that covers all use cases forming the basis for handling user media with an application (e.g. chip cards).

7.1.1.3 Basic Bundle Terminal - Extended Logging

This optional functionality bundle allows the extended logging for an application or an entitlement. The extended logging can also be used for static entitlements.

7.1.1.4 Basic Bundle CCP-Terminal - Foundation

Functionality bundle that covers CCP terminal use cases forming the basis for handling sales, payment methods and CICO.

7.1.1.5 Basic Bundle CCP-Terminal - UM with Application

Functionality bundle that covers use cases forming the basis for handling sales, payment methods, etc. when using user media with an application (e.g. chip cards).

7.1.1.6 Basic Bundle CCP-Terminal - UM in Customer Center

This functionality bundle includes use cases that are normally only carried out in the customer centre.

7.1.1.7 Basic Bundle CCP-Terminal - UM with Customer Data

This optional functionality bundle includes use cases that manage extended customer data on the user medium.

7.1.1.8 Basic Bundle CCP-Terminal - UM with Password

This optional functionality bundle covers all use cases for the password or PIN handling on a user medium with application. Inside this bundle, the use cases are mandatory.

Note: This functionality bundle depends on [Basic Bundle CCP-Terminal - UM with Customer Data](#).

7.1.1.9 Executing CCP

Basic functionality bundles that are relevant when implementing a terminal of an executing CCP.

7.1.1.10 Basic Bundle SO-Terminal

Functionality bundle that covers SO terminal use cases forming the basis for handling inspection, etc.

7.1.2 Back-Office Systems

Basic functionality bundles for back-office systems.

7.1.2.1 Basic Bundle Back-Office System

This functionality bundle contains use cases that all back-office systems of the CCP, SO and PO must implement.

7.1.2.2 Basic Bundle Terminal Operator System - Foundation

This functionality bundle contains use cases that all back-office systems of CCP and SO must implement. CCP and SO are terminal operators whose systems interact with the respective terminals.

7.1.2.3 Basic Bundle Terminal Operator System - Extended Logging

This optional functionality bundle allows the processing of extended logging notifications for an application or an entitlement. The extended logging can also be used for static entitlements.

7.1.2.4 Basic Bundle Terminal Operator System - UM with Application

Functionality bundle that covers all the basic use cases required for a terminal operator system that works with user media with an application.

7.1.2.5 Basic Bundle CCP-System - Foundation

Functionality bundle that covers all the basic use cases required for the CCP back-office system, regardless of the user media employed.

7.1.2.6 Basic Bundle CCP-System - UM with Application

Functionality bundle that covers all the basic use cases required for the CCP back-office system when employing user media with an application (e.g. chip cards).

7.1.2.7 Basic Bundle CCP-System - UM in Customer Center

This functionality bundle includes use cases that are normally only carried out in the customer centre.

7.1.2.8 Basic Bundle CCP-System - UM with Customer Data

This optional functionality bundle includes use cases that manage extended customer data on the user medium in interaction between the back-office system with the terminal.

7.1.2.9 Basic Bundle CCP-System - Extension for UM with Application

Functionality bundle that covers all the basic use cases required for the CCP back-office system when employing user media with an application (e.g. chip cards) and extended functionality.

7.1.2.10 Executing CCP

Basic functionality bundles that are relevant when implementing a back-office system of an executing CCP.

7.1.2.11 SO

Basic functionality bundles suitable for a SO back-office system.

7.1.2.12 Basic Bundle PO-System

Functionality bundle that covers all the basic use cases required for the PO back-office system.

7.2 Electronic Ticket

Functionality bundle for electronic tickets on user media with application (e.g. chip cards).

7.2.1 Electronic Ticket Bundle CCP-System

Functionality bundle that covers the use cases for a CCP back-office system with electronic tickets placed on a user medium with application (e.g. chip card).

7.2.2 Electronic Ticket Bundle CCP-Terminal

Functionality bundle that covers the use cases for a CCP terminal with electronic tickets placed on a user medium with application (e.g. chip card).

7.2.3 Electronic Ticket Bundle SO-Terminal

Functionality bundle that covers the use cases for a SO terminal with electronic tickets placed on a user medium with application (e.g. chip card).



7.2.4 Electronic Ticket Bundle SO-System

Functionality bundle that covers the use cases for a SO back-office system with electronic tickets placed on a user medium with application (e.g. chip card).

7.2.5 Electronic Ticket Bundle PO-System

Functionality bundle that covers the use cases for a PO back-office system with electronic tickets placed on a user medium with application (e.g. chip card).

7.3 Account-Based Payment

Functionality bundle for account-based payment methods on user media with application (e.g. chip cards).

7.3.1 Account-Based Payment Bundle CCP-Terminal

Functionality bundle that covers the use cases for a CCP terminal forming the basis for using the account-based payment method.

7.3.2 Account-Based Payment Bundle CCP-System

Functionality bundle that covers the use cases for a CCP back-office system forming the basis for using the account-based payment method.

7.3.3 Account-Based Payment Bundle SO-System

No payment method use cases for a SO System.

7.3.4 Account-Based Payment Bundle SO-Terminal

No payment method use cases for a SO terminal.

7.3.5 Account-Based Payment Bundle PO-System

Functionality bundle that covers the use cases for a PO back-office system forming the basis for using the account-based payment method.

7.4 Stored-Value Payment

Functionality bundle for stored-value payment methods on user media with application (e.g. chip cards).

7.4.1 Stored-Value Payment Bundle CCP-System

Functionality bundle that covers the use cases for a CCP back-office system forming the basis for using the stored-value payment method.

7.4.2 Stored-Value Payment Bundle CCP-Terminal

Functionality bundle that covers the use cases for a CCP terminal forming the basis for using the account-based payment method.

7.4.3 Stored-Value Payment Bundle SO-System

No payment method use cases for a SO system.

7.4.4 Stored-Value Payment Bundle SO-Terminal

No payment method use cases for a SO terminal.

7.4.5 Stored-Value Payment Bundle PO-System

Functionality bundle that covers the use cases for a PO back-office system forming the basis for using the stored-value payment method.

7.5 Sale Electronic Ticket via Account-Based Payment

Functionality bundle for the sale of electronic tickets via account-based payment methods on user media with application (e.g. chip cards).

7.5.1 Sale Electronic Ticket via Account-Based Payment Bundle CCP-Terminal

Functionality bundle that provides CCP terminal use cases for selling or purchasing an electronic ticket to added to the user medium using the account-based payment method, which is also located on the user medium.

7.5.2 Sale Electronic Ticket via Account-Based Payment Bundle CCP-System

Functionality bundle that provides CCP back-office system use cases for selling or purchasing electronic tickets via an account-based payment method on a user medium.

7.5.3 Sale Electronic Ticket via Account-Based Payment Bundle SO-Terminal

No payment method use cases for a SO terminal.

7.5.4 Sale Electronic Ticket via Account-Based Payment Bundle SO-System

No payment method use cases for a SO System.

7.5.5 Sale Electronic Ticket via Account-Based Payment Bundle PO-System

Functionality bundle that provides PO back-office system use cases for selling or purchasing electronic tickets via an account-based payment method on a user medium.

7.6 Sale Electronic Ticket via Stored-Value Payment

Functionality bundle for the sale of electronic tickets via stored-value payment methods on user media with application (e.g. chip cards).

7.6.1 Sale Electronic Ticket via Stored-Value Payment Bundle CCP-Terminal

Functionality bundle that provides CCP terminal use cases for selling or purchasing an electronic ticket to added to the user medium using the stored-value payment method, which is also located on the user medium.

7.6.2 Sale Electronic Ticket via Stored-Value Payment Bundle CCP-System

Functionality bundle that provides CCP back-office system use cases for selling or purchasing electronic tickets via a stored-value payment method on a user medium.

7.6.3 Sale Electronic Ticket via Stored-Value Payment Bundle SO-Terminal

No payment method use cases for a SO terminal.

7.6.4 Sale Electronic Ticket via Stored-Value Payment Bundle SO-System

No payment method use cases for a SO System.

7.6.5 Sale Electronic Ticket via Stored-Value Payment Bundle PO-System

Functionality bundle that provides PO back-office system use cases for selling or purchasing electronic tickets via a stored-value payment method on a user medium.

7.7 IN-OUT-Systems

Functionality bundle for employing stored-value or account-based payment methods for check-in and check-out.

7.7.1 IN-OUT Bundle CCP-Terminal

Functionality bundle that provides CCP terminal use cases for IN-OUT functionality. The term CICO is also used.

7.7.2 IN-OUT Bundle CCP-System

Functionality bundle that provides CCP back-office system use cases for IN-OUT functionality. The term CICO is also used.

7.7.3 IN-OUT Bundle SO-Terminal

Functionality bundle that provides SO terminal use cases for IN-OUT functionality. The term CICO is also used.

7.7.4 IN-OUT Bundle SO-System

Functionality bundle that provides SO back-office system use cases for IN-OUT functionality. The term CICO is also used.

7.7.5 IN-OUT Bundle PO-System

Functionality bundle that provides PO back-office system use cases for IN-OUT functionality. The term CICO is also used.

7.8 Ordered Action Management

A functionality bundle for using action management to remotely order and locally execute actions on a user medium with an application (e.g. a chip card).

7.8.1 Ordered Action Management Bundle Executing CCP-Terminal

Functionality bundle that covers use cases to implement CCP terminal functionality for executing (ordered) actions located in a distributed action list.

7.8.2 Ordered Action Management Bundle Executing-CCP-System

Functionality bundle that covers use cases to implement CCP back-office system functionality for executing (ordered) actions located in a distributed action list.

7.8.3 Ordered Action Management Bundle Ordering-CCP-System

Functionality bundle that covers the use cases to implement the CCP back-office system functionality for working with remote action lists. This could include ordering or cancelling actions, for example.

7.8.4 Ordered Action Management Bundle SO-Terminal

No use cases in SO terminal for ordered action management.

7.8.5 Ordered Action Management Bundle SO-System

No use cases in SO back-office system for ordered action management.

7.8.6 Ordered Action Management Bundle PO-System

Functionality bundle that covers the use cases to implement the PO back-office system functionality for working with action lists. This could include ordering or cancelling actions, provisioning of action lists, distribution of retrieval configuration, etc.

7.9 Static Entitlements

Functionality bundle for employing static entitlements as electronic tickets.

7.9.1 Static Entitlements Bundle CCP-Terminal

Functionality bundle that covers CCP terminal use cases for working with static entitlements.

7.9.2 Static Entitlements Bundle CCP-System

Functionality bundle that covers CCP back-office system use cases for working with static entitlements.

7.9.3 Static Entitlements Bundle SO-Terminal

Functionality bundle that covers SO terminal use cases for working with static entitlements.

7.9.4 Static Entitlements Bundle SO-System

Functionality bundle that covers SO back-office system use cases for working with static entitlements.

7.9.5 Static Entitlements Bundle PO-System

Functionality bundle that covers PO back-office system use cases for working with static entitlements.

7.10 Miscellaneous

Bundle for various functionality.

7.10.1 Miscellaneous Bundle CCP-Terminal

Functionality bundle that covers miscellaneous CCP terminal use cases.

7.10.2 Miscellaneous Bundle CCP-System

Functionality bundle that covers miscellaneous CCP back-office system use cases.

7.10.3 Miscellaneous Bundle SO-Terminal

Functionality bundle that covers miscellaneous SO terminal use cases.

7.10.4 Miscellaneous Bundle SO-System

Functionality bundle that covers miscellaneous SO back-office system use cases.

7.10.5 Miscellaneous Bundle PO-System

Functionality bundle that covers miscellaneous PO back-office system use cases.

8 Entitlement Categories

The following chapter shows the possible categories of entitlements that use the same data structure but implement entities for different purposes.

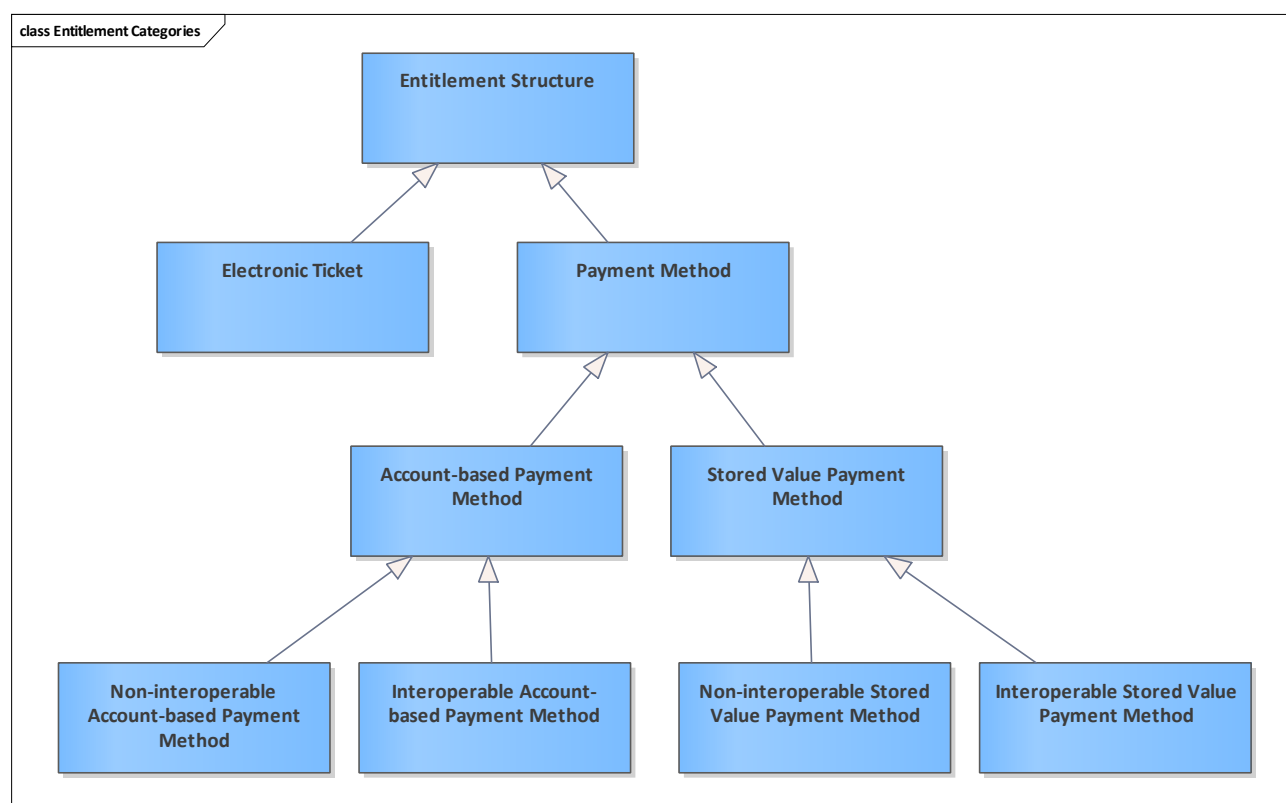


Figure 14: Entitlement Categories

8.1 Entitlement Structure

General structure to hold the data of a derived entitlement. This data structure is the same for

- [Electronic Tickets](#)
- [Payment Methods](#)

8.2 Electronic Ticket

Journey entitlement in the form of an electronic ticket.

8.3 Payment Method

Payment method for public transport services. The payment method can be user-medium-based (stored value payment method) or processed via an account in the background system (account-based payment method).

8.4 Account-based Payment Method

The customer can use this [Payment Method](#) to purchase or use public transport services. Payment is made after notification and, if necessary, collection of corresponding messages in the CCP back-office system.

The customer has an account there, which in turn is linked to a payment method of a payment service provider.

Depending on the tariff or product characteristics, this payment authorisation can be used interoperably or only be permitted for local regions.

8.5 Stored Value Payment Method

The customer can use this [Payment Method](#) to purchase or use public transport services. Payment is made directly by adjusting a credit balance stored on the user medium. Corresponding messages are sent to the CCP back-office system for the purpose of booking. The customer does not necessarily have to be stored there with its data.

Depending on the tariff or product characteristics, this payment authorisation can be used interoperably or only be permitted for local regions.

8.6 Interoperable Account-based Payment Method

[Account-based Payment Method](#) that can be employed interoperably, e.g. nationwide.

8.7 Interoperable Stored Value Payment Method

[Stored Value Payment Method](#) that can be employed interoperably, e.g. nationwide.

8.8 Non-interoperable Account-based Payment Method

[Account-based Payment Method](#) that cannot be used interoperably, e.g. can only be used for the tariff area of the local transport association.

8.9 Non-interoperable Stored Value Payment Method

[Stored Value Payment Method](#) that cannot be used interoperably, e.g. can only be used for the tariff area of the local transport association.

9 Layer 1 and 2 - Workflows and Basic Processes in BPMN

This chapter contains more or less the high-level perspective on the processes in ((etiCORE. All business processes are modelled in BPMN notation which allows a quick overview concerning participants and exchanged information between the participants and systems.

Due to the distributed systems character of ((etiCORE, these diagrams and descriptions give a common understanding in a more global view, which would not be possible with regarding only use cases or basic processes in a standalone way.

((etiCORE does not describe in detail all activities and use cases which are needed for the whole process. Some activities or actions only serve as glue to bind the process steps together and depend strongly on the implementation of the underlying system.

Nevertheless, these activities or actions have to be performed to enable the next activity or action, which might be an ((etiCORE specified one.

Thus, these process steps are high-level descriptions and can be implemented in different systems in different ways.

9.1 Layer 1 - Workflows as BPMN Choreography

This chapter describes workflow scenarios that show basic processes with in relation to other basic processes and the participants involved. The BPMN Choreography is used to show these interactions.

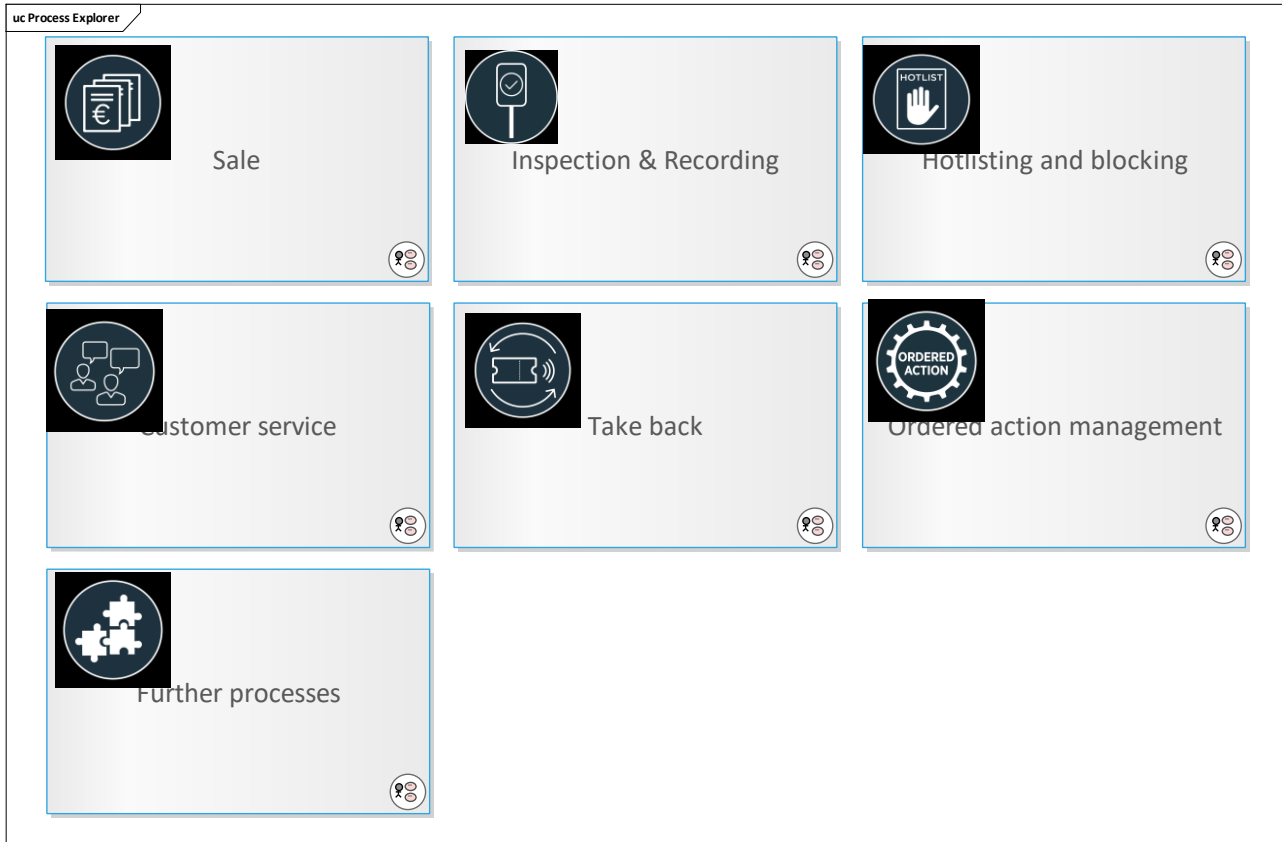


Figure 15: Process Explorer

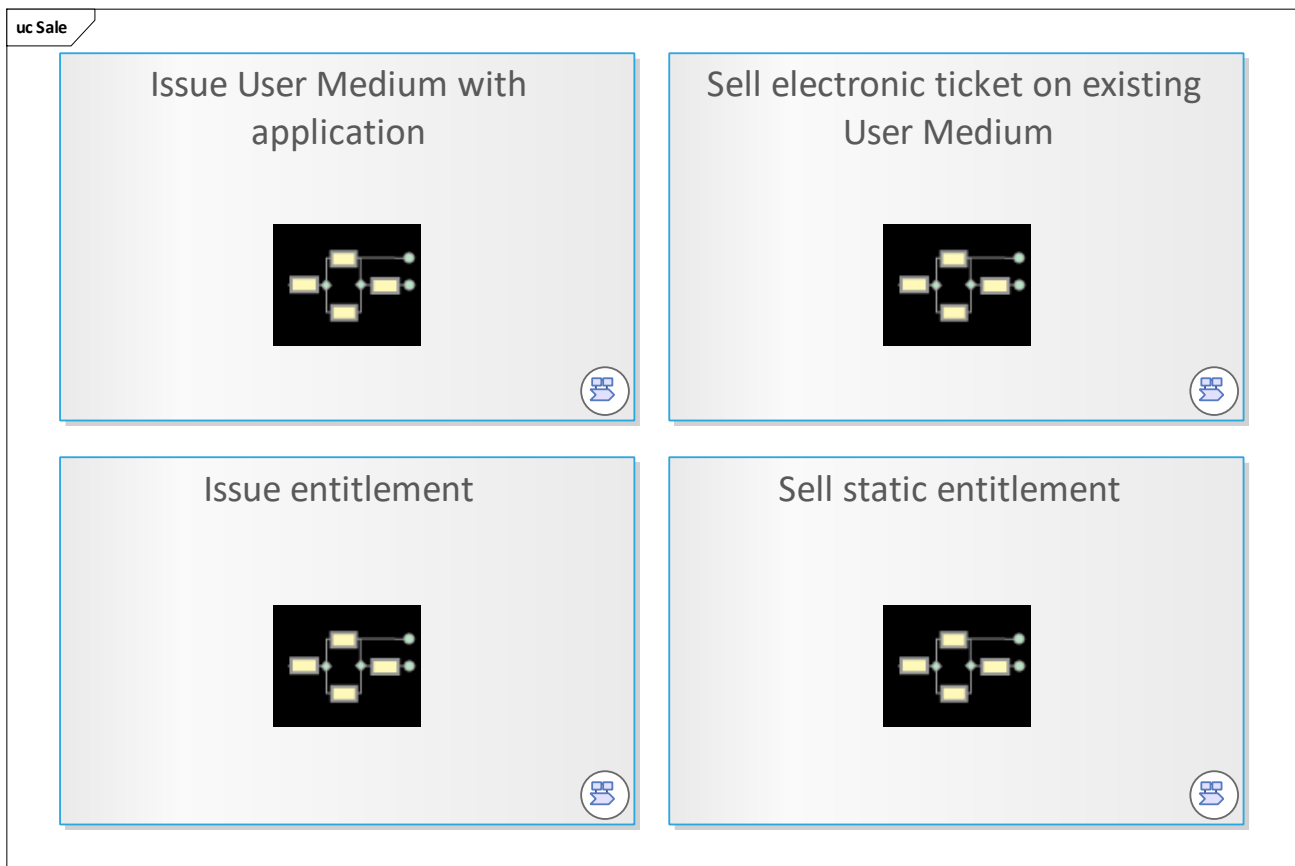


Figure 16: Sale

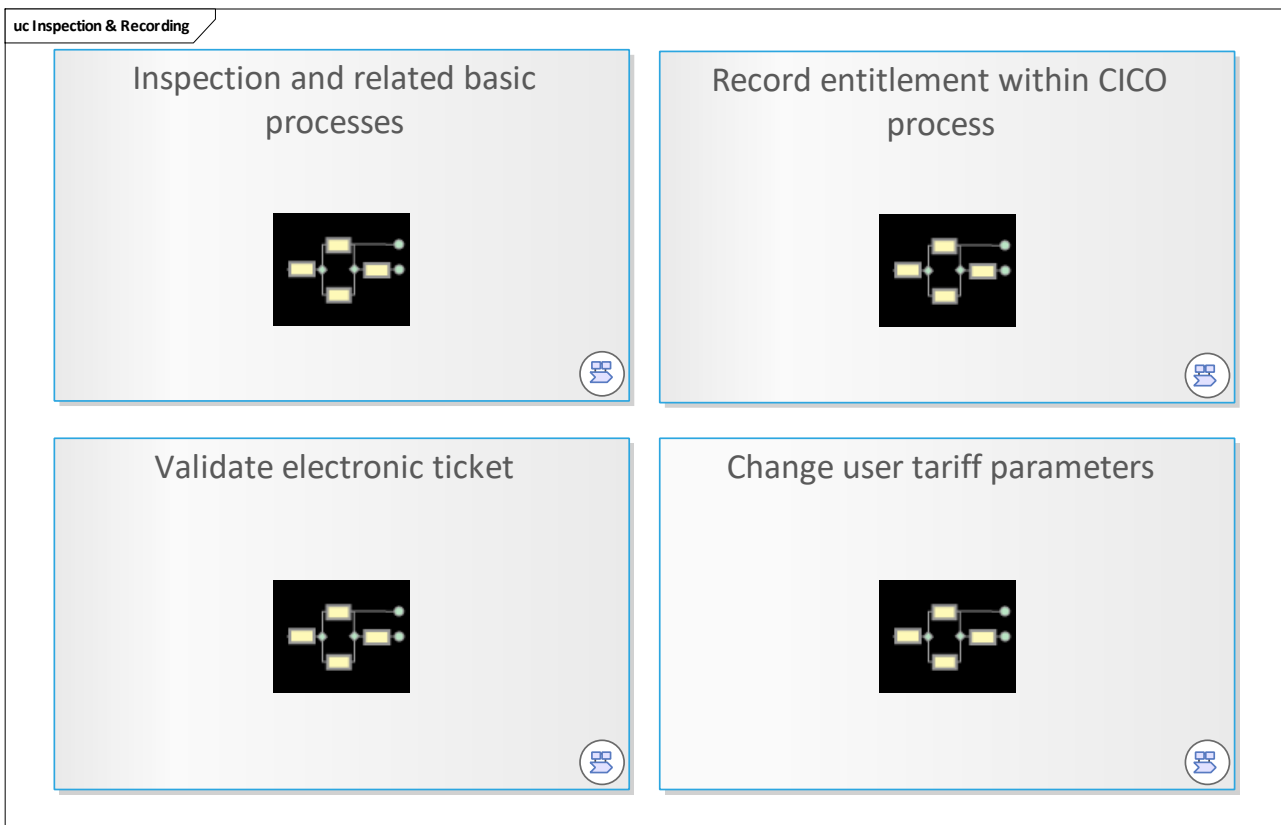


Figure 17: Inspection & Recording

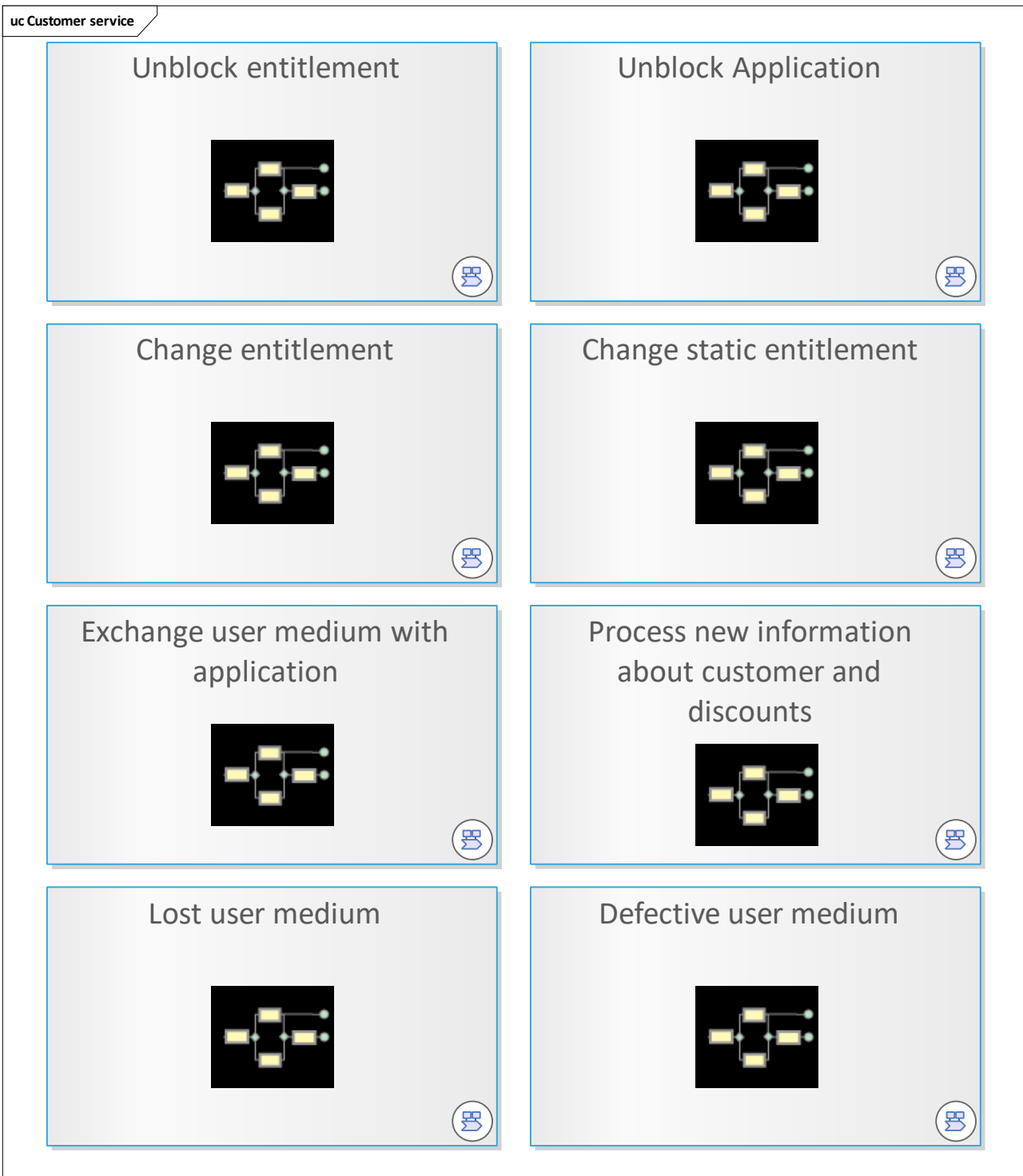
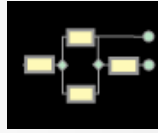


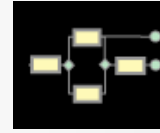
Figure 18: Customer service

uc Hotlisting and blocking

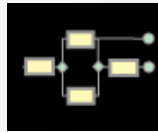
Hotlist and block application



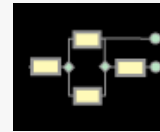
Hotlist and block entitlement



Hotlist a SAM



Hotlist an organisation



Hotlist an authentication key

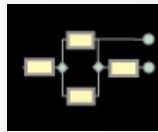
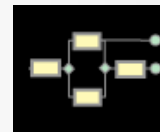
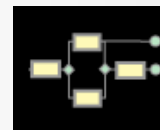
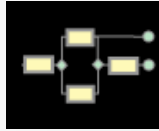
Update hotlist inventories
with external verificationUpdate hotlist inventories
with external verification (SM
checks PO)

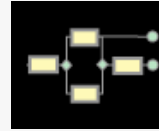
Figure 19: Hotlisting and blocking

uc Take back

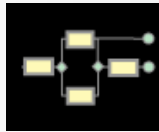
Take back entitlement



Take back application



Take back static entitlement



Reset application

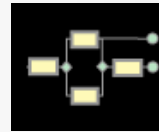


Figure 20: Take back

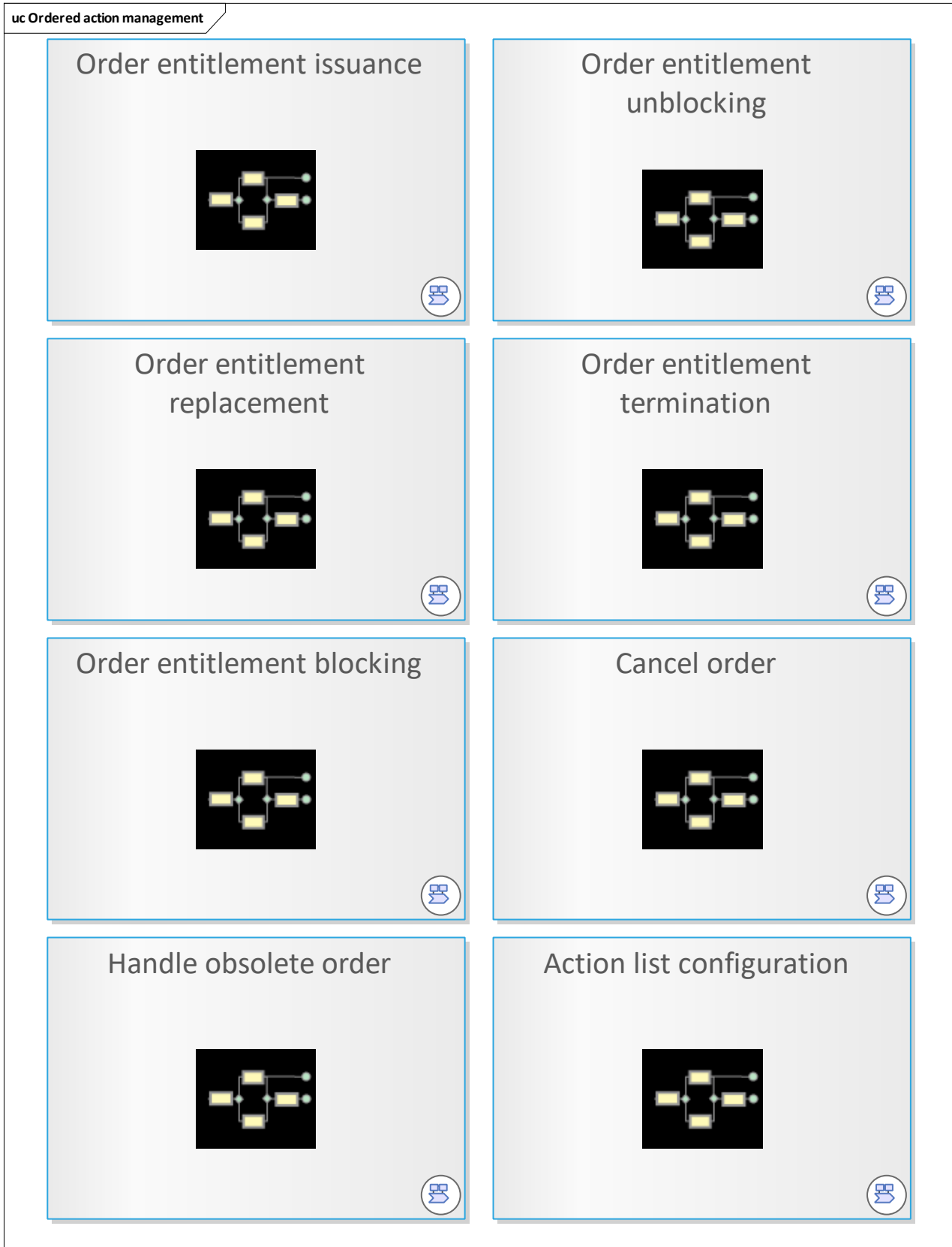


Figure 21: Ordered action management

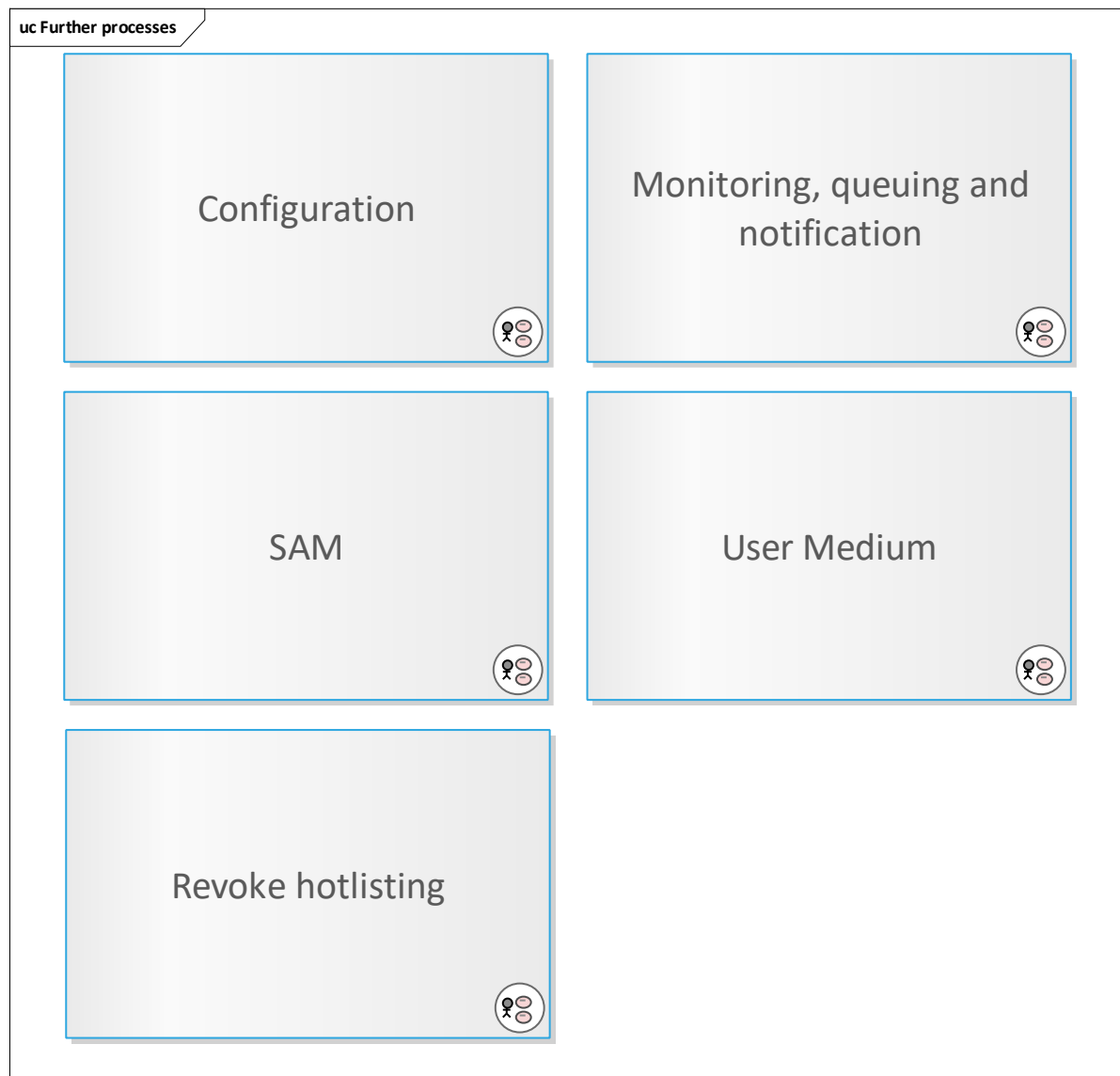


Figure 22: Further processes

9.1.1 Sale

This chapter describes the basic processes in the context of sales in a choreography model.

9.1.2 Issue user medium with application

Shows a high-level view of the basic processes in the context of issuing a user medium with the first configuration steps performed by the card manufacturer and the [Scheme Manager](#) ordered by the [Primary Customer Contract Partner](#). The process ends with issuing the configured and initialised user medium to the [Customer](#).

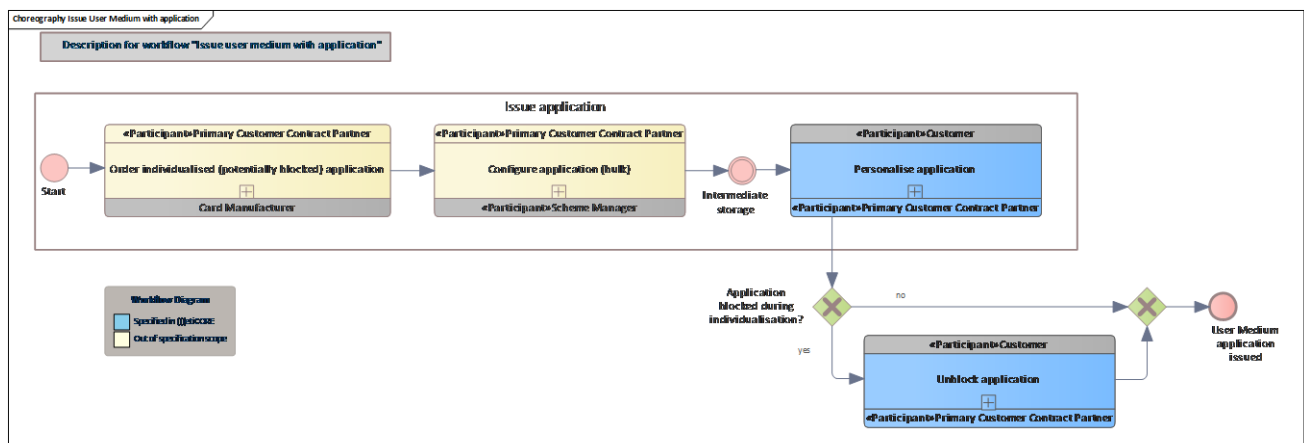


Figure 23: Issue User Medium with application

9.1.2.1 Intermediate storage

At this point, the user medium application may be put into / taken from intermediate storage. Situations where a user medium application may be stocked:

- A new user medium application was configured but is not directly needed
- An existing user medium application was taken back and de-personalised and is now in a configured state

To further protect stock user media, their owner may decide to store them in a blocked state.

9.1.2.2 Order individualised (potentially blocked) application

The user medium application is individualised so that it contains common and individualisation data, which may consist of:

- the application instance ID
- the initial application status
- its private and public keys
- the certificates of the root CAs

For further details, see [etiCORE UM Specification](#).

9.1.2.3 Configure application (bulk)

The user medium application is configured via a configuration script provided by the Media Management System per the [Primary Customer Contract Partner](#) request.

Configuration ensures that the application obtains (among other things) its validity period, certificates, ownership information and symmetric keys.

Described in detail in the context of the Media Management System.

For further details, see [etiCORE UM Specification](#).

This step can be skipped if the user media is configured individually during personalisation. However, since User Media are usually ordered in a configured state, the step is not shown as optional here to keep the diagram simple.

9.1.2.4 Personalise application

The application has to be initialised by the [Primary Customer Contract Partner](#). After this step, the user medium is ready for usage and can be issued to the customer. The initialisation and its exchanged data can be tailored to the requirements of the customer contract partner and may depend on the intended purpose.

See [Initialise user medium with application for customer](#)

9.1.2.5 Unblock application

The user medium owner may decide to block the application before stocking it for later (re-)use. This may provide an additional security aspect for the stored user media. Additionally, the unblocking attestation created during (re-)issuance of the user medium can be used as a standardised, secure and timestamped information set upon which to base the change of possession process.

See [Unblock application](#).

9.1.3 Sell electronic ticket on existing user medium

Shows a high-level view of the basic processes in the context of selling an electronic ticket to be issued to an existing user medium. In this case, the user medium works with an ((etiCORE application. Depending on the payment method, the order of payment and issuing of the electronic ticket varies, as seen in [Sell electronic ticket on existing user medium](#).

The workflow focuses on the sales process and does not consider the common checks done with the user medium that may lead to the blocking of an entitlement/application, etc.

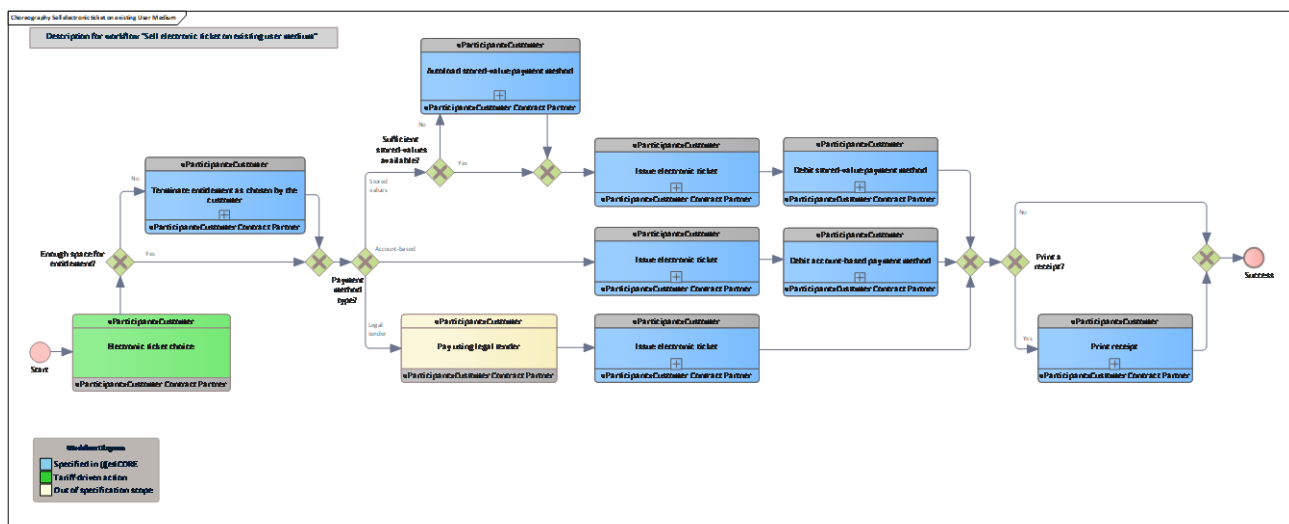


Figure 24: Sell electronic ticket on existing User Medium

9.1.3.1 Electronic ticket choice

The customer chooses his electronic ticket and shapes its properties such as the validity period, location validity, price etc.

During this process, the customer information is read from the ser medium if necessary.



9.1.3.2 Terminate entitlement as chosen by the customer

When all terminated or expired entitlements have been deleted and there still is not enough space for the new entitlement, the customer can choose to terminate entitlements to make room for the new entitlement.

9.1.3.3 Autoload stored-value payment method

If the chosen balance of the stored-value payment method is too low, the stored-value payment method needs to be recharged. This can be done via autoload. A manual, also possible, recharge process is not shown here.

9.1.3.4 Issue electronic ticket

The electronic ticket is issued, but the transaction is not committed until payment is completed. See [Issue entitlement starting in terminal](#).

9.1.3.5 Debit stored-value payment method

The chosen stored-value payment method is debited as a means to pay for the electronic ticket.

9.1.3.6 Issue electronic ticket

The electronic ticket is issued, but the transaction is not committed until payment is completed. See [Issue entitlement starting in terminal](#).

9.1.3.7 Debit account-based payment method

The chosen account-based payment method is debited as a means to pay for the electronic ticket.

9.1.3.8 Pay using legal tender

Payment is made using legal tender.

9.1.3.9 Issue electronic ticket

The electronic ticket is issued and the transaction is committed directly. See [Issue entitlement starting in terminal](#).

9.1.3.10 Print receipt

A receipt is printed if the customer so wishes.

9.1.4 Sell static entitlement

Shows a high-level view of the basic process of selling a static entitlement. The workflow is similar to [Sell electronic ticket on existing user medium](#) but in this case, no chip with an etiCORE application is involved.

The technical structure of the static entitlement is identical to the one written onto a chip of a user medium with the (((etiCORE application.

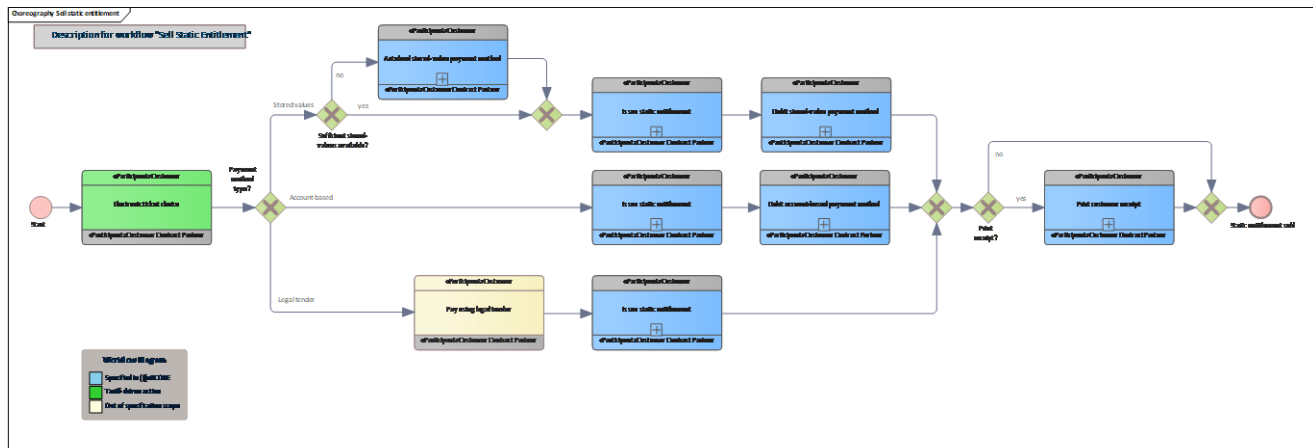


Figure 25: Sell static entitlement

9.1.4.1 Electronic ticket choice

The customer chooses his electronic ticket and shapes its properties such as the validity period, location validity, price etc.

9.1.4.2 Autoload stored-value payment method

If there is an insufficient balance on the stored-value, an autoload option can be triggered before the static entitlement is sold via a stored-value payment method.

9.1.4.3 Issue static entitlement

See [Issue static entitlement](#).

9.1.4.4 Debit stored-value payment method

The chosen stored-value payment method is debited as a means to pay for the static entitlement.

9.1.4.5 Issue static entitlement

See [Issue static entitlement](#).

9.1.4.6 Debit account-based payment method

The chosen account-based payment method is debited as a means to pay for the static entitlement.

9.1.4.7 Pay using legal tender

Payment is made using legal tender.

9.1.4.8 Issue static entitlement

See [Issue static entitlement](#).

9.1.4.9 Print customer receipt

A receipt is printed for the customer.

9.1.5 Issue entitlement

Shows a high-level view of the basic processes in the context of issuing an entitlement to a user medium.

This workflow might be used for bringing a certain payment method to the user medium (a payment method has the same technical structure and is considered as an entitlement in (((etiCORE).

The issuance could also occur when replacing an existing entitlement or during a re-issue due to a defective user medium.

In all these cases, no sales process is involved.

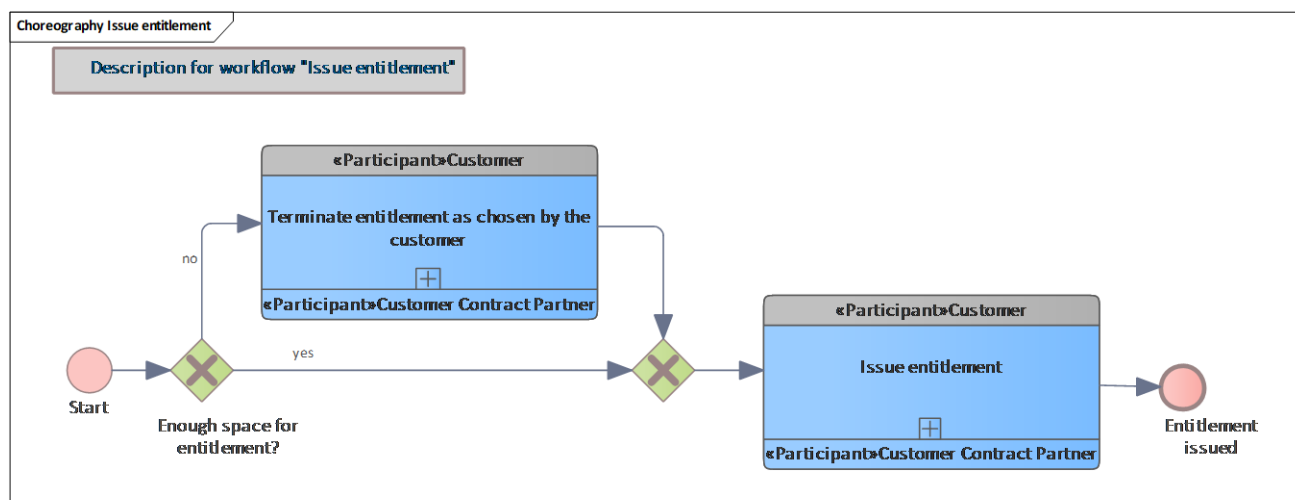


Figure 26: Issue entitlement

9.1.5.1 Terminate entitlement as chosen by the customer

When all terminated or expired entitlements have been deleted and there still is not enough space for the new entitlement, the customer can choose to terminate entitlements to make room for the new entitlement.

9.1.5.2 Issue entitlement

The entitlement is issued to the user medium.

9.1.6 Inspection

This chapter describes the basic process "inspection" in a choreography model.

9.1.7 Inspection and related basic processes

In the choreography model for the inspection, it will be shown which basic processes interact with each other as part of the whole inspection process.

Please note that the inspection process has been divided into two processes (inspect user medium with application and inspect user medium without application) due to technical differences.

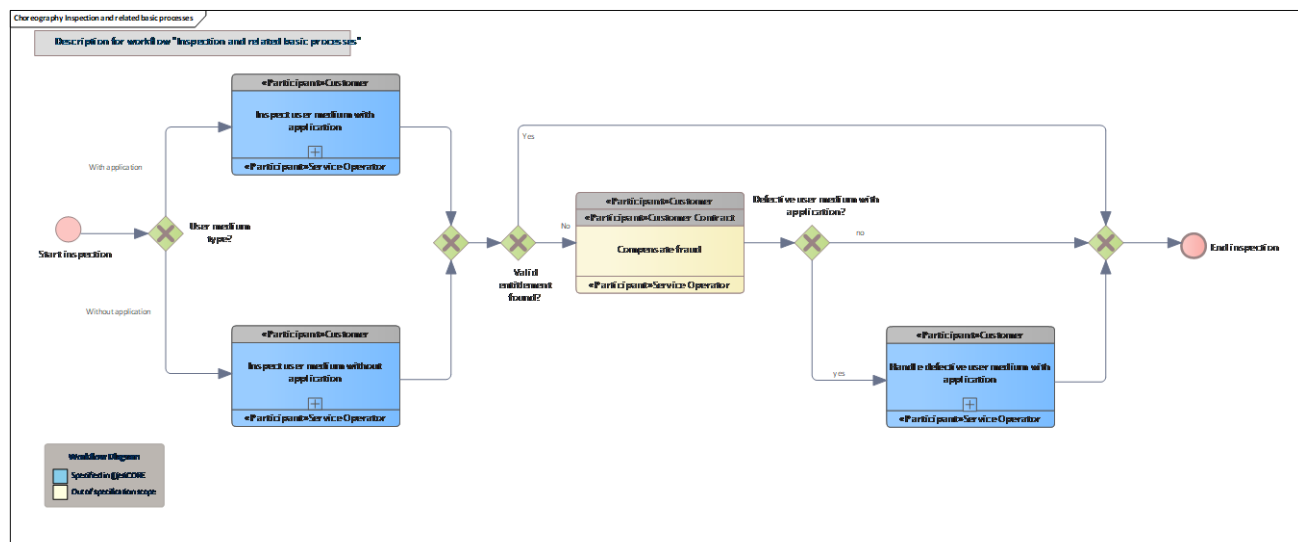


Figure 27: Inspection and related basic processes

9.1.7.1 Inspect user medium with application

The basic process **Inspection** realises the verification of the existence and validity of an entitlement.

In this process, only entitlements **within** the application are inspected.

9.1.7.2 Inspect user medium without application

The basic process **Inspection** realises the verification of the existence and validity of an entitlement.

In this process, only entitlements (static entitlements) **without** application are inspected.

9.1.7.3 Compensate fraud

Register the result of the inspection and process further steps with the customer according to the result (for staff-operated inspection only):

- Obtain confirmation from the customer that all entitlement information may be read out from the user medium
- Read and display the full list of entitlements including expired and blocked entitlements and entitlements from other regions



- The inspector decides whether any of the entitlements read can be recognised as a courtesy case
- If this is not the case, a penalty fare notice will be initiated

Note: As there can be more than one entitlement on a user medium without an application, the above procedure also applies in this case.

9.1.7.4 Handle defective user medium with application

When a user medium with an application is defective, the back-office system is informed. The application must be added to the application hotlist of the central hotlist service.

9.1.8 Hotlisting and blocking

This chapter describes the basic processes "hotlisting and blocking" in a BPMN choreography model.

9.1.9 Hotlist and block application

In the choreography model "Hotlist and blocking application", it will be shown which basic processes interact with each other.

The workflow shows the lifecycle and the result of a hotlist entry of an application. The first step results in a new hotlist entry for the specified application instance.

If the participants update their application hotlists, the hotlist containing the new entry is distributed to the terminals.

If the involved user medium contacts one of these terminals, the application will be blocked and the user medium can no longer be used until it is unblocked by a [Primary Customer Contract Partner](#).

Note: due to delays in the process steps, it takes some time before a new hotlist entry in the terminal can lead to the application instance being blocked. In addition to a possible check at the pCCP, the hotlist entry only becomes active with the next hotlist cycle.

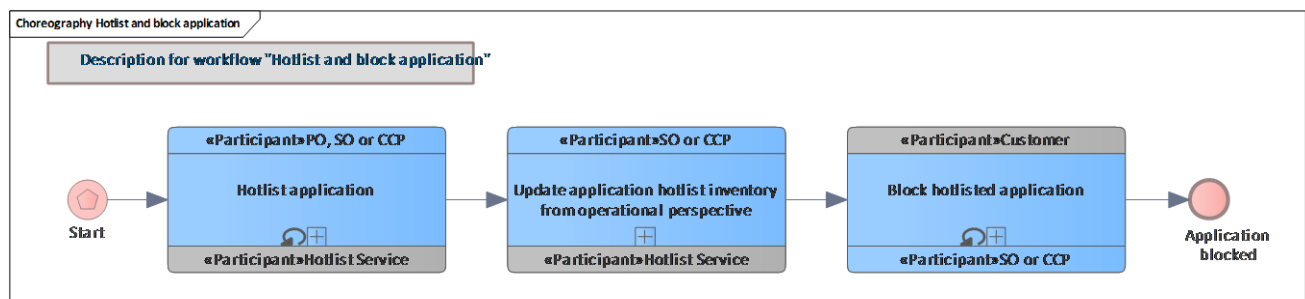


Figure 28: Hotlist and block application

9.1.9.1 Hotlist application

The application can be hotlisted in the following ways:

1. A demand will be submitted by SO, PO or sCCP to the pCCP and the pCCP orders an application hotlisting in the hotlist service
2. the pCCP directly orders an application hotlisting in the hotlist service

The result of the hotlisting order/demand is sent to the requestor.

This application can be a user medium application or a MOTICS app.

9.1.9.2 Update application hotlist inventory from operational perspective

[Service Operators](#) and [Customer Contract Partners](#) must retrieve the application hotlists at contractually defined intervals. The participants update their hotlist inventory with the new data and update the hotlists on their terminals.

To avoid excessively redundant data transfer, the use of an incremental application hotlist is offered in addition to the full hotlist.

9.1.9.3 Start

An application can be hotlisted due to several reasons:

- defective user medium
- lost user medium
- stolen user medium
- exchange of user medium
- monitoring

9.1.9.4 Block hotlisted application

SO or CCP blocks a hotlisted application on the user medium and other partners will be notified about the result of the blocking process.

Note: in the case of a MOTICS application instance, this application instance will not be blocked physically but refused by each terminal.

9.1.10 Hotlist and block entitlement

In the choreography model "Hotlist and blocking entitlement", it will be shown which basic processes interact with each other.

The workflow shows the lifecycle and the result of a hotlist entry of an entitlement. The first step results in a new hotlist entry for the specified entitlement.

If the participants update their entitlement hotlists, the hotlist containing the new entry is distributed to the terminals.

If the involved user medium contacts one of these terminals, the entitlement will be blocked and can no longer be used until it is unblocked by a [Primary Customer Contract Partner](#).

Note: due to delays in the process steps, it takes some time before a new hotlist entry in the terminal can lead to the entitlement being locked. In addition to a possible check at the pCCP, the hotlist entry only becomes active with the next hotlist cycle.

Note: The process of [Hotlist configuration](#) impacts the entitlement hotlist inventories.

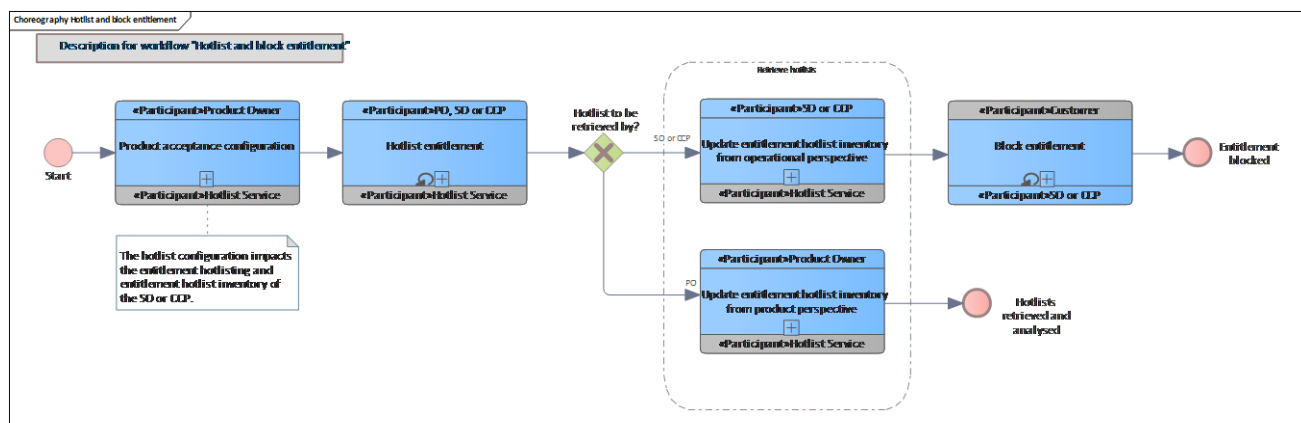


Figure 29: Hotlist and block entitlement

9.1.10.1 Hotlist entitlement

PO, sCCP or SO (e.g. in the case of penalty fare notice, deactivated product etc.) send their hotlisting demand for an entitlement to the pCCP.

This process is the same for all entitlement and product types.

After the pCCP has analysed the demand for hotlisting, it can decide to instruct the hotlist service system to add the entitlement to the hotlist.

Note: if the pCCP itself determines the need for a hotlisting (e.g. due to payment delay), it instructs the hotlist service system directly.

9.1.10.2 Update entitlement hotlist inventory from operational perspective

[Service Operators](#) and [Customer Contract Partners](#) must retrieve the entitlement hotlist within contractually defined intervals. The participants update their hotlist inventory with the new data and update the hotlists on their terminals.

To avoid excessively redundant data transfer, the use of an incremental entitlement hotlist is offered in addition to the full hotlist.

9.1.10.3 Update entitlement hotlist inventory from product perspective

A [Product Owner](#) must retrieve the entitlement hotlist at contractually defined intervals. The [Product Owner](#) updates its hotlist inventory with the new data.

To avoid excessively redundant data transfer, the use of an incremental entitlement hotlist is offered in addition to the total hotlist.

Hotlists are to be analysed from a product perspective for monitoring purposes.

9.1.10.4 Block entitlement

This basic process implements the blocking of an entitlement on a user medium. Blocking is executed when the entitlement is found on the hotlist.

The result of blocking entitlement is sent to the

- PO, after that to pCCP, if the blocking is executed by SO or sCCP
- PO and hotlist service, if the blocking is executed by pCCP.

After that, the pCCP can request the removal of the entitlement from the hotlist, if required.

Note: for entitlements that are hotlisted temporarily the entitlement is refused by the terminal but not physically blocked. The same happens with static entitlements that might have been hotlisted.

9.1.11 Hotlist a SAM

The SAM hotlisting process puts a Secure Application Module (SAM) onto the SAM hotlist. This concerns all objects that were attested by the hotlisted SAM.

The relevant participants have to update their SAM hotlists. The CCP and the SO distribute these hotlists to their terminals.

The next contact with an involved user medium causes the blocking of an entitlement or an application as a result.

Note: due to delays in the process steps, it takes some time before a new SAM hotlist entry in the terminal may affect involved entitlements being blocked. In addition to a possible check at the SAM owner, the hotlist entry only becomes active with the next hotlist cycle.

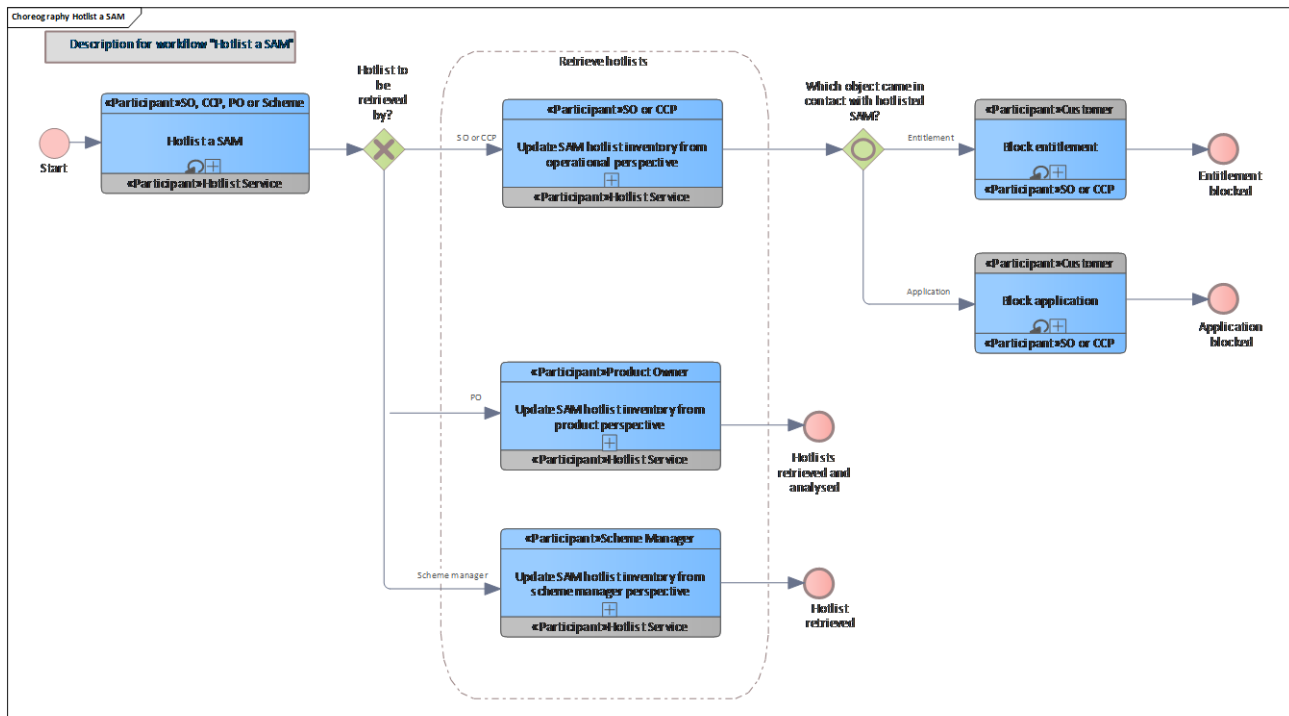


Figure 30: Hotlist a SAM

9.1.11.1 Hotlist a SAM

A SAM needs to be hotlisted.

- A demand will be submitted by PO, SO or CCP to another SAM owner.
 - The SAM owner decides whether to submit a SAM hotlisting order to the hotlist service.
- The result of the hotlisting order/demand is sent to the requestor.

Please note that the [Scheme Manager](#) may, in certain cases such as hotlisted organisations, hotlist SAMs.

9.1.11.2 Update SAM hotlist inventory from operational perspective

[Service Operators](#) and [Customer Contract Partners](#) must retrieve the SAM hotlist at contractually defined intervals. The participants update their hotlist inventory with the new data and update their terminal hotlists.

9.1.11.3 Update SAM hotlist inventory from product perspective

The [Product Owner](#) must retrieve the SAM hotlist at contractually defined intervals. The participant updates its hotlist inventory with the new data.

9.1.11.4 Update SAM hotlist inventory from scheme manager perspective

The [Scheme Manager](#) retrieves the SAM hotlist for monitoring and security purposes. The scheme manager updates its hotlist inventory with the new data.

9.1.11.5 Block entitlement

If the SO or CCP encounters an entitlement that was attested by a hotlisted SAM, the entitlement will be blocked on the user medium and other participants will be informed. See also [Block entitlement](#).

9.1.11.6 Block application

The SO or CCP blocks an application on the user medium if the last transaction on the UM was performed by a hotlisted SAM. Other participants will be notified about the result of the blocking process.

9.1.12 Hotlist an organisation

In the choreography model "Hotlist organisation", it will be shown which basic processes interact with each other.

The workflow shows the lifecycle and the result of a hotlist entry of an organisation. The [Scheme Manager](#) hotlists an organisation. This step results in a new hotlist entry for the specified organisation ID.

If the participants update their organisation hotlist, the hotlist containing the new entry is distributed to the terminals.

All SAMs configured for this organisation may also be hotlisted. If so, the SAM hotlists will be updated (not shown here).

All user media configured for this organisation and/or all entitlements issued by this organisation will be physically blocked, if an involved user medium contacts one of these terminals. Neither the application instance nor entitlement can be used any longer.

Normally, blocked entities due to organisation hotlist entries will not be unblocked again.

Note: due to delays in the process steps, it takes some time before a new organisation hotlist entry or a new SAM hotlist entry in the terminal may affect to involved entitlements or applications being blocked. In addition to a possible check at the SAM owner and the scheme manager, the hotlist entry only becomes active with the next hotlist cycle.

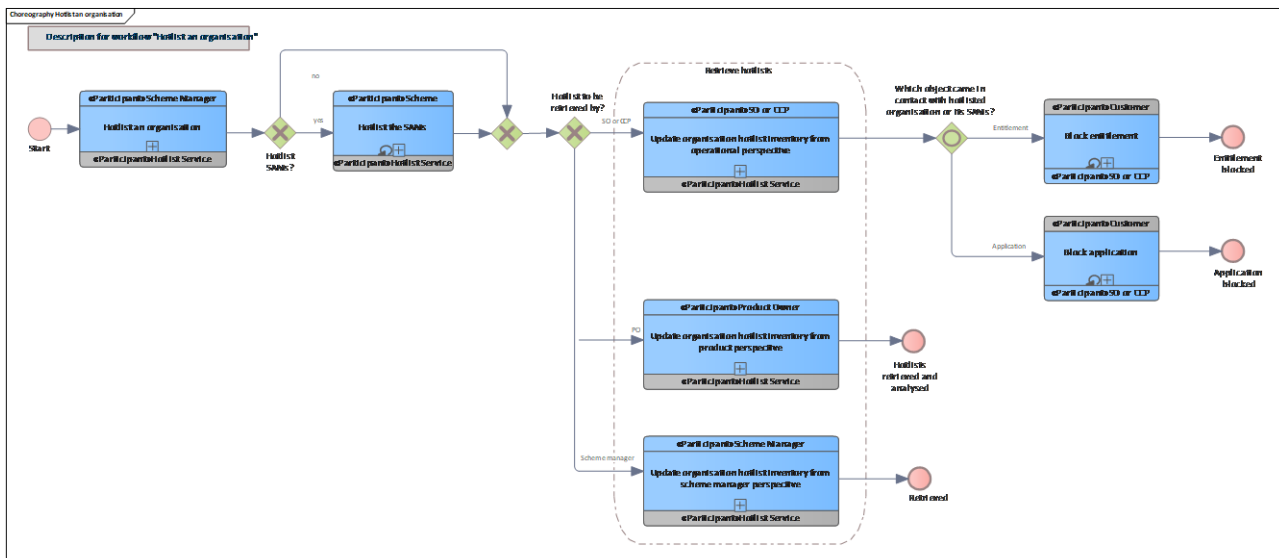


Figure 31: Hotlist an organisation

9.1.12.1 Hotlist an organisation

The organisation is hotlisted. This can be done only by the scheme manager.

9.1.12.2 Hotlist the SAMs

The SAMs of a hotlisted organisation are hotlisted by the scheme manager based on the decisions of those responsible for this at VDV-ETS.

If any SAM of the organisation is hotlisted, the SAM hotlists must also be retrieved. The relevant choreograph models would be

- Update SAM hotlist inventory from an operational perspective
- Update SAM hotlist inventory from a product perspective
- Update SAM hotlist inventory from the scheme manager perspective

9.1.12.3 Update organisation hotlist inventory from operational perspective

Service operators and customer contract partners must retrieve the hotlists of organisations at contractually defined intervals. The participants update their hotlist inventory with the new data and update their terminal hotlists.

9.1.12.4 Update organisation hotlist inventory from product perspective

The [Product Owner](#) must retrieve the hotlists of organisations at contractually defined intervals. The PO then updates its hotlist inventory with the new data.

9.1.12.5 Update organisation hotlist inventory from scheme manager perspective

The [Scheme Manager](#) retrieves the hotlist of organisations for monitoring and security purposes. The participant updates its hotlist inventory with the new data.

9.1.12.6 Block entitlement

If the SO or CCP encounters an entitlement that is owned by a hotlisted organisation, the entitlement will be blocked on the user medium and the partners will be notified.

9.1.12.7 Block application

The SO or CCP blocks an application instance that is owned by a hotlisted organisation on the user medium and other partners will be notified about the result of the blocking process.

9.1.13 Hotlist an authentication key

In the choreography model "Hotlist an authentication key", it will be shown which basic processes interact with each other.

The authentication key is located on all SAMs and each user medium. This symmetric key allows the establishment of a secure session between the user medium and the SAM.

The [Scheme Manager](#) can hotlist an authentication key.

The [Service Operator](#) and [Customer Contract Partner](#) have to update the hotlist for authentication keys and distribute it to their terminals.

Since several generations of authentication keys are located on SAM/user medium, aided by the key version, both can switch to the next authentication key for session establishing if the previous key is hotlisted.

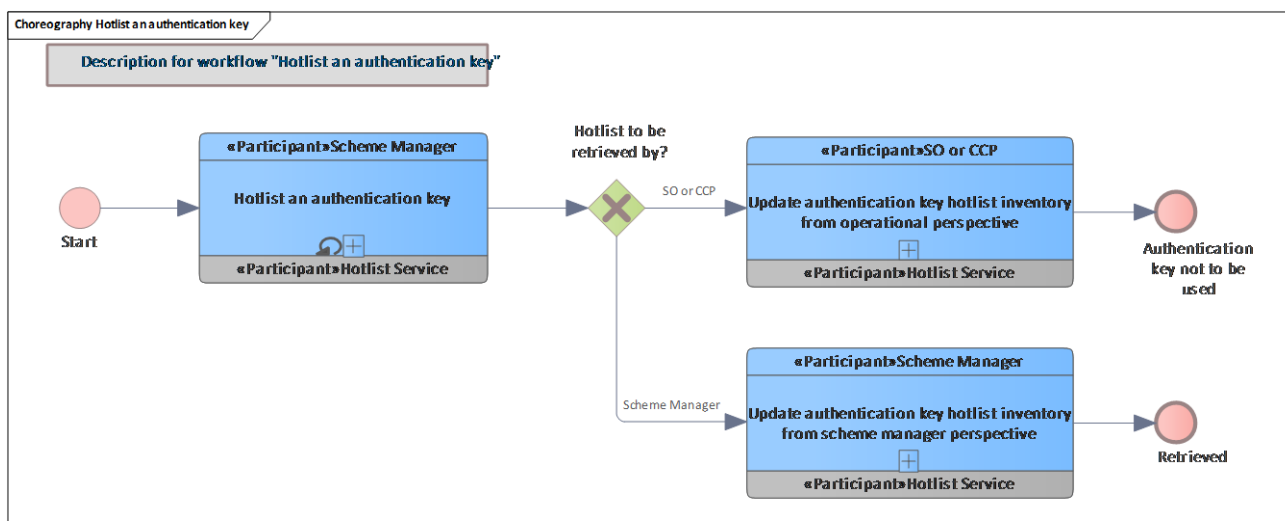


Figure 32: Hotlist an authentication key

9.1.13.1 Hotlist an authentication key

The authentication key is hotlisted. This can be only done by the [Scheme Manager](#).

9.1.13.2 Update authentication key hotlist inventory from scheme manager perspective

The [Scheme Manager](#) retrieves the hotlist of authentication keys for monitoring and security purposes. He updates his hotlist inventory with the new data.

9.1.13.3 Update authentication key hotlist inventory from operational perspective

The [Customer Contract Partner](#) and [Service Operator](#) must retrieve the hotlist of authentication keys at contractually defined intervals. These participants update their hotlist inventory with the new data and update their terminal hotlists.

Hotlisted authentication keys will no longer be used for secured sessions with a user medium in a terminal.

9.1.13.4 Authentication key not to be used

Please note that the authentication key cannot be used in a terminal anymore if it is hotlisted. Please see [Terminal startup procedure](#) for details.

9.1.14 Update hotlist inventories with external verification

This choreography model shows that all participants are obliged to regularly collect the hotlists intended for them and update their data. To verify this, the control authority PO can obtain information about the collection of the lists for its CCPs and SOs. This also applies to the scheme manager.

In addition, the scheme manager can monitor the collection behaviour of the POs, as it receives the information for all participants, including the POs.

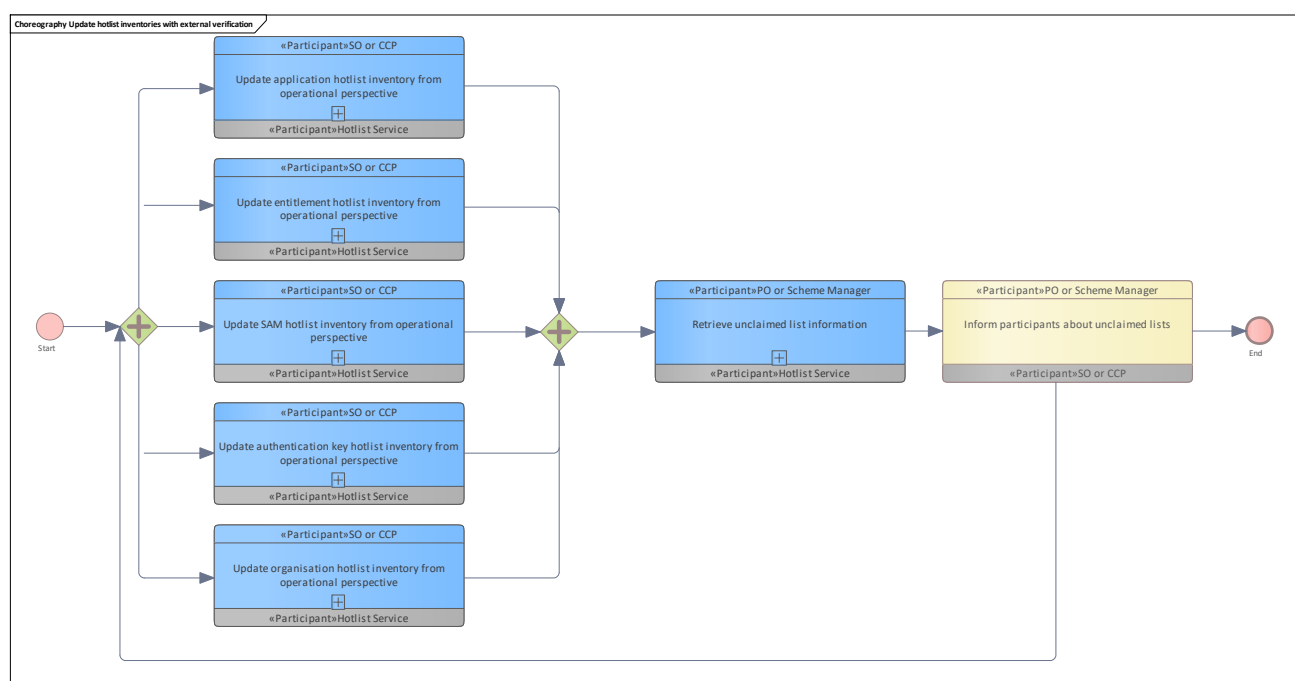


Figure 33: Update hotlist inventories with external verification

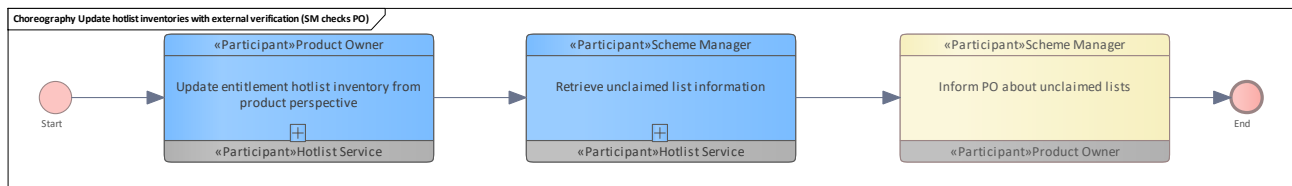


Figure 34: Update hotlist inventories with external verification (SM checks PO)

9.1.14.1 Inform participants about unclaimed lists

The Scheme Manager oder PO informs the CCP or SO that not all intended hotlists have been retrieved.

9.1.14.2 Inform PO about unclaimed lists

The Scheme Manager informs the PO that not all intended hotlists have been retrieved.

9.1.14.3 Retrieve unclaimed list information

See also [Retrieve unclaimed list information](#).

The PO or scheme manager verifies that each CCP or SO regularly retrieves all hotlists intended for it.

9.1.14.4 Retrieve unclaimed list information

See also [Retrieve unclaimed list information](#).

The scheme manager verifies that each PO regularly retrieves all hotlists intended for it.

9.1.15 Revoke hotlisting

This chapter describes the workflows with their basic processes for revocation or removal of hotlist entries.

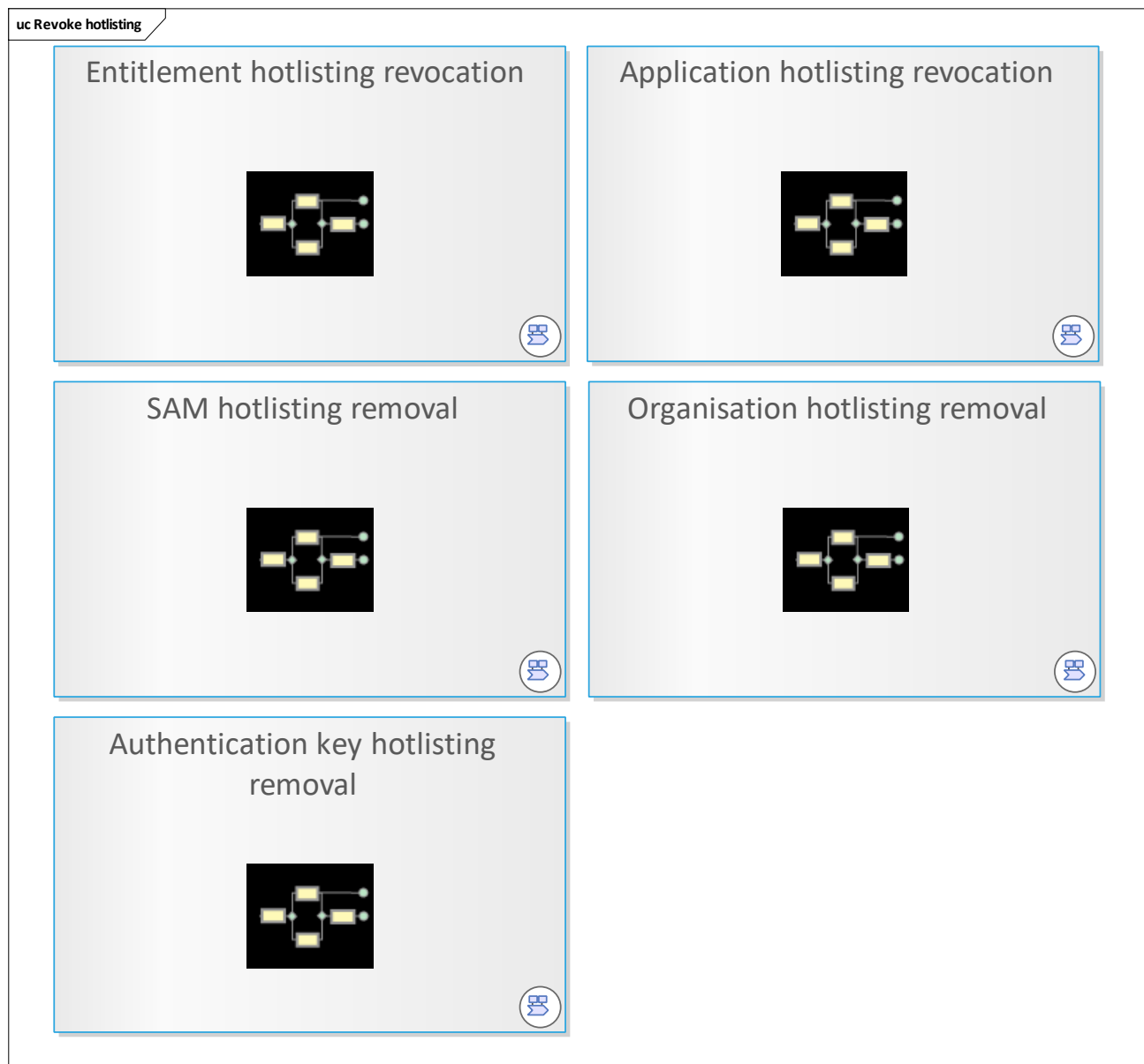


Figure 35: Revoke hotlisting

9.1.16 Application hotlisting revocation

In the choreography model "Application hotlisting revocation", it will be shown which basic processes interact with each other.

In this case, a revocation of an application hotlist entry is performed. This can be initiated by a third party or the [Primary Customer Contract Partner](#) itself.

The result is that the application instance is no longer on the application hotlist.

In order to bring this to effect, [Customer Contract Partner](#) and [Service Operator](#) systems have to update their application hotlists and distribute them to their terminals.

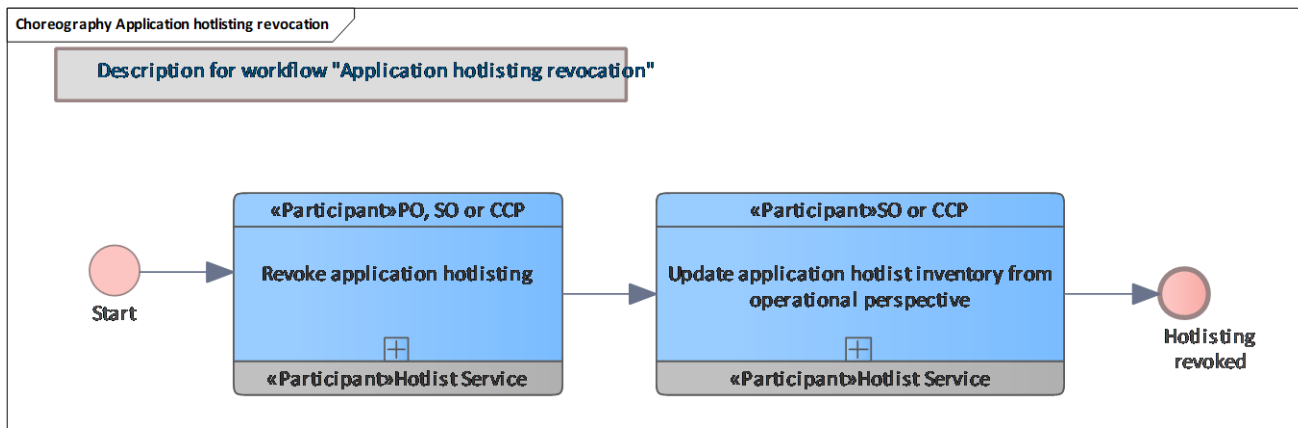


Figure 36: Application hotlisting revocation

9.1.16.1 Revoke application hotlisting

The SO, PO or CCP might want to revoke its hotlisting demand due to e.g. wrongly submitted hotlisting demands.

If the SO, PO or CCP requests a hotlisting demand revocation for an application, the pCCP checks the revocation demand and will remove the initial demand from the list for an application.

The pCCP keeps track of the hotlisting demands for the applications for which it is responsible. If all demands for a particular application have been revoked, the pCCP will request a hotlist entry removal from the hotlist service system.

9.1.16.2 Update application hotlist inventory from operational perspective

[Service operators](#) and [customer contract partners](#) must retrieve the newest application hotlist, update their hotlist inventory with the new data and update the hotlists on their terminals. Without these updates, an outdated hotlist entry might unintentionally block the involved application.

9.1.17 Entitlement hotlisting revocation

In the choreography model "Entitlement hotlisting revocation", it will be shown which basic processes interact with each other.

In this case, a revocation of an entitlement hotlist entry is performed. This can be initiated by a third party or the [Primary Customer Contract Partner](#) itself.

The result is that the entitlement is no longer on the entitlement hotlist.

To bring this to effect, [Customer Contract Partner](#) and [Service Operator](#) systems have to update their entitlement hotlists and distribute them to their terminals.

The [Product Owner](#) must update its hotlist to avoid monitoring issues and to be able to provide correct information about entitlements.

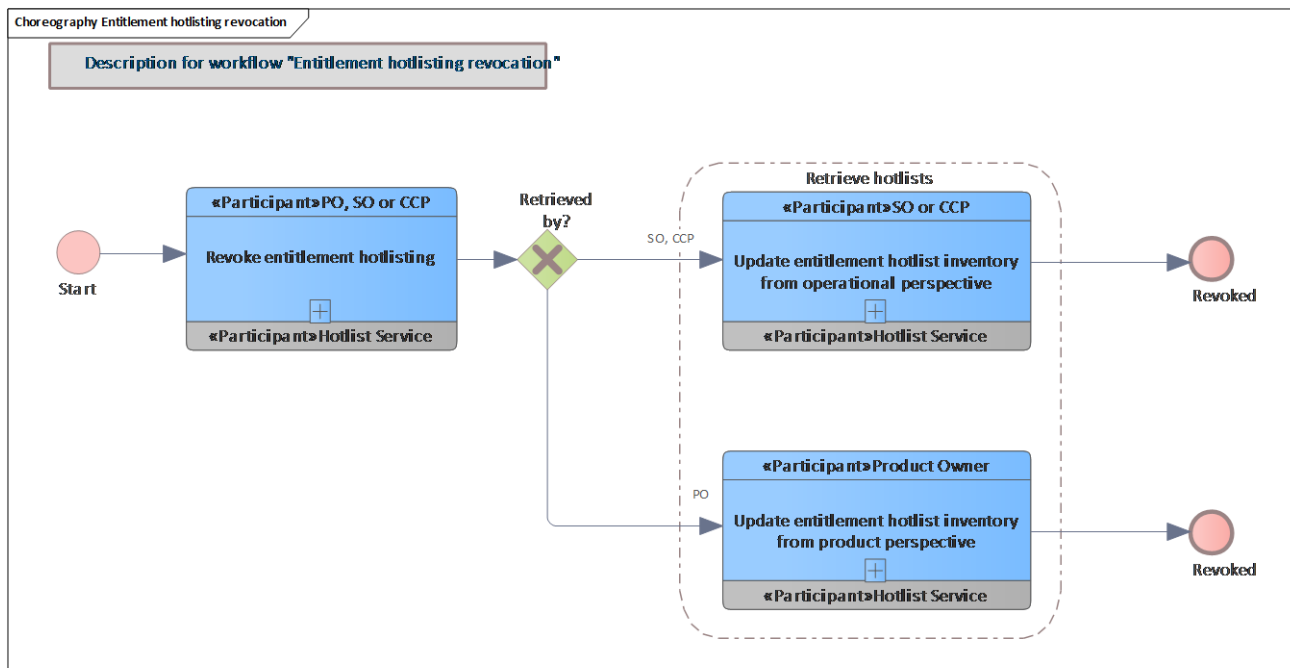


Figure 37: Entitlement hotlisting revocation

9.1.17.1 Revoke entitlement hotlisting

The SO, PO or CCP might want to revoke its hotlisting demand due to e.g. wrongly submitted hotlisting demands.

If the SO, PO or CCP requests hotlisting demand revocation for an entitlement, the pCCP checks the revocation demand and will remove the initial demand from the list for the entitlement.

The pCCP keeps track of the hotlisting demands for the entitlements for which it is responsible. If all demands for a particular entitlement have been revoked, the pCCP will request hotlist removal from the hotlist service system.

9.1.17.2 Update entitlement hotlist inventory from operational perspective

[Service operators](#) and [customer contract partners](#) must retrieve the newest entitlement hotlist, update their hotlist inventory with the new data and update the hotlists on their terminals. Without these updates, an outdated hotlist entry might unintentionally block the involved entitlement.

9.1.17.3 Update entitlement hotlist inventory from product perspective

A [Product Owner](#) must retrieve the entitlement hotlist at contractually defined intervals. The [Product Owner](#) updates its hotlist inventory with the new data. Hotlists are to be analysed from a product perspective for monitoring purposes. For this reason, removed hotlist entries must not cause monitoring issues, since they still exist on non-updated hotlists. Furthermore, if a [Customer Contract Partner](#) requests information about entitlements

on a certain user medium for replacement purposes, wrongly hotlisted entitlements are not considered.

9.1.18 SAM hotlisting removal

Revoking a SAM hotlisting is supposed to remove a SAM from the SAM hotlist. After the hotlist has been changed, it needs to be retrieved and updated in the SO and CCP back-office systems and the terminals. Furthermore, the [Product Owner](#) and the [Scheme Manager](#) have to update their hotlists for monitoring purposes.

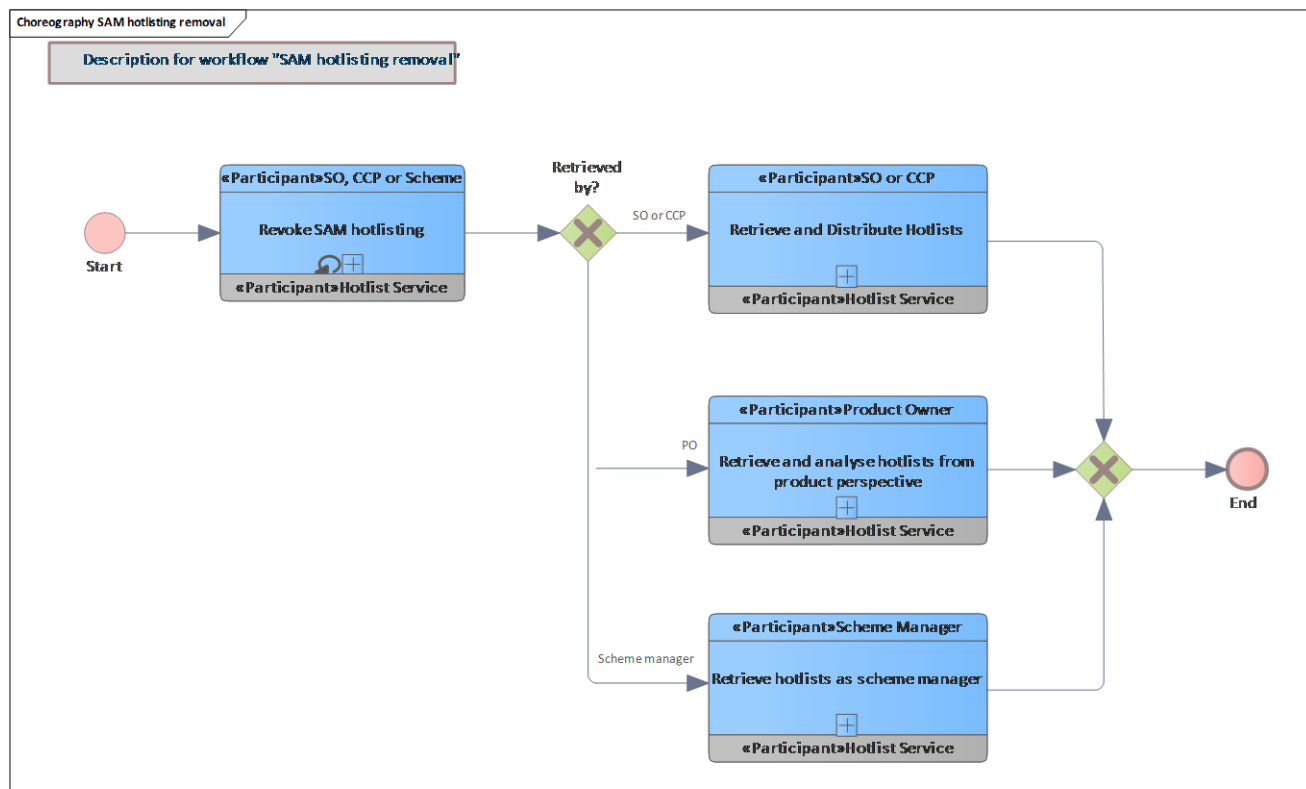


Figure 38: SAM hotlisting removal

9.1.18.1 Retrieve and analyse hotlists from product perspective

The [Product Owner](#) must retrieve the hotlists of entitlements, SAMs and organisations at contractually defined intervals for monitoring purposes.

9.1.18.2 Retrieve hotlists as scheme manager

The [Scheme Manager](#) retrieves the hotlists of SAMs, organisations and authentication keys for monitoring and security purposes.

9.1.18.3 Revoke SAM hotlisting

If a SAM was hotlisted, removal from the hotlist is not allowed for the SO and CCP.

The scheme manager is allowed to remove a SAM from the hotlist in the production and staging environment.

Note: hotlisted SAMs can not be reconfigured since their certificate has been revoked in the media management system.

9.1.18.4 Retrieve and Distribute Hotlists

Participants must retrieve the hotlists of applications, entitlements, SAMs, organisations and authentication keys at contractually defined intervals.

To avoid excessively redundant data transfer, the use of an incremental hotlist is offered in addition to the total hotlist.

9.1.19 Organisation hotlisting removal

Revoking an organisation hotlisting is supposed to remove an organisation from the organisation hotlist. Only the [Scheme Manager](#) is allowed to remove a hotlist entry for an organisation.

After the hotlist has been changed, it needs to be retrieved and updated in the SO and CCP back-office systems as well as in the PO and scheme manager system.

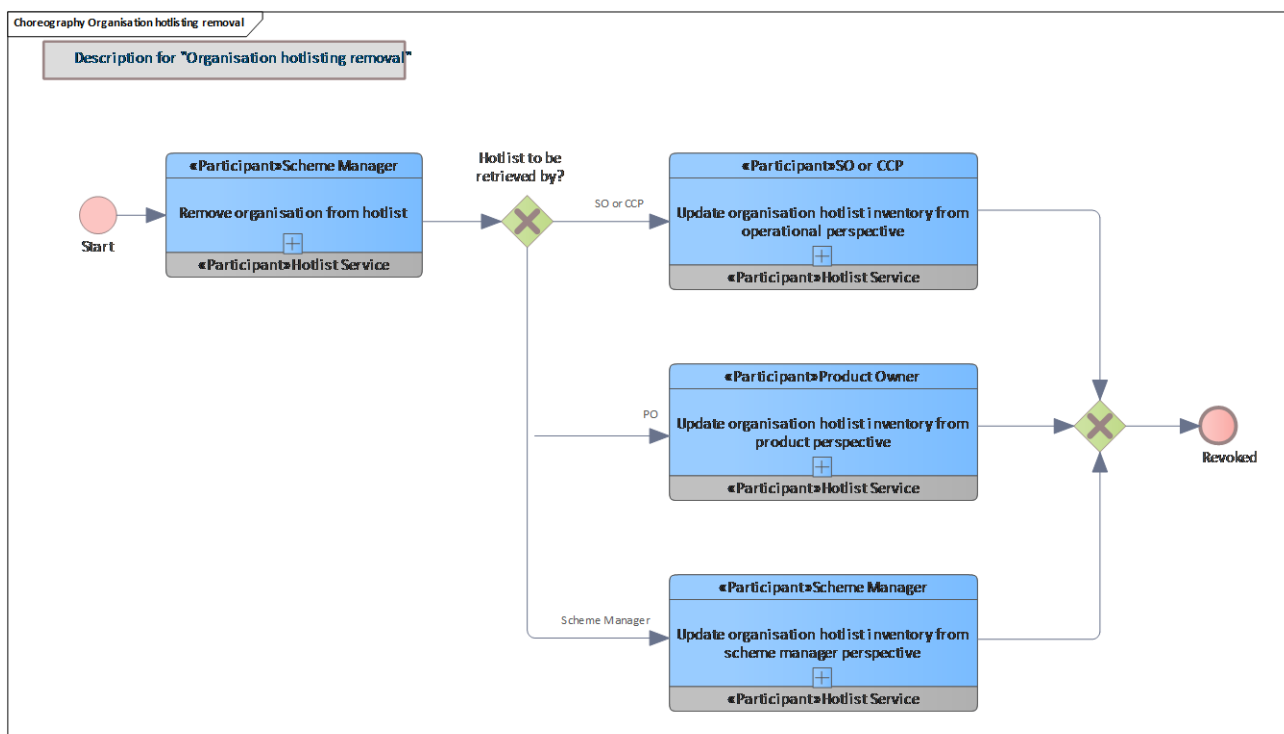


Figure 39: Organisation hotlisting removal

9.1.19.1 Remove organisation from hotlist

The [Scheme Manager](#) removes a hotlisted organisation from the hotlist due to e.g. wrongly submitted hotlisting.

9.1.19.2 Update organisation hotlist inventory from operational perspective

Service operators and customer contract partners must retrieve the hotlists of organisations at contractually defined intervals. The participants update their hotlist inventory with the new data and update their terminal hotlists.

The new hotlist no longer contains the entry for the organisation.

9.1.19.3 Update organisation hotlist inventory from product perspective

The [Product Owner](#) must retrieve the hotlist of organisations at contractually defined intervals. The participant updates its hotlist inventory with the new data.

The new hotlist no longer contains the entry for the organisation.

9.1.19.4 Update organisation hotlist inventory from scheme manager perspective

The [Scheme Manager](#) retrieves the hotlist of organisations for monitoring and security purposes. The participant updates its hotlist inventory with the new data.

The new hotlist no longer contains the entry for the organisation.

9.1.20 Authentication key hotlisting removal

Only the scheme manager is allowed to remove a hotlist entry for an authentication key. This process should nearly never happen and could be possible for wrongly hotlisted keys or after a long time when a next generation of SAMs and user media is active.

After the hotlist has been changed, it needs to be retrieved and updated in the SO and CCP back-office systems as well as the scheme manager for monitoring purposes.

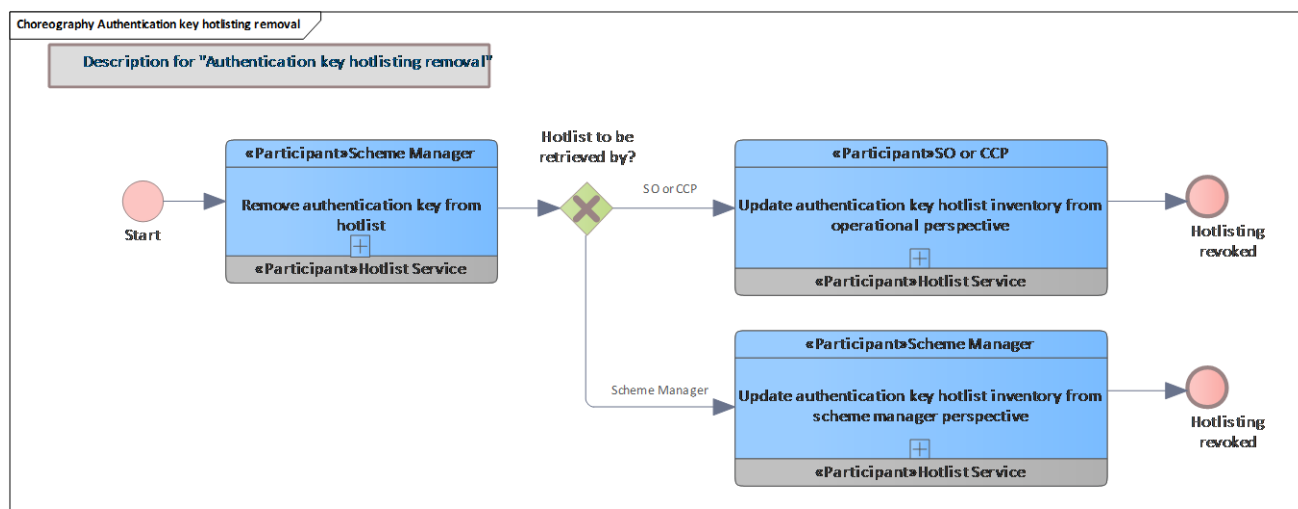


Figure 40: Authentication key hotlisting removal

9.1.20.1 Remove authentication key from hotlist

The [Scheme Manager](#) removes a hotlisted authentication key from the hotlist due to e.g. a cleanup process, in case an authentication key is no longer contained in any user medium or SAM.

9.1.20.2 Update authentication key hotlist inventory from scheme manager perspective

The [Scheme Manager](#) retrieves the hotlist of authentication keys for monitoring and security purposes. He updates his hotlist inventory with the new data.

9.1.20.3 Update authentication key hotlist inventory from operational perspective

The [Customer Contract Partner](#) and [Service Operator](#) must retrieve the hotlist of authentication keys at contractually defined intervals. These participants update their hotlist inventory with the new data and update their terminal hotlists.

Authentication keys removed from the hotlist will no longer be considered for comparisons with key versions in SAMs and user media to prevent their usage in terminal processes.

9.1.21 Customer service

This chapter describes basic processes related to customer services in a BPMN choreography model.

9.1.22 Defective user medium

Shows a high-level view of the basic processes in the context of handling a defective user medium with an application.

A defective user medium may be noticed during the inspection of an SO or the customer may be in the customer centre of a CCP. After an optional lookup for the application instance ID (pass the medium ID, if the medium ID is not equal to the application instance ID), the application has to be hotlisted.

Afterwards, the customer may receive a new user medium, if he is in the customer centre of the pCCP. This is not shown here.

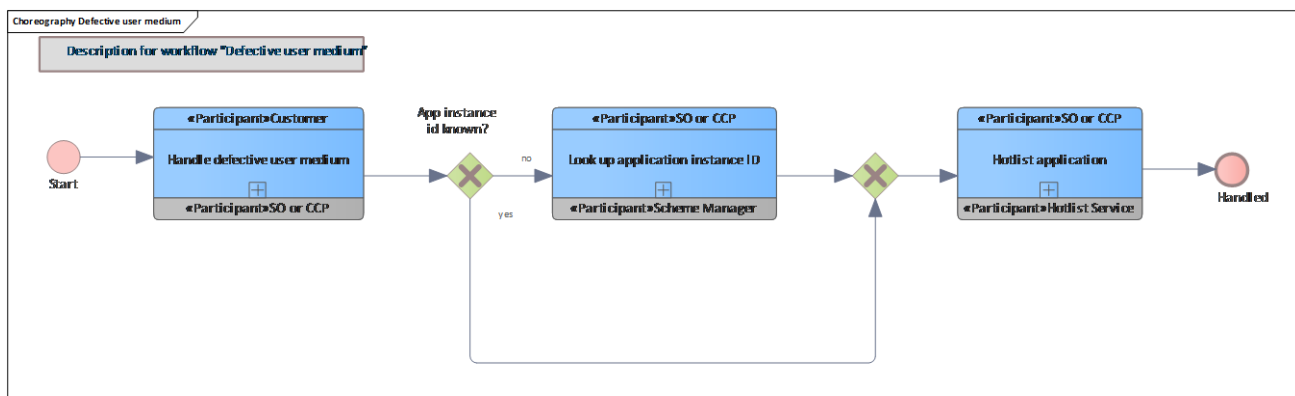


Figure 41: Defective user medium

9.1.22.1 Handle defective user medium

A defective user medium has been detected. Therefore, the time and location will be logged when the defective user medium was detected, together with its medium ID, which has to be recorded manually.

9.1.22.2 Look up application instance ID

Ask the [Scheme Manager](#) for the application instance ID related to a medium ID. See also [Look up application instance ID](#).

9.1.22.3 Hotlist application

The application needs to be hotlisted in one of the following ways depending on the appearance location of the defective user medium:

- A demand will be submitted by the SO or sCCP to the pCCP which orders an application hotlisting in the hotlist service (see [Hotlist non-owned application](#))
- The pCCP directly orders an application hotlisting in the hotlist service (see [Hotlist owned application](#))

9.1.23 Lost user medium

Shows a high-level view of the basic processes in the context of handling a lost user medium with an application.

A lost user medium with an application can only be handled by the pCCP of the user medium with application since the PO does not have information about blocked applications on user media. In the case of another sCCP, the PO could list entitlements which are located in a blocked application, and the sCCP has no chance to verify this.

Note that this process is only possible if the customer is known in the back-office system of the pCCP.

If the back-office system of the pCCP only knows the medium ID, an optional lookup for the application instance ID can be done.

The application of the lost user medium is added to the hotlist.

The customer needs a new user medium. A new one is issued. The entitlements owned by the pCCP can be issued directly to the new medium. If the customer has other entitlements that have been issued by another CCP, the responsible PO can be requested to list these entitlements.

Aided by this list, the pCCP of the user medium can issue replacement entitlements.

For all re-issued entitlements, the pCCP must notify the old entitlement ID as replaced.

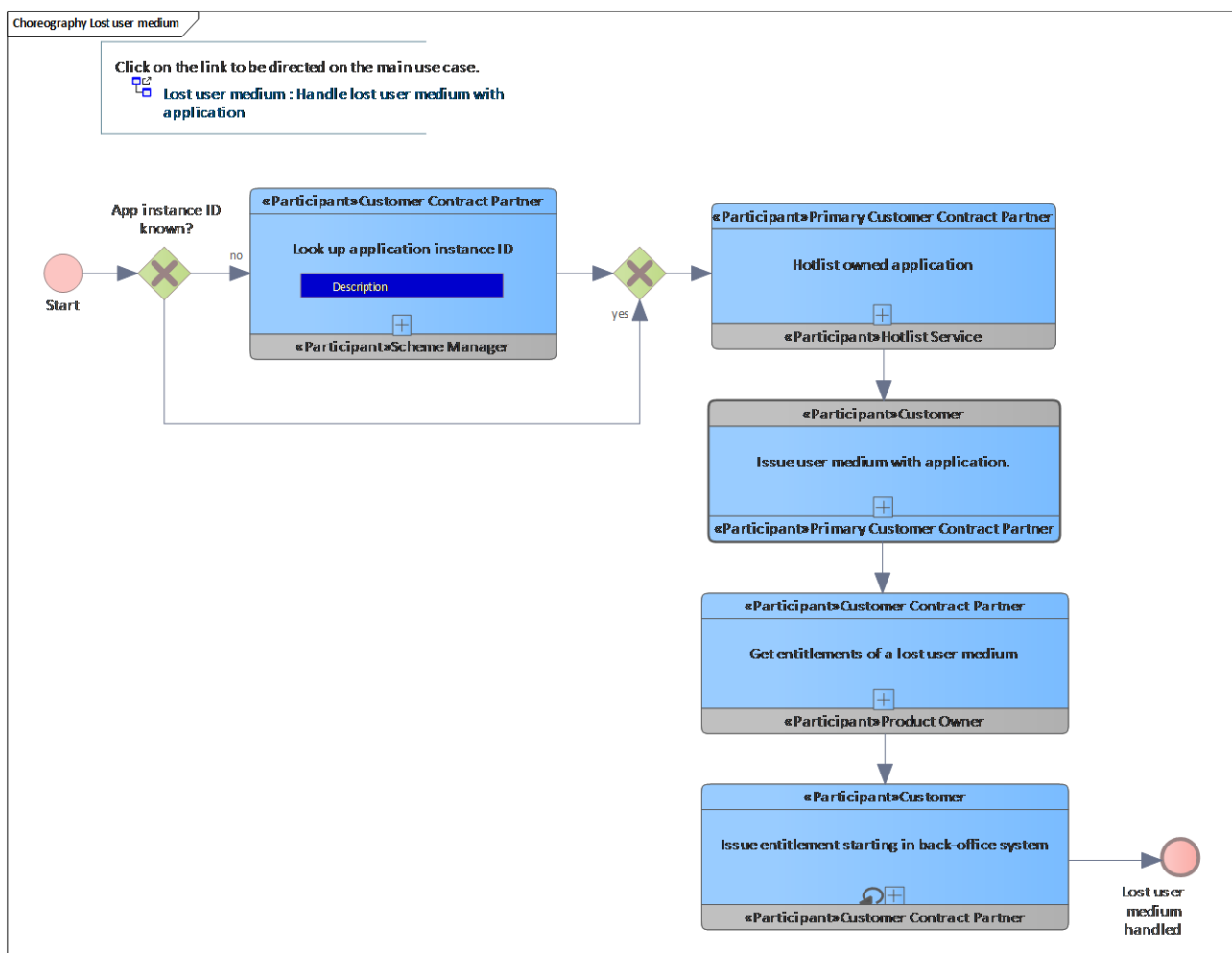


Figure 42: Lost user medium

9.1.23.1 Look up application instance ID

Ask the [Scheme Manager](#) for the application instance ID related to a medium ID. See also [Look up application instance ID](#).

9.1.23.2 Hotlist owned application

Hotlist an owned application by requesting the Hotlist Service to directly adding it to the hotlist. See also [Hotlist owned application](#).

9.1.23.3 Get entitlements of a lost user medium

If the entitlements belonging to the lost user medium are not registered in the pCCP's database, the information regarding entitlements can be retrieved from the [Product Owner](#) based on the contracts between CCP and PO.

See also [Get entitlements of a lost user medium](#).

9.1.23.4 Issue user medium with application.

Issue a new (((etiCORE application. See [Issue user medium with application](#).

9.1.23.5 Issue entitlement starting in back-office system

Issuance of entitlements which can be retrieved from own database and/or from product owners that might come in question.

See also [Issue entitlement starting in back-office system](#).

9.1.23.6 App instance ID known?

Please note that the medium ID must be known in the back-office system database or with the help of customer receipt, at least, if the application instance ID is not known.

9.1.24 Unblock application

The workflow "unblock application" is considered an independent basic process, as there are no other basic processes involved. For further information please refer the description of the basic process [Unblock application](#).

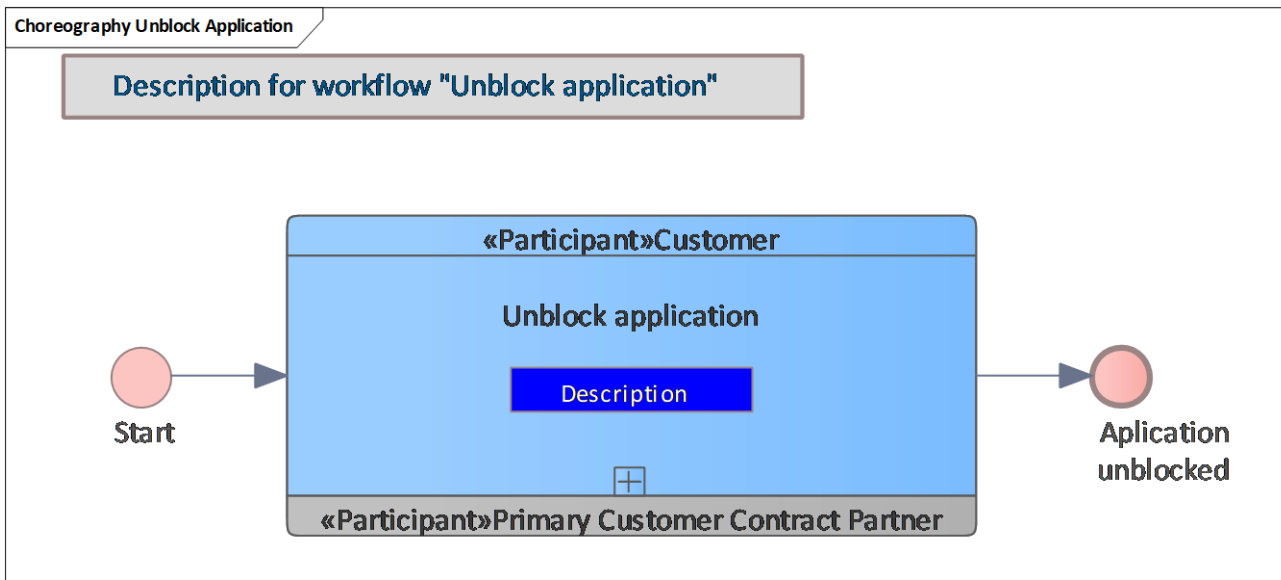


Figure 43: Unblock Application

9.1.24.1 Unblock application

The application is already marked as blocked on the user medium, and the hotlist entry has already been removed.

The block on the user medium must be removed again by the pCCP when the blocking reason no longer exists.

9.1.25 Unblock entitlement

The workflow "unblock entitlement" is considered an independent basic process, as there are no other basic processes involved.

For further information please refer the description of the basic process [Unblock entitlement](#).

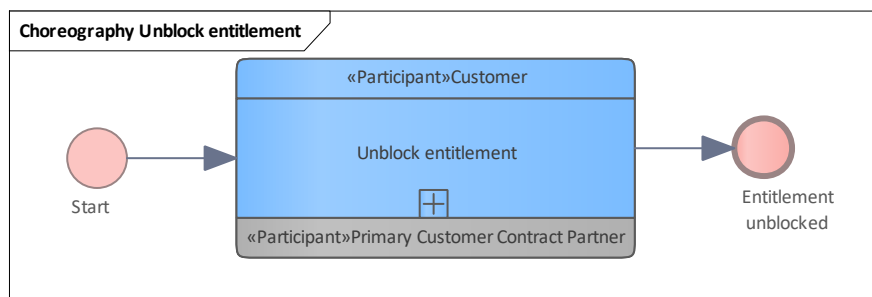


Figure 44: Unblock entitlement

9.1.25.1 Unblock entitlement

The entitlement is already marked as blocked on the user medium, and the hotlist entry has already been removed.

The block of the entitlement on the user medium must be removed again by the pCCP, when the blocking reason no longer exists.

9.1.26 Exchange user medium with application

A user medium with an application is replaced with a one, for example, due to a broken medium or its validity which will expire soon.

In these cases, the user medium has to be replaced by a new one. All active entitlements on the old user medium have to be terminated.

Then these entitlements have to be issued with the same data on the new user medium (if necessary, with a new expiry date). This user medium has to be prepared in advance. The application instance must be configured for the current customer contract partner and, depending on the underlying contract, customer data has to be stored.

Note: if at least one entitlement does not belong to the customer contract partner, this entitlement and the old application must not be terminated. The customer has to go to the responsible customer contract partner to transfer this entitlement(s) to his new user medium.

Please note that it is assumed that there is no need for any payment transaction.

Please note that a broken medium does not mean that it is defective as in the basic process "[Defective user medium](#)". The user medium may be broken on the edge, and the application on the user medium is still working.

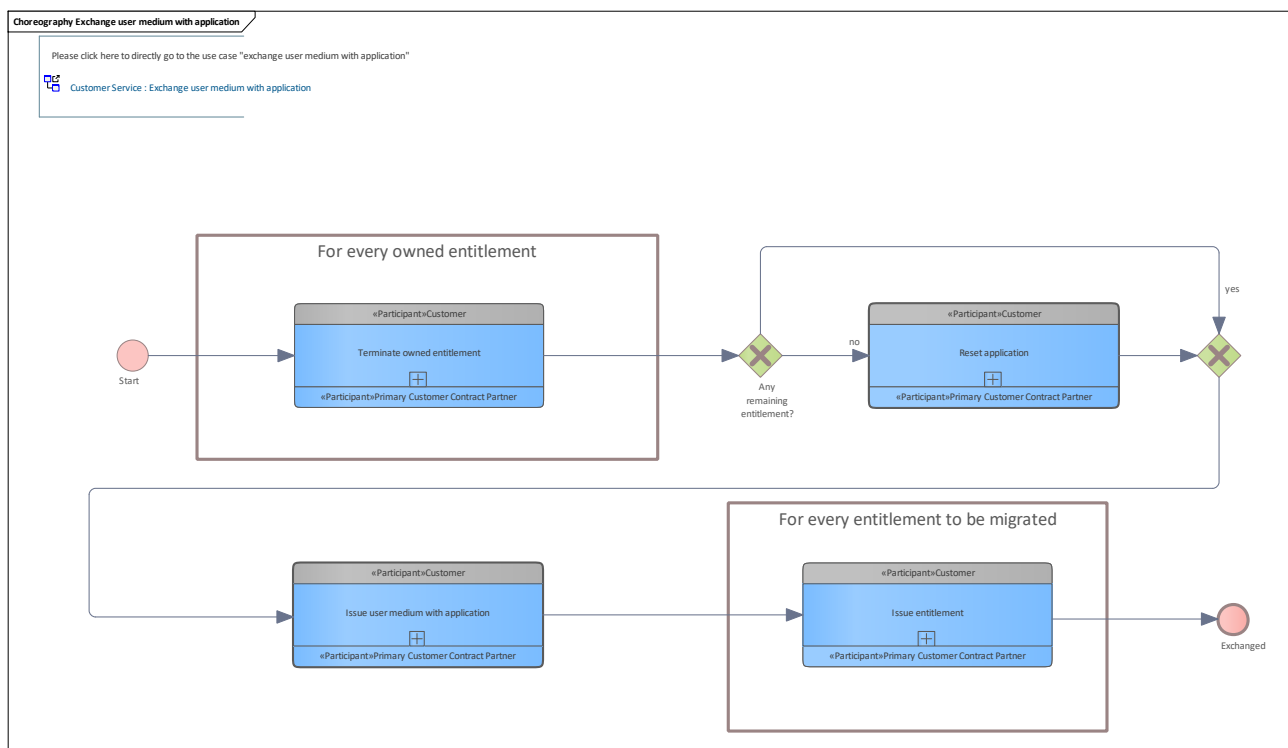


Figure 45: Exchange user medium with application

9.1.26.1 Terminate owned entitlement

Terminates an entitlement owned by the customer contract partner. Termination means that the entitlement is given a final state and can no longer be used.

This has to be done for all entitlements if more than one entitlement exists on the user medium.

Note: cache all entitlements before terminating them so that the master information on entitlement can be used by issuing new entitlement on the new user medium with application.

9.1.26.2 Reset application

Resets the application on the user medium. See [Reset application](#).

9.1.26.3 Issue user medium with application

See [Issue user medium with application](#).

9.1.26.4 Issue entitlement

Entitlement is issued, as needed.

See [Issue entitlement starting in terminal](#).

9.1.27 Process new information about customer and discounts

Shows a high-level view of the basic processes in the context of processing new information about customer and discounts data on a user medium with an application.

The workflow changes customer data and/or discounts for concessionary tickets. If the current customer has entitlements that are personalised, the change of the customer data may involve the existing entitlements. In this case, the entitlements have to be changed (terminate the old one, re-issue the new one with the changed customer's personal data).

Please note that it is assumed that there is no need for any payment transaction.

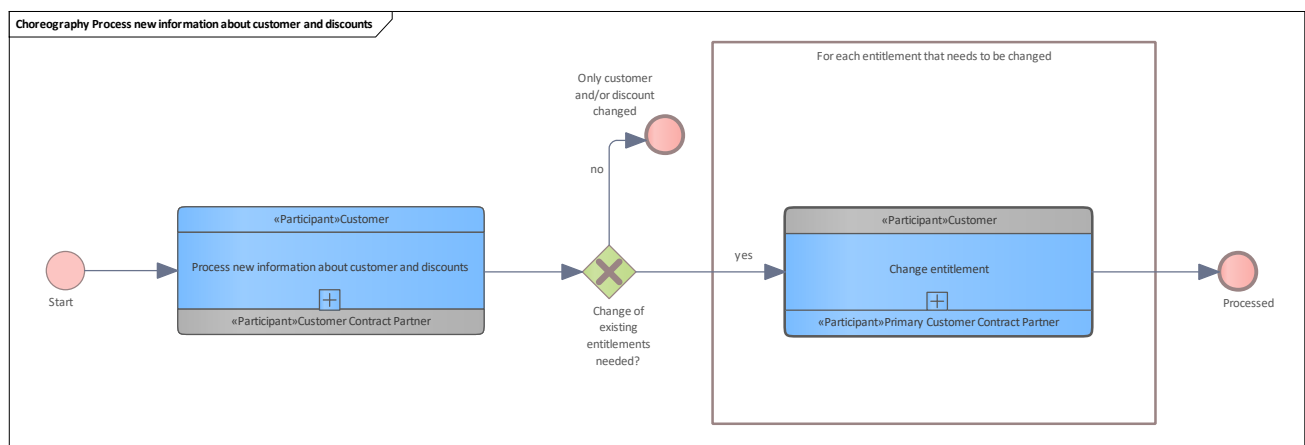


Figure 46: Process new information about customer and discounts

9.1.27.1 Process new information about customer and discounts

Customer and discount information is read, changed and re-written to a user medium with an application.

9.1.27.2 Change entitlement

Terminates an owned entitlement and then issue a new entitlement according to changed information about customer and discounts.

Note: cache all entitlements before terminating so that the master information on entitlement can be used by issuing new entitlement on the new user medium with application. See also [Change entitlement](#).

9.1.28 Change entitlement

Shows a high-level view of the basic processes in the context of changing an entitlement on a user medium.

If the customer contract must be changed (for example due to changed customer data), then the entitlement can be changed by terminating the existing one and issuing a new entitlement with corresponding parameters.

Please note that it is assumed that there is no need for any payment transaction.

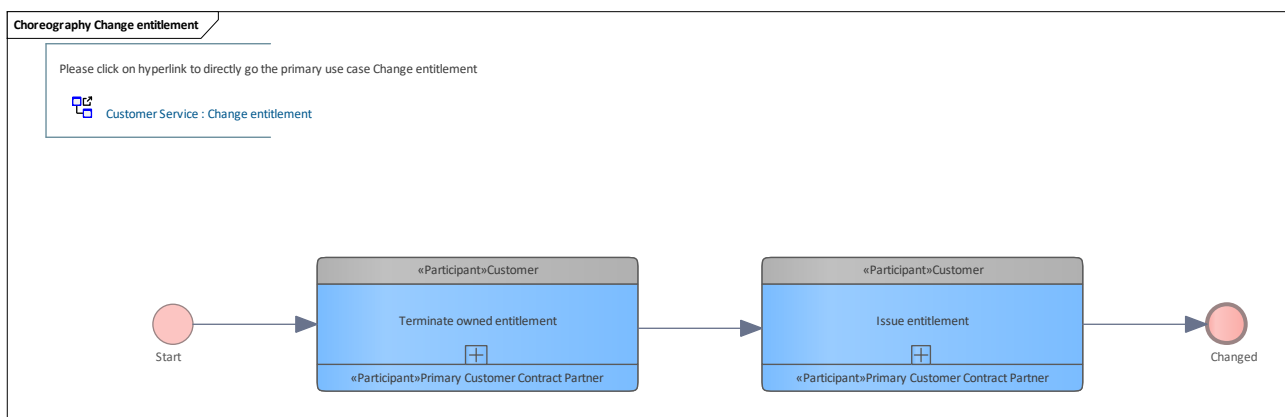


Figure 47: Change entitlement

9.1.28.1 Terminate owned entitlement

Terminates an owned entitlement. The entitlement changes to a final state, so it can no longer be used. See also [Take back owned entitlement](#).

Note: Cache all entitlement data that needs to be changed before terminating so that the master information of the entitlement can be used by issuing new entitlement on the user medium with the application.

9.1.28.2 Issue entitlement

The entitlement is issued in its desired form. All cached data from the old entitlement is transferred to the new entitlement together with the data that have to be changed.

See [Issue entitlement starting in terminal](#)

9.1.29 Change static entitlement

Shows a high-level view of the basic processes in the context of changing a static entitlement on a user medium.

If the customer contract must be changed (for example due to changed customer data), then the entitlement can be changed by terminating the existing one and issuing a new entitlement with corresponding parameters.

Please note that it is assumed that there is no need for any payment transaction.

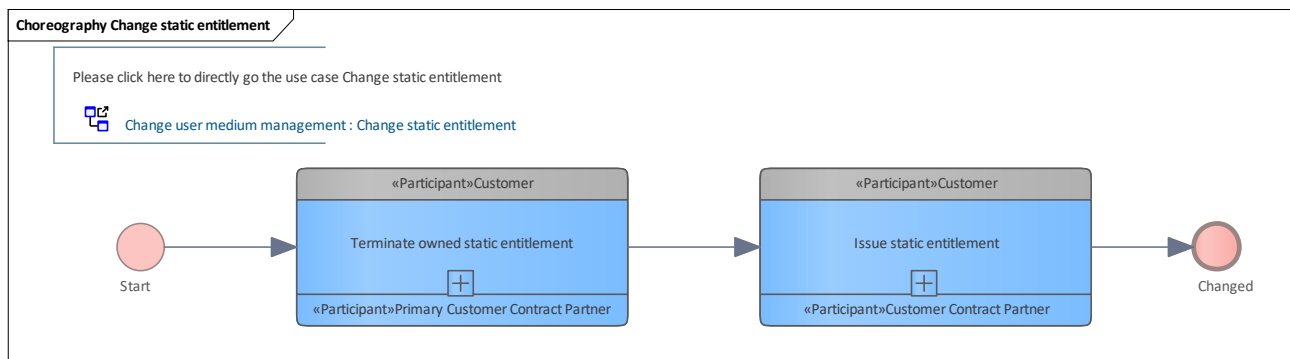


Figure 48: Change static entitlement

9.1.29.1 Terminate owned static entitlement

Terminates an owned static entitlement. This involves no state transition but notifying the entitlement as terminated resulting in subsequent monitoring and hotlisting.

Note: cache all entitlement data that needs to be changed before terminating so that the master information of the entitlement can be used by issuing the new static entitlement.

9.1.29.2 Issue static entitlement

Issues a new static entitlement by using the data from the old static entitlement together with the data to be changed.

9.1.30 Take back

This chapter describes the basic processes in the context of taking back applications and entitlements in a BPMN choreography model.

9.1.31 Take back application

Workflow that combines the basic processes [Take back entitlement](#) and [Reset application](#) for taking back a user medium application.

An application is taken back when the underlying user medium is taken back from a customer or when the (((etiCORE-specific application is removed from a customer-owned user medium. This can only be done by the [Primary Customer Contract Partner](#). Before taking back the application, all entitlements have to be terminated.

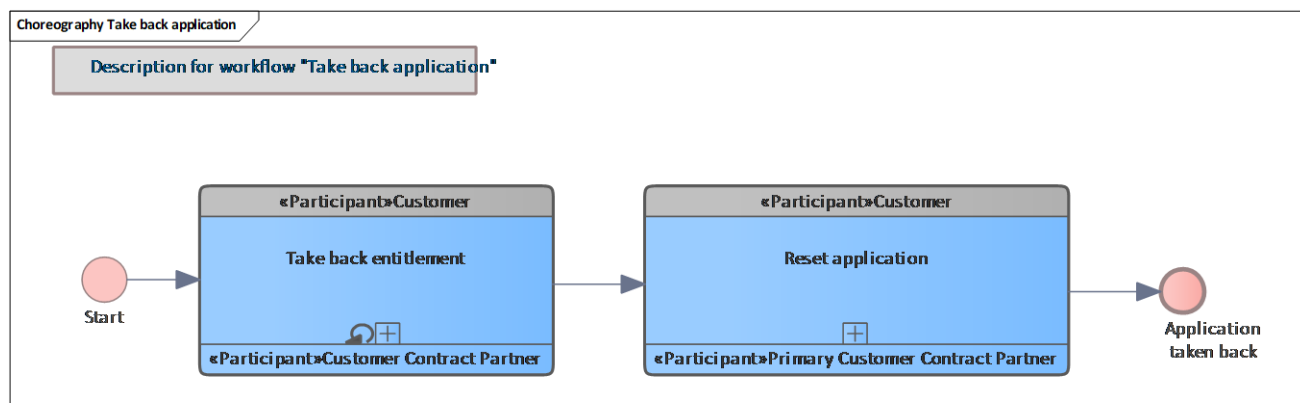


Figure 49: Take back application

9.1.31.1 Take back entitlement

Takes back every entitlement that has not yet expired. This is also possible for non-owned entitlements. See also [Take back entitlement](#).

9.1.31.2 Reset application

Resets the application. See also [Reset application](#).

9.1.32 Reset application

The basic processes involved during application reset are shown.

An application is terminated when the underlying user medium is taken back from a customer or when the (((etiCORE-specific application is removed from a customer-owned user medium. The application can be terminated, which means bringing it to a final state without the possibility of re-using it.

Alternatively, the application can be blocked if the user medium is provided for later re-use. Independent of re-using the user medium or not, the customer data has to be removed from the user medium.

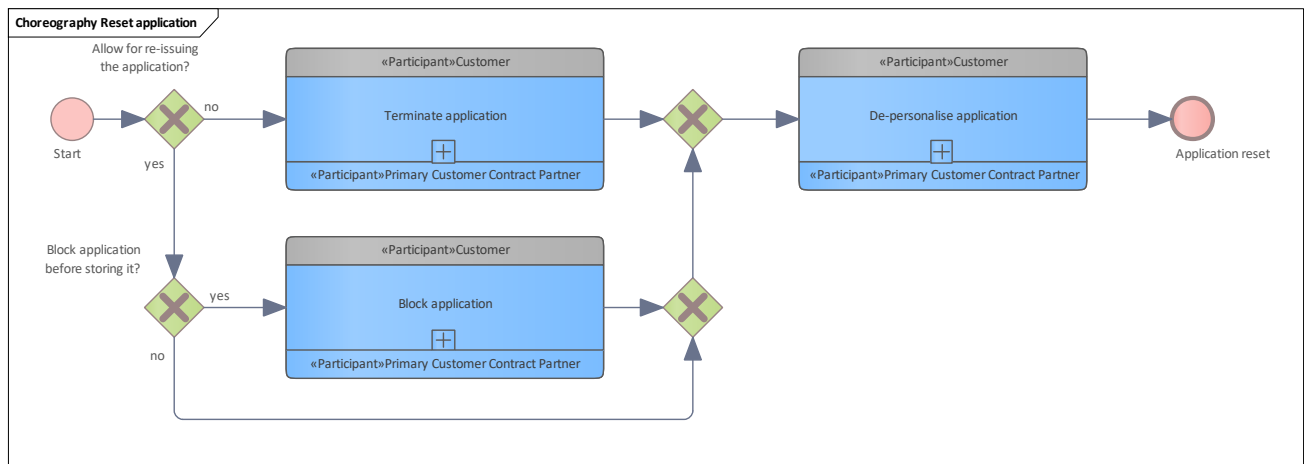


Figure 50: Reset application

9.1.32.1 Block application

The user medium owner may decide to block the user medium application before storing it for later re-use. This may provide an additional security aspect for the stored user media. Additionally, the unblocking attestation created during the re-issuance of the user medium can be used as standardised, secure and time-stamped information upon which to base the change of possession process.

Note: since the user medium application may not actually be hotlisted in this case, the blocking process deviates in this regard and the related blocking attestation does not provide a valid hotlist cycle.

9.1.32.2 De-personalise application

The process of de-personalisation is used to remove all customer-specific information from an etiCORE application.

See [De-personalise application](#)

9.1.32.3 Allow for re-issuing the application?

The application does not necessarily need to be terminated. Termination definitely ends the lifetime of this user medium application. If the possibility of using this application again later on (potentially for a different customer) is to be preserved, the application must not be terminated.

9.1.32.4 Terminate application

See [Perform application termination and notify](#).

9.1.33 Take back entitlement

The basic processes involved in taking back an entitlement are shown.

An entitlement may be taken back when the underlying user medium is taken back from a customer. The other possible use case is re-newing an entitlement due to expiry reasons or due to tariff changes. In this case, the old entitlement is terminated and a new one is issued.

Taking back an entitlement means the combination of a possible reimbursement and a downstream termination of the entitlement.

The reimbursement follows the legal requirement that the same payment means has to be used for reimbursing as the one for the original payment.

When taking back a stored-value payment method, the he payment is made with legal tender. If no reimbursement is needed, the entitlement can be terminated directly.

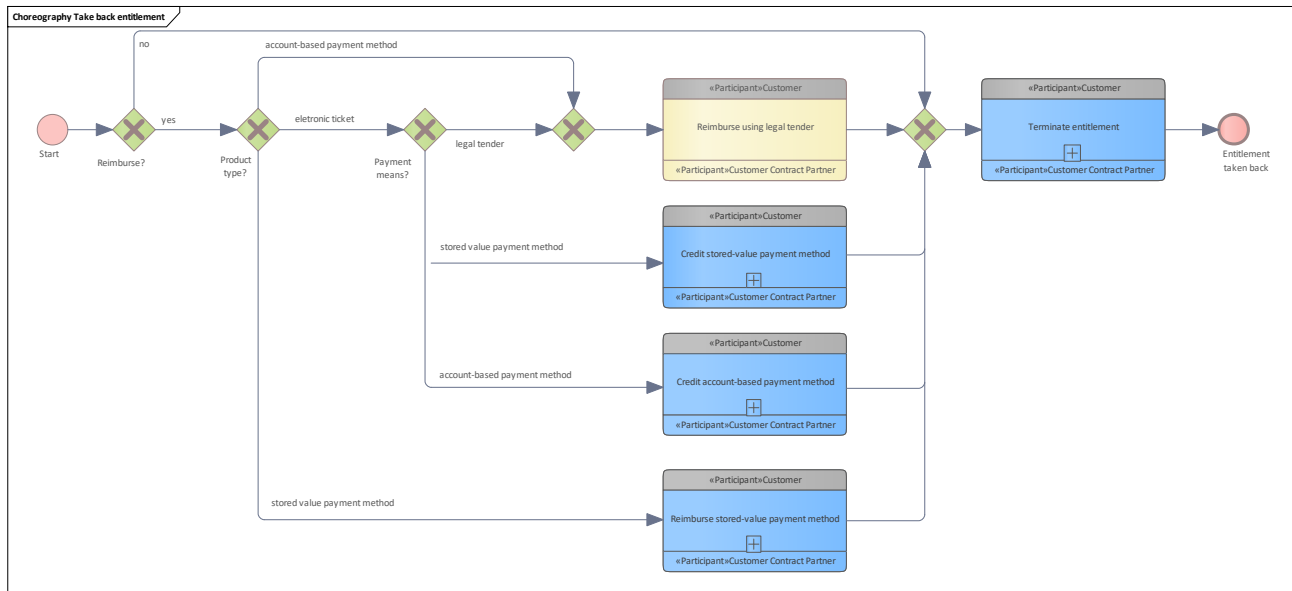


Figure 51: Take back entitlement

9.1.33.1 Reimburse using legal tender

Reimburse using legal tender. The easiest way to reimburse an amount for an electronic ticket that is taken back.

9.1.33.2 Credit stored-value payment method

If the amount for the electronic ticket was originally paid with a stored-value payment method, this amount is posted back to this payment method.

9.1.33.3 Credit account-based payment method

If the amount for the electronic ticket was originally paid with an account-based payment method, this amount is posted back to this payment method.

9.1.33.4 Reimburse stored-value payment method

If the entitlement to be terminated is a stored-value payment method, the remaining amount must be paid out before.

9.1.33.5 Terminate entitlement

An entitlement is marked as terminated to prevent any further usage of the entitlement. Usually, the entitlement is deleted from the user medium application directly afterwards.

The termination can be performed by any CCP. This means that the CCP may be the owner of the entitlement or not. See also [Take back non-owned entitlement](#) and [Take back owned entitlement](#).

9.1.34 Take back static entitlement

The basic processes involved in taking back a static entitlement are shown. The workflow includes reimbursement and termination.

A static entitlement can be taken back only by its pCCP.

A possible use case is re-newing a static entitlement due to expiry reasons or due to tariff changes. In this case, the old static entitlement is terminated and a new one is issued. Another possible scenario is the purchase of a wrong ticket.

Taking back a static entitlement means the combination of a possible reimbursement and a downstream termination of the entitlement.

The reimbursement follows the legal requirement, that the same payment means has to be used for reimbursing as the one for the original payment.

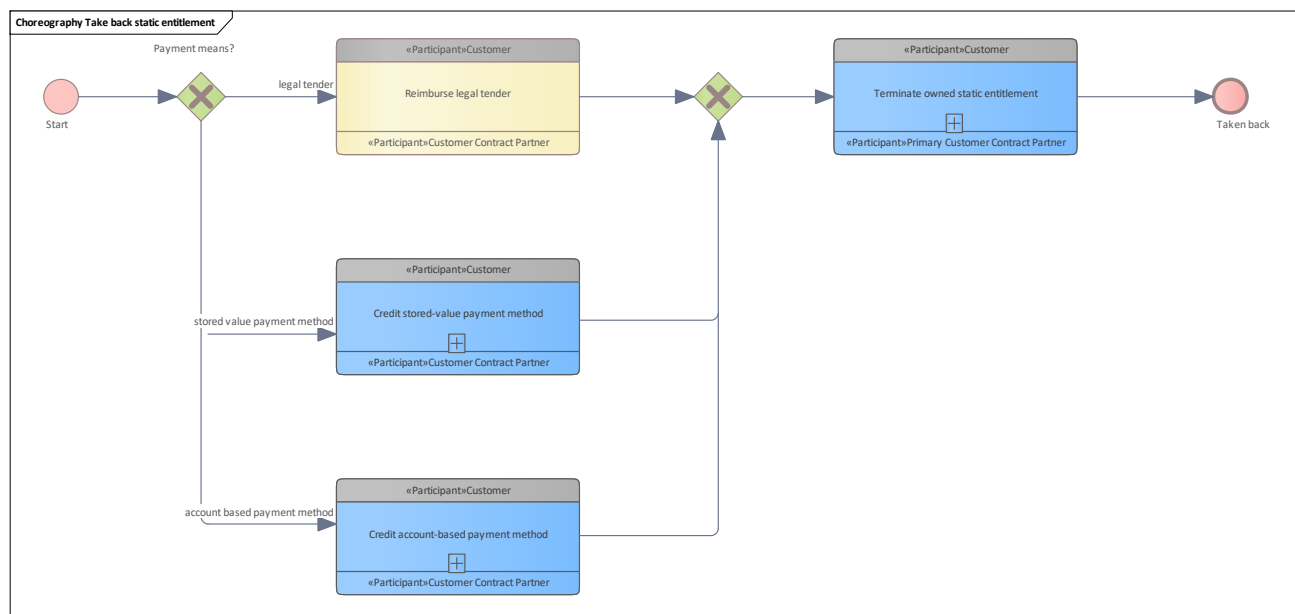


Figure 52: Take back static entitlement

9.1.34.1 Reimburse legal tender

Reimburse using legal tender. The easiest way to reimburse an amount for an electronic ticket as static entitlement that is taken back.

9.1.34.2 Credit stored-value payment method

If the amount for the electronic ticket was originally paid with a stored-value payment method, this amount is posted back to this payment method.



9.1.34.3 Credit account-based payment method

If the amount for the electronic ticket was originally paid with an account-based payment method, this amount is posted back to this payment method.

9.1.34.4 Terminate owned static entitlement

Terminates an owned static entitlement. This involves no state transition but notifying the entitlement as terminated resulting in subsequent monitoring and hotlisting. The termination can be done only by the pCCP.

9.1.35 Ordered action management

This chapter describes the basic processes of ordered action execution in a BPMN choreography model.

9.1.36 Action list configuration

Workflow that shows the dependency between the basic process of action list retrieval configuration as first step and the basic process to retrieve action lists, update the action inventory and distribution of action lists to the involved terminals.

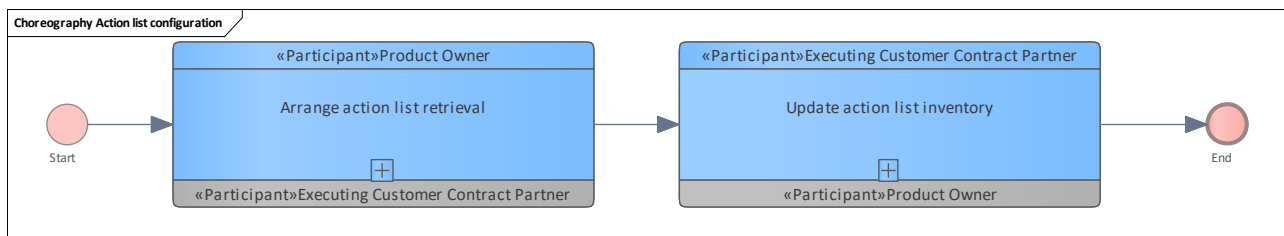


Figure 53: Action list configuration

9.1.36.1 Arrange action list retrieval

See [Distribute action list retrieval configuration](#).

9.1.36.2 Update action list inventory

See [Update action list inventory](#).

9.1.37 Order entitlement issuance

The process of ordered entitlement issuance is used to decouple the sales process from the processes of writing the entitlement onto a user medium and the related notification aspects. This enables sales processes in web shops or call centres to target the user medium as a secure entitlement container.

All parameters regarding the entitlement (including, e. g., its validity period) have to be determined before the order is sent.

Triggered by the customer, a new entitlement is needed. The [Ordering Customer Contract Partner](#) will place a new order for this entitlement in the action list by requesting the [Product Owner Action Management](#). The [Executing Customer Contract Partner](#) will update its action list inventory in its terminals and the next contact of the involved user medium with a suitable terminal triggers the issuance of the entitlement with all downstream notifications.

Since the ordering of an action for issuing an entitlement, is in principle, also possible after the start of the entitlement's validity period, the following must be taken into account:

- the lead time it takes for the action to reach the terminals of Executing Customer Contract Partners
- the eligibility of the customer to use the ordered entitlement only after the corresponding lead time

This must be properly communicated to the customer.

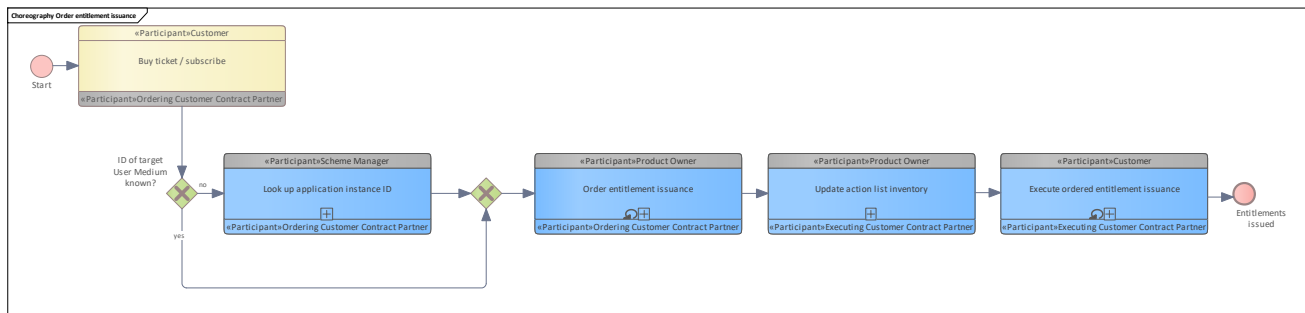


Figure 54: Order entitlement issuance

9.1.37.1 Buy ticket / subscribe

The customer buys a ticket or takes out a public transport subscription.

9.1.37.2 Look up application instance ID

Ask the [Scheme Manager](#) for the application instance ID related to a medium ID. See also [Look up application instance ID](#).

9.1.37.3 Order entitlement issuance

The [Ordering Customer Contract Partner](#) orders a new entitlement issuance. See [Order entitlement issuance](#).

9.1.37.4 Update action list inventory

See [Update action list inventory](#).

9.1.37.5 Execute ordered entitlement issuance

See [Execute ordered entitlement issuance](#).

9.1.38 Order entitlement termination

The process of ordered entitlement termination is used to decouple the contract termination process from the processes of marking the entitlement on a user medium as terminated and the related notification aspects. This enables contract termination processes in web shops or call centres for entitlements stored on user media that are not currently accessible for their systems.

Note that no crediting actions with the user medium can be performed for reimbursement of the terminated entitlement. Reimbursement thus has to be handled by other means, e. g., bank transfer.

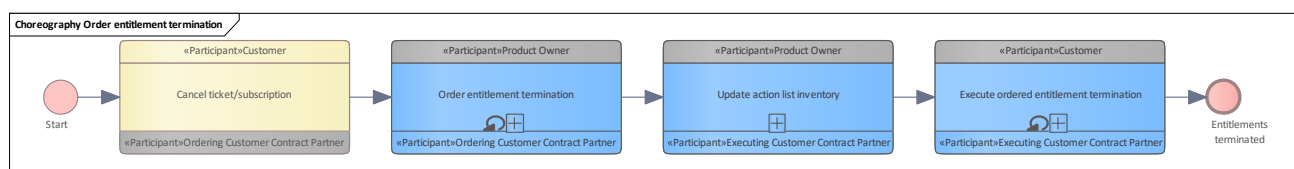


Figure 55: Order entitlement termination

9.1.38.1 Cancel ticket/subscription

The customer cancels his ticket/subscription.

9.1.38.2 Order entitlement termination

The [Ordering Customer Contract Partner](#) orders a new entitlement termination. See [Order entitlement termination](#).

9.1.38.3 Update action list inventory

See [Update action list inventory](#).

9.1.38.4 Execute ordered entitlement termination

See [Execute ordered entitlement termination](#).

9.1.39 Order entitlement blocking

The process of ordered entitlement blocking is similar to the hotlisting processes, but for the special case where the entitlement ID of the target is unknown as the entitlement has not yet been issued or the issue notification has not yet been received and processed.

This is the case when an issuance order was placed, whose execution status is unknown to the ordering CCP and which should either not be executed any more or (in case it already was executed) shall not enable the corresponding customer to use the entitlement.

Without the entitlement ID, the regular hotlisting process can not be used. Instead, this process is based on the order ID of the pending issuance order.

This process can be used in conjunction with issuance order cancellation. In contrast to hotlist entries, a blocking order used this way should expire already when it can safely be assumed that all terminals have the action list without the cancelled issuance order.

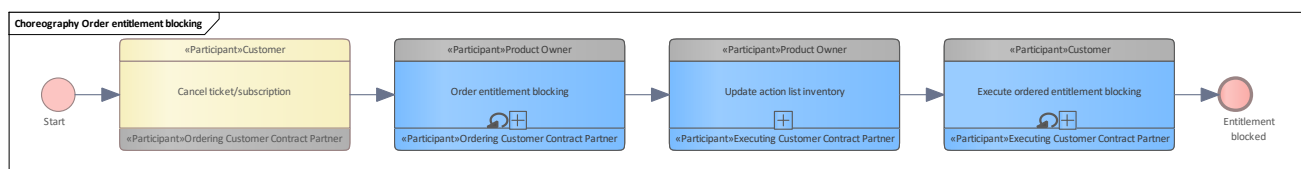


Figure 56: Order entitlement blocking

9.1.39.1 Cancel ticket/subscription

The customer cancels a ticket/subscription which he has previously ordered and which has not yet been issued.

9.1.39.2 Order entitlement blocking

See [Order entitlement blocking](#).

9.1.39.3 Update action list inventory

See [Update action list inventory](#).

9.1.39.4 Execute ordered entitlement blocking

See [Execute ordered entitlement blocking](#).

9.1.40 Order entitlement unblocking

The process of ordered entitlement unblocking is used to decouple the process of determining whether an entitlement should still be blocked from the processes of removing the block mark from the entitlement on a user medium and the related notification aspects.

This may be useful to save the customer the trip to a service centre in case one of his entitlements may be unblocked.

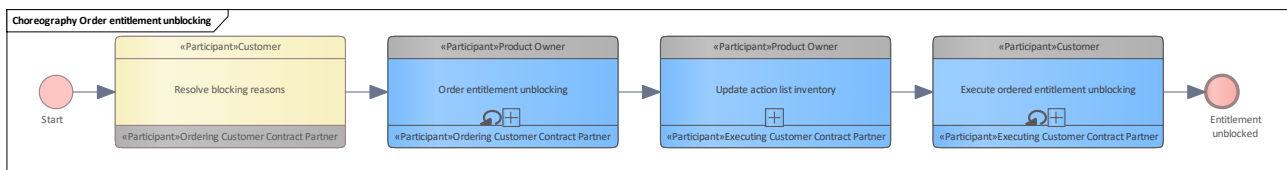


Figure 57: Order entitlement unblocking

9.1.40.1 Resolve blocking reasons

The customer resolves the reasons that led to the blocking of his ticket.

9.1.40.2 Order entitlement unblocking

See [Order entitlement unblocking](#).

9.1.40.3 Update action list inventory

See [Update action list inventory](#).

9.1.40.4 Execute ordered entitlement unblocking

See [Execute ordered entitlement unblocking](#).

9.1.41 Order entitlement replacement

The entitlement replacement process replaces entitlements with different entitlements. This process can be used for many different scenarios, e.g. for tariff adjustments triggered by the product owner or for reflecting changes in a user subscription. Technically, any number of entitlements on a single user medium can be replaced by any number of entitlements to be issued.

The replacement order comprises termination and issuance orders that are logically grouped. This logical grouping is still available to terminals (with enabled ordered action execution) and can be used to execute all contained orders in the group within a single transaction.

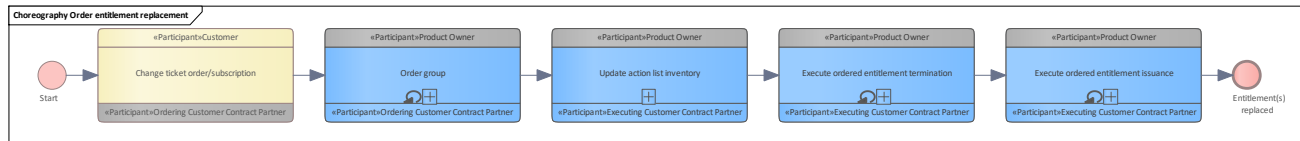


Figure 58: Order entitlement replacement

9.1.41.1 Change ticket order/subscription

The customer wants to change the ticket he bought or his public transport subscription.

9.1.41.2 Start

This process can be used for many different scenarios, e.g.

- tariff adjustments triggered by the product owner
- change the entitlements on the user medium to reflect changes in user subscription

9.1.41.3 Order group

See [Order group](#).

To change an entitlement effective immediately, order an entitlement termination (for the current entitlement) and order an entitlement issuance for the version of the entitlement that reflects the changes.

To change an entitlement effective at a given point in time, order an entitlement termination (for the current entitlement), order an entitlement issuance for the original entitlement with the entitlement expiration time set to the above-given point in time, and order an entitlement issuance for the version of the entitlement that reflects the changes and has the above-given point in time as entitlement effective time.

9.1.41.4 Update action list inventory

See [Update action list inventory](#).

9.1.41.5 Execute ordered entitlement termination

See [Execute ordered entitlement termination](#).

9.1.41.6 Execute ordered entitlement issuance

See [Execute ordered entitlement issuance](#).

9.1.42 Cancel order

The ordering CCP cancels an order.

Cancelling an order will remove it from future action lists.

The desired behaviour removes the entry from the future action lists before the action could be executed by a terminal on the involved user medium.

Note: due to possible race conditions, the order cancellation may not be able to prevent the order from being executed before the next action list has reached all relevant terminals. In particular, the order may already have been executed and the execution notification may still be on its way to the relevant systems.

This has to be considered in monitoring.

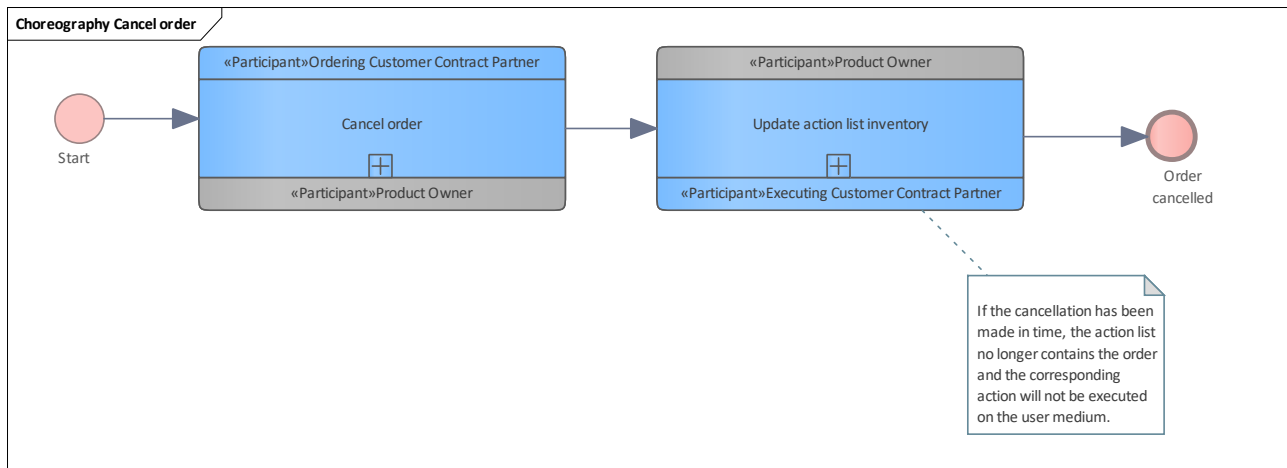


Figure 59: Cancel order

9.1.42.1 Cancel order

See [Cancel order](#).

9.1.42.2 Update action list inventory

See [Update action list inventory](#). The new action list no longer contains the cancelled order.

9.1.43 Handle obsolete order

Workflow with only one basic process performed between the product owner and the ordering customer contract partner.

During monitoring, action lists are searched for orders for unblocking an entitlement. If a notification message of the involved entitlement is received by the PO which indicates that the entitlement has already been unlocked, a related order for unblocking this entitlement can be removed from the action list.

This is also announced to the ordering customer contract partner.

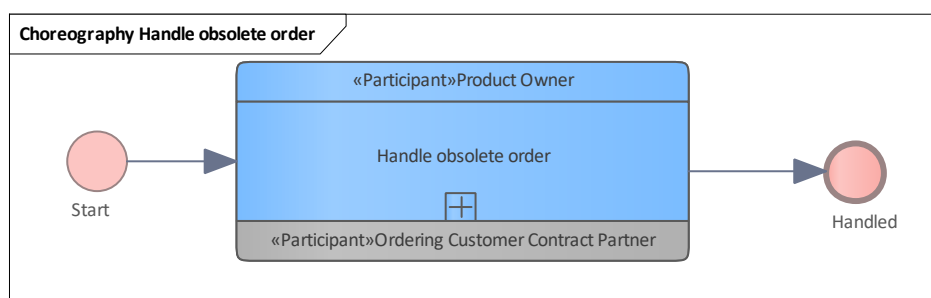




Figure 60: Handle obsolete order

9.1.43.1 Handle obsolete order

See [Handle obsolete order](#).

9.1.44 CICO

This chapter describes the basic process related to check-in/check-out processes in a BPMN choreography model.

9.1.45 Record entitlement within the CICO process

This workflow describes and shows a high-level perspective on recording a check-in (CI) and a check-out (CO) done with a payment method on the user medium. The payment method can be either a stored-value payment method (SVPM) or an account-based one (ABPM).

- In the case of a stored-value payment method, the amount for the current trip is charged directly to the user medium.
- In the case of an account-based payment method, the CICO events are collected and charged later, e.g. monthly.

Optionally, the user tariff parameters can be changed for the next trip/journey. For a stored-value payment method, an autoload option may be used.

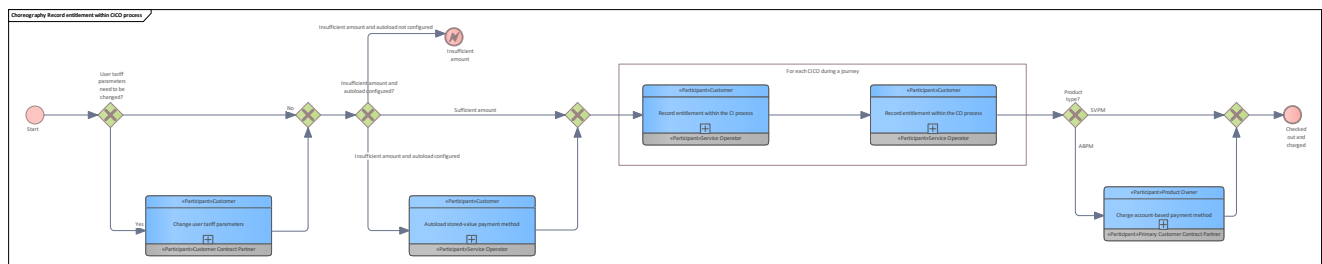


Figure 61: Record entitlement within CICO process

9.1.45.1 For each CICO during a journey

A journey may have more than one check-in and check-out.

9.1.45.2 Change user tariff parameters

A customer changes user tariff parameters (UTP) such as service class, number of passengers etc.

Please note that in order to set them to default values before entering a new journey, the customer needs to have UTPs changed on a CCP-terminal.

This can be done on each CCP terminal, thus, from the CCP's point of view, it can be an owned or a non-owned entitlement.

See [Change user tariff parameters of an owned entitlement](#) and [Change user tariff parameters of a non-owned entitlement](#).

9.1.45.3 Autoload stored-value payment method

The stored-value payment method can be configured with an autoload option. In this case, an autoload is triggered if the amount on the user medium is not sufficient for the trip or journey. The autoload can be performed on each CCP terminal, thus, from the CCP's point of view, it can be an owned or a non-owned stored-value payment method that is charged.

See also [Autoload owned stored-value payment method](#) and [Autoload non-owned stored-value payment method](#).

9.1.45.4 Record entitlement within the CI process

An entitlement is recorded within check-in process.

Please note that

- a user medium is allowed to have at most one payment method
- checking in with an electronic ticket is out of specification

See also [Record entitlement within check-in process](#).

9.1.45.5 Record entitlement within the CO process

An entitlement is recorded within check-out process.

See also [Record entitlement within check-out process](#).

9.1.45.6 Charge account-based payment method

The price of a journey based on its check-in and check-out notifications are determined and the pCCP is informed about the journey and its price to be charged.

See also [Charge account-based payment method](#).

9.1.45.7 Insufficient amount and autoload configured?

That is only relevant for the stored-value payment method.

9.1.45.8 User tariff parameters need to be changed?

In the CICO world, it is possible to change user tariff parameters (UTP) (e.g. service class, number of passengers) before checking in. If customer a decides to change them on a CCP terminal, he is asked to continue with "yes", otherwise with "no".

9.1.46 Change user tariff parameters

The customer wants to change his user tariff parameters (e.g. he would like to take a bicycle with him for the next trip/journey).

This can be done on each CCP terminal, thus, from the CCP's point of view, it can be an owned or a non-owned entitlement. The changed parameters remain valid until the customer actively resets them.

See [Change user tariff parameters of an owned entitlement](#) and [Change user tariff parameters of a non-owned entitlement](#).

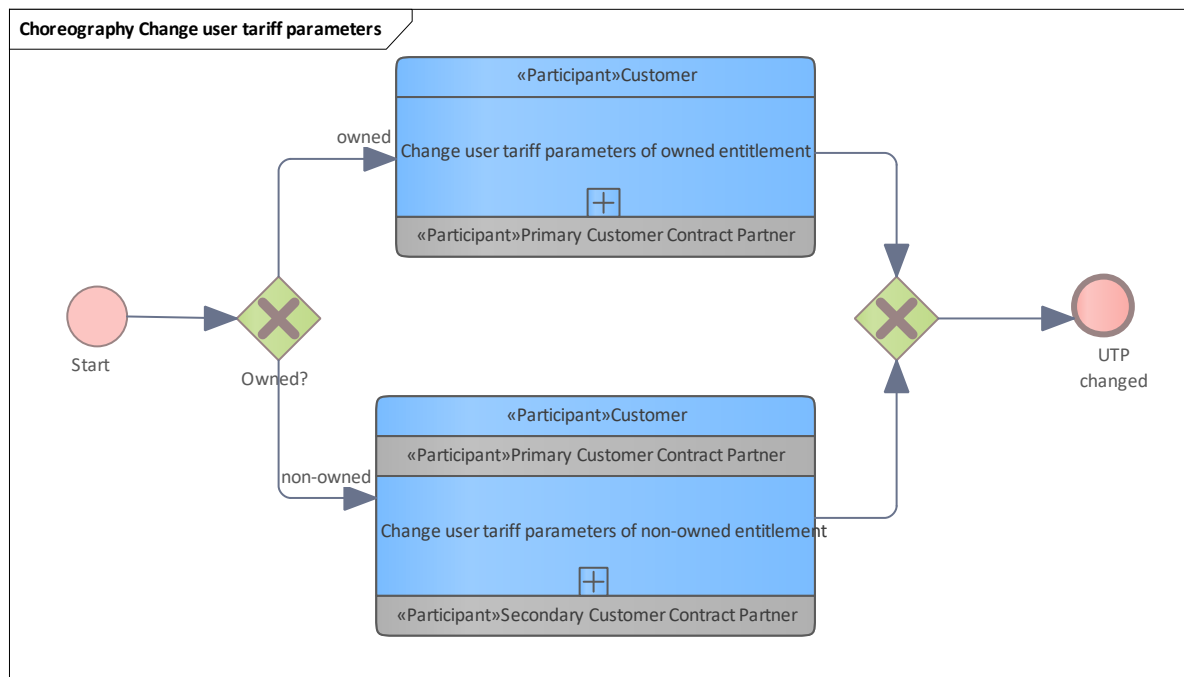


Figure 62: Change user tariff parameters

9.1.46.1 Change user tariff parameters of non-owned entitlement

Change user tariff parameters of non-owned entitlement in a user medium with an application.
See [Change user tariff parameters of a non-owned entitlement](#).

9.1.46.2 Change user tariff parameters of owned entitlement

Change user tariff parameters of an owned entitlement in a user medium with an application.
See [Change user tariff parameters of an owned entitlement](#).

9.1.47 Validation

This chapter describes the participants and the activities within the workflow "validation". BPMN Choreography is used.

9.1.48 Validate electronic ticket

Validation process for an electronic ticket. Validation makes the electronic ticket valid for the upcoming journey or trip. See [Validate electronic ticket](#).

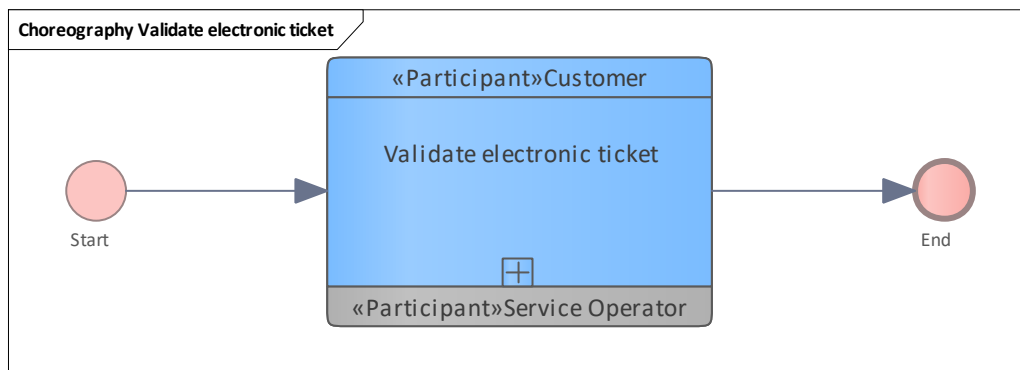


Figure 63: Validate electronic ticket

9.1.48.1 Validate electronic ticket

Validation step for an electronic ticket. Validation done by the SO terminal makes the electronic ticket valid for the upcoming journey or trip. See [Validate electronic ticket](#).

The terminal of the SO must check whether the entitlement in question of the customer's user medium is suitable for validation.

9.1.49 Monitoring and notification

This chapter describes the participants and the activities within the workflow for monitoring and monitoring-triggered notifications.

BPMN Choreography is used.

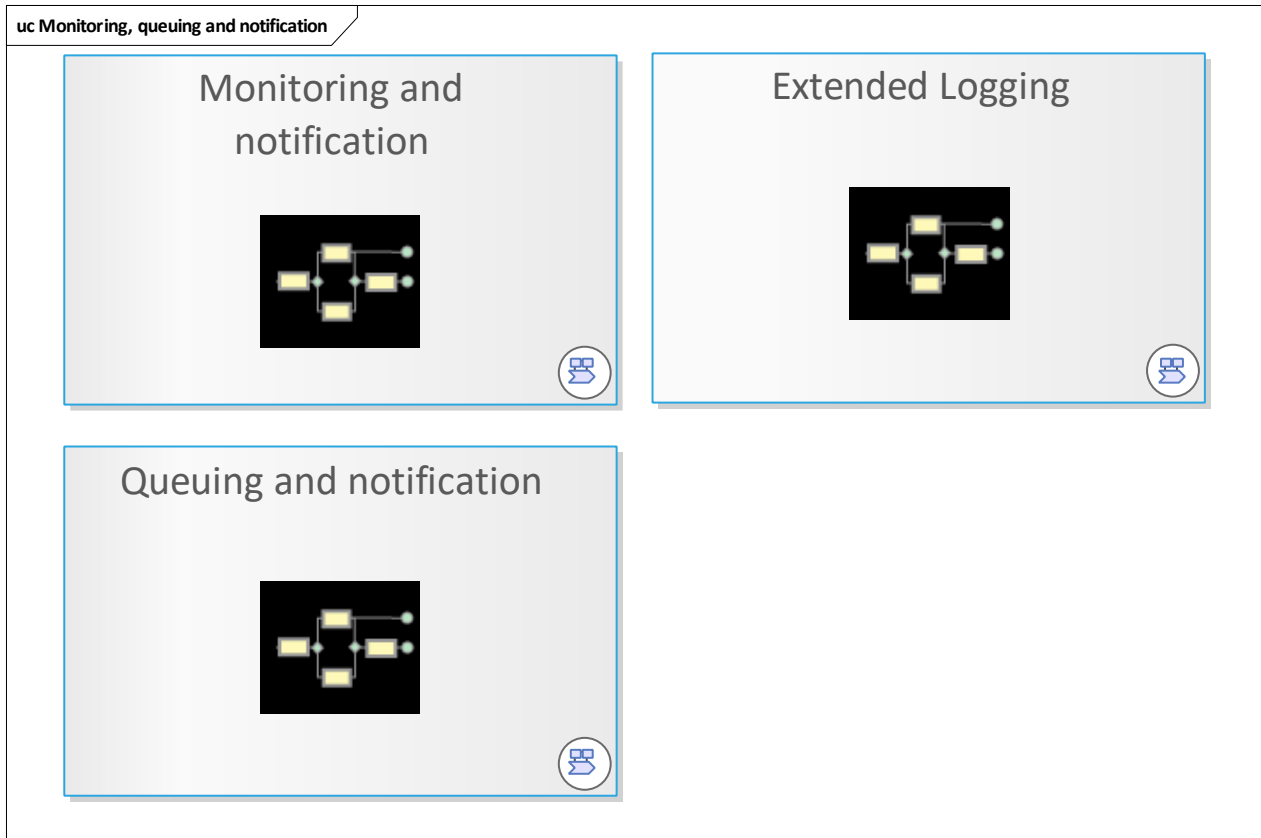


Figure 64: Monitoring, queuing and notification

9.1.50 Extended Logging

This workflow shows the extended logging in the case that invalid applications on a chip card or invalid entitlements are detected by a terminal. This is reported to the related back-office systems of the current terminal operator (SO or CCP).

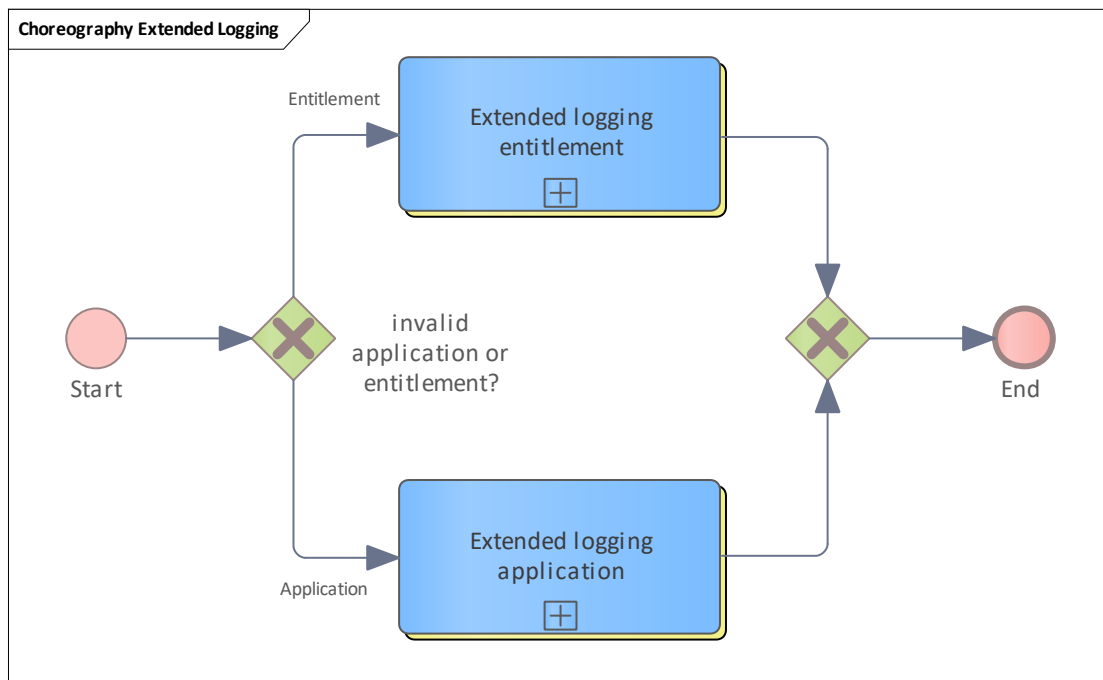


Figure 65: Extended Logging

9.1.50.1 Extended logging application

See [Extended logging for an application](#).

9.1.50.2 Extended logging entitlement

See [Extended logging for an entitlement](#).

9.1.51 Monitoring and notification

This workflow shows the monitoring-triggered notification of the relevant participants.

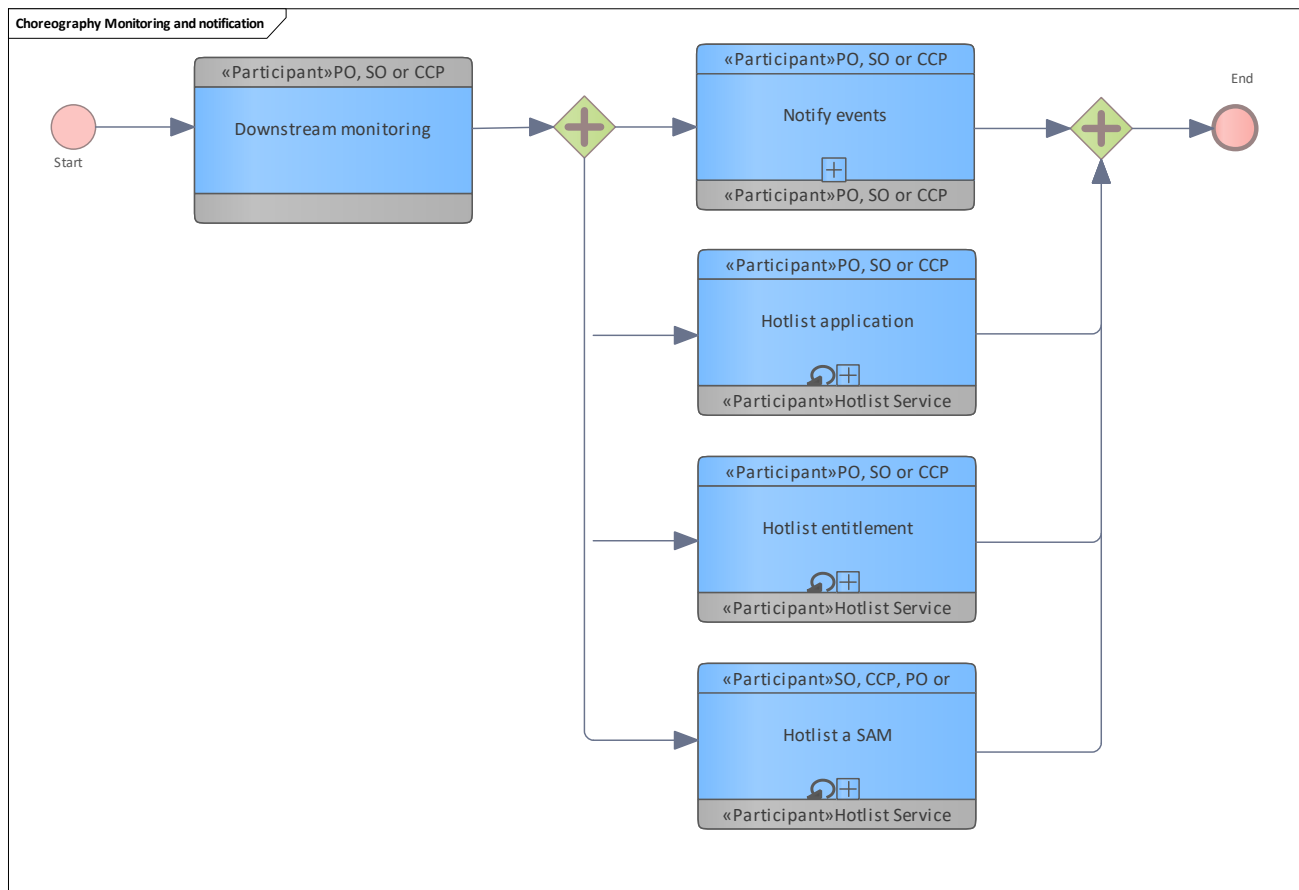


Figure 66: Monitoring and notification

9.1.51.1 Downstream monitoring

See package [Downstream checks](#). The downstream monitoring of the participant detects problems that have to be reported to the originator of the problem.

9.1.51.2 Notify events

See [Notify events](#).

9.1.52 Queuing and notification

This workflow shows the notification of the participants in the case that the CRE finally could not deliver the message(s) to the intended message recipient.

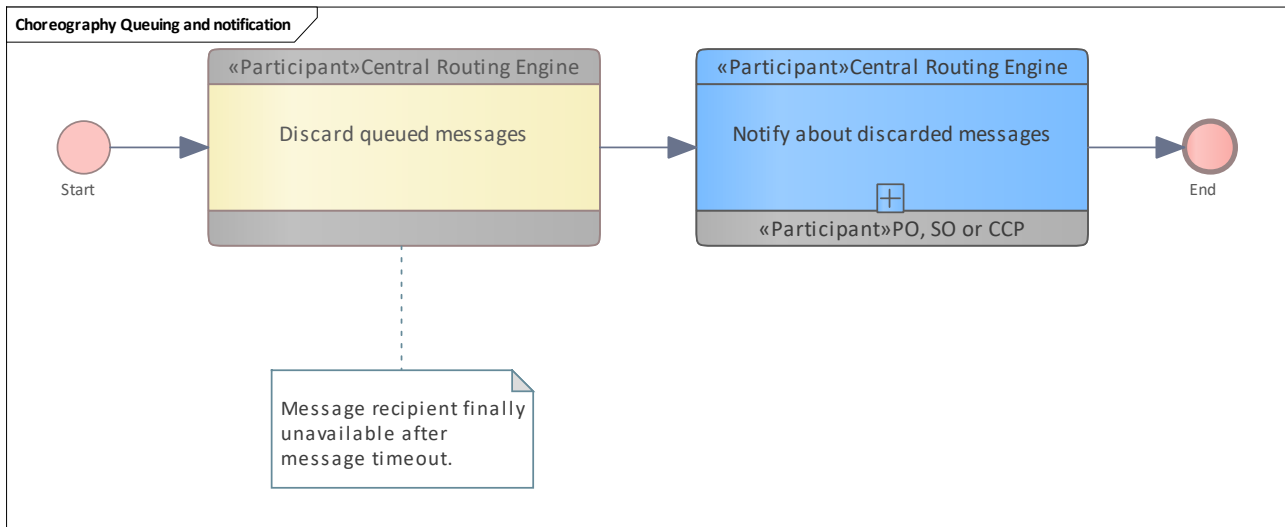


Figure 67: Queuing and notification

9.1.52.1 Discard queued messages

See Use Case [Discard queued messages](#).

9.1.52.2 Notify about discarded messages

See [Discarded messages](#).

9.1.53 Configuration

This package contains BPMN Choreography workflows all about the configuration of back-office- and hotlist-service-system as well as terminals.

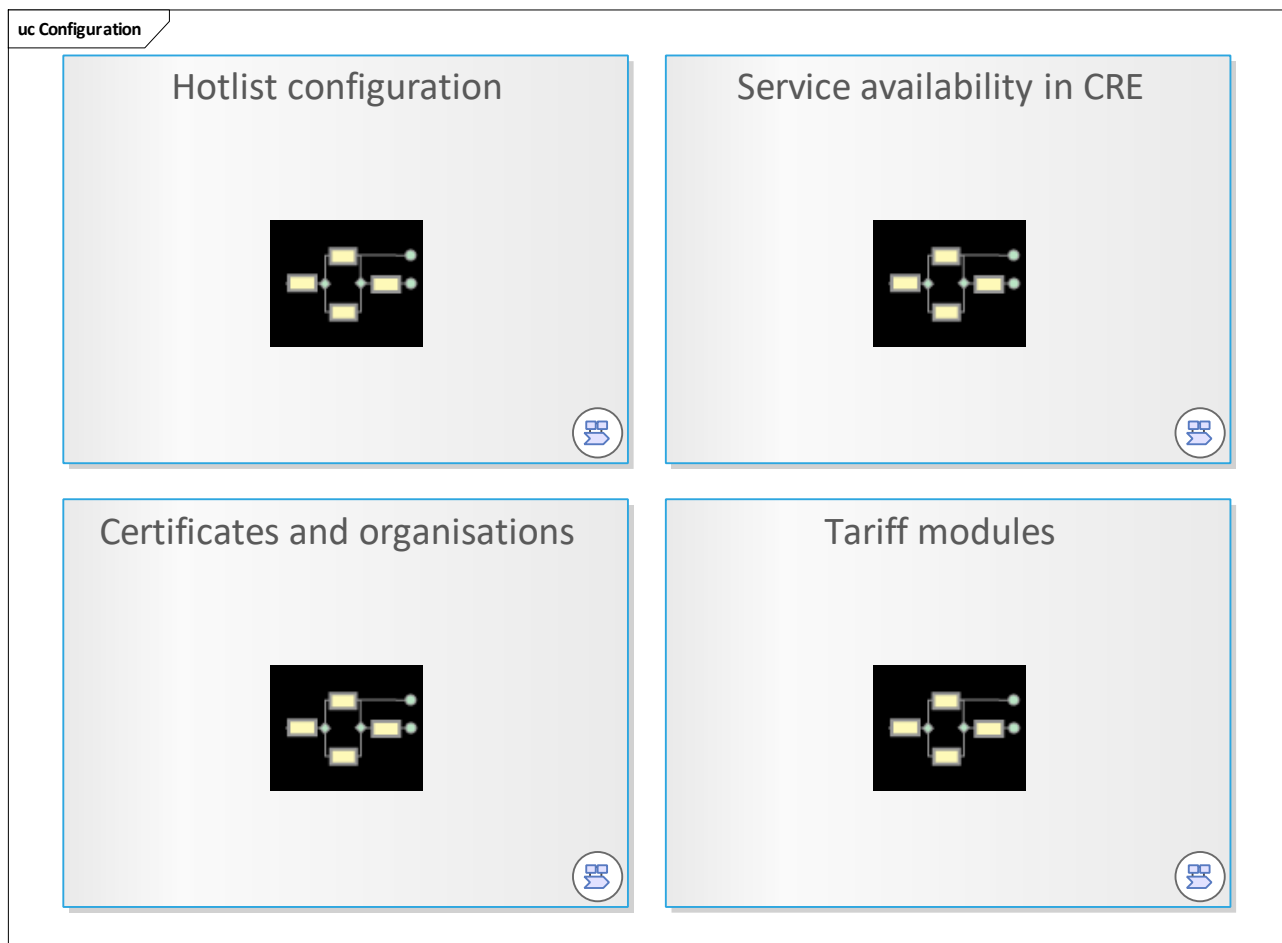


Figure 68: Configuration

9.1.54 Certificates and organisations

The BPMN Choreography-based workflow shows how to retrieve certificate and organisation data from the scheme manager as a participant being SO, PO or CCP.

This updated configuration data is used for the signature verification of attestations and further monitoring steps using the updated configuration data.

Furthermore, the common user medium checks in the terminal or further inspection tasks use the updated configuration data. See also [Check user medium with application](#) or [Check user medium without application](#) and [Inspect user medium with application](#) or [Inspect user medium without application](#).

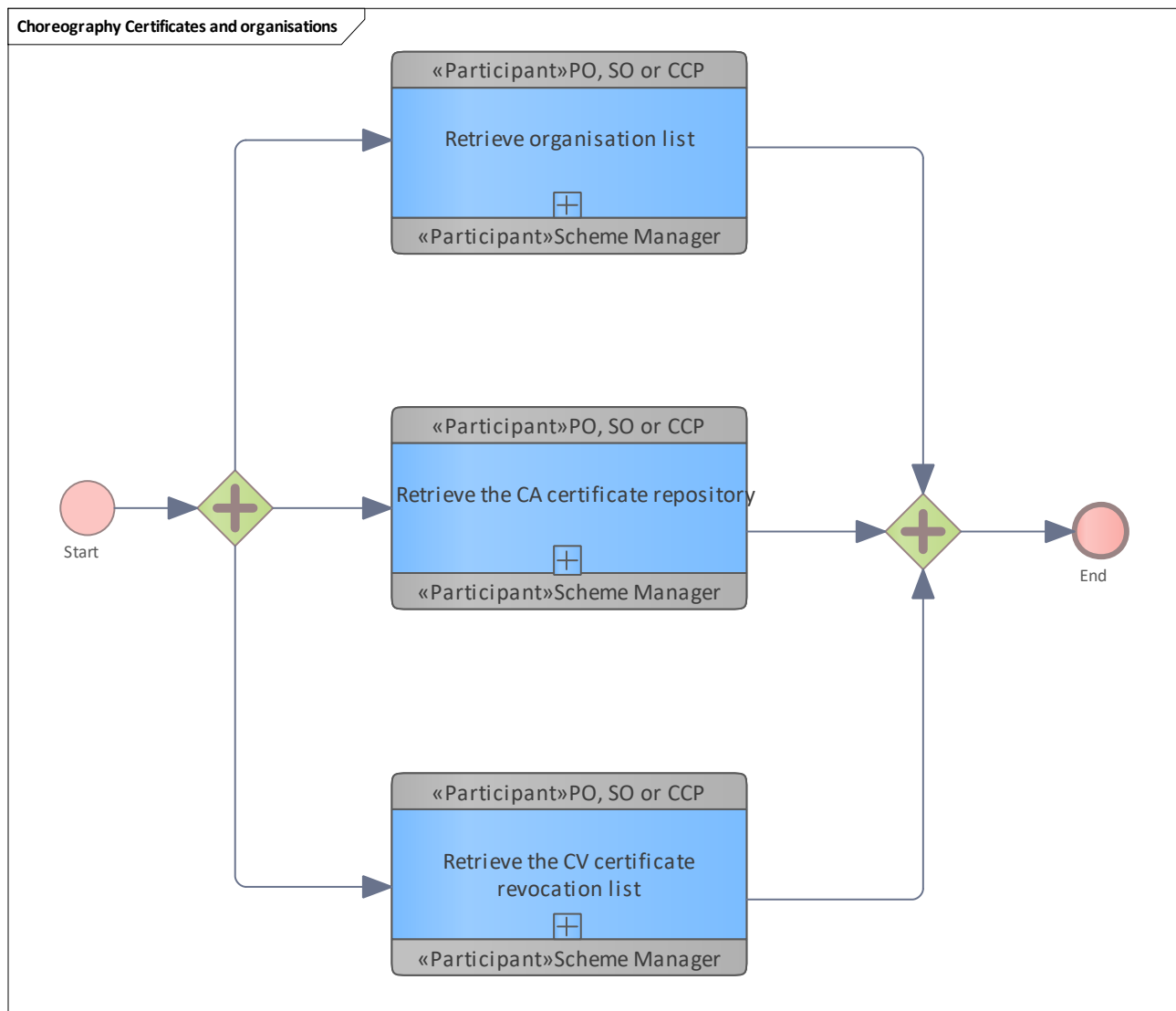


Figure 69: Certificates and organisations

9.1.54.1 Retrieve organisation list

See [Retrieve organisation list](#).

9.1.54.2 Retrieve the CA certificate repository

See [Retrieve the CA certificate repository](#)

9.1.54.3 Retrieve the CV certificate revocation list

See [Retrieve the CV certificate revocation list](#)

9.1.55 Service availability in CRE

The BPMN Choreography-based workflow shows how a participant configures the availability of his services.

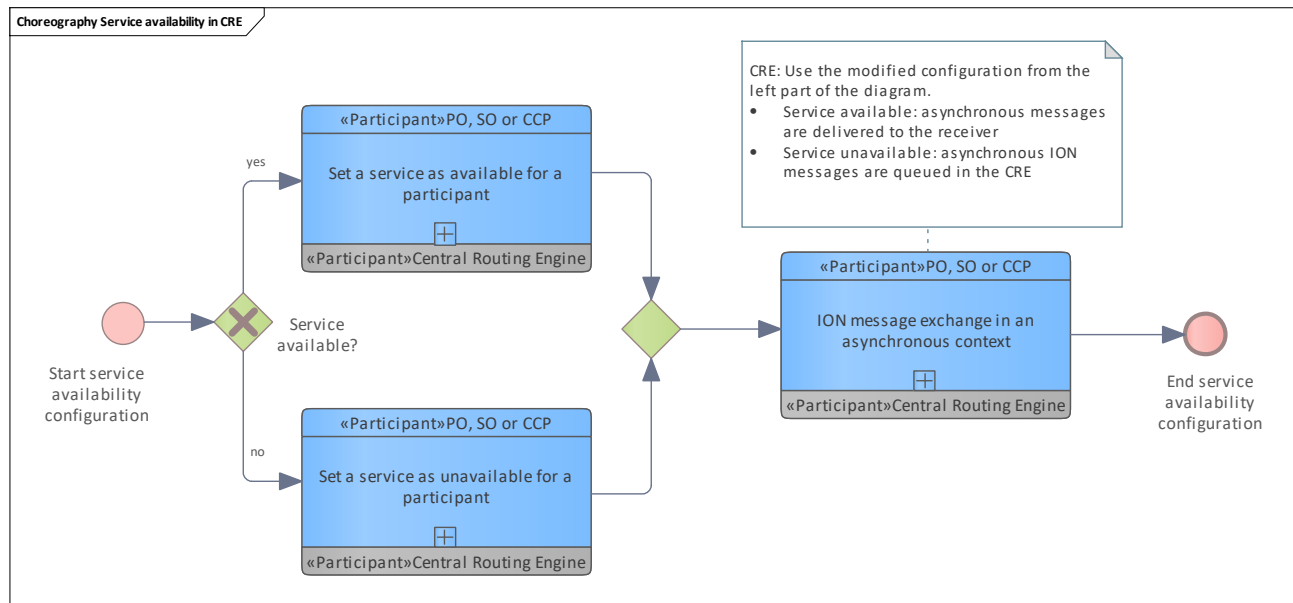


Figure 70: Service availability in CRE

9.1.55.1 ION message exchange in an asynchronous context

See [Process in an asynchronous context](#).

9.1.55.2 Set a service as available for a participant

See [Set a service as available for a participant](#).

9.1.55.3 Set a service as unavailable for a participant

See [Set a service as unavailable for a participant](#)

9.1.56 Hotlist configuration

The BPMN Choreography-based workflow shows how a product owner configures the hotlist service due to the product owner's products a related public transport companies that supports these products.

The modified acceptance configuration will effect the SO's and CCP's entitlement hotlist inventory and their hotlisting processes .

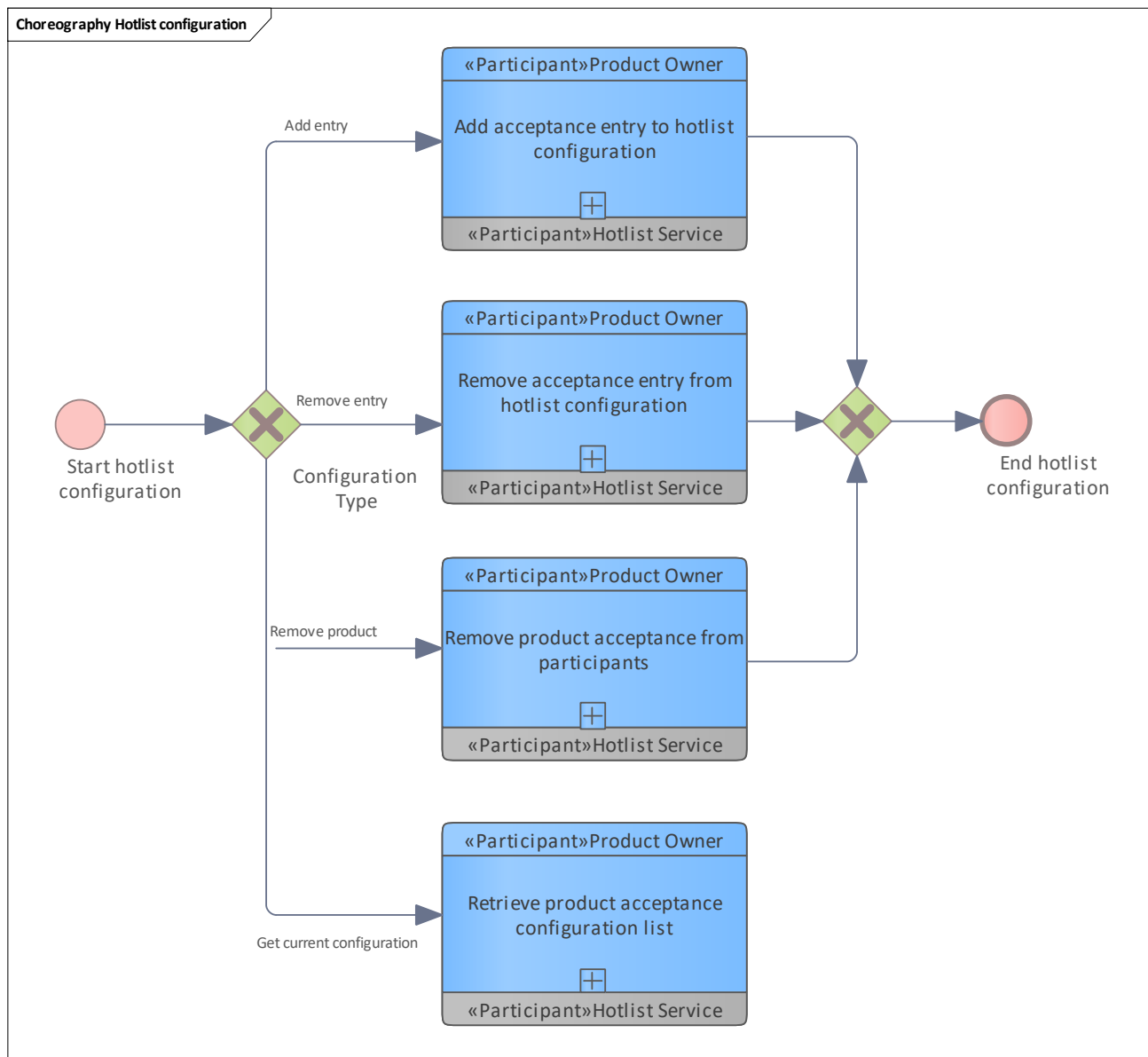


Figure 71: Hotlist configuration

9.1.56.1 Add acceptance entry to hotlist configuration

See [Add acceptance entry to hotlist configuration](#)

9.1.56.2 Remove acceptance entry from hotlist configuration

See [Remove acceptance entry from hotlist configuration](#)

9.1.56.3 Remove product acceptance from participants

See [Remove product acceptance from participants](#)

9.1.56.4 Retrieve product acceptance configuration list

See [Retrieve product acceptance configuration list](#).

9.1.57 Tariff modules

See [Distribute tariff modules](#).

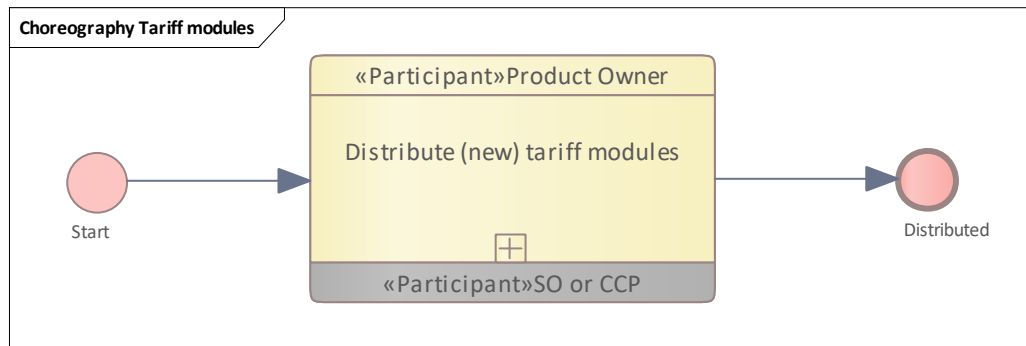


Figure 72: Tariff modules

9.1.57.1 Distribute (new) tariff modules

See [Distribute tariff modules](#).

9.1.58 SAM

This chapter describes the basic process related to SAM processes in a BPMN choreography model.

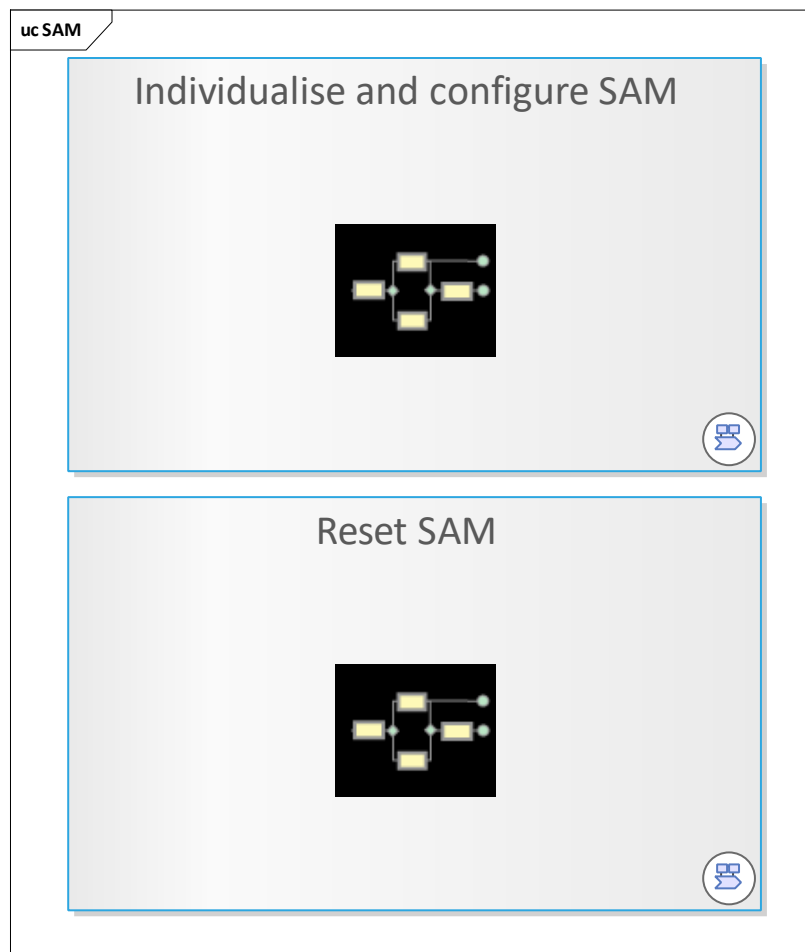


Figure 73: SAM

9.1.59 Individualise and configure SAM

Workflow for the SAM order process, starting from the Scheme Manager and Card Manufacturer to the SAM Owner. The SAMs are delivered from the card manufacturer to the scheme manager with the state Individualised.

The SAM owner orders a certain amount of SAMs together with configuration scripts. These scripts prepare the SAMs for later usage and switch their state to Configured.

The configuration of the SAMs takes place in the terminals of the SAM owner in the built-in state.

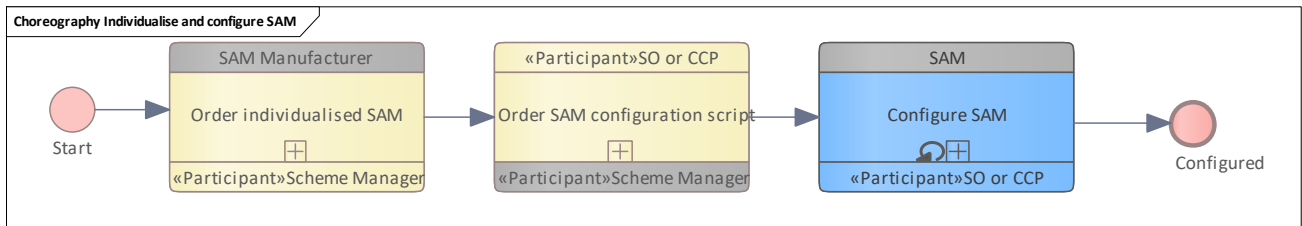


Figure 74: Individualise and configure SAM

9.1.59.1 Order individualised SAM

The scheme manager orders individualised SAMs.

See also [Individualise SAM](#).

Please note that this model gives an idea for the future version of MMS/ESH (starting in 2026).

9.1.59.2 Order SAM configuration script

SAM configuration script is requested by a SAM owner, which may take the role of SO or CCP.

See also [Demand SAM configuration script](#).

Please note that this model gives an idea for the future version of MMS/ESH (from 2026).

9.1.59.3 Configure SAM

The SAM and its configuration script are distributed. The SAM is updated accordingly.

See also [Distribute and update SAM](#).

9.1.60 Reset SAM

In this workflow, the SAM is already installed in a terminal or the stock of the SAM owner.

Resetting a SAM brings the SAM's state from *Configured* back to *Individualised*.

This may be used if a SAM should temporarily unusable or as preparation for a re-configuration.

The SAM owner orders reset scripts for its SAMs in the ESH of the Scheme Manager. Then the scripts are distributed and executed, normally directly in the terminals which are fitted with the involved SAMs.

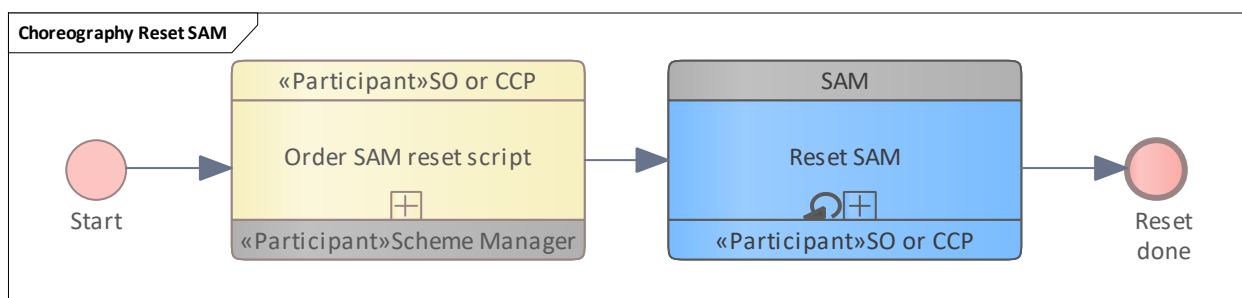


Figure 75: Reset SAM

9.1.60.1 Order SAM reset script

SAM reset script is requested by a SAM owner, which may take the role of SO or CCP. See also [Demand SAM reset script](#).

Please note that this model gives an idea for the future version of MMS/ESH (2026).

9.1.60.2 Start

The configuration of the SAM can be reset for different reasons - for example:

- the SAM will be transferred from one company to another company
- the SAM is no longer in use

9.1.60.3 Reset SAM

The reset script is received and distributed. The SAM is reset accordingly. See also [Distribute and reset SAM](#).

9.1.61 User Medium

This chapter describes the basic process related to user medium processes in a BPMN choreography model.

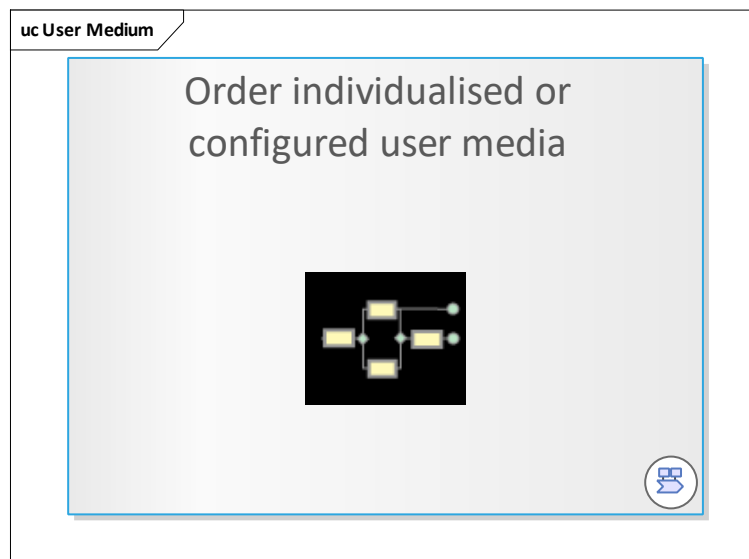


Figure 76: User Medium

9.1.62 Order individualised or configured user media

Workflow for the user medium order process, starting from the CCP via Scheme Manager and Card Manufacturer and/or Mass Personaliser.

The CCP can order user media ready-to-use with certificates and optionally personalised.

The other possibility is the order individualised user media without certificates and/or personal data. In this case, that has to be done in a downstream process.

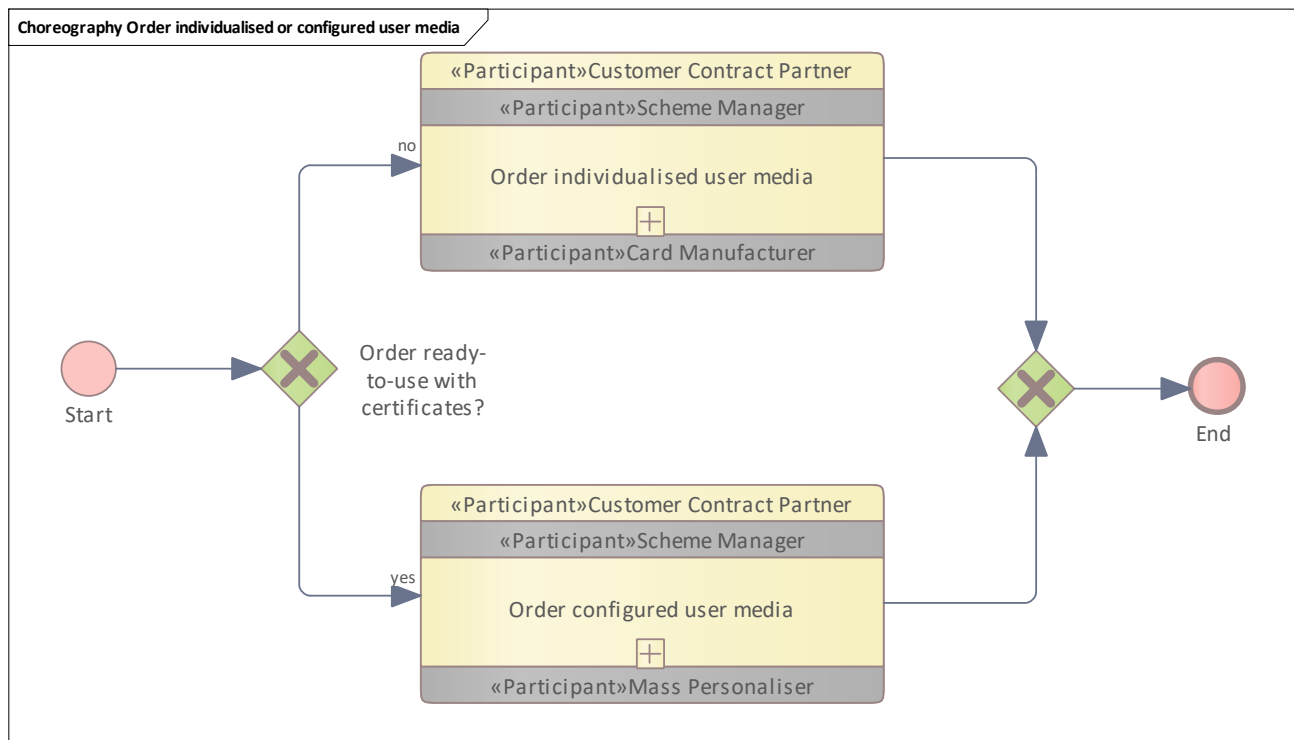


Figure 77: Order individualised or configured user media

9.1.62.1 Order configured user media

See [Configure UM](#)

9.1.62.2 Order individualised user media

See [Individualise UM](#)

9.2 Layer 2 - Basic Processes as BPMN Collaboration

This chapter describes the participants and the activities within a basic process. BPMN Collaboration is used to show the interaction of lanes and pools belonging to the different participants.

9.2.1 Notification process patterns

This chapter deals with the patterns which are widely used in the model especially for notification processes.

If you understand these patterns, you will understand the structure of most of the basic processes and use cases.

The different actions that can be executed by the user medium fall into different categories with respect to who is authorised to execute them. The important distinction here is whether the executing organisation is also the owner of the target object (UM application for actions targeting the application, entitlement for actions targeting the entitlement).

For some actions, the executing organisation is never the object owner (i.e. for all actions only relevant to Service Operators, which are never owners of UMs or entitlements). Others may only be performed by the owner, i.e. unblocking (outside of ordered action execution). The rest may be performed by organisations that may or may not be the object owner, e.g. debiting a payment method or blocking entitlement or application.

The actions (and their notifications) fall into four different categories:

- Entitlement owned
- Entitlement non-owned
- Application owned
- Application non-owned

Depending on the category into which the action falls, the handling of their notifications with respect to potential forwarding to the owner and with respect to when the contractual processing happens may be different.

The above distinction also results in a division with regard to the perspectives that are very often used in the specification model:

- The perspective from an operational point of view: after the message has been reported by a terminal, this message must be processed in the sense of a registration and certain checks on the consistency and authenticity of the message. Afterwards, the message is reported to the product owner.
- The perspective from the product owner's point of view: after the message has been received, extensive checks are carried out. If necessary, the message is then forwarded to the primary CCP.
- The perspective from the point of view of the primary CCP: after the notification receipt, a few more checks are carried out. Afterwards, processes outside the (((etiCORE specification have to be initiated, such as the booking.

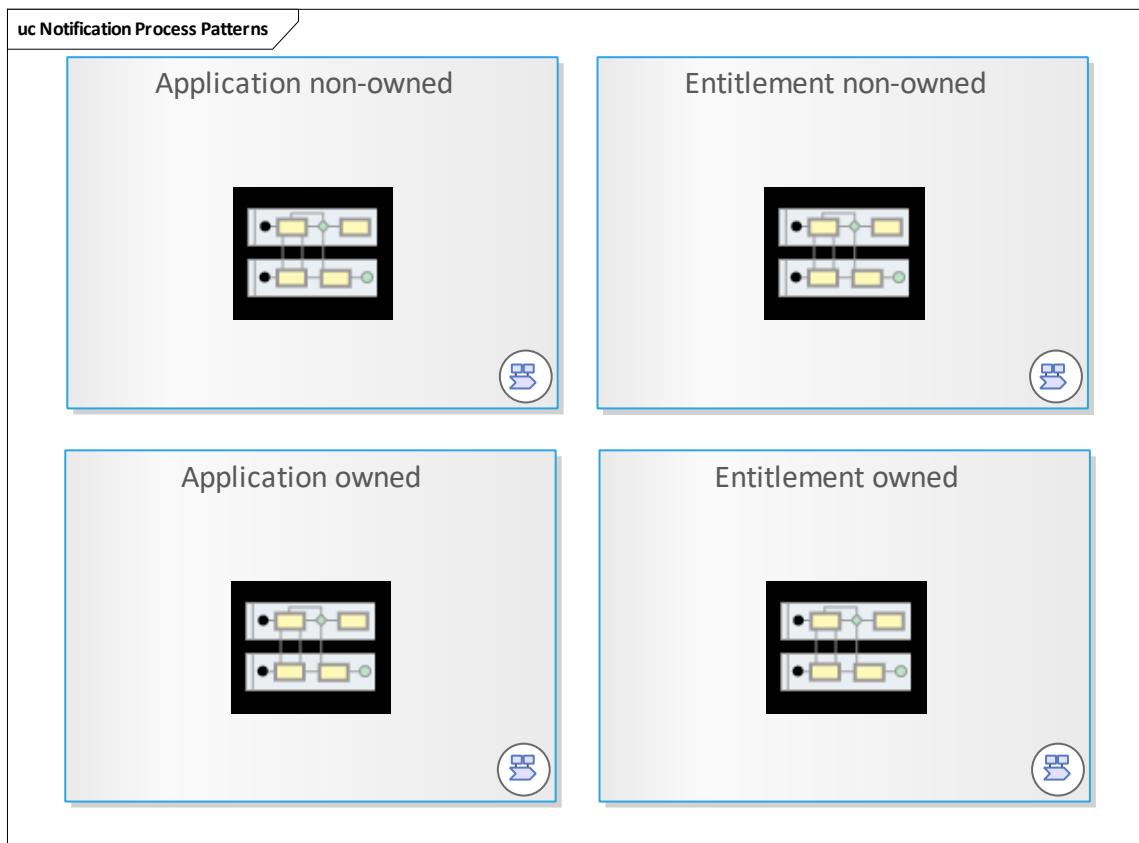


Figure 78: Notification Process Patterns

9.2.2 Application non-owned

The basic process that shows a notification chain starting in a terminal of an organisation which is not the owner of the application instance.

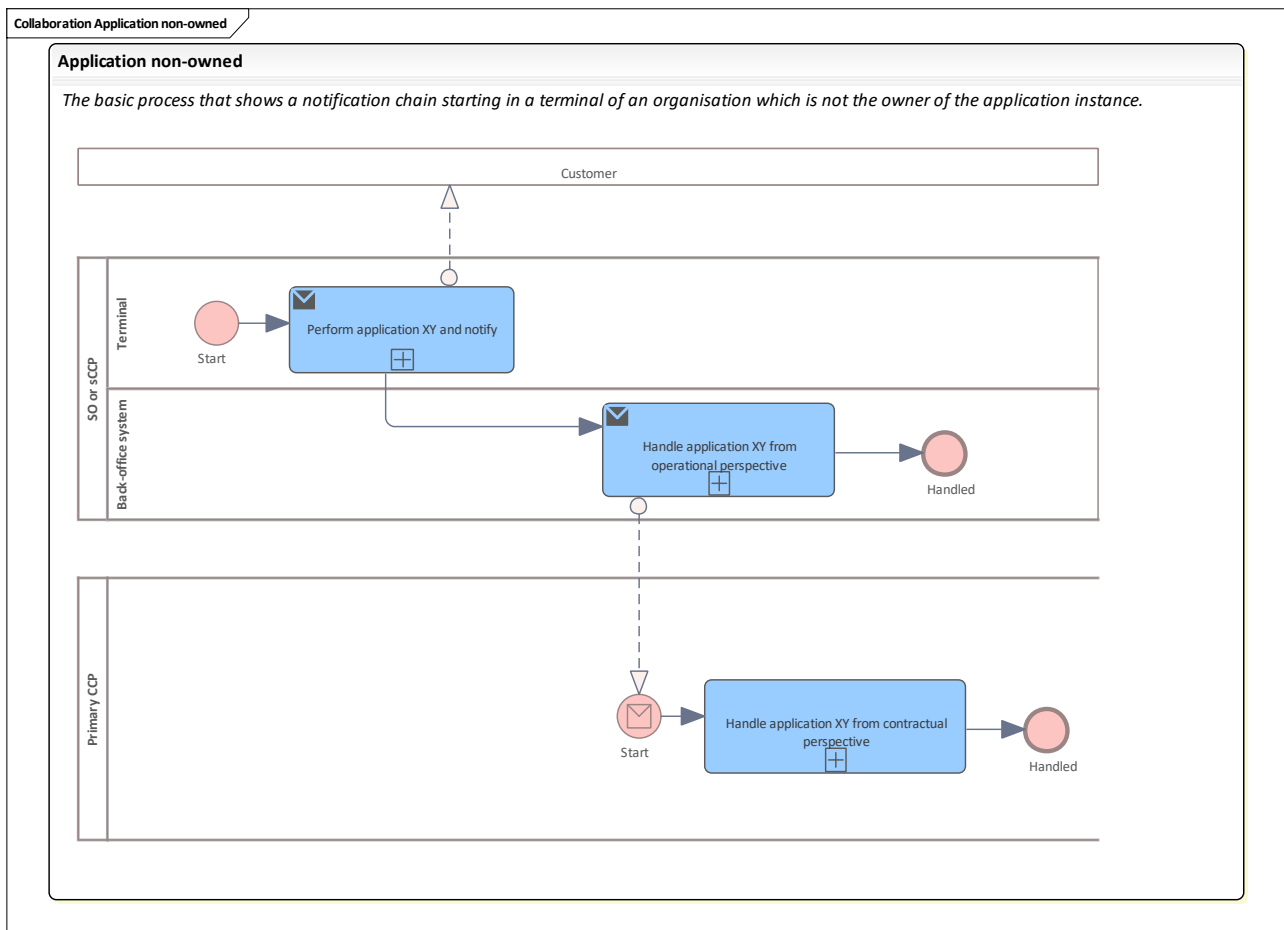


Figure 79: Application non-owned

9.2.2.1 Primary CCP

Primary customer contract partner (always contacted in "non-owned" scenarios).

9.2.2.1.1 Handle application XY from contractual perspective

See [Handle application XY notification from contractual perspective](#)

9.2.2.2 SO or sSCP

Third-party service operator or secondary customer contract partner in non-owned scenarios.

9.2.2.2.1 Back-office system

Lane for back-office system

1.1.1.1.1.1 Handle application XY from operational perspective

See [Handle application XY notification from operational perspective](#)

9.2.2.2.2 Terminal

Lane for terminal

1.1.1.1.1.2 Perform application XY and notify

See [Perform application XY and notify](#)

9.2.2.3 Customer

Customer for all scenarios.

9.2.3 Application owned

Basic process that shows a notification chain starting in a terminal of an organisation which is the owner of the application instance.

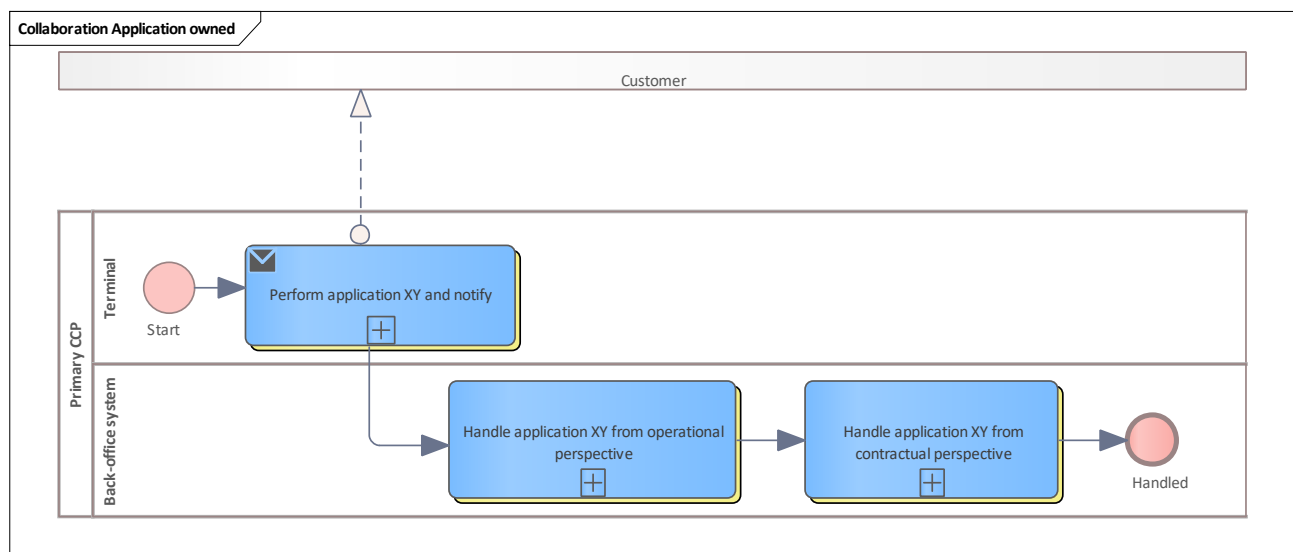


Figure 80: Application owned

9.2.3.1 Customer

Customer for all scenarios.

9.2.3.2 Primary CCP

Primary customer contract partner (always initiator in "owned" scenarios).

9.2.3.2.1 Back-office system

Lane for back-office system

1.1.1.1.1.3 Handle application XY from contractual perspective

See [Handle application XY notification from contractual perspective](#)

1.1.1.1.1.4 Handle application XY from operational perspective

See [Handle application XY notification from operational perspective](#)

9.2.3.2.2 Terminal

Lane for terminal

1.1.1.1.1.5 Perform application XY and notify

See [Perform application XY and notify](#)

9.2.4 Entitlement non-owned

The basic process that shows a notification chain starting in a terminal of an organisation which is not the owner of the entitlement.

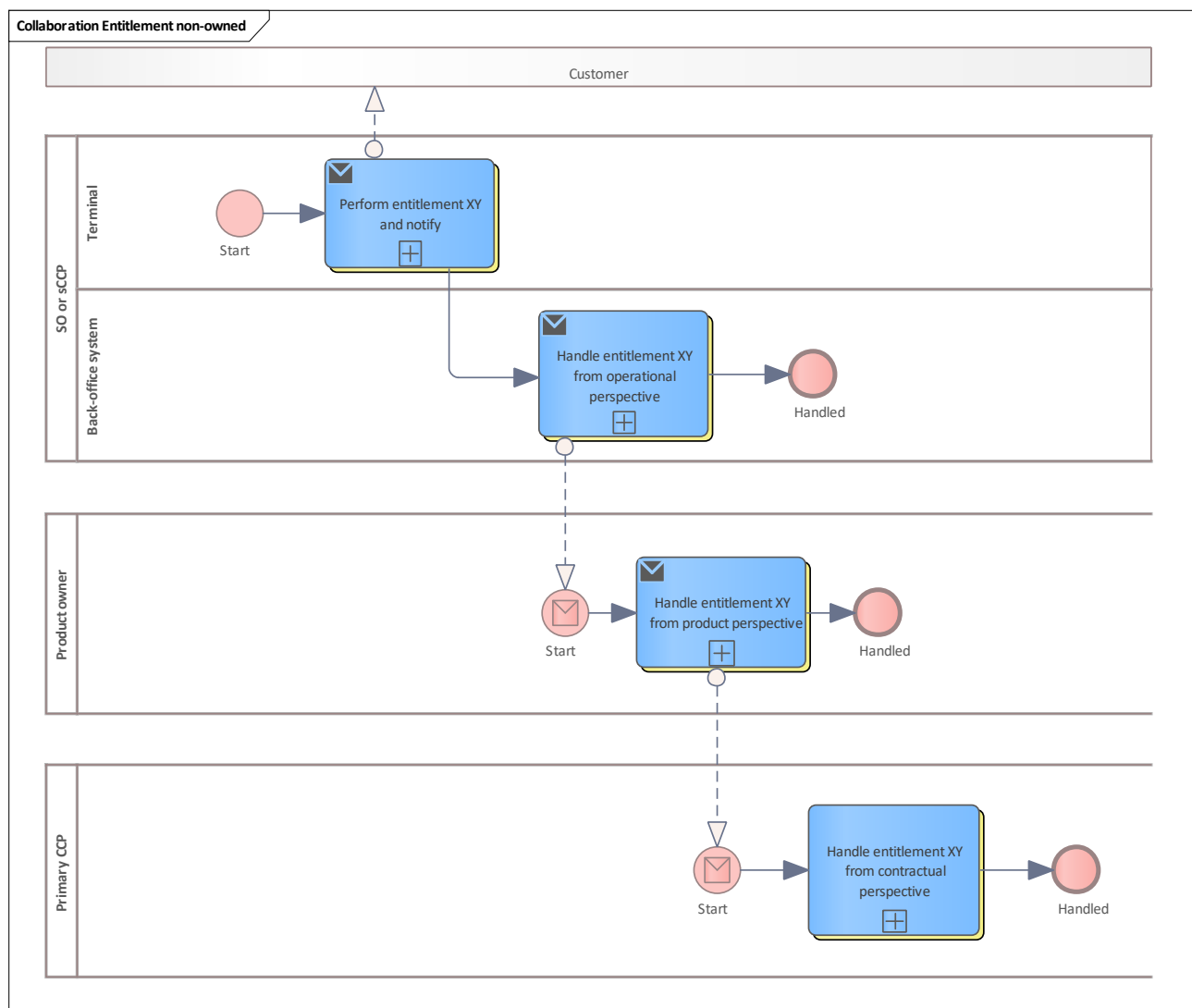


Figure 81: Entitlement non-owned

9.2.4.1 SO or sCCP

Third-party service operator or secondary customer contract partner in non-owned scenarios.

9.2.4.1.1 Terminal

Lane for terminal

1.1.1.1.1.6 Perform entitlement XY and notify

See [Perform entitlement XY and notify](#)

9.2.4.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.7 Handle entitlement XY from operational perspective

See [Handle entitlement XY notification from operational perspective](#)

9.2.4.2 Product owner

Product owner to be notified for all entitlement scenarios (owned or non-owned scenario).

9.2.4.2.1 Handle entitlement XY from product perspective

See [Handle entitlement XY notification from product perspective](#)

9.2.4.3 Primary CCP

Primary customer contract partner (always contacted in "non-owned" scenarios).

9.2.4.3.1 Handle entitlement XY from contractual perspective

See [Handle entitlement XY notification from contractual perspective](#)

9.2.5 Entitlement owned

The basic process that shows a notification chain starting in a terminal of an organisation which is the owner of the entitlement.

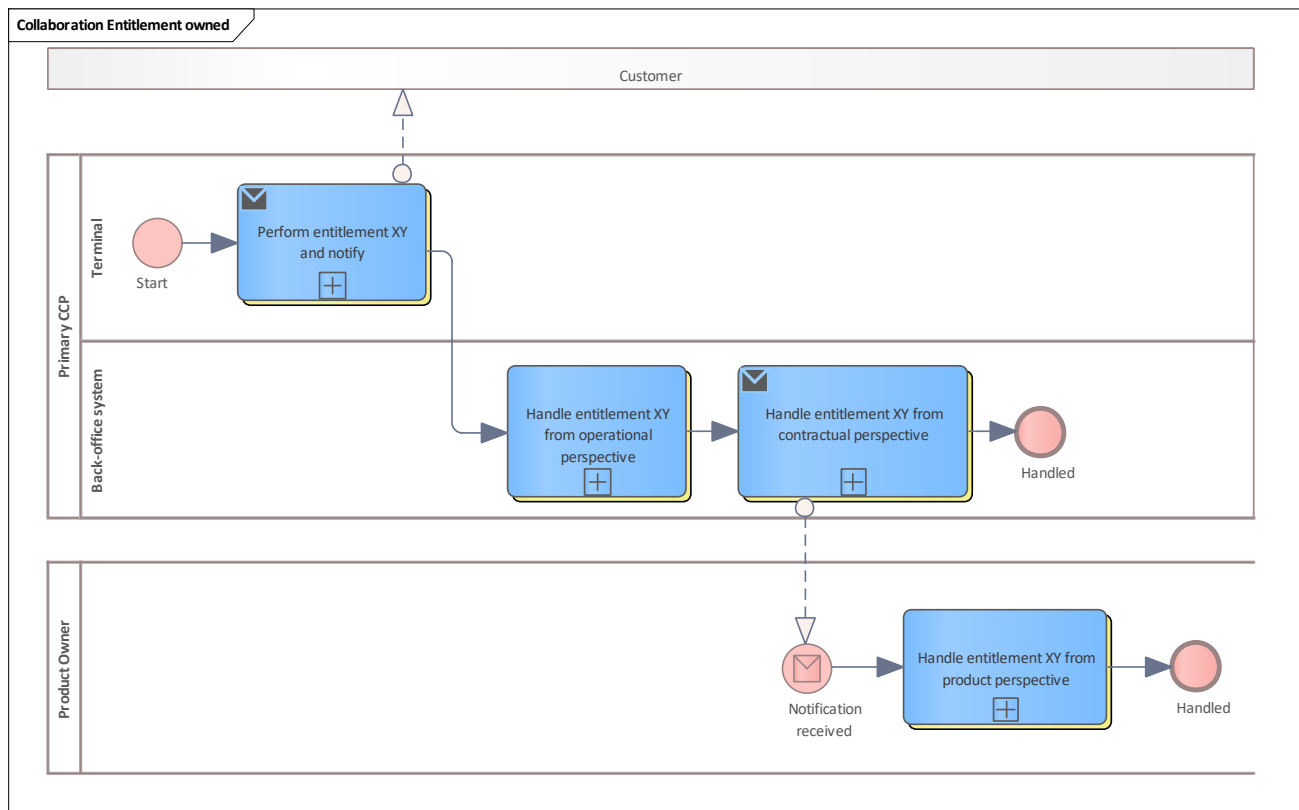


Figure 82: Entitlement owned

9.2.5.1 Primary CCP

Primary customer contract partner (always in "owned" scenarios).

9.2.5.1.1 Terminal

Lane for terminal

1.1.1.1.1.8 Perform entitlement XY and notify

See [Perform entitlement XY and notify](#)

9.2.5.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.9 Handle entitlement XY from operational perspective

See [Handle entitlement XY notification from operational perspective](#)

1.1.1.1.1.10 Handle entitlement XY from contractual perspective

See [Handle entitlement XY notification from contractual perspective](#)



9.2.5.2 Product Owner

Product owner to be notified for entitlement-owned scenarios.

9.2.5.2.1 Handle entitlement XY from product perspective

See [Handle entitlement XY notification from product perspective](#)

9.2.6 Sale

This chapter contains the basic processes that are related to any sale scenarios.

9.2.7 Issue entitlement starting in back-office system

Issue an entitlement to a user medium.

Technical name due to [ProcessNameEnum](#): **IssueEntitlement**

This basic process shows the issuance of an entitlement when the customer is in the customer centre of the CCP. The customer may be new in the system and will be registered. The new entitlement data is collected in the CCP back-office system and the data is transferred to the terminal to write it to the user medium.

The next step is notifying the PO about the new entitlement together with a registration of the entitlement in the CCP back-office system for paying, booking and monitoring purposes.

Note: the special case of issuing an entitlement triggered by action lists (action management) is not represented here.

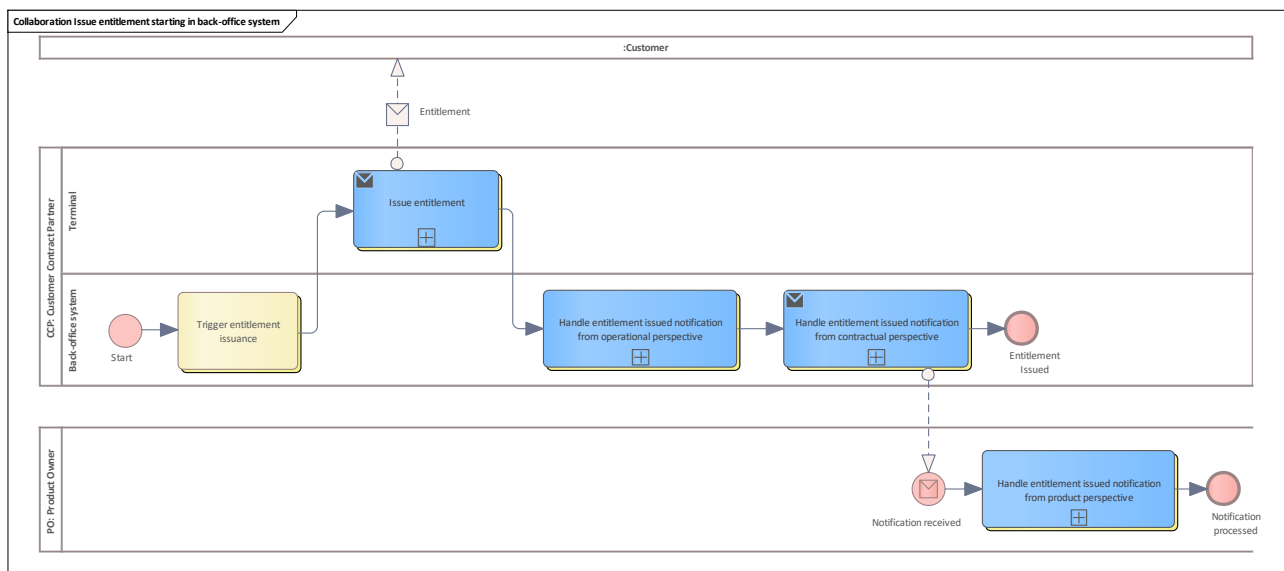


Figure 83: Issue entitlement starting in back-office system

9.2.7.1 CCP

See [Customer Contract Partner](#)

9.2.7.1.1 Terminal

Lane for terminal

1.1.1.1.1.11 Issue entitlement

See [Issue entitlement](#).

9.2.7.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.12 Trigger entitlement issuance

The back-office system prepares data for a new entitlement due to a customer request or other internal reasons (e.g. automatic renewal of an entitlement triggered by the entitlement management module of the CCP back-office system).

1.1.1.1.1.13 Handle entitlement issued notification from contractual perspective

See [Handle entitlement issued notification from contractual perspective](#).

1.1.1.1.1.14 Handle entitlement issued notification from operational perspective

See [Handle entitlement issued notification from operational perspective](#).

9.2.7.2 PO

See [Product Owner](#)

9.2.7.2.1 Handle entitlement issued notification from product perspective

See [Handle entitlement issued notification from product perspective](#).

9.2.8 Issue entitlement starting in terminal

Issue an entitlement to a user medium.

Technical name due to [ProcessNameEnum](#): **IssueEntitlement**

This basic process shows the issuance of an entitlement when the customer orders a new entitlement using a CCP terminal.

The data of the new entitlement that was written to the user medium is transferred to the CCP back-office system.

The next step is notifying the PO about the new entitlement together with a registration of the entitlement in the CCP back-office system for paying, booking and monitoring purposes.

Note: the special case of issuing an entitlement triggered by action lists (action management) is not represented here.

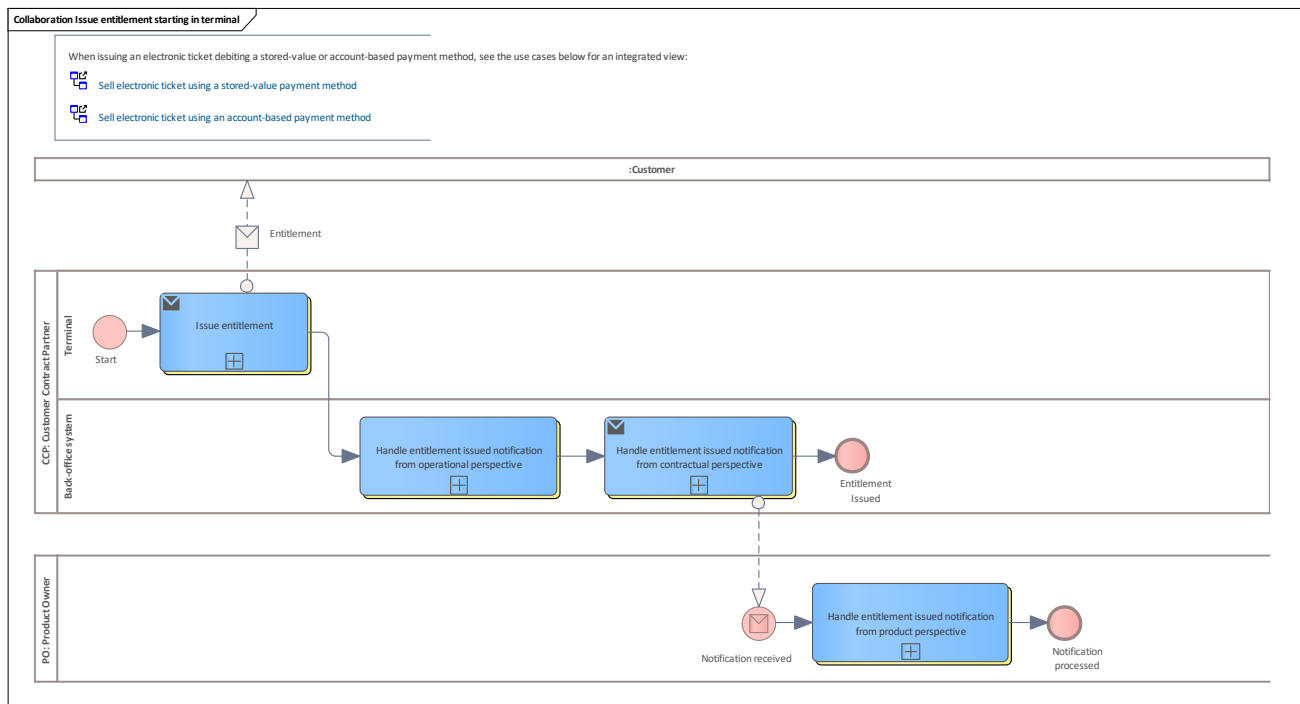


Figure 84: Issue entitlement starting in terminal

9.2.8.1 CCP

See [Customer Contract Partner](#)

9.2.8.1.1 Terminal

Lane for terminal

1.1.1.1.1.15 Issue entitlement

See [Issue entitlement](#).

9.2.8.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.16 Handle entitlement issued notification from contractual perspective

See [Handle entitlement issued notification from contractual perspective](#).

1.1.1.1.1.17 Handle entitlement issued notification from operational perspective

See [Handle entitlement issued notification from operational perspective](#).

9.2.8.2 PO

See [Product Owner](#)

9.2.8.2.1 Handle entitlement issued notification from product perspective

See [Handle entitlement issued notification from product perspective](#).

9.2.9 Issue static entitlement

Issue a static entitlement as a barcode or binary structure e.g. to a smartphone.

Technical name due to [ProcessNameEnum](#): **IssueStaticEntitlement**

This basic process shows the issuance of an entitlement when the customer orders a new static entitlement using a CCP terminal.

The data of the new static entitlement is transferred to the CCP back-office system.

The next step is notifying the PO about the new static entitlement together with a registration of the static entitlement in the CCP back-office system for paying, booking and monitoring purposes.

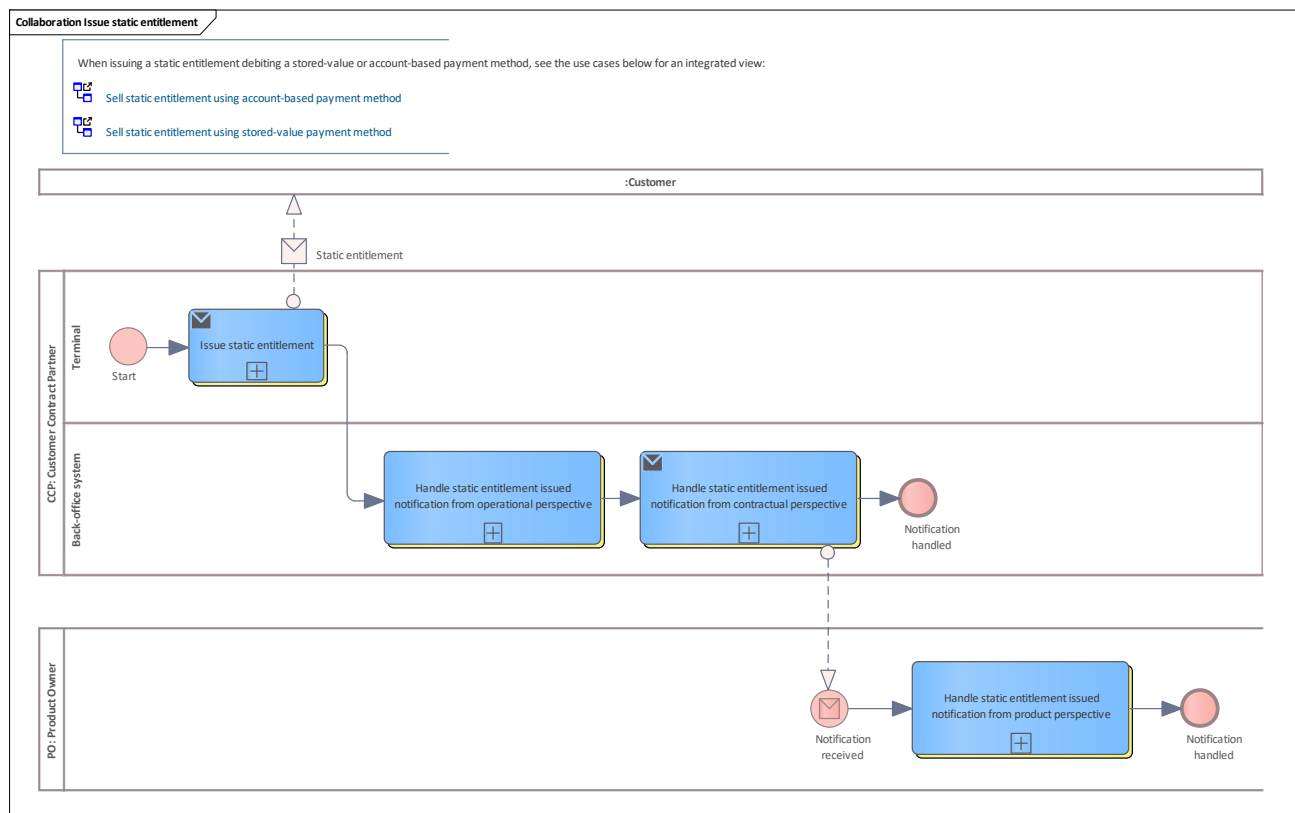


Figure 85: Issue static entitlement

9.2.9.1 CCP

See [Customer Contract Partner](#)

9.2.9.1.1 Terminal

Lane for terminal

1.1.1.1.1.18 Issue static entitlement

See [Issue static entitlement](#).

9.2.9.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.19 Handle static entitlement issued notification from contractual perspective

See [Handle static entitlement issued notification from contractual perspective](#).

1.1.1.1.1.20 Handle static entitlement issued notification from operational perspective

See [Handle static entitlement issued notification from operational perspective](#).

9.2.9.2 PO

See [Product Owner](#)

9.2.9.2.1 Handle static entitlement issued notification from product perspective

See [Handle static entitlement issued notification from product perspective](#).

9.2.10 Recharge non-owned stored-value payment method

This basic process describes a customer-triggered recharge of a stored-value payment method. Technical name due to [ProcessNameEnum](#): **RechargeStoredValuePaymentMethod**.

The process starts in a CCP terminal. The payment method was not issued by the CCP that operates the terminal.

In this case, the typical template process for [Entitlement non-owned](#) is the foundation of this basic process.

After recharging the stored-value payment method on the user medium by a certain amount, the terminal sends the data to the CCP back-office system.

This CCP back-office system does some basic monitoring steps and notifies this transaction to the PO back-office system. The PO back-office system receives this notification and does checks and monitoring before forwarding it to the pCCP back-office system.

Finally, the pCCP back-office system registers the recharge action and does the contractual checks and monitoring.

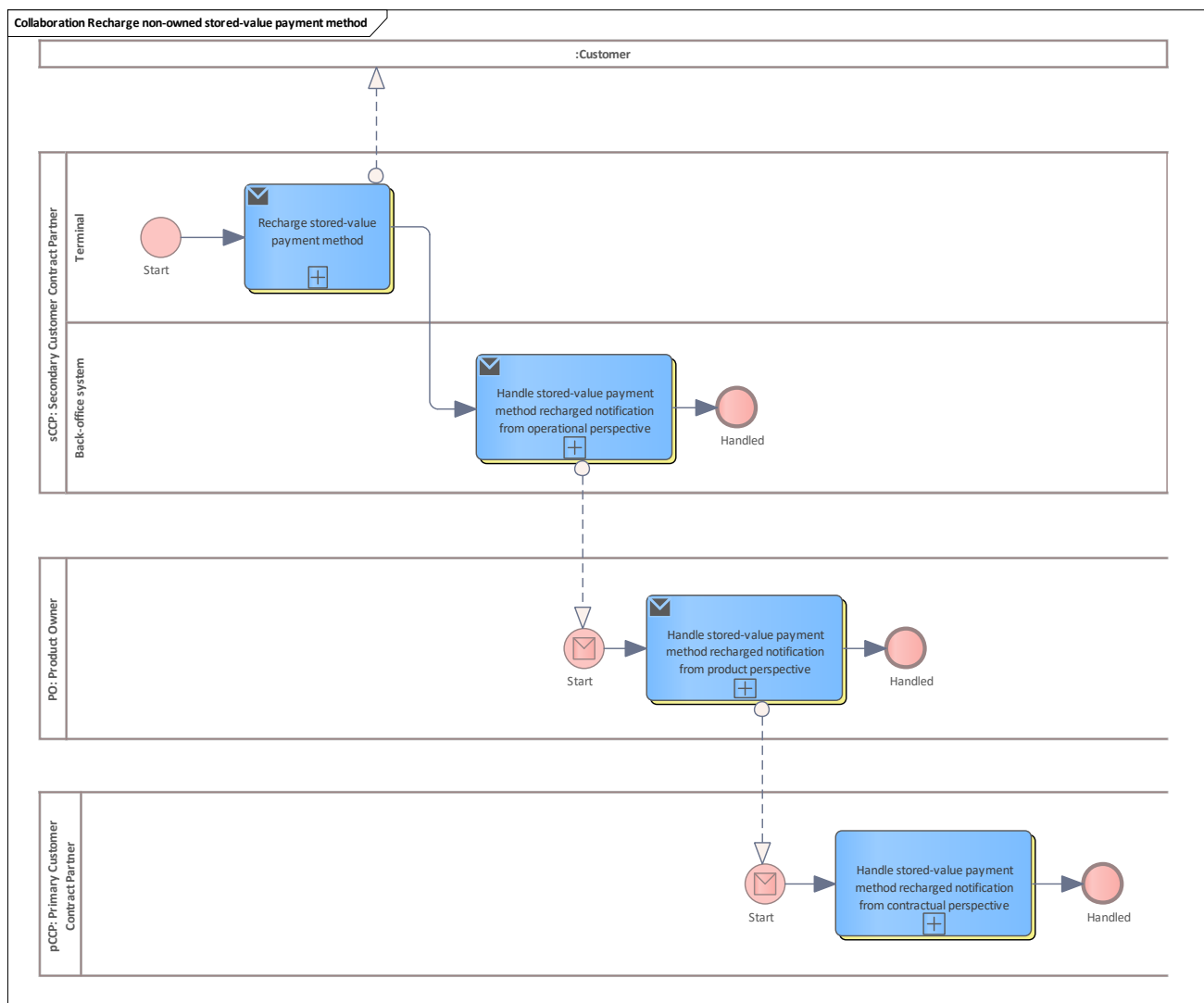


Figure 86: Recharge non-owned stored-value payment method

9.2.10.1 sCCP

See [Secondary Customer Contract Partner](#)

9.2.10.1.1 Terminal

Lane for terminal

1.1.1.1.1.21 Recharge stored-value payment method

See [Recharge stored-value payment method](#).

9.2.10.1.2 Back-office system

Lane for back-office system



1.1.1.1.1.22 Handle stored-value payment method recharged notification from operational perspective

See [Handle stored-value payment method recharged notification from operational perspective](#).

9.2.10.2 PO

See [Product Owner](#)

9.2.10.2.1 Handle stored-value payment method recharged notification from product perspective

See [Handle stored-value payment method recharged notification from product perspective](#).

9.2.10.3 pCCP

See [Primary Customer Contract Partner](#)

9.2.10.3.1 Handle stored-value payment method recharged notification from contractual perspective

See [Handle stored-value payment method recharged notification from contractual perspective](#).

9.2.11 Recharge owned stored-value payment method

This basic process describes a customer-triggered recharge of a stored-value payment method. The process starts in a CCP terminal.

Technical name due to [ProcessNameEnum](#): **RechargeStoredValuePaymentMethod**.

The payment method was issued by the CCP that operates the terminal.

In this case, the typical template process for [Entitlement owned](#) is the foundation of this basic process.

After recharging the stored-value payment method on the user medium by a certain amount, the terminal sends the data to the pCCP back-office system.

This pCCP back-office system does some basic monitoring steps. Since the pCCP is the owner, the pCCP back-office system registers the recharge action and does the contractual checks and monitoring.

Then the pCCP notifies this transaction to the PO back-office system. The PO back-office system receives this notification and does checks and monitoring.

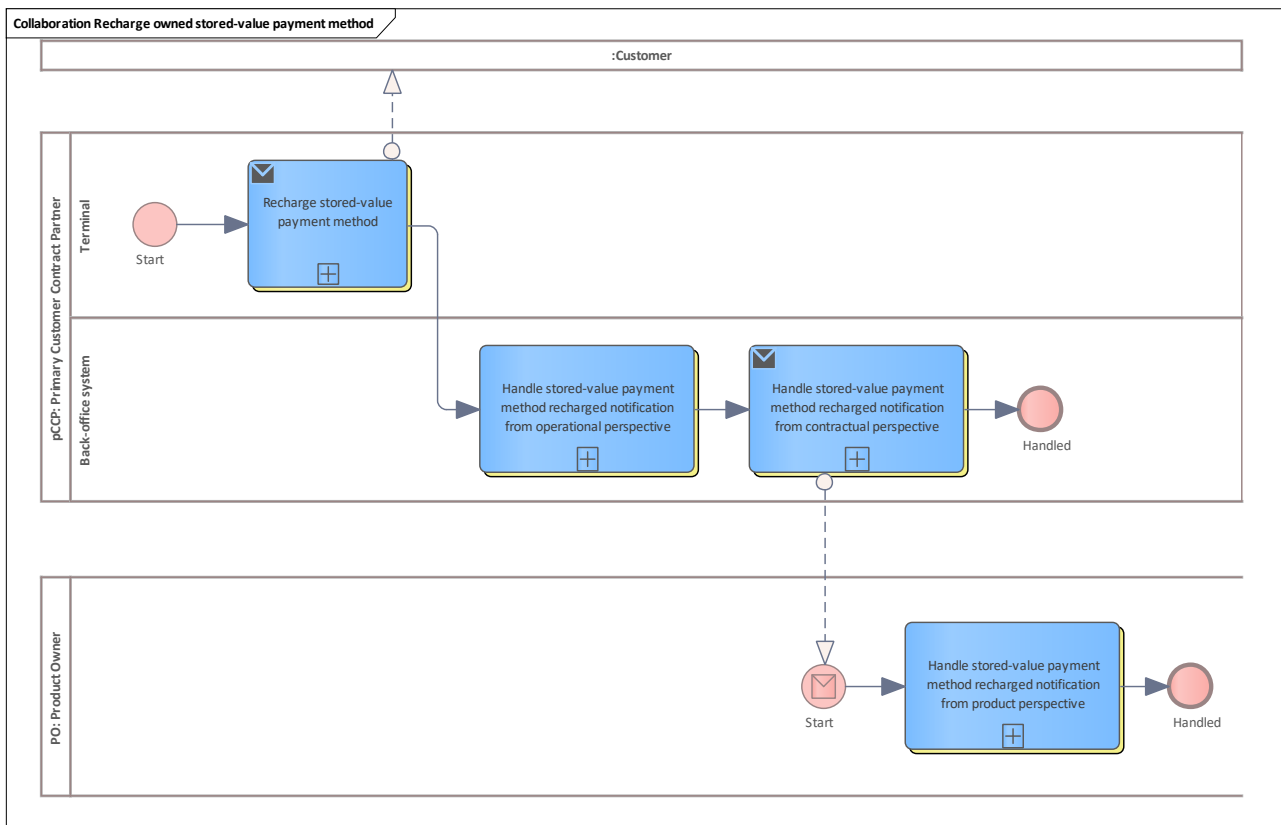


Figure 87: Recharge owned stored-value payment method

9.2.11.1 pCCP

See [Primary Customer Contract Partner](#)

9.2.11.1.1 Terminal

Lane for terminal

1.1.1.1.1.23 Recharge stored-value payment method

See [Recharge stored-value payment method](#)

9.2.11.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.24 Handle stored-value payment method recharged notification from contractual perspective

See [Handle stored-value payment method recharged notification from contractual perspective](#)

1.1.1.1.1.25 Handle stored-value payment method recharged notification from operational perspective

See [Handle stored-value payment method recharged notification from operational perspective](#)

9.2.11.2 PO

See [Product Owner](#)

9.2.11.2.1 Handle stored-value payment method recharged notification from product perspective

See [Handle stored-value payment method recharged notification from product perspective](#)

9.2.12 Autoload non-owned stored-value payment method

This basic process describes an automatically triggered ("autoload") recharge of a stored-value payment method.

Technical name due to [ProcessNameEnum](#): **AutoloadStoredValuePaymentMethod**

The process starts in a CCP (or CICO-) terminal if the amount on the user medium is detected as being not sufficient for the next payment-based action.

Without any further payment by the customer, the stored-value payment method will be recharged with the possible autoload amount. This amount is configured in the payment method data structure on the user medium.

The payment method was not issued by the CCP that operates the terminal.

In this case, the typical template process for [Entitlement non-owned](#) is the foundation of this basic process.

After recharging the stored-value payment method on the user medium by a certain amount, the terminal sends the data to the CCP back-office system.

This CCP back-office system does some basic monitoring steps and notifies this transaction to the PO back-office system. The PO back-office system receives this notification and does checks and monitoring before forwarding it to the pCCP back-office system.

Furthermore, the pCCP back-office system registers the autoload recharge action and does the contractual checks and monitoring.

Finally, the amount which was recharged by the terminal is booked from the customer's account (out of specification).

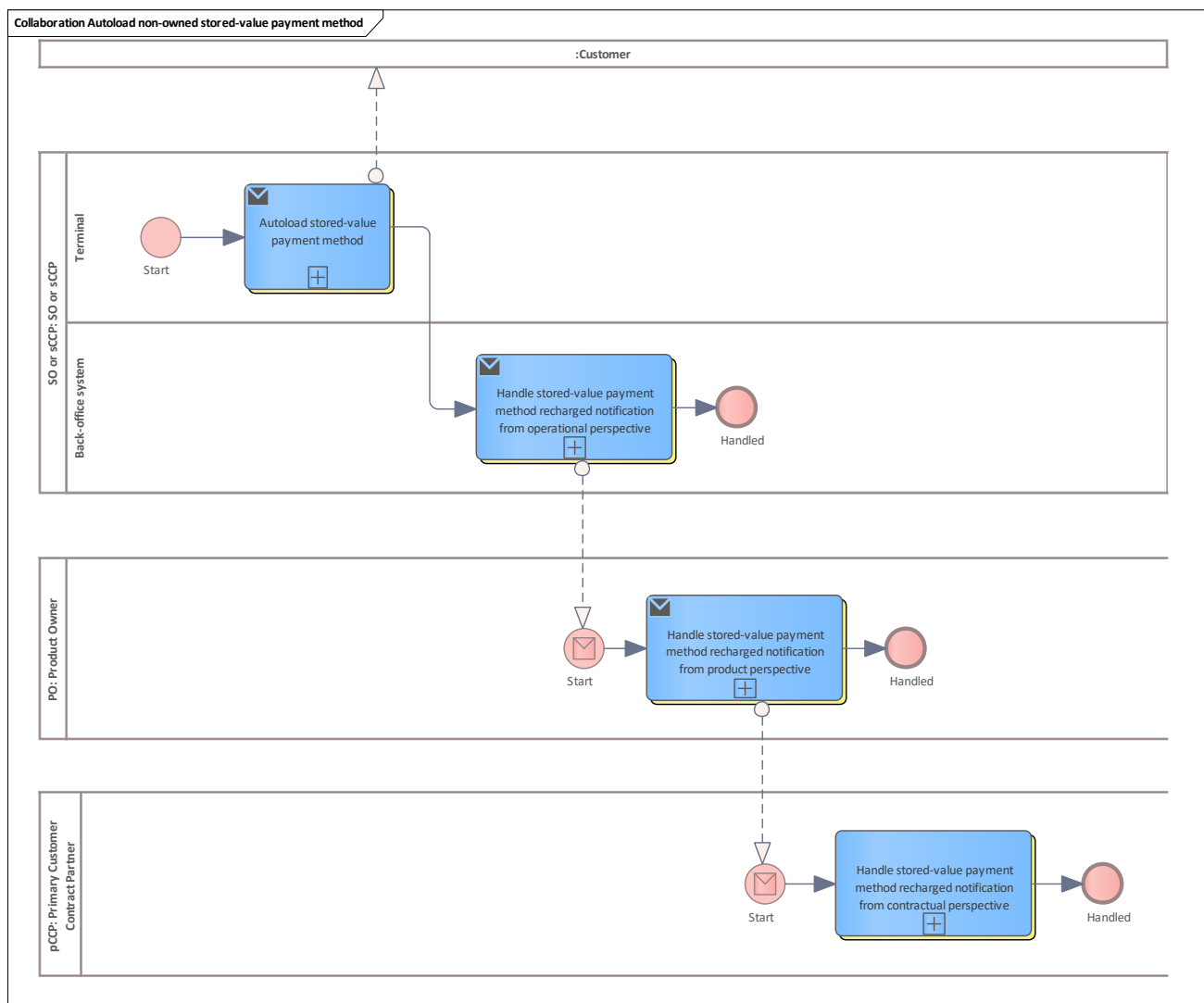


Figure 88: Autoload non-owned stored-value payment method

9.2.12.1 SO or sCCP

See [SO or sCCP](#)

9.2.12.1.1 Terminal

Lane for terminal

1.1.1.1.1.26 Autoload stored-value payment method

See [Autoload stored-value payment method](#)

9.2.12.1.2 Back-office system

Lane for back-office system



1.1.1.1.1.27 Handle stored-value payment method recharged notification from operational perspective

See [Handle stored-value payment method recharged notification from operational perspective](#)

9.2.12.2 PO

See [Product Owner](#)

9.2.12.2.1 Handle stored-value payment method recharged notification from product perspective

See [Handle stored-value payment method recharged notification from product perspective](#)

9.2.12.3 pCCP

See [Primary Customer Contract Partner](#)

9.2.12.3.1 Handle stored-value payment method recharged notification from contractual perspective

See [Handle stored-value payment method recharged notification from contractual perspective](#)

9.2.13 Autoload owned stored-value payment method

This basic process describes an automatically triggered ("autoload") recharge of a stored-value payment method.

Technical name due to [ProcessNameEnum](#): **AutoloadStoredValuePaymentMethod**

The process starts in a CCP (or CICO-) terminal if the amount on the user medium is detected as being not sufficient for the next payment-based action. Without any further payment by the customer, the stored-value payment method will be recharged with the possible autoload amount. This amount is configured in the stored-value payment method data structure on the user medium.

The stored-value payment method was issued by the CCP that operates the terminal.

In this case, the typical template process for [Entitlement owned](#) is the foundation of this basic process.

After recharging the stored-value payment method on the user medium by a certain amount, the terminal sends the data to the pCCP back-office system.

This pCCP back-office system does some basic monitoring steps. Since the pCCP is the owner, the pCCP back-office system registers the autoload recharge action and does the contractual checks and monitoring.

Then the pCCP notifies this transaction to the PO back-office system. The PO back-office system receives this notification and does checks and monitoring.

Finally, the amount which was recharged by the terminal is booked from the customer's account (out of specification).

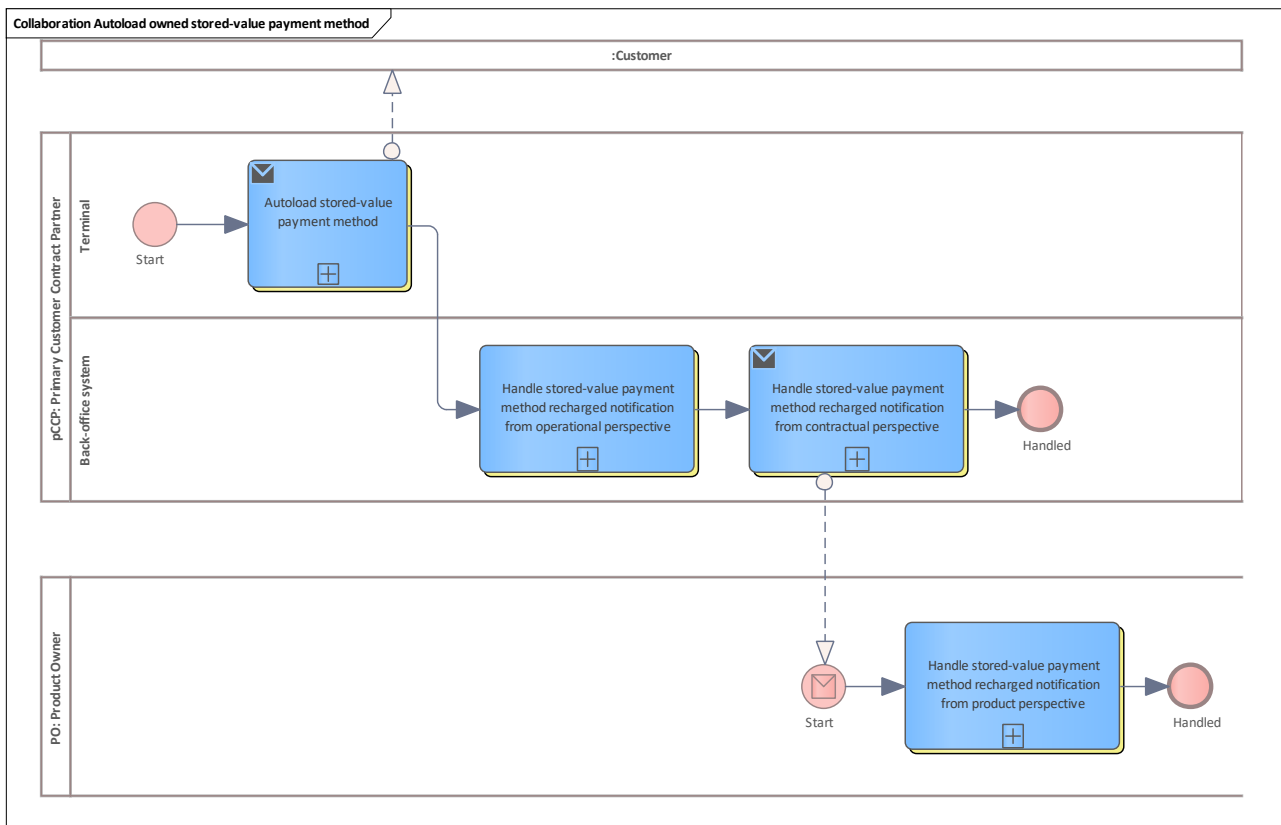


Figure 89: Autoload owned stored-value payment method

9.2.13.1 pCCP

See [Primary Customer Contract Partner](#)

9.2.13.1.1 Terminal

Lane for terminal

1.1.1.1.1.28 Autoload stored-value payment method

See [Autoload stored-value payment method](#)

9.2.13.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.29 Handle stored-value payment method recharged notification from contractual perspective

See [Handle stored-value payment method recharged notification from contractual perspective](#)

1.1.1.1.1.30 Handle stored-value payment method recharged notification from operational perspective

See [Handle stored-value payment method recharged notification from operational perspective](#)

9.2.13.2 PO

See [Product Owner](#)

9.2.13.2.1 Handle stored-value payment method recharged notification from product perspective

See [Handle stored-value payment method recharged notification from product perspective](#)

9.2.14 Personalise application

The basic process that shows the interaction between the customer (with his user medium), the terminal and the back-office system when personalising an application.

This process is located at the pCCP. The customer may be in the customer centre. Downstream shipping to the customer is also possible.

This process starts in the background system and has the aim of providing the customer with a new user medium and a new application instance.

In order to assign the user medium and the application instance to the customer, the user medium is initially read. The next step configures the user medium application, if not already done. This step also determines the validity period.

Dependent on the contract, personal data coming from the back-office system is transferred to the user medium.

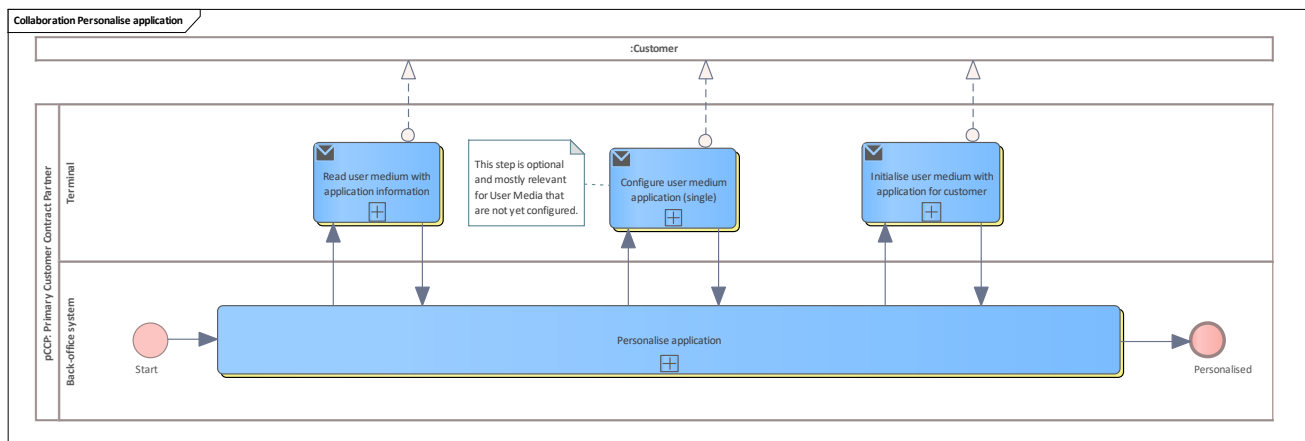


Figure 90: Personalise application

9.2.14.1 pCCP

See [Primary Customer Contract Partner](#)

9.2.14.1.1 Terminal

Lane for terminal



1.1.1.1.1.31 Configure user medium application (single)

See [Configure user medium application](#).

1.1.1.1.1.32 Initialise user medium with application for customer

See [Initialise User Medium with application for customer](#).

1.1.1.1.1.33 Read user medium with application information

See [Read User Medium with application information](#).

9.2.14.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.34 Personalise application

See [Personalise user medium application](#).

9.2.15 Debit non-owned stored-value payment method

In this basic process, due to a payment-based action, the stored-value payment method has been debited for a certain amount in a CCP terminal.

The stored-value payment method was not issued by the CCP that operates the terminal.

In this case, the typical template process for [Entitlement non-owned](#) is the foundation of this basic process.

After debiting the stored-value payment method on the user medium by a certain amount, the terminal sends the data to the CCP back-office system.

This CCP back-office system does some basic monitoring steps and notifies this debit transaction to the PO back-office system. The PO back-office system receives this notification and does checks and monitoring before forwarding it to the pCCP back-office system.

The pCCP back-office system registers the debit action and does the contractual checks and monitoring.

Finally, the amount is booked internally (out of specification).

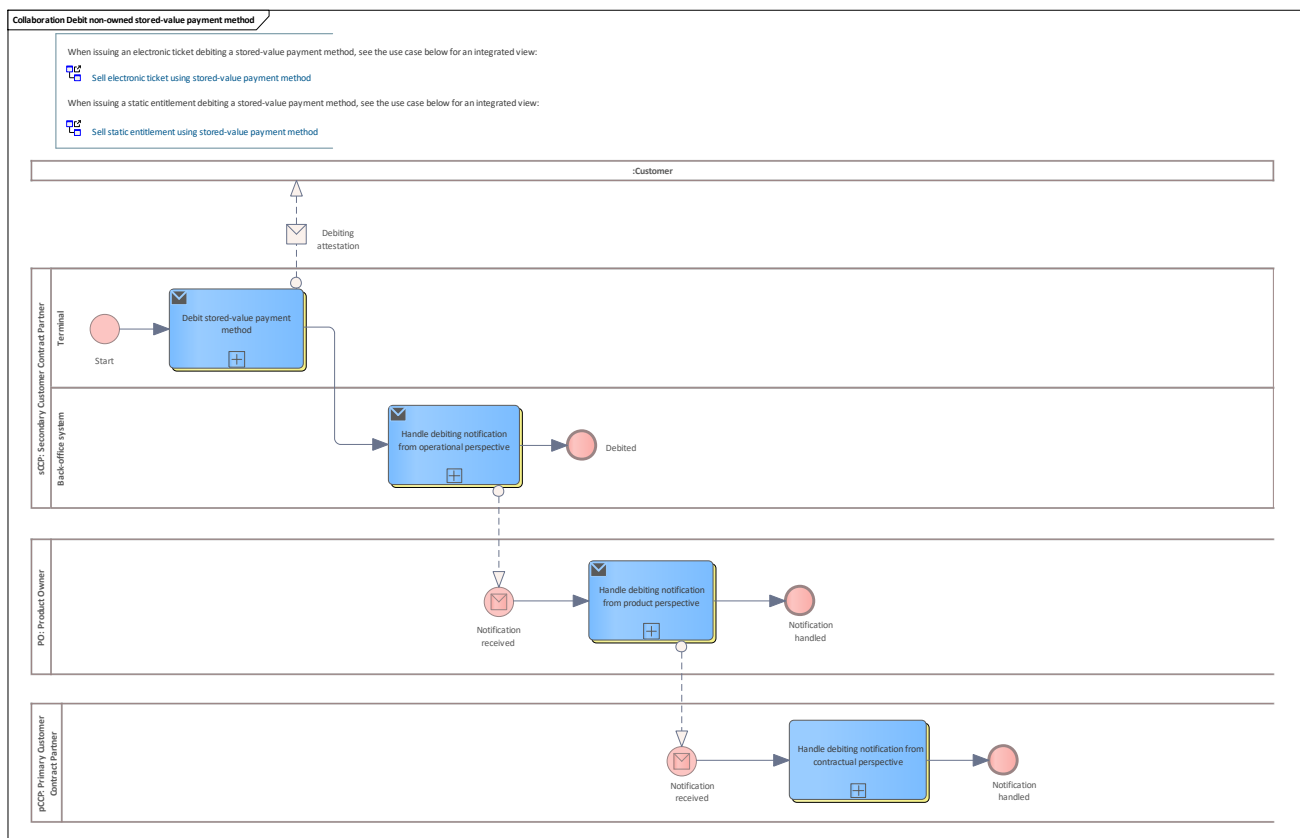


Figure 91: Debit non-owned stored-value payment method

9.2.15.1 sCCP

See [Secondary Customer Contract Partner](#)

9.2.15.1.1 Terminal

Lane for terminal

1.1.1.1.1.35 Debit stored-value payment method

See [Debit stored-value payment method](#).

9.2.15.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.36 Handle debiting notification from operational perspective

See [Handle stored-value payment method debited notification from operational perspective](#).

9.2.15.2 PO

See [Product Owner](#)

9.2.15.2.1 Handle debiting notification from product perspective

See [Handle stored-value payment method debited notification from product perspective](#).

9.2.15.3 pCCP

See [Primary Customer Contract Partner](#)

9.2.15.3.1 Handle debiting notification from contractual perspective

See [Handle stored-value payment method debited notification from contractual perspective](#).

9.2.16 Debit owned stored-value payment method

In this basic process, due to a payment-based action, the stored-value payment method has been debited by a certain amount in a CCP terminal.

The stored-value payment method was issued by the CCP that operates the terminal.

In this case, the typical template process for [Entitlement owned](#) is the foundation of this basic process.

After debiting the stored-value payment method on the user medium by a certain amount, the terminal sends the data to the pCCP back-office system.

This pCCP back-office system does some basic monitoring steps. Since the pCCP is the owner, the pCCP back-office system registers the debit action and does the contractual checks and monitoring.

Then the pCCP notifies this debit transaction to the PO back-office system. The PO back-office system receives this notification and does checks and monitoring.

Finally, the amount is booked internally (out of specification).

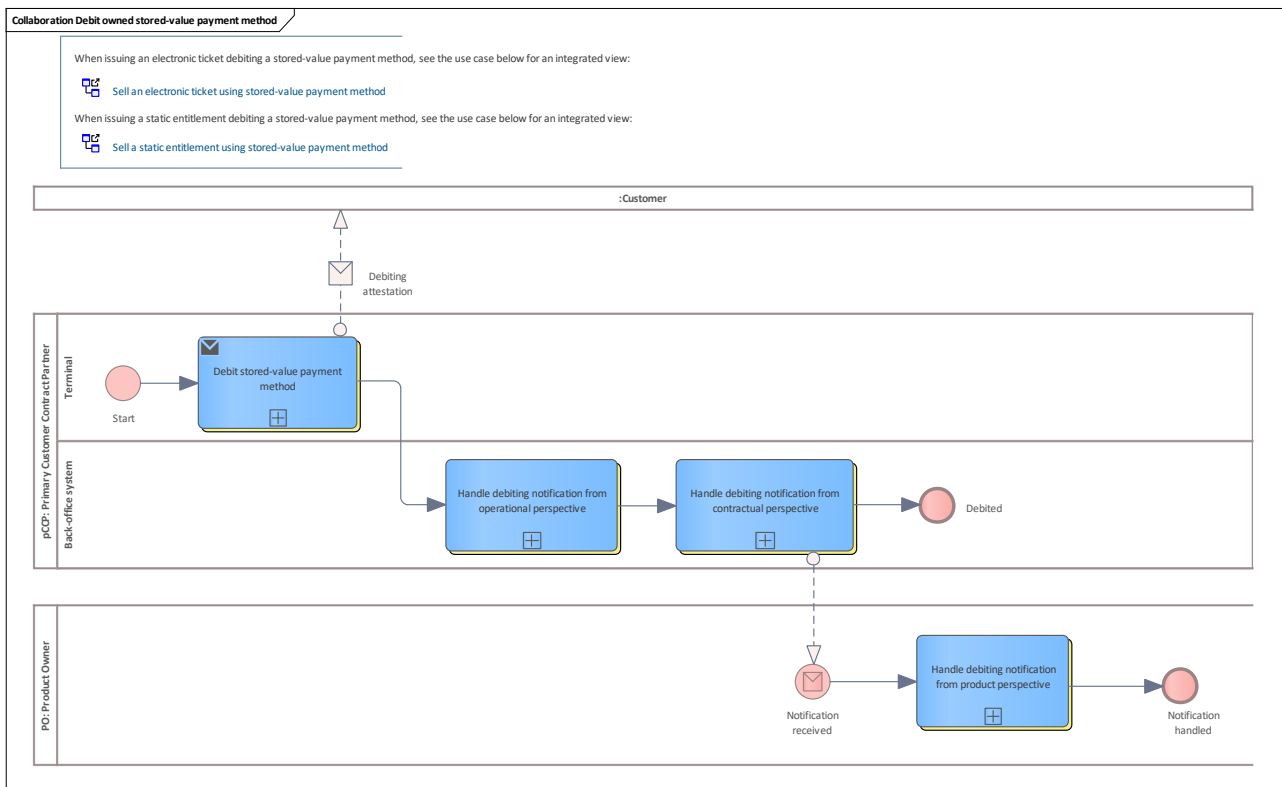


Figure 92: Debit owned stored-value payment method

9.2.16.1 pCCP

See [Primary Customer Contract Partner](#)

9.2.16.1.1 Terminal

Lane for terminal

1.1.1.1.1.37 Debit stored-value payment method

See [Debit stored-value payment method](#).

9.2.16.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.38 Handle debiting notification from operational perspective

See [Handle stored-value payment method debited notification from operational perspective](#).

1.1.1.1.1.39 Handle debiting notification from contractual perspective

See [Handle stored-value payment method debited notification from contractual perspective](#).

9.2.16.2 PO

See [Product Owner](#)

9.2.16.2.1 Handle debiting notification from product perspective

See [Handle stored-value payment method debited notification from product perspective](#).

9.2.17 Debit non-owned account-based payment method

In this basic process, due to a payment-based action, the account-based payment method has been debited by a certain amount in a CCP terminal.

The account-based payment method was not issued by the CCP that operates the terminal. In this case, the typical template process for [Entitlement non-owned](#) is the foundation of this basic process.

After debiting the account-based payment method on the user medium by a certain amount, the terminal sends the data to the CCP back-office system.

This CCP back-office system does some basic monitoring steps and notifies this debit transaction to the PO back-office system. The PO back-office system receives this notification and does checks and monitoring before forwarding it to the pCCP back-office system.

The pCCP back-office system registers the debit action and does its contractual checks and monitoring. Finally, the amount is booked from the customer's account (out of specification).

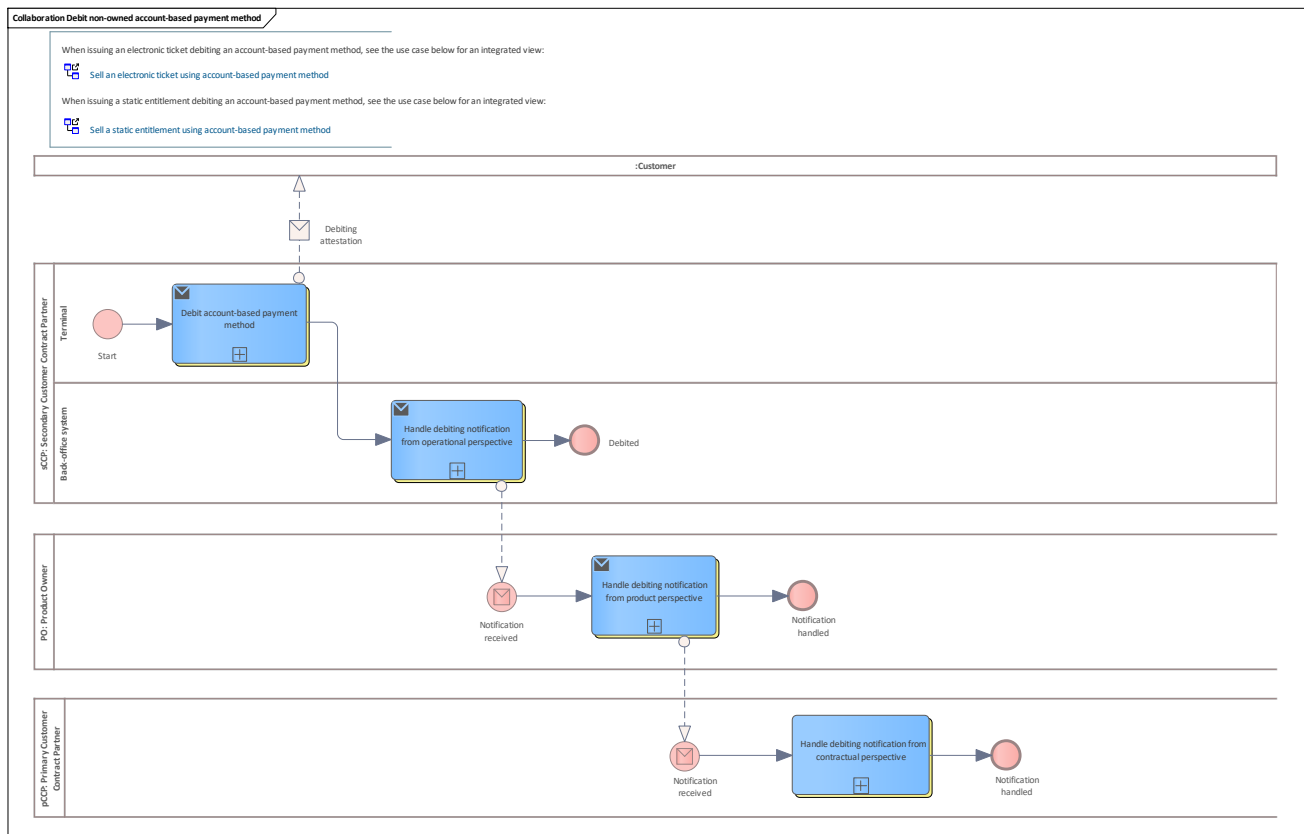


Figure 93: Debit non-owned account-based payment method

9.2.17.1 sCCP

See [Secondary Customer Contract Partner](#)

9.2.17.1.1 Terminal

Lane for terminal

1.1.1.1.1.40 Debit account-based payment method

See [Debit account-based payment method](#).

9.2.17.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.41 Handle debiting notification from operational perspective

See [Handle account-based payment method debited notification from operational perspective](#).



9.2.17.2 PO

See [Product Owner](#)

9.2.17.2.1 Handle debiting notification from product perspective

See [Handle account-based payment method debited notification from product perspective](#).

9.2.17.3 pCCP

See [Primary Customer Contract Partner](#)

9.2.17.3.1 Handle debiting notification from contractual perspective

See [Handle account-based payment method debited notification from contractual perspective](#).

9.2.18 Debit owned account-based payment method

In this basic process, due to a payment-based action, the account-based payment method has been debited by a certain amount in a CCP terminal.

The account-based payment method was issued by the CCP that operates the terminal.

In this case, the typical template process for [Entitlement owned](#) is the foundation of this basic process.

After debiting the account-based payment method on the user medium by a certain amount, the terminal sends the data to the CCP back-office system.

This pCCP back-office system does some basic monitoring steps. Since the pCCP is the owner, the pCCP back-office system registers the debit action and does the contractual checks and monitoring.

Then the pCCP notifies this transaction to the PO back-office system. The PO back-office system receives this notification and does checks and monitoring.

Finally, the amount is booked from the customer's account (out of specification).

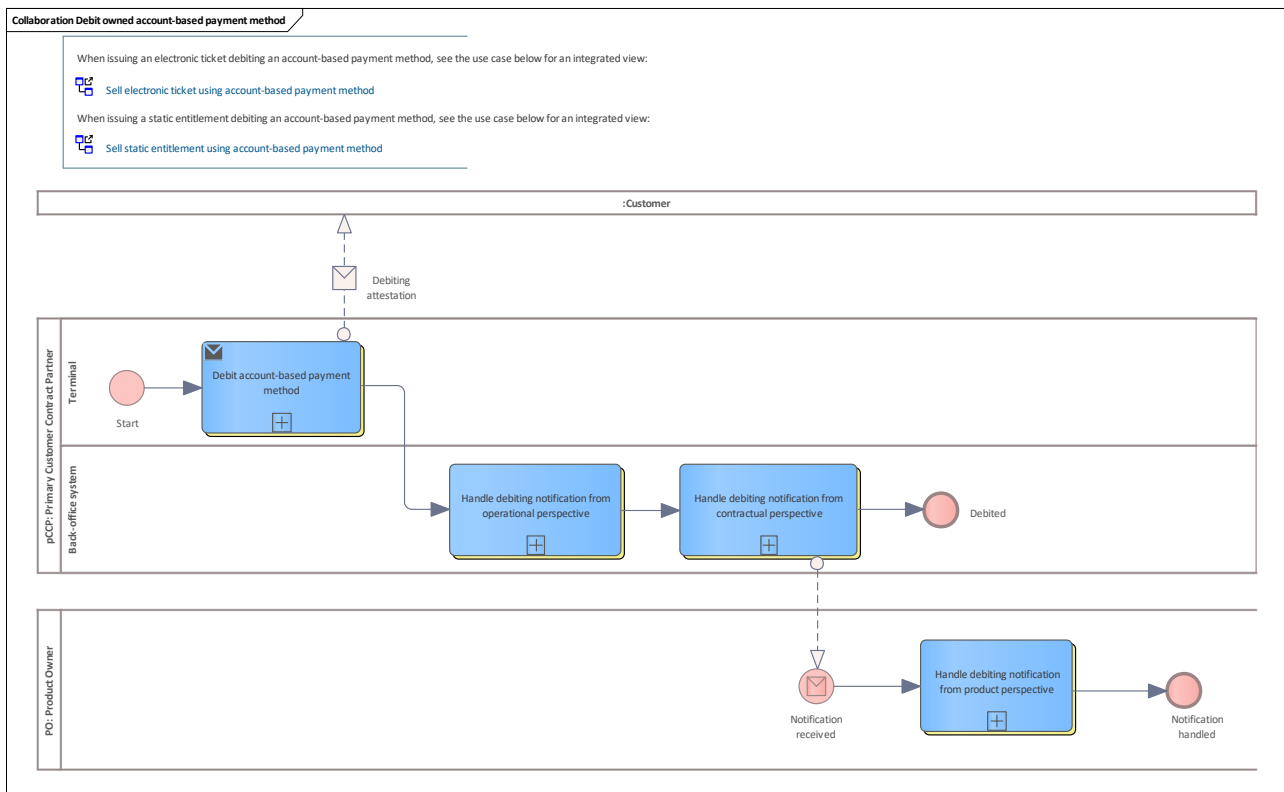


Figure 94: Debit owned account-based payment method

9.2.18.1 pCCP

See [Primary Customer Contract Partner](#)

9.2.18.1.1 Terminal

Lane for terminal

1.1.1.1.1.42 Debit account-based payment method

See [Debit account-based payment method](#).

9.2.18.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.43 Handle debiting notification from contractual perspective

See [Handle account-based payment method debited notification from contractual perspective](#)

1.1.1.1.1.44 Handle debiting notification from operational perspective

See [Handle account-based payment method debited notification from operational perspective](#)



9.2.18.2 PO

See [Product Owner](#)

9.2.18.2.1 Handle debiting notification from product perspective

See [Handle account-based payment method debited notification from product perspective](#)

9.2.19 Inspection

This chapter describes the participants and the activities within the basic process "inspection". BPMN Collaboration is used.

9.2.20 Inspect user medium with application

Inspects entitlements on a user medium with application (mostly a chip card) to find out if the passenger is allowed to ride at the current time and place.

This is always done by the SO.

The basic process starts in the terminal of the SO which interacts with the passenger's user medium. The inspection in the terminal has 3 phases.

- The first phase performs an authenticity check and does common checks against the hotlists.
- The second phase filters out the entitlement(s) with a matching validity period and performs an initial check if it is valid in the area of the current public transport association.
- The third phase performs checks according to the tariff modules to be applied for the filtered entitlement(s).

For monitoring and notification purposes, the inspection of a certain entitlement is notified through the systems starting in the SO system, followed by the PO system and finally forwarded to the pCCP system of the CCP which is the owner (issuer) of the inspected entitlement. Each system does dedicated monitoring and plausibility checks.

In this case, the typical template process for [Entitlement non-owned](#) is the foundation of this basic process.

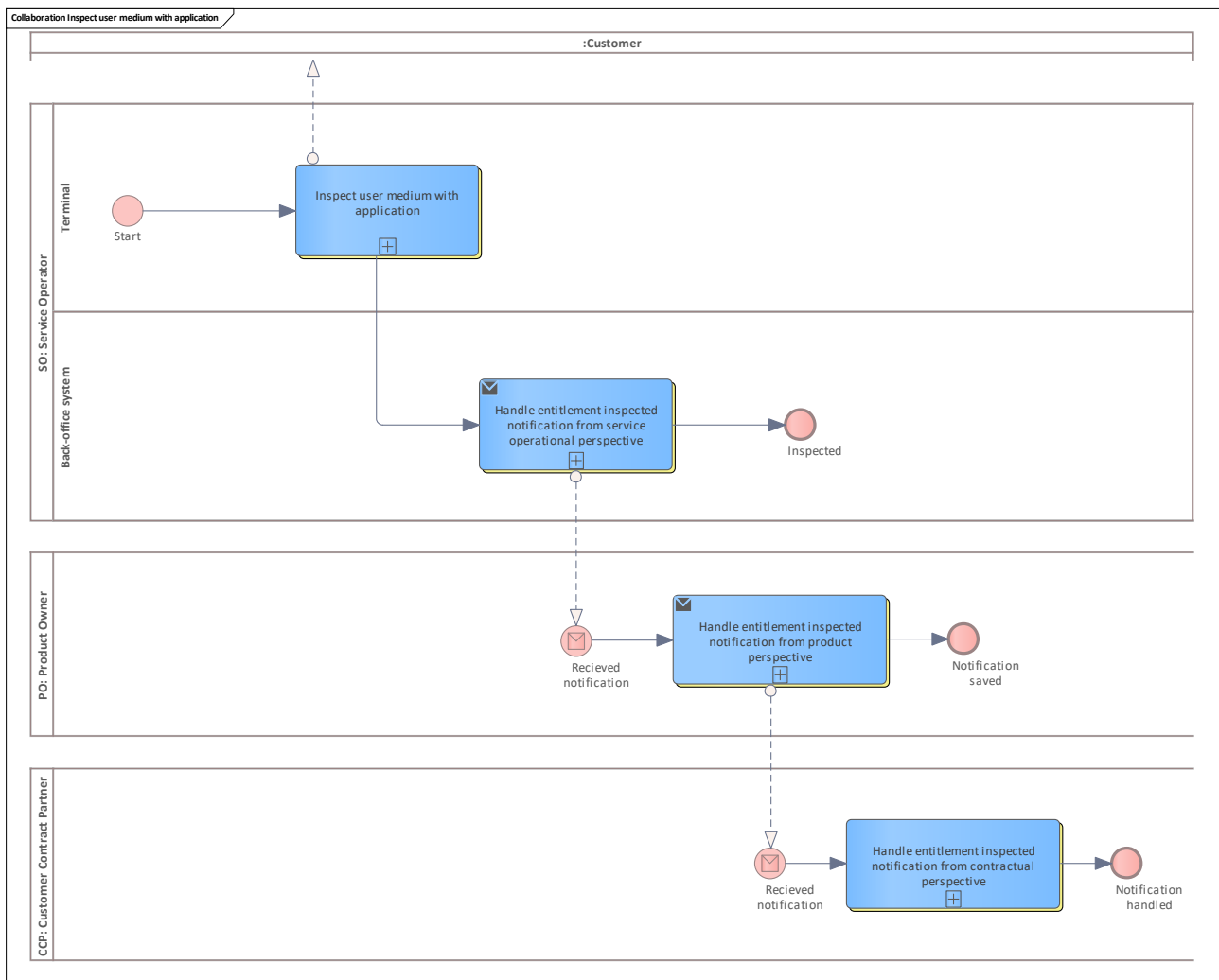


Figure 95: Inspect user medium with application

9.2.20.1 SO

See [Service Operator](#)

9.2.20.1.1 Terminal

Lane for terminal

1.1.1.1.1.45 Inspect user medium with application

See [Inspect user medium with application](#).

9.2.20.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.46 Handle entitlement inspected notification from service operational perspective

See [Handle entitlement inspected notification from operational perspective](#).

9.2.20.2 PO

See [Product Owner](#)

9.2.20.2.1 Handle entitlement inspected notification from product perspective

See [Handle entitlement inspected notification from product perspective](#).

9.2.20.3 CCP

See [Customer Contract Partner](#)

9.2.20.3.1 Handle entitlement inspected notification from contractual perspective

See [Handle entitlement inspected notification from contractual perspective](#).

9.2.21 Inspect user medium without application

Inspects static entitlement(s) on a user medium without an application (mostly a smartphone or a paper ticket) to find out, if the passenger is allowed to ride at the current time and place. The static entitlement can be realised a 2D barcode for an optical interface (bar-code on paper or picture in a smartphone) or as a binary structure transferred and verified with NFC (only smartphone).

This is always done by the SO.

The basic process starts in the SO's terminal which interacts with the passenger's user medium. The inspection in the terminal has 3 phases.

- The first phase performs an authenticity check (only smartphone + NFC) and does common checks against the hotlists.
- If more than 1 static entitlement is in the container, the second phase filters out the entitlement(s) with a matching validity period and performs an initial check if it is valid in the area of the current public transport association.
- The third phase performs checks according to the tariff modules to be applied for the filtered entitlement(s).

For monitoring and notification purposes, the inspection of a certain static entitlement is notified through the systems starting in the SO system, followed by the PO system and finally forwarded to the pCCP system of the CCP that is the owner (issuer) of the inspected static entitlement.

Each system does dedicated monitoring and plausibility checks.

In this case, the typical template process for [Entitlement non-owned](#) is the foundation of this basic process.

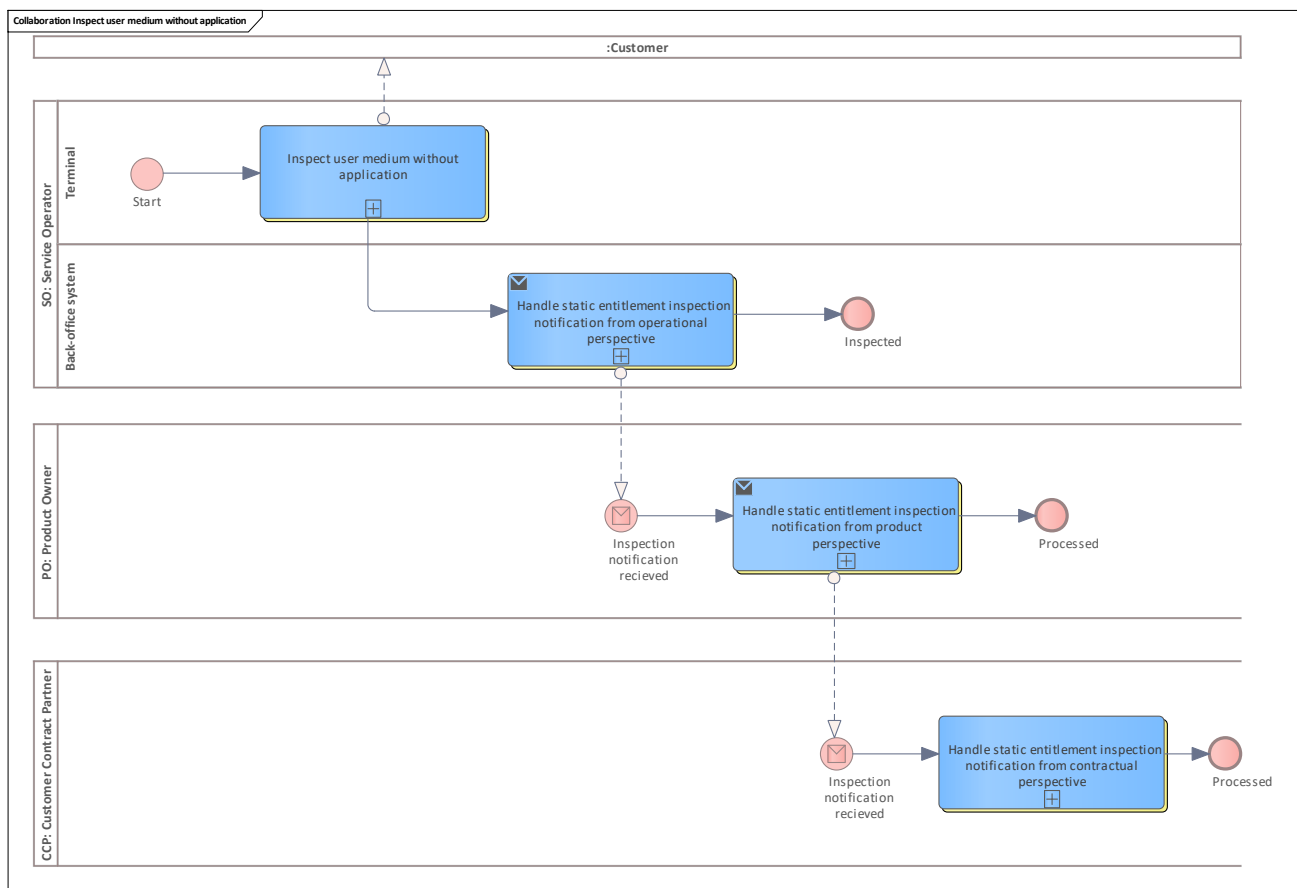


Figure 96: Inspect user medium without application

9.2.21.1 SO

See [Service Operator](#)

9.2.21.1.1 Terminal

Lane for terminal

1.1.1.1.1.47 Inspect user medium without application

See [Inspect user medium without application.](#)

9.2.21.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.48 Handle static entitlement inspection notification from operational perspective

See [Handle static entitlement inspection notification from operational perspective](#).



9.2.21.2 PO

See [Product Owner](#)

9.2.21.2.1 Handle static entitlement inspection notification from product perspective

See [Handle static entitlement inspection notification from product perspective](#).

9.2.21.3 CCP

See [Customer Contract Partner](#)

9.2.21.3.1 Handle static entitlement inspection notification from contractual perspective

See [Handle static entitlement inspection notification from contractual perspective](#).

9.2.22 Hotlisting and blocking

This chapter describes the participants and activities within the basic processes "hotlisting" and "blocking" of entitlement, application, SAM, organisation and authentication key. BPMN Collaboration is used.

9.2.22.1 Hotlist and block application

This chapter describes the hotlisting and blocking processes of an application instance, aided by BPMN collaboration diagrams.

9.2.22.2 Block non-owned hotlisted application

This basic process describes blocking an application instance issued by a third party such as an SO or an sCCP.

In this case, the typical template process for [Application non-owned](#) is the foundation of this basic process.

Normally, this process will be triggered by a match in the hotlist for an application instance on the user medium.

The application instance is blocked physically by switching the state. Except for limited data access, the application instance can no longer be used until it is unblocked.

The block transaction with the application is notified from the terminal to the terminal operator system which may be an SO or sCCP back-office system. After a few checks and some monitoring, the notification is forwarded to the owner of the application instance.

The pCCP system of the owner registers the application blocking and also does some checks and monitoring.

Finally, the pCCP is responsible for removing the application instance from the hotlist.

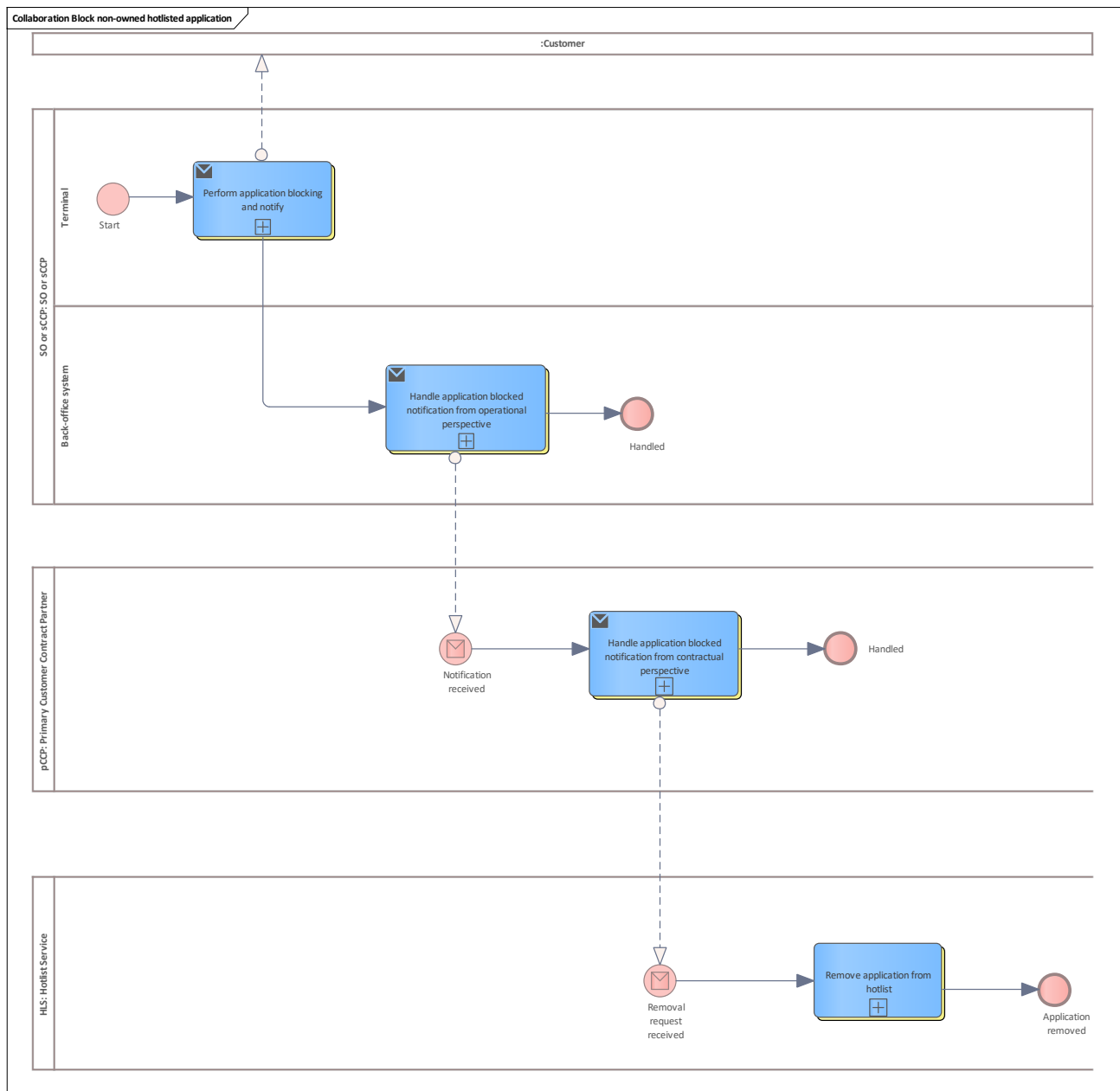


Figure 97: Block non-owned hotlisted application

9.2.22.2.1 SO or sCCP

See [SO or sCCP](#)

1.1.1.1.1.49 Back-office system

Lane for back-office system

1.1.1.1.1.49.1 Handle application blocked notification from operational perspective

See [Handle application blocked notification from operational perspective](#).

1.1.1.1.1.50 Terminal

Lane for terminal

1.1.1.1.1.50.1 Perform application blocking and notify

See [Perform application blocking and notify](#).

9.2.22.2.2 pCCP

See [Primary Customer Contract Partner](#)

1.1.1.1.1.51 Handle application blocked notification from contractual perspective

See [Handle application blocked notification from contractual perspective](#).

1.1.1.1.1.52 Notification received

The start point can be triggered by

- application blocked notification from third party
- or pCCP where pCCP has reason to block the application

9.2.22.2.3 HLS

See [Hotlist Service](#)

1.1.1.1.1.53 Remove application from hotlist

See [Remove application from hotlist](#).

9.2.22.3 Block owned hotlisted application

This basic process describes blocking an application instance that was issued by the CCP itself (pCCP).

In this case, the typical template process for [Application owned](#) is the foundation of this basic process.

Normally, this process will be triggered by a match in the hotlist for an application instance on the user medium.

The application instance is blocked physically by switching the state. Except for limited data access, the application instance can no longer be used until it is unblocked.

The application block transaction is notified from the terminal to the pCCP back-office system. The pCCP system of the owner registers the application blocking and also does some checks and monitoring.

Finally, the pCCP is responsible for removing the application instance from the hotlist.

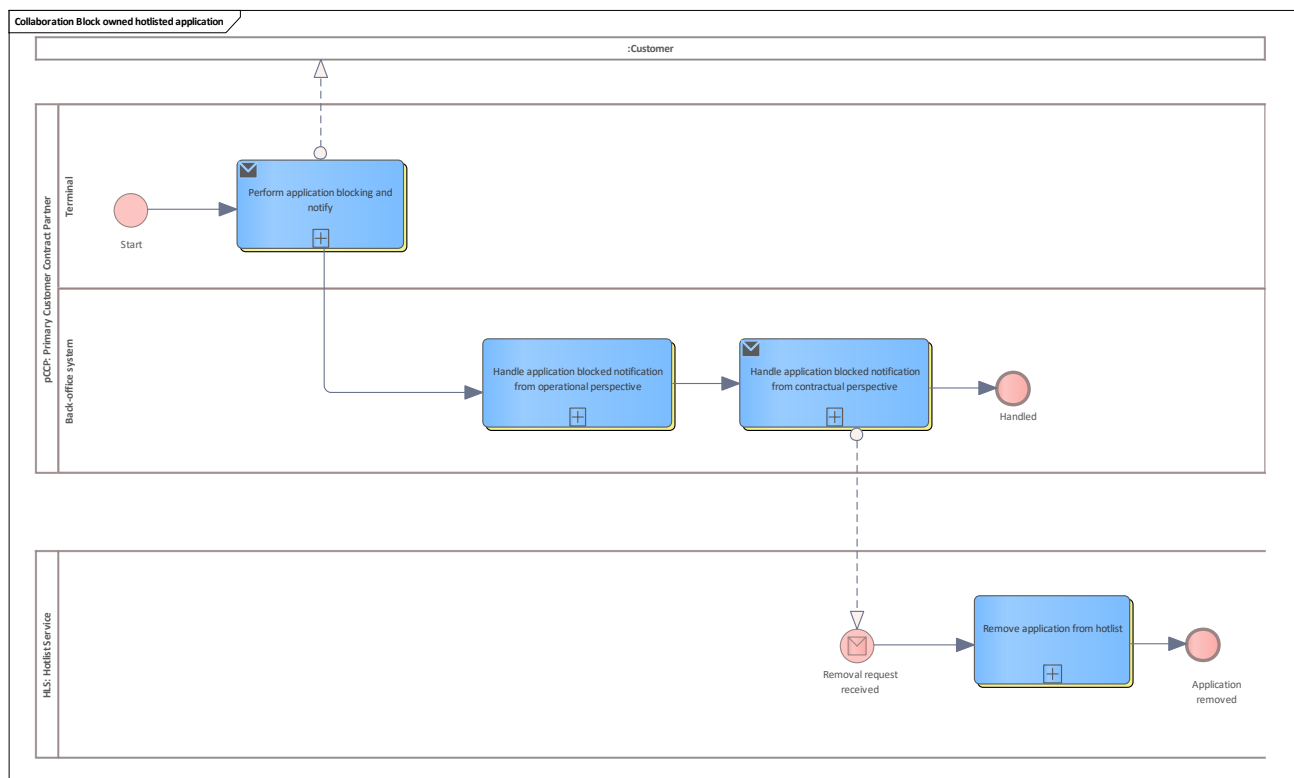


Figure 98: Block owned hotlisted application

9.2.22.3.1 pCCP

See [Primary Customer Contract Partner](#)

1.1.1.1.1.54 Terminal

Lane for terminal

1.1.1.1.1.54.1 Perform application blocking and notify

See [Perform application blocking and notify](#).

1.1.1.1.1.55 Back-office system

Lane for back-office system

1.1.1.1.1.55.1 Handle application blocked notification from operational perspective

See [Handle application blocked notification from operational perspective](#).

1.1.1.1.1.55.2 Handle application blocked notification from contractual perspective

See [Handle application blocked notification from contractual perspective](#).

9.2.22.3.2 HLS

See [Hotlist Service](#)

1.1.1.1.1.56 Remove application from hotlist

See [Remove application from hotlist](#).

9.2.22.4 Hotlist non-owned application

The basic process performs the hotlisting of an application issued by a third party. The demand for hotlisting can be an SO, sCCP or PO.

In most cases, the monitoring processes will trigger this process. Another reason might be a non-readable user medium. If any problems with the user medium are detected that require the blocking of the application, the demand is sent to the owner of the application instance.

The owner of the application instance (the pCCP) then verifies the hotlist demand and requests adding the application instance to the application hotlist.

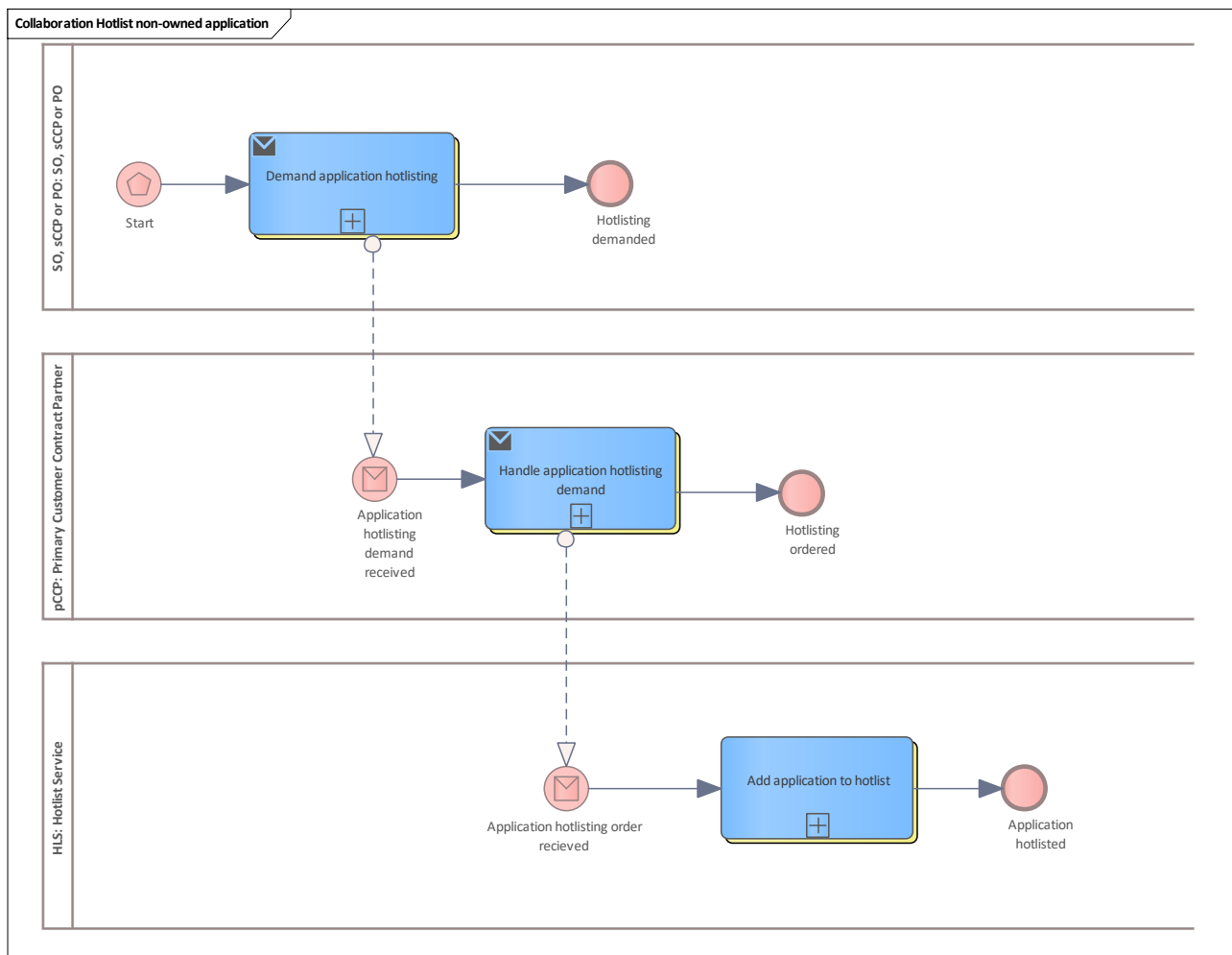


Figure 99: Hotlist non-owned application

9.2.22.4.1 SO, sCCP or PO

See [SO, sCCP or PO](#)

1.1.1.1.1.57 Demand application hotlisting

See [Demand application hotlisting](#)

1.1.1.1.1.58 Start

A participant can demand application hotlisting due to several reasons, such as stolen, lost or defective user medium etc.

9.2.22.4.2 pCCP

See [Primary Customer Contract Partner](#)

1.1.1.1.1.59 Handle application hotlisting demand

See [Handle application hotlisting demand](#)

1.1.1.1.1.60 Application hotlisting demand received

The pCCP has received a demand from a third party for hotlisting an application.

9.2.22.4.3 HLS

See [Hotlist Service](#)

1.1.1.1.1.61 Add application to hotlist

See [Add application to hotlist](#)

9.2.22.5 Hotlist owned application

The basic process performs the hotlisting of the application instance performed by the application instance owner itself (pCCP).

Either the internal monitoring process might trigger this process, it might have contractual reasons or the user medium is lost or not readable.

These cases should result in blocking the application, thus adding the application instance to the application hotlist should be requested.

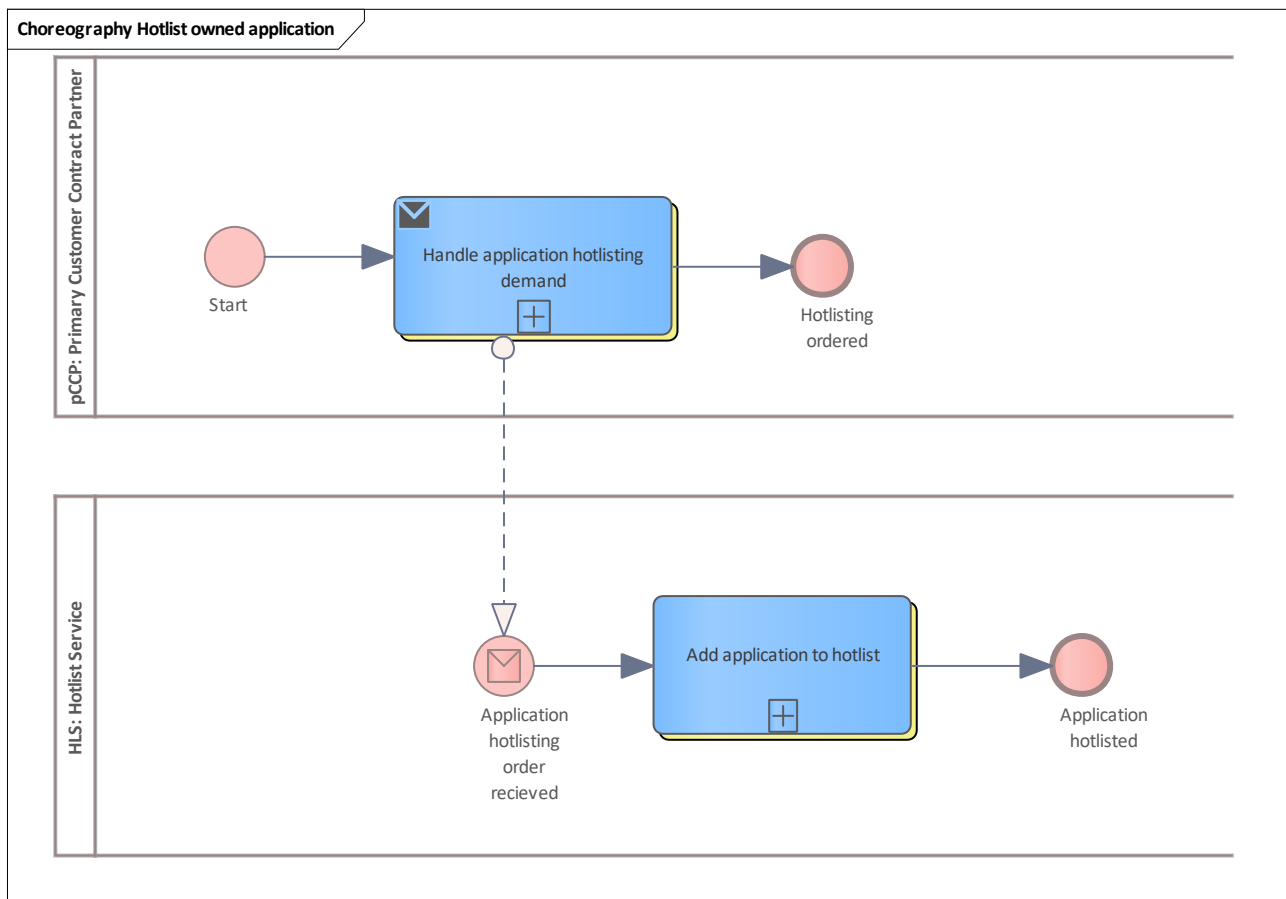


Figure 100: Hotlist owned application

9.2.22.5.1 pCCP

See [Primary Customer Contract Partner](#)

1.1.1.1.1.62 Handle application hotlisting demand

See [Add application to hotlist](#)

1.1.1.1.1.63 Start

A pCCP can order to hotlist an application due to several reasons, such as stolen, lost or defective user medium etc.

A demand to hotlist the application is created internally by the pCCP.

9.2.22.5.2 HLS

See [Hotlist Service](#)

1.1.1.1.1.64 Add application to hotlist

See [Add application to hotlist](#).



9.2.22.6 Hotlist and block entitlement

This chapter describes hotlisting and blocking processes of an entitlement, aided by BPMN collaboration diagrams.

9.2.22.7 Block non-owned entitlement

This basic process describes blocking an entitlement by a third party that did not issue the entitlement, such as an SO or an sCCP.

In this case, the typical template process for [Entitlement non-owned](#) is the foundation of this basic process.

This process will normally be triggered by a match in the hotlist for an entitlement on the user medium.

The entitlement is blocked physically by switching the state. Except for limited data access, the entitlement can no longer be used until it is unblocked.

The entitlement block transaction is notified from the terminal to the terminal operator system which may be an SO or sCCP back-office system. After a few checks and some monitoring the notification is forwarded to the PO.

The PO registers the entitlement blocking and performs some checks and further monitoring. Then the notification is forwarded to the entitlement issuer (the pCCP).

The pCCP system registers the entitlement blocking and also does some contractual checks and monitoring.

Finally, the pCCP is responsible for removing the entitlement from the hotlist.

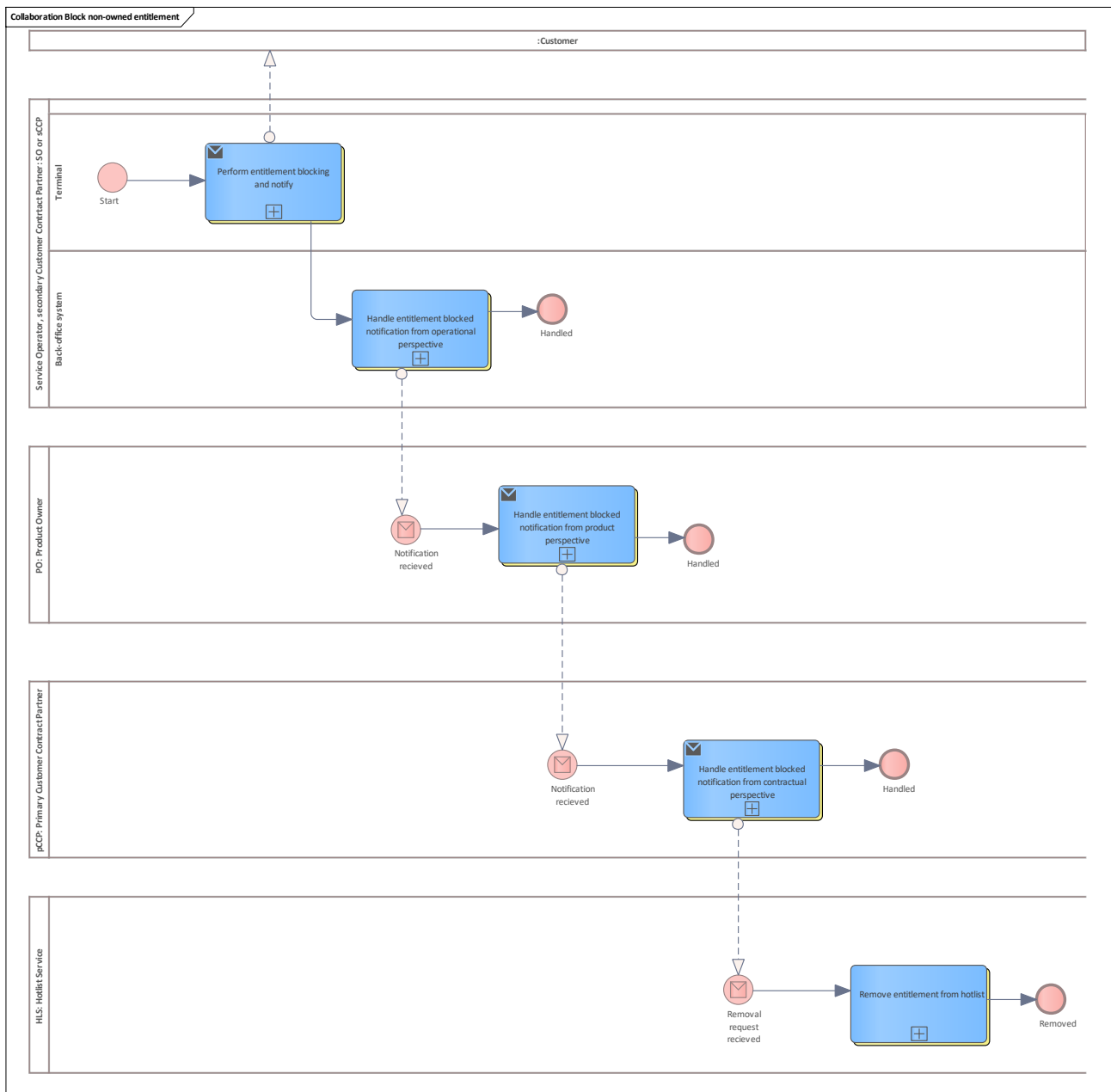


Figure 101: Block non-owned entitlement

9.2.22.7.1 Service Operator, secondary Customer Contract Partner

See [SO or sCCP](#)

1.1.1.1.1.65 Terminal

Lane for terminal

1.1.1.1.1.65.1 Perform entitlement blocking and notify

See [Perform entitlement blocking and notify](#)

1.1.1.1.1.66 Back-office system

Lane for back-office system

9.2.22.7.2 PO

See [Product Owner](#)

1.1.1.1.1.67 Handle entitlement blocked notification from product perspective

See [Handle entitlement blocked notification from product perspective](#)

9.2.22.7.3 pCCP

See [Primary Customer Contract Partner](#)

1.1.1.1.1.68 Handle entitlement blocked notification from contractual perspective

See [Handle entitlement blocked notification from contractual perspective](#)

9.2.22.7.4 HLS

See [Hotlist Service](#)

1.1.1.1.1.69 Remove entitlement from hotlist

See [Remove entitlement from hotlist](#)

9.2.22.8 Block owned entitlement

This basic process describes blocking an entitlement by the entitlement issuer itself, the pCCP. In this case, the typical template process for [Entitlement owned](#) is the foundation of this basic process.

This process will normally be triggered by a match in the hotlist for an entitlement on the user medium.

The entitlement is blocked physically by switching the state. Except for limited data access, the entitlement can no longer be used until it is unblocked.

The entitlement block transaction is notified from the terminal to the pCCP back-office system. The pCCP system registers the entitlement blocking and also does some operational and contractual checks and monitoring. Then the notification is forwarded to the PO.

The PO registers the entitlement blocking and performs some checks and further monitoring. Finally, the pCCP is responsible for removing the entitlement from the hotlist.

Note: this can be done at the same time as the forwarding of the blocking notification.

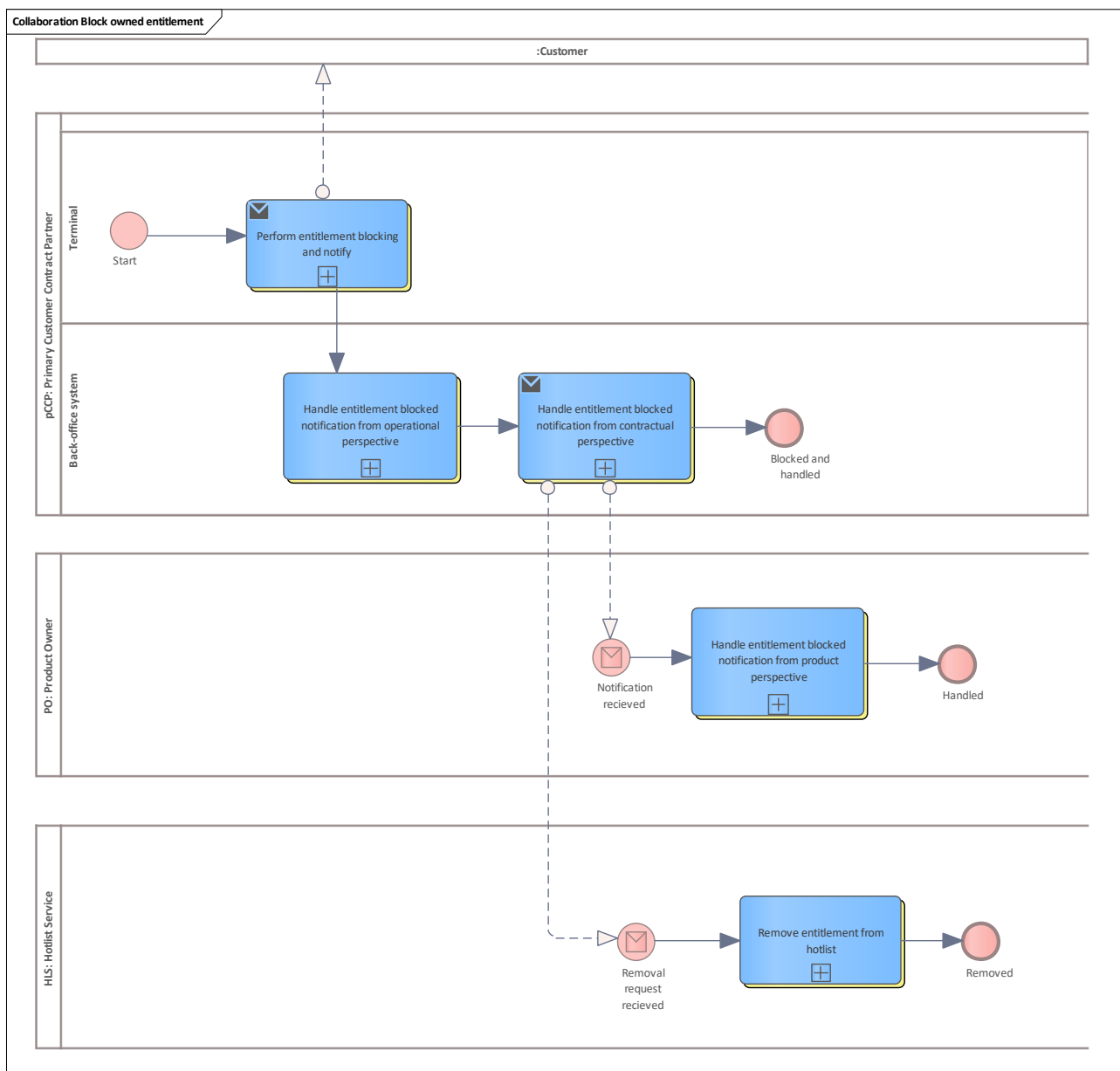


Figure 102: Block owned entitlement

9.2.22.8.1 pCCP

See [Primary Customer Contract Partner](#)

1.1.1.1.1.70 Terminal

Lane for terminal

1.1.1.1.1.70.1 Perform entitlement blocking and notify

See [Perform entitlement blocking and notify](#)

1.1.1.1.1.71 Back-office system

Lane for back-office system

9.2.22.8.2 PO

See [Product Owner](#)

1.1.1.1.1.72 Handle entitlement blocked notification from product perspective

See [Handle entitlement blocked notification from product perspective](#)

9.2.22.8.3 HLS

See [Hotlist Service](#)

1.1.1.1.1.73 Remove entitlement from hotlist

See [Remove entitlement from hotlist](#)

9.2.22.9 Hotlist non-owned entitlement

The basic process performs the hotlisting of an entitlement issued by a third party. The demand for hotlisting can be an SO, sCCP or PO.

In most cases, the monitoring processes will trigger this process. If any problems with the entitlement are detected that require the blocking, the demand is sent to the owner of the entitlement.

The owner of the entitlement (the pCCP) then verifies the hotlist demand and requests adding the entitlement to the entitlement hotlist.

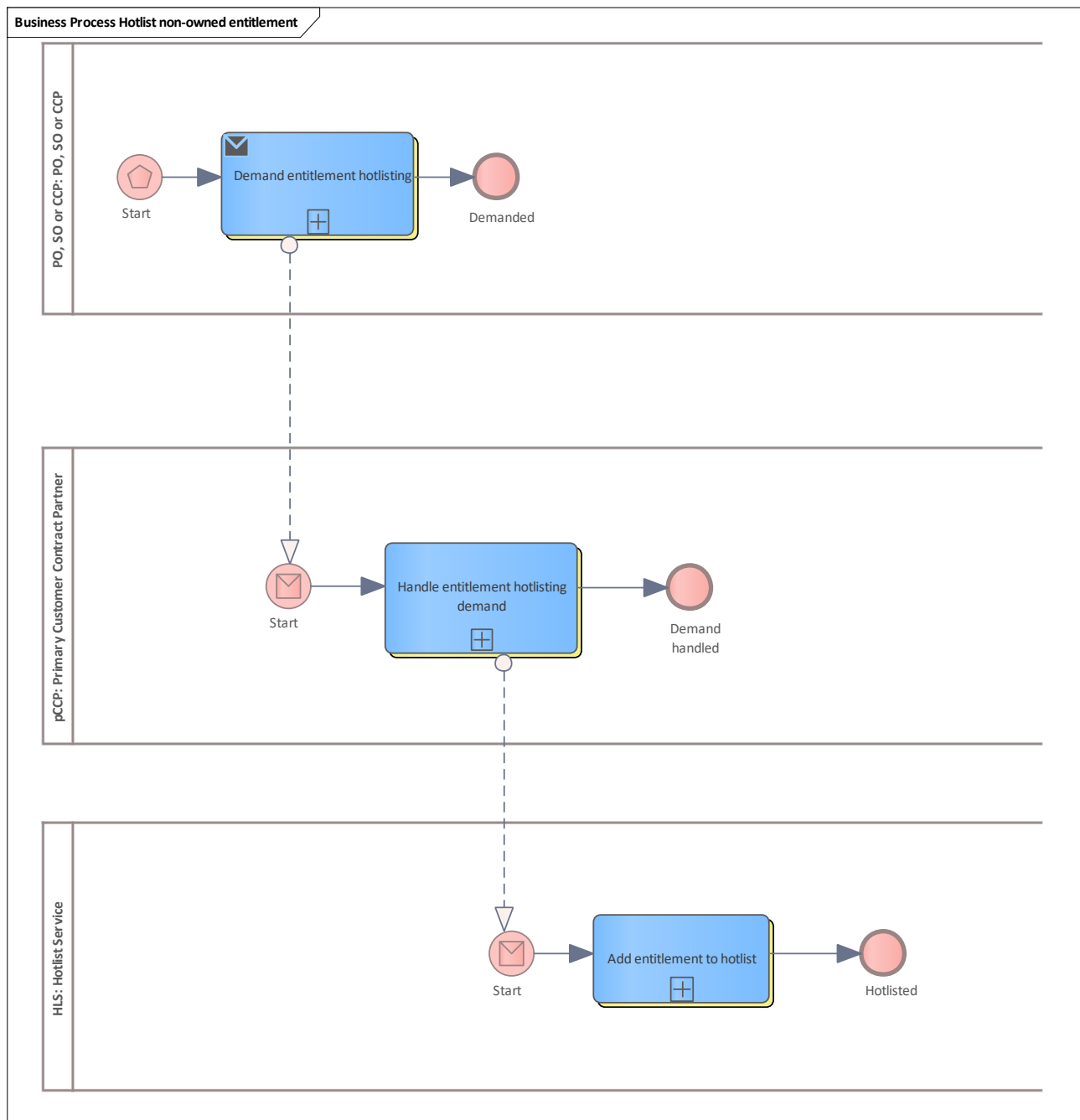


Figure 103: Hotlist non-owned entitlement

9.2.22.9.1 PO, SO or CCP

Describes activities of the SO ([Service Operator](#)), [PO\(Product Owner\)](#) or sCCP([secondary customer contract partner](#)) who is not the issuer of the application or entitlement of the current customer.

1.1.1.1.1.74 Demand entitlement hotlisting

See [Demand entitlement hotlisting](#).

1.1.1.1.1.75 Start

This activity initiates the entitlement hotlisting demand. Depending on the internal implementation of the initiating system the demand starts with a certain state. This activity depends on the implementation of the underlying system and is not specified in detail.

A hotlisting process starts, e.g., due to customer misbehaviour such as an offence against carriage terms or fraud.

The roles authorised to initiate such a process are:

- [Secondary Customer Contract Partner](#)
- [Service Operator](#)
- [Product Owner](#)

9.2.22.9.2 pCCP

See [Primary Customer Contract Partner](#).

1.1.1.1.1.76 Handle entitlement hotlisting demand

See [Handle entitlement hotlisting demand](#).

9.2.22.9.3 HLS

See [Hotlist Service](#).

1.1.1.1.1.77 Add entitlement to hotlist

See [Add entitlement to hotlist](#)

9.2.22.10 Hotlist owned entitlement

The basic process performs the entitlement hotlisting of the entitlement owner itself (pCCP). Either the internal monitoring process might trigger this process or it might have contractual or other reasons.

These cases should finally result in blocking the entitlement, thus adding the entitlement to the entitlement hotlist must be requested as first step.

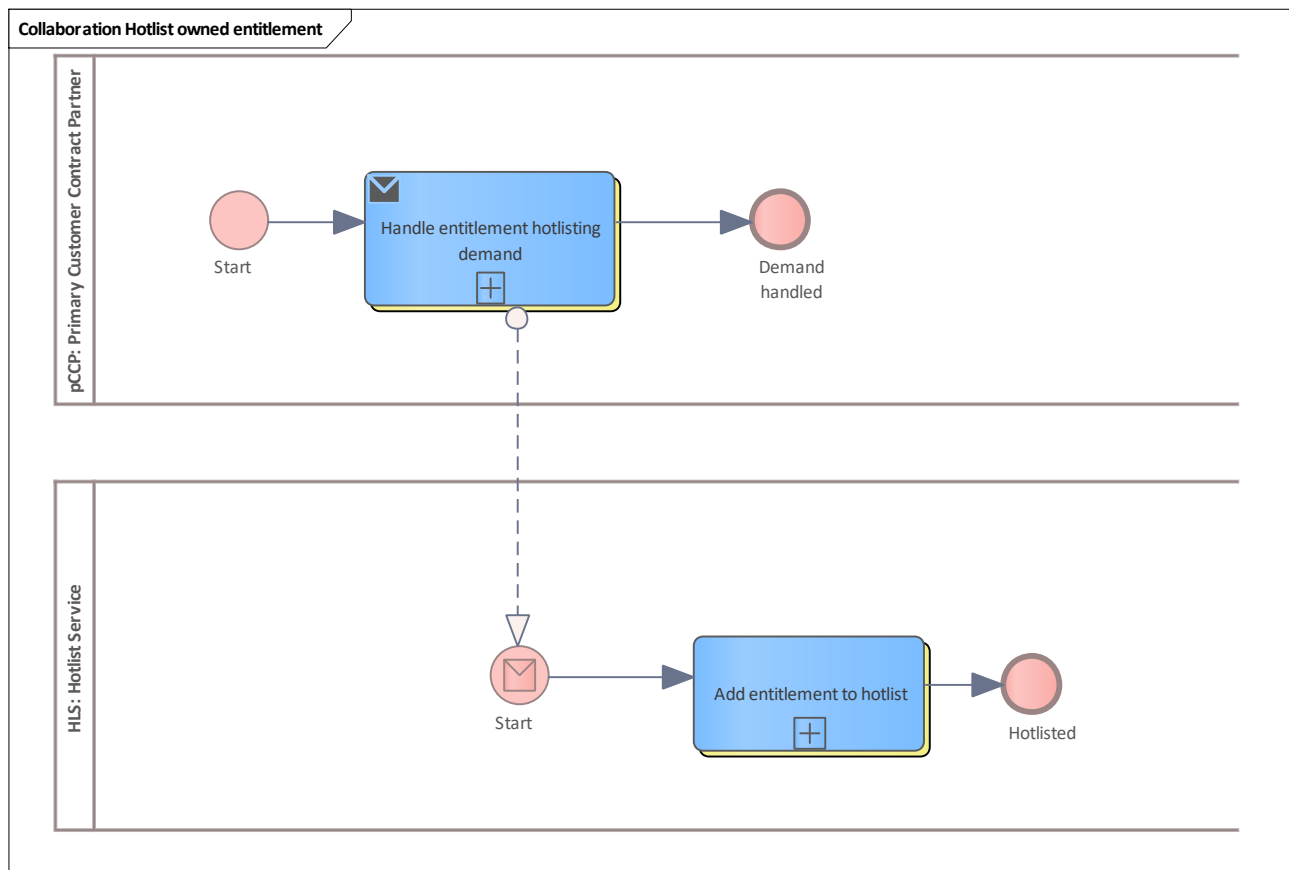


Figure 104: Hotlist owned entitlement

9.2.22.10.1 pCCP

See [Primary Customer Contract Partner](#)

1.1.1.1.1.78 Handle entitlement hotlisting demand

See [Handle entitlement hotlisting demand](#).

1.1.1.1.1.79 Start

A pCCP can order to hotlist an entitlement due to several reasons, such as stolen, lost or defective user medium etc.

A demand to hotlist the entitlement is internally created by the pCCP. That is not in the scope of etiCORE.

9.2.22.10.2 HLS

See [Hotlist Service](#)

1.1.1.1.1.80 Add entitlement to hotlist

See [Add entitlement to hotlist](#).



9.2.22.11 Hotlist authentication key

This chapter describes the hotlisting process of an authentication key, aided by BPMN collaboration diagrams.

9.2.22.12 Hotlist authentication key

This basic process describes the hotlisting process of an authentication key. This can only be done by the [Scheme Manager](#) for security reasons.

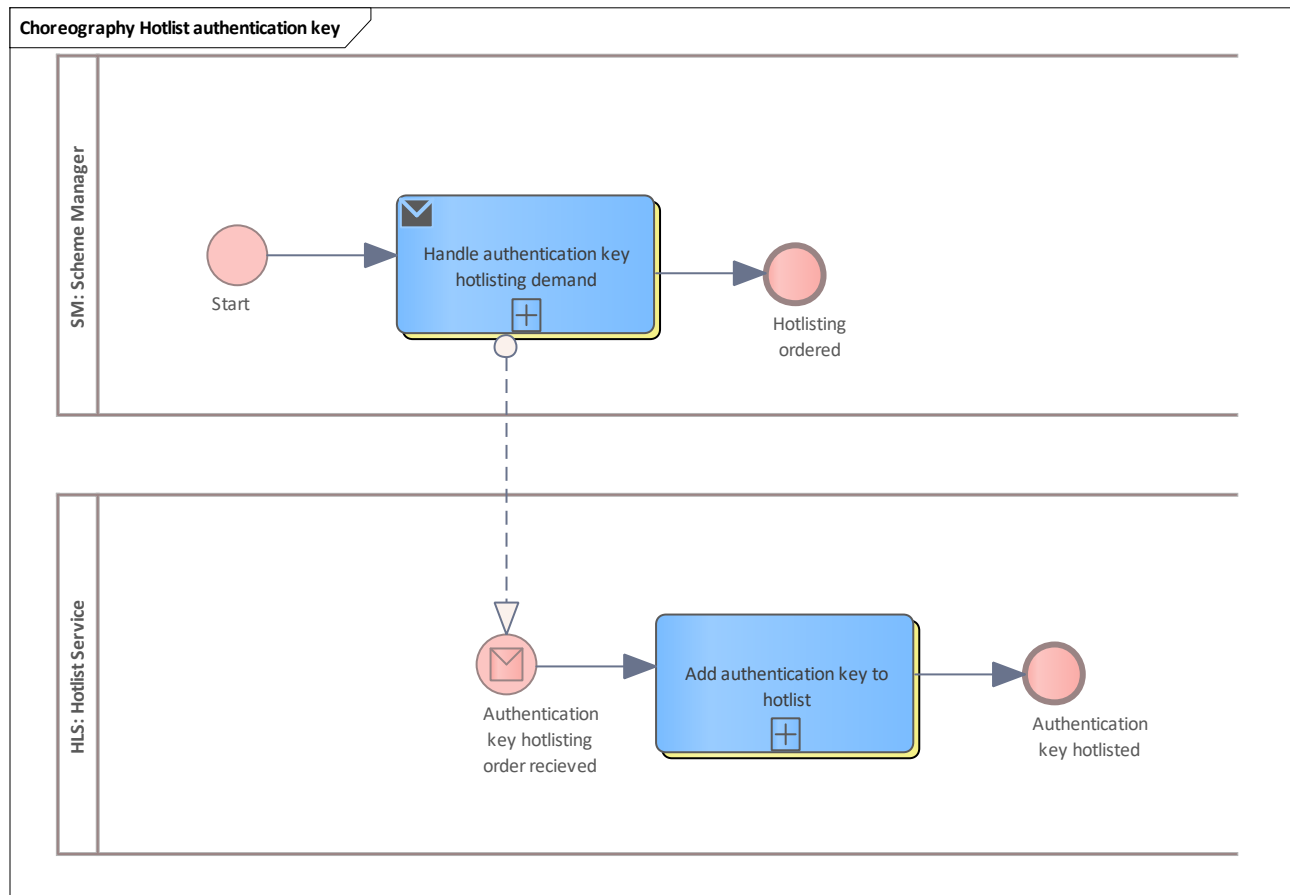


Figure 105: Hotlist authentication key

9.2.22.12.1 SM

See [Scheme Manager](#)

1.1.1.1.1.81 Handle authentication key hotlisting demand

See [Handle authentication key hotlisting demand](#)

9.2.22.12.2 HLS

See [Hotlist Service](#)



1.1.1.1.1.82 Add authentication key to hotlist

See [Add authentication key to hotlist](#)

9.2.22.13 Hotlist organisation

This chapter describes the hotlisting process of an organisation, aided by BPMN collaboration diagrams.

9.2.22.14 Hotlist organisation

This basic process describes the hotlisting process of an organisation. This can only be done by the [Scheme Manager](#).

An external request may exist (not shown here) that was added in the VDV-ETS service management.

An authorised employee checks the request and adds the organisation to the hotlist.

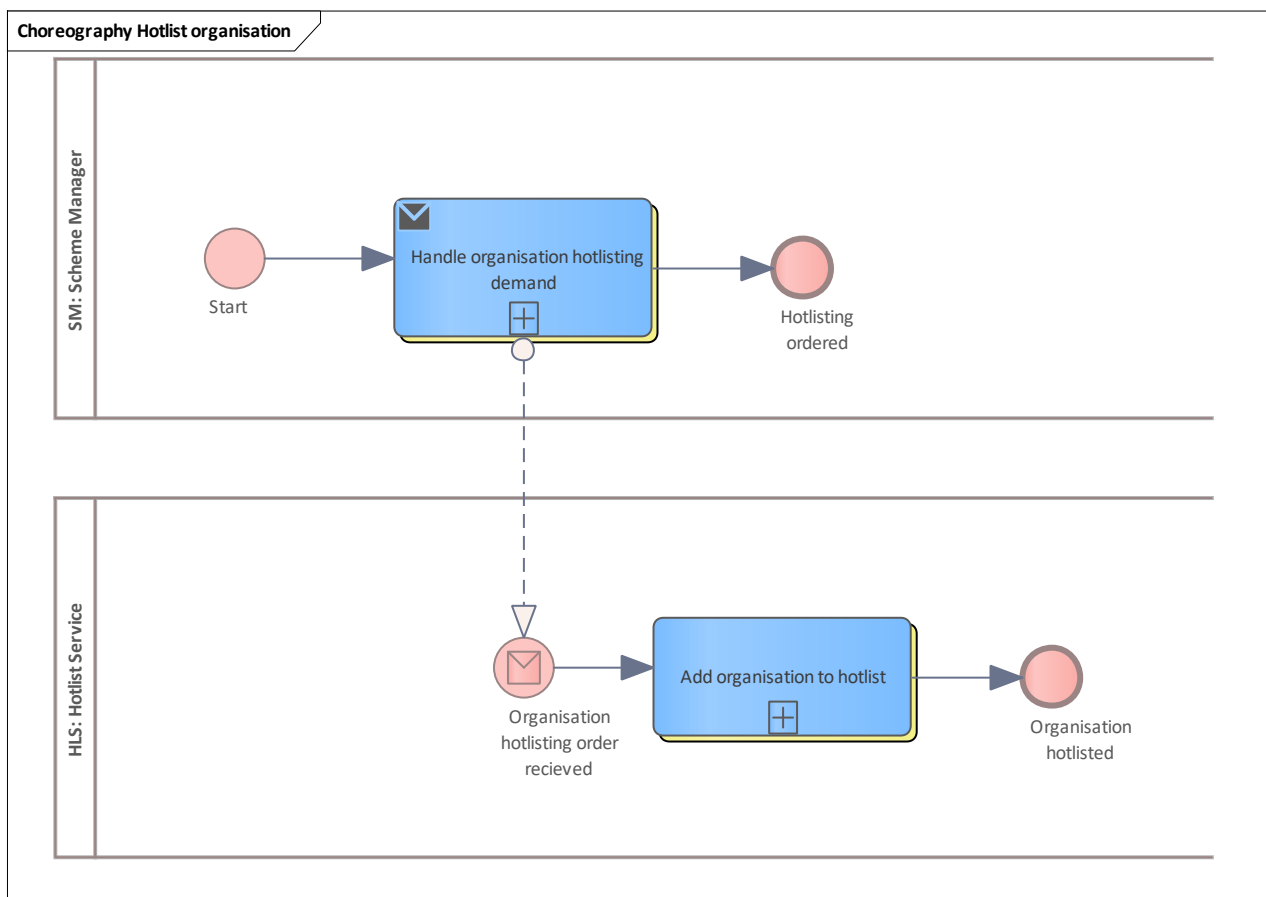


Figure 106: Hotlist organisation

9.2.22.14.1 SM

See [Scheme Manager](#)

1.1.1.1.1.83 Handle organisation hotlisting demand

See [Handle organisation hotlisting demand](#).



1.1.1.1.1.84 Start

9.2.22.14.2 HLS

See [Hotlist Service](#)

1.1.1.1.1.85 Add organisation to hotlist

See [Add organisation to hotlist](#).

9.2.22.15 Hotlist SAM

This chapter describes the hotlisting process of a SAM, aided by BPMN collaboration diagrams. The hotlisting can be initiated by the SAM owner itself or by a third party (e.g. monitoring of a PO system).

Furthermore, the scheme manager can be requested to add a SAM to the SAM hotlist or he might perform this due to internal security reasons.

9.2.22.16 Hotlist non-owned SAM

The basic process performs the hotlisting of a SAM triggered by a third party (not the SAM owner). The demand for hotlisting can be done by an SO, sCCP or PO.

In most cases, the monitoring processes will trigger this process. If any problems with the SAM are detected that require the hotlisting of the SAM, the demand is sent to the SAM owner.

The owner of the SAM (the CCP or SO) then verifies the hotlist demand and requests adding the SAM to the SAM hotlist.

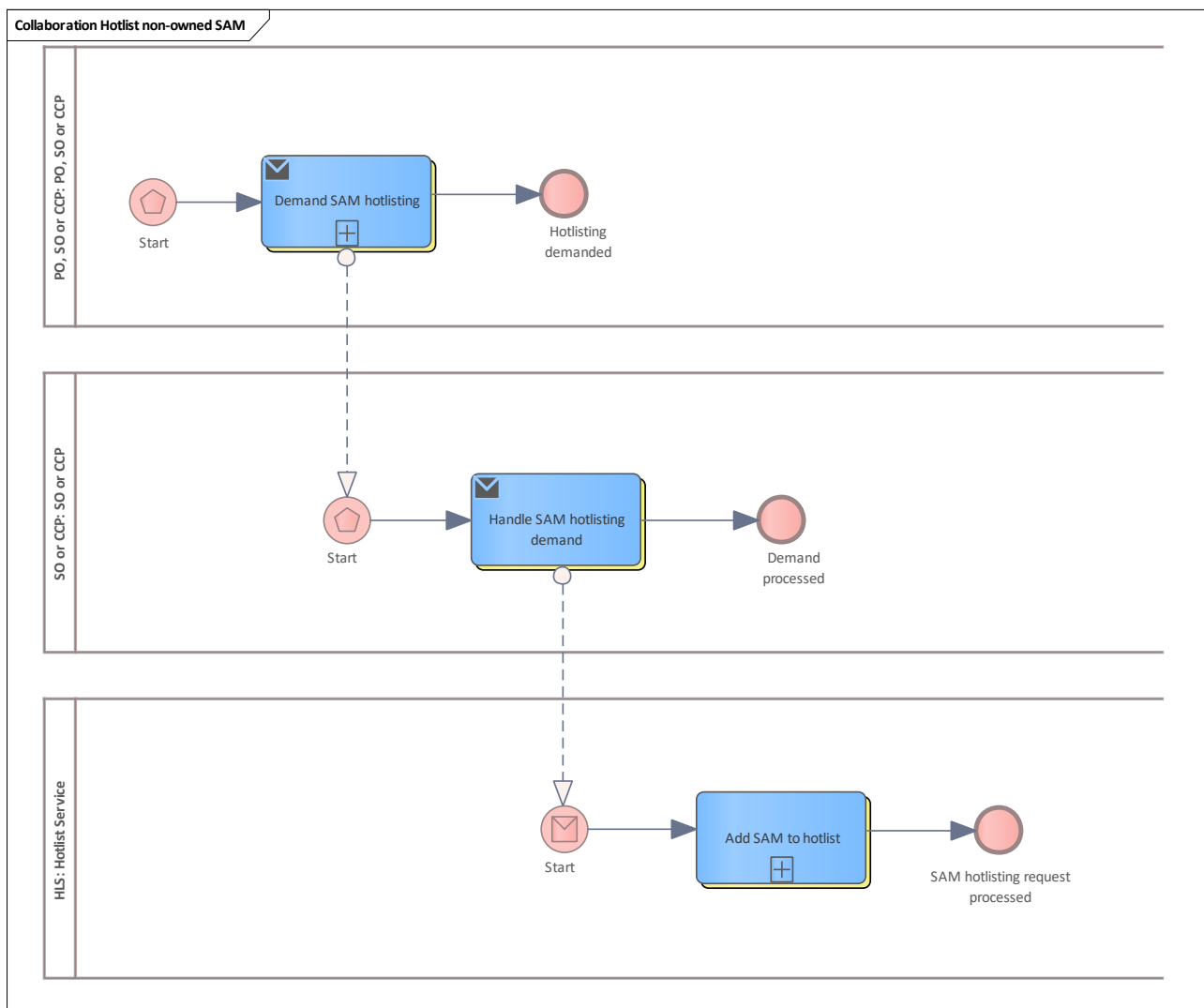


Figure 107: Hotlist non-owned SAM

9.2.22.16.1 PO, SO or CCP

See [PO, SO or CCP](#)

1.1.1.1.1.86 Demand SAM hotlisting

See [Demand SAM hotlisting](#).

1.1.1.1.1.87 Start

The demand to hotlist a SAM can be triggered by a PO, SO or CCP, that does not actually own the SAM.

9.2.22.16.2 SO or CCP

See [SO or CCP](#)

1.1.1.1.1.88 Handle SAM hotlisting demand

See [Handle SAM hotlisting demand](#).

1.1.1.1.1.89 Start

The processing of a SAM hotlisting demand is performed by the SAM owner. It can either be triggered by an external entity or by the SAM owner itself.

9.2.22.16.3 HLS

See [Hotlist Service](#)

1.1.1.1.1.90 Add SAM to hotlist

See [Add SAM to hotlist](#).

9.2.22.17 Hotlist owned SAM

The basic process performs the hotlisting of a SAM triggered by the SAM owner itself. The internal monitoring process might trigger this process. Another reason might be a stolen or lost SAM. If any problems with the SAM are detected that require the hotlisting of the SAM, the owner of the SAM (the CCP or SO) requests adding the SAM to the SAM hotlist.

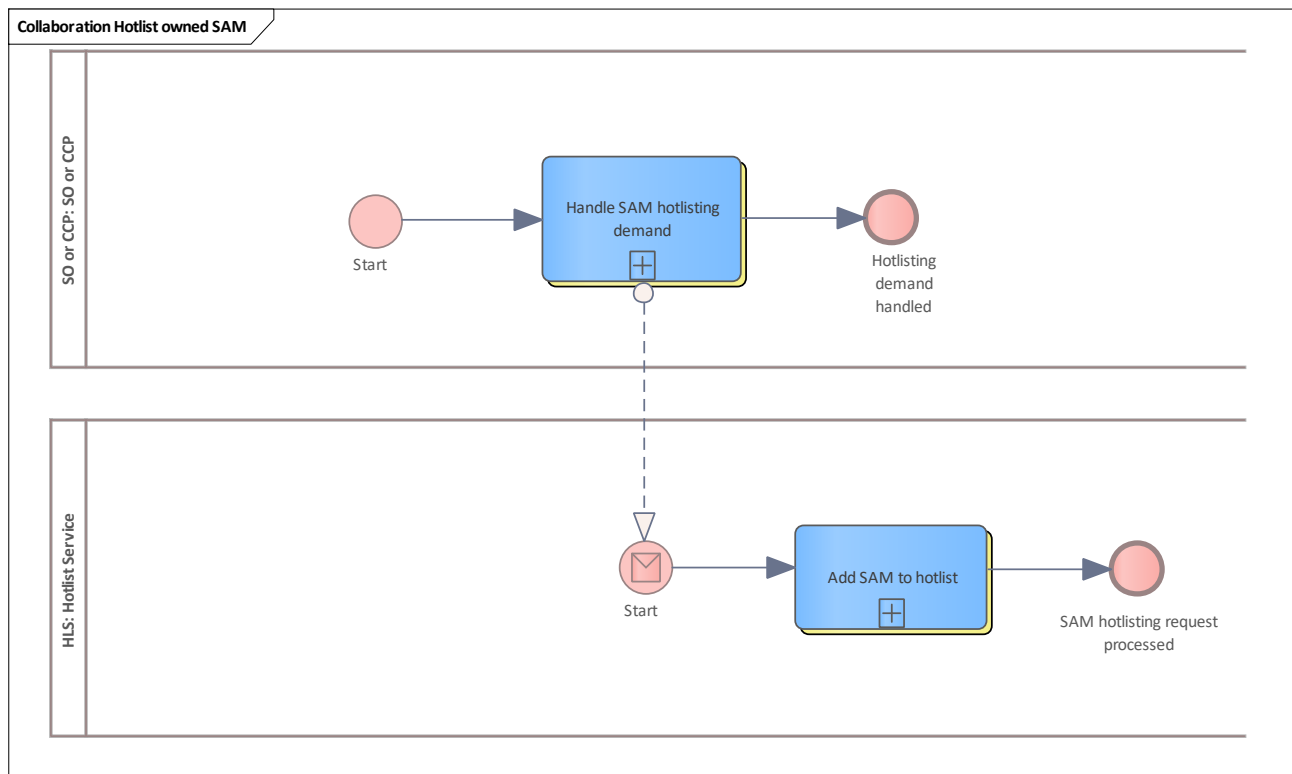


Figure 108: Hotlist owned SAM

9.2.22.17.1 SO or CCP

See [SO or CCP](#)

1.1.1.1.1.91 Handle SAM hotlisting demand

See [Handle SAM hotlisting demand](#).

9.2.22.17.2 HLS

See [Hotlist Service](#)

1.1.1.1.1.92 Add SAM to hotlist

See [Add SAM to hotlist](#).

9.2.22.18 Hotlist SAM as Scheme Manager

The basic process performs the hotlisting of a SAM triggered by the [Scheme Manager](#). The scheme manager can hotlist a SAM if the hotlist demand due to a monitoring process was ignored and the scheme manager was requested by service management to hotlist the involved SAM. Another reason might be caused by internal security monitoring.

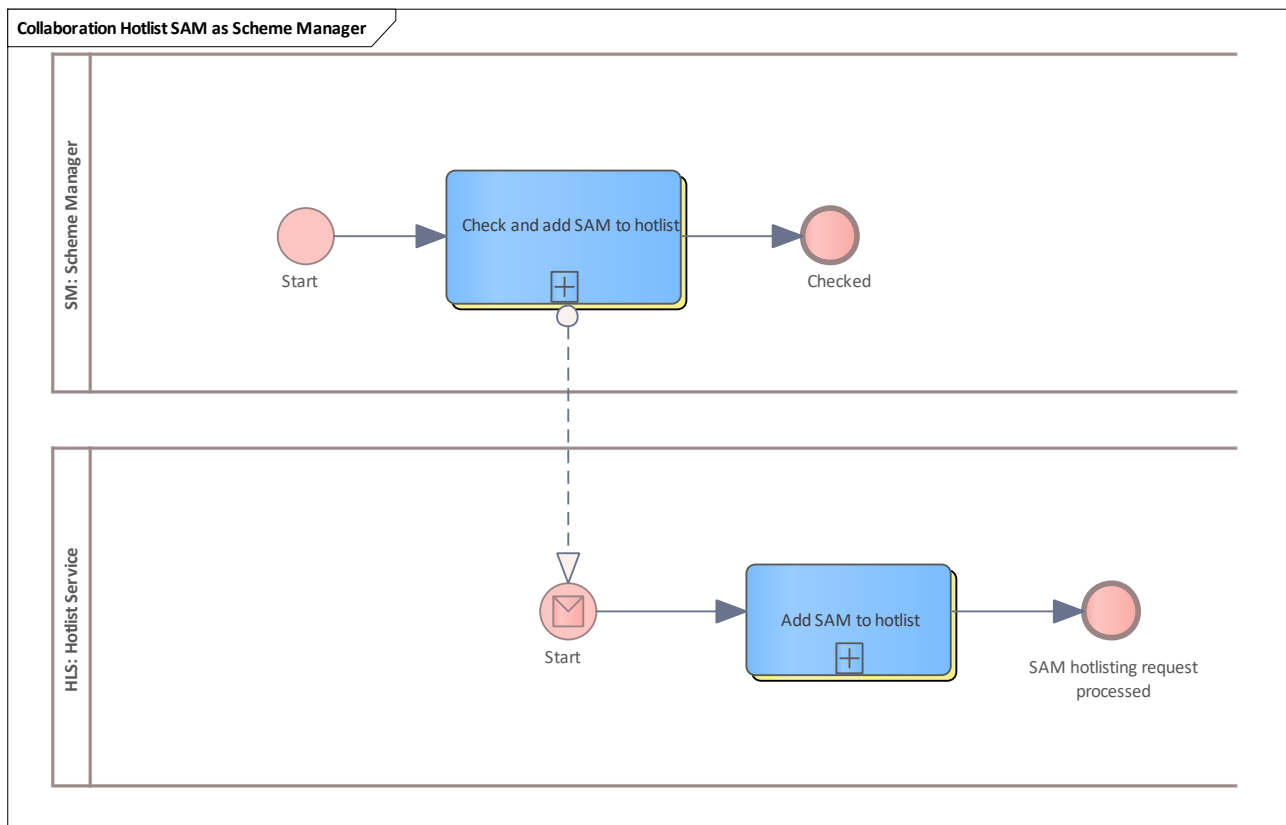


Figure 109: Hotlist SAM as Scheme Manager

9.2.22.18.1 SM

See [Scheme Manager](#)

1.1.1.1.1.93 Check and add SAM to hotlist

See [Check and add SAM to hotlist](#).

9.2.22.18.2 HLS

See [Hotlist Service](#)

1.1.1.1.1.94 Add SAM to hotlist

See [Add SAM to hotlist](#).

9.2.22.19 Retrieve hotlists

This chapter describes the process of retrieving hotlists and next steps, aided by BPMN collaboration diagrams.

Five different hotlists exist which have to be requested by several parties and distributed regularly (SO and CCP as terminal operators).

All hotlists are renewed on a cyclic base, thus all hotlists can be retrieved and updated at the same time.

The cycle interval for the production environment is 24 hours (note: for the staging environment 15 minutes). The cycle is renewed each night (short after 0:00) and accurate to the day before.

This means that all hotlist entries which are active at the moment of the renewal of the cycle (until 0:00) are in the scope of the retrieved hotlists for the next 24 hours and these hotlists will not change until the next renewal of the cycle.

For incremental lists, all changes (additions or removals) of hotlist entries compared with a reference cycle (normally the current cycle - 1) will be in the list.

9.2.22.19.1 Application

Retrieve application hotlist as terminal owner, such as SO or CCP.

9.2.22.19.2 Update application hotlist inventory from operational perspective

Basic process of the retrieval and distribution of application hotlists.

Only terminal operators (CCP and SO) need the application hotlist. Due to the size of the list it is also possible to work with incremental application hotlists. Either the full application hotlist or the incremental application hotlist has to be retrieved regularly.

Optionally, if using the incremental application hotlist, verification can be triggered by the terminal operator system. The system updates its inventory with the incremental elements and calculates a checksum. This checksum is sent to the hotlist service that calculates the checksum over the full hotlist (which the requestor should have) and compares the checksum values. Finally, the internal hotlist inventory is updated with the new hotlist information and the application hotlist is distributed to the terminals.

Note: for process and data efficiency, the distribution of the hotlists to the terminals takes place when all new hotlists are available in the back-office system of the terminal operator.

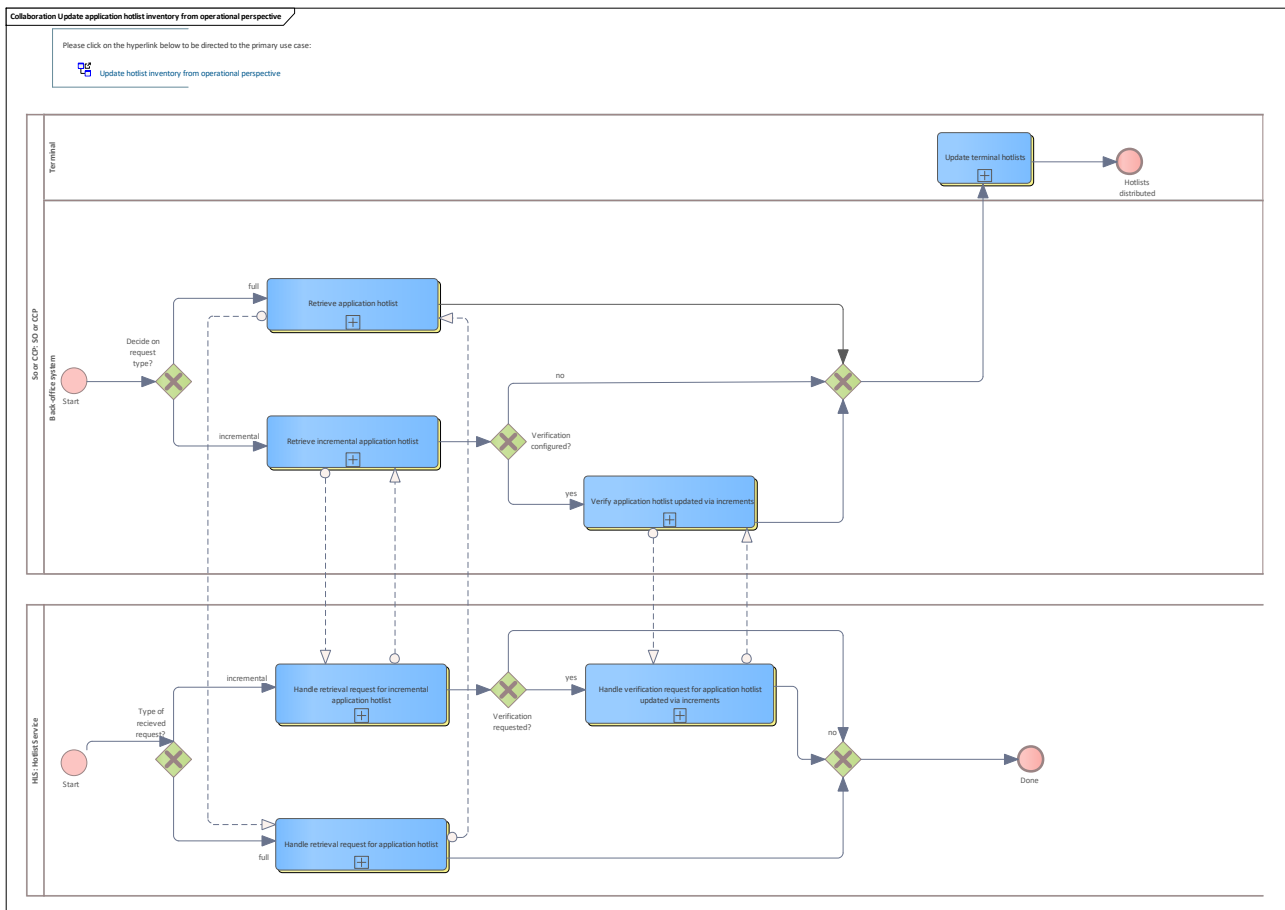


Figure 110: Update application hotlist inventory from operational perspective

1.1.1.1.1.95 So or CCP

See [SO or CCP](#)

1.1.1.1.1.95.1 Back-office system

Lane for a back-office system

1.1.1.1.1.95.1.1 Retrieve application hotlist

See [Retrieve application hotlist](#)

1.1.1.1.1.95.1.2 Retrieve incremental application hotlist

See [Retrieve incremental application hotlist](#)

1.1.1.1.1.95.1.3 Verify application hotlist updated via increments

See [Verify application hotlist updated via increments](#)



1.1.1.1.1.95.2 Terminal

Lane for terminal

1.1.1.1.1.95.2.1 Update terminal hotlists

See [Update terminal hotlists](#)

1.1.1.1.1.96 HLS

See [Hotlist Service](#)

1.1.1.1.1.96.1 Handle verification request for application hotlist updated via increments

See [Handle verification request for application hotlist updated via increments](#)

1.1.1.1.1.96.2 Handle retrieval request for application hotlist

See [Handle retrieval request for application hotlist](#)

1.1.1.1.1.96.3 Handle retrieval request for incremental application hotlist

See [Handle retrieval request for incremental application hotlist](#)

9.2.22.19.3 Entitlement

Retrieve entitlement hotlist from different perspectives (participants with different roles). Due to the size of the list, it is also possible to work with incremental entitlement hotlists. Either the full entitlement hotlist or the incremental entitlement hotlist has to be retrieved regularly. Optionally, if using the incremental entitlement hotlist, verification can be triggered by the requesting system. The requesting system updates its inventory with the incremental elements and calculates a checksum. This checksum is sent to the hotlist service that calculates the checksum over the full hotlist (which the requestor should have due to its configuration) and compares the checksum values.

9.2.22.19.4 Update entitlement hotlist inventory from operational perspective

Basic process of the retrieval and distribution of entitlement hotlists for a terminal operator (CCP or SO).

CCP, SO and PO need the entitlement hotlist. The operational perspective, however, only considers the CCP and SO as terminal operators. The perspective of the PO is described in [Update entitlement hotlist inventory from product perspective](#).

After retrieving the hotlist, the internal hotlist inventory is updated with the new hotlist information and the entitlement hotlist is distributed to the terminals.

Note: for process and data efficiency, the distribution of the hotlists to the terminals takes place when all new hotlists are available in the back-office system of the terminal operator.

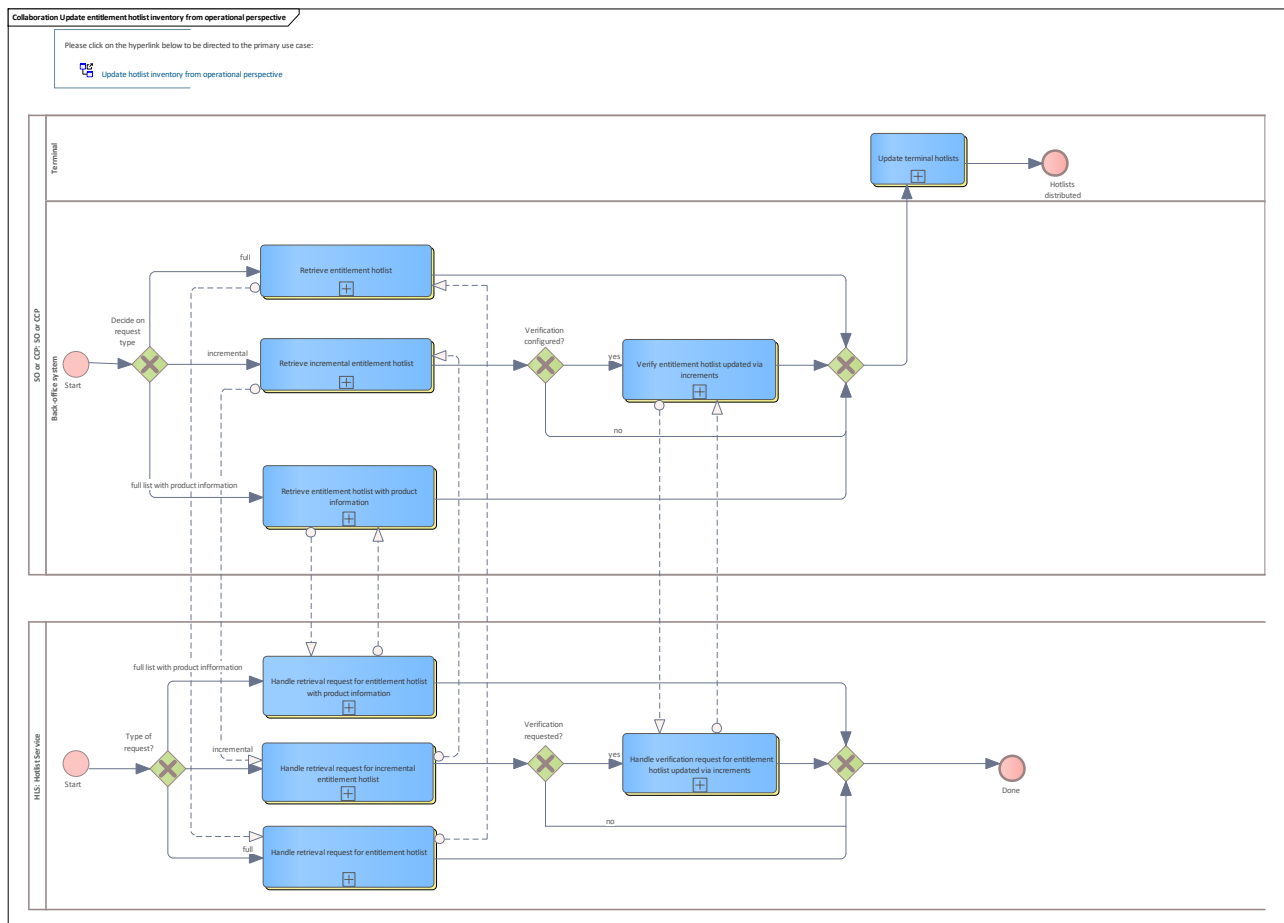


Figure 111: Update entitlement hotlist inventory from operational perspective



1.1.1.1.1.97 SO or CCP

See [SO or CCP](#)

1.1.1.1.1.97.1 Back-office system

Lane for back-office system

1.1.1.1.1.97.1.1 Retrieve entitlement hotlist

See [Retrieve entitlement hotlist](#)

1.1.1.1.1.97.1.2 Retrieve entitlement hotlist with product information

See [Retrieve entitlement hotlist with product information](#)

1.1.1.1.1.97.1.3 Retrieve incremental entitlement hotlist

See [Retrieve incremental entitlement hotlist](#)

1.1.1.1.1.97.1.4 Verify entitlement hotlist updated via increments

See [Verify entitlement hotlist updated via increments](#).

1.1.1.1.1.97.2 Terminal

Lane for terminal

1.1.1.1.1.97.2.1 Update terminal hotlists

See [Update terminal hotlists](#)

1.1.1.1.1.98 HLS

See [Hotlist Service](#)

1.1.1.1.1.98.1 Handle verification request for entitlement hotlist updated via increments

See [Handle verification request for entitlement hotlist updated via increments](#)

1.1.1.1.1.98.2 Handle retrieval request for entitlement hotlist

See [Handle retrieval request for entitlement hotlist](#)

1.1.1.1.1.98.3 Handle retrieval request for entitlement hotlist with product information

See [Handle retrieval request for entitlement hotlist with product information](#)

1.1.1.1.1.98.4 Handle retrieval request for incremental entitlement hotlist

See [Handle retrieval request for incremental entitlement hotlist](#)

9.2.22.19.5 Update entitlement hotlist inventory from product perspective

The basic process of the retrieval of entitlement hotlists as a PO.

CCP, SO and PO need the entitlement hotlist. The product perspective, however, only considers the PO as a monitoring instance.

After retrieving the hotlist, the PO's internal hotlist inventory is updated with the new hotlist information.

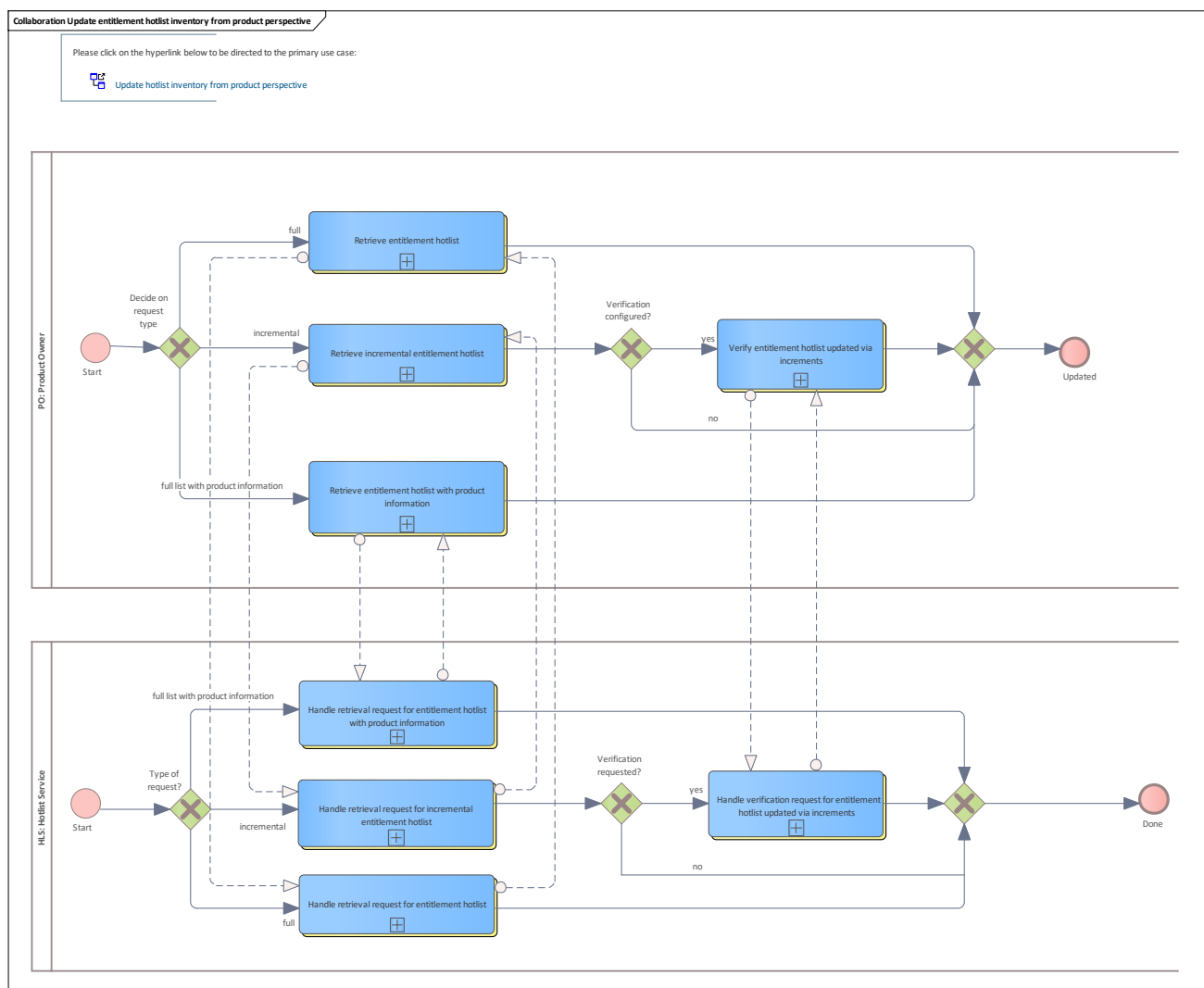


Figure 112: Update entitlement hotlist inventory from product perspective

1.1.1.1.1.99 PO

See [Product Owner](#)

1.1.1.1.1.99.1 Retrieve entitlement hotlist

See [Retrieve entitlement hotlist](#)

1.1.1.1.1.99.2 Retrieve entitlement hotlist with product information

See [Retrieve entitlement hotlist with product information](#)

1.1.1.1.1.99.3 Retrieve incremental entitlement hotlist

See [Retrieve incremental entitlement hotlist](#)

1.1.1.1.1.99.4 Verify entitlement hotlist updated via increments

See [Verify entitlement hotlist updated via increments](#)

1.1.1.1.1.100 HLS

See [Hotlist Service](#)

1.1.1.1.1.100.1 Handle retrieval request for entitlement hotlist

See [Handle retrieval request for entitlement hotlist](#)

1.1.1.1.1.100.2 Handle retrieval request for entitlement hotlist with product information

See [Handle retrieval request for entitlement hotlist with product information](#)

1.1.1.1.1.100.3 Handle retrieval request for incremental entitlement hotlist

See [Handle retrieval request for incremental entitlement hotlist](#)

1.1.1.1.1.100.4 Handle verification request for entitlement hotlist updated via increments

See [Handle verification request for entitlement hotlist updated via increments](#)

9.2.22.19.6 SAM

Retrieve the SAM hotlist from various perspectives (participants with different roles). CCP, SO, PO and the Scheme Manager need the SAM hotlist. The SAM hotlist is always delivered as a full hotlist since the size is expected to be small.

9.2.22.19.7 Update SAM hotlist inventory from operational perspective

Basic process of the retrieval and distribution of a SAM hotlist for a terminal operator (CCP or SO).

The operational perspective only considers the CCP and SO as terminal operators. The perspective of the PO is described in [Update SAM hotlist inventory from product perspective](#). After retrieving the hotlist, the internal hotlist inventory of the terminal operator is updated with the new hotlist information and the SAM hotlist is distributed to the terminals.

Note: for process and data efficiency, the distribution of the hotlists to the terminals takes place when all new hotlists are available in the back-office system of the terminal operator.

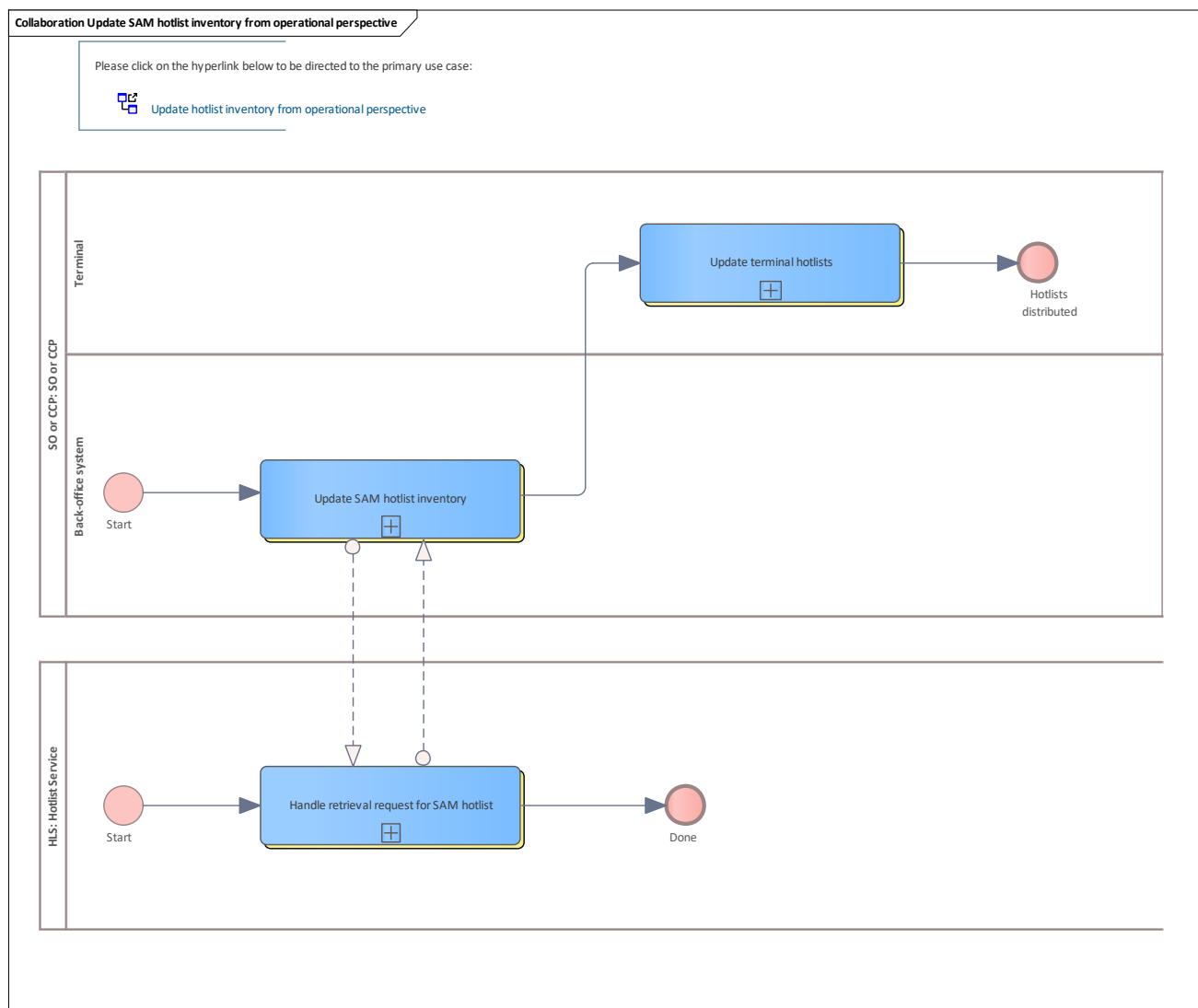


Figure 113: Update SAM hotlist inventory from operational perspective

1.1.1.1.1.101 SO or CCP

See [SO or CCP](#)

1.1.1.1.1.101.1 Back-office system

Lane for a back-office system

1.1.1.1.1.101.1.1 Update SAM hotlist inventory

See [Update SAM hotlist inventory](#)

1.1.1.1.1.101.2 Terminal

Lane for terminal

1.1.1.1.1.101.2.1 Update terminal hotlists

See [Update terminal hotlists](#)

1.1.1.1.1.102 HLS

See [Hotlist Service](#)

9.2.22.19.8 Update SAM hotlist inventory from product perspective

Basic process of the retrieval of a SAM hotlist as a PO.

The product perspective only considers the PO as a monitoring instance.

After retrieving the hotlist, the internal hotlist inventory of the PO system is updated with the new hotlist information.

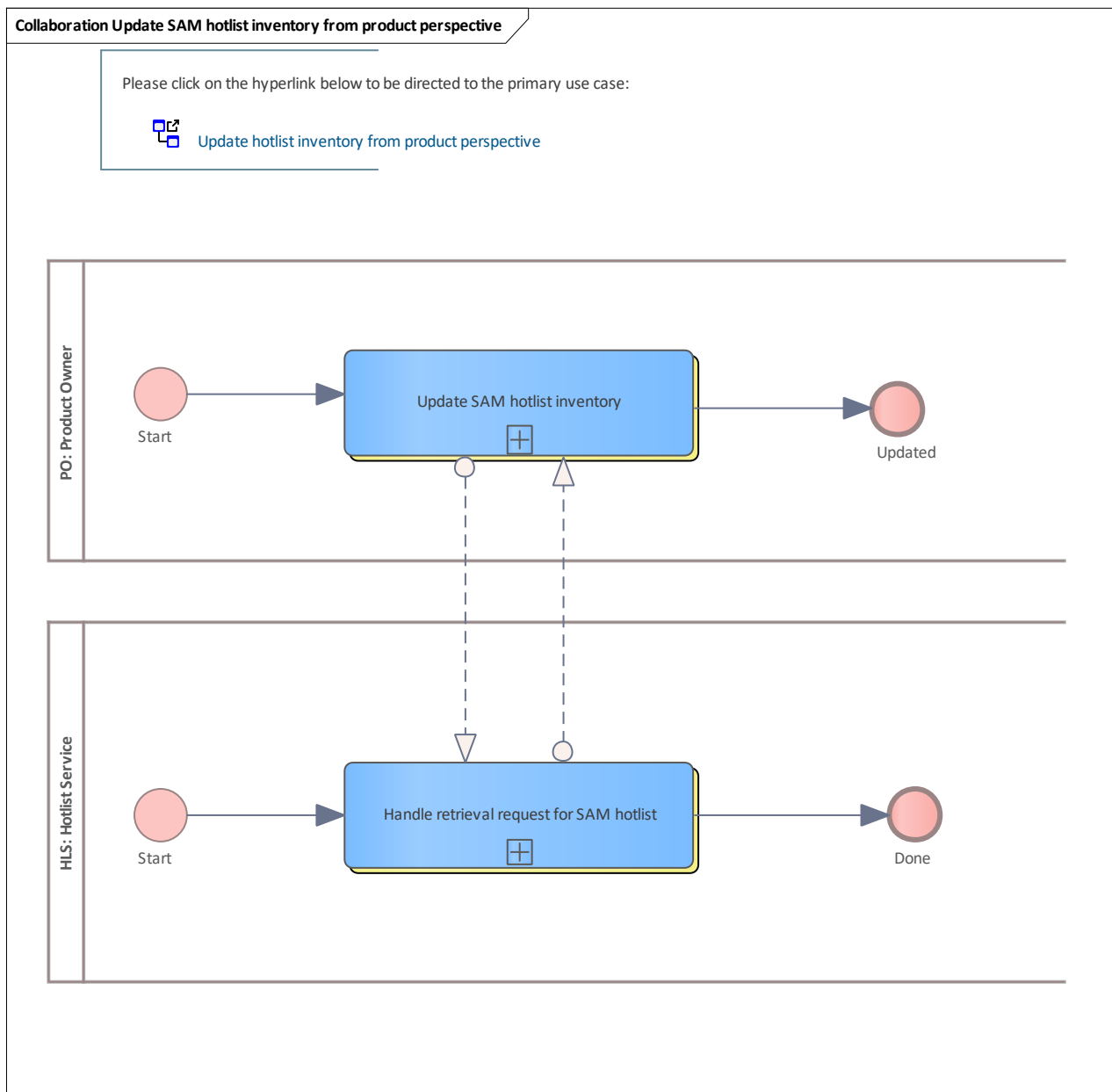


Figure 114: Update SAM hotlist inventory from product perspective

1.1.1.1.1.103 PO

See [Product Owner](#)

1.1.1.1.1.103.1 Update SAM hotlist inventory

See [Update SAM hotlist inventory](#)

1.1.1.1.1.104 HLS

See [Hotlist Service](#)

1.1.1.1.1.104.1 Handle retrieval request for SAM hotlist

See [Handle retrieval request for SAM hotlist](#)

9.2.22.19.9 Update SAM hotlist inventory from scheme manager perspective

Basic process of the retrieval of a SAM hotlist as the [Scheme Manager](#).

The ESH of the scheme manager will fetch the SAM hotlist and forward it to the Media Management System (MMS). The ESH uses the information to provide it for the users (staff of the public transport companies).

The Media Management System MMS will deactivate the SAMs that are contained in the SAM hotlist. These SAMs cannot be reconfigured and the validity cannot be extended.

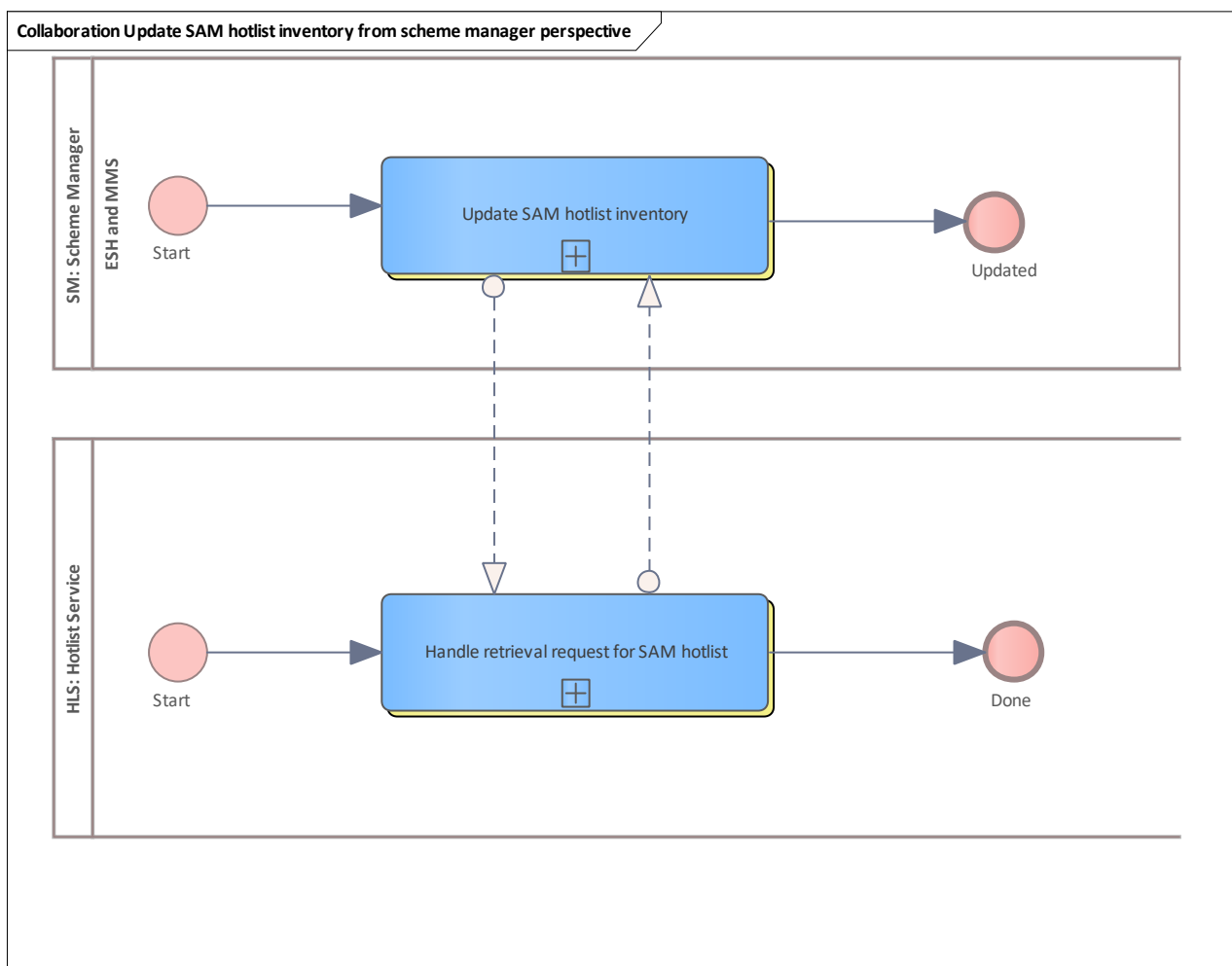


Figure 115: Update SAM hotlist inventory from scheme manager perspective

1.1.1.1.1.105 SM

See [Scheme Manager](#)



1.1.1.1.1.105.1 ESH and MMS

Lane for ESH and MMS

1.1.1.1.1.105.1.1 Update SAM hotlist inventory

See [Update SAM hotlist inventory](#)

1.1.1.1.1.106 HLS

See [Hotlist Service](#)

1.1.1.1.1.106.1 Handle retrieval request for SAM hotlist

See [Handle retrieval request for SAM hotlist](#)

9.2.22.19.10 Organisation

Retrieve the organisation hotlist from various perspectives (participants with different roles). CCP, SO, PO and the Scheme Manager need the organisation hotlist.

The organisation hotlist is always delivered as a full hotlist since the size is expected to be very small.

9.2.22.19.11 Update organisation hotlist inventory from operational perspective

Basic process of the retrieval and distribution of an organisation hotlist for a terminal operator (CCP or SO).

After retrieving the hotlist, the internal hotlist inventory of the terminal operator is updated with the new hotlist information and the organisation hotlist is distributed to the terminals.

Note: for process and data efficiency, the distribution of the hotlists to the terminals takes place when all new hotlists are available in the back-office system of the terminal operator.

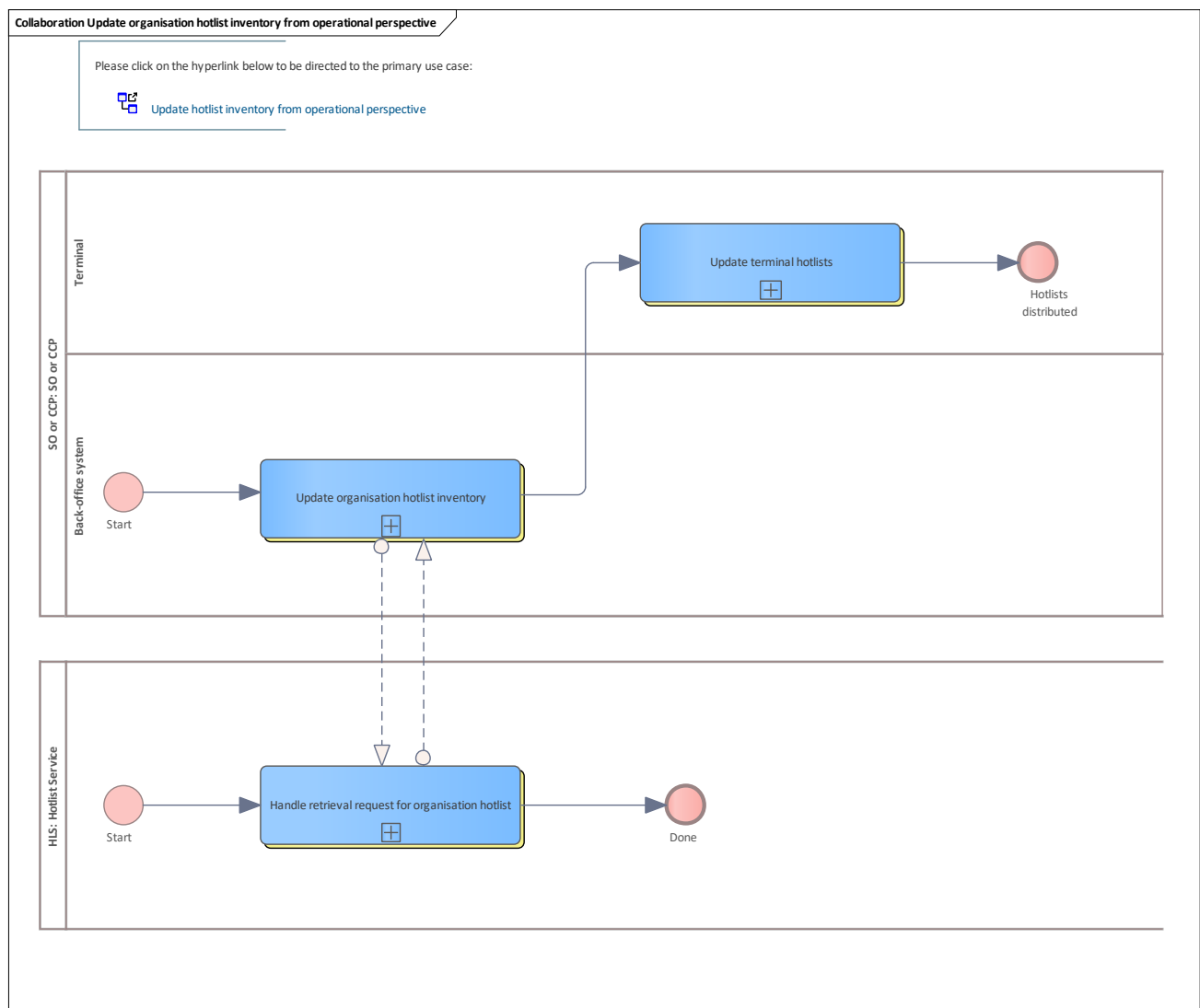


Figure 116: Update organisation hotlist inventory from operational perspective

**1.1.1.1.1.107 SO or CCP**

See [SO or CCP](#)

1.1.1.1.1.107.1 Back-office system

Lane for a back-office system

1.1.1.1.1.107.1.1 Update organisation hotlist inventory

See [Update organisation hotlist inventory](#)

1.1.1.1.1.107.2 Terminal

Lane for terminal

1.1.1.1.1.107.2.1 Update terminal hotlists

See [Update terminal hotlists](#)

1.1.1.1.1.108 HLS

See [Hotlist Service](#)

1.1.1.1.1.108.1 Handle retrieval request for organisation hotlist

See [Handle retrieval request for organisation hotlist.](#)

9.2.22.19.12 Update organisation hotlist inventory from product perspective

Basic process of the retrieval and distribution of an organisation hotlist as a PO.
After retrieving the hotlist, the internal hotlist inventory of the PO system is updated with the new hotlist information for monitoring purposes.

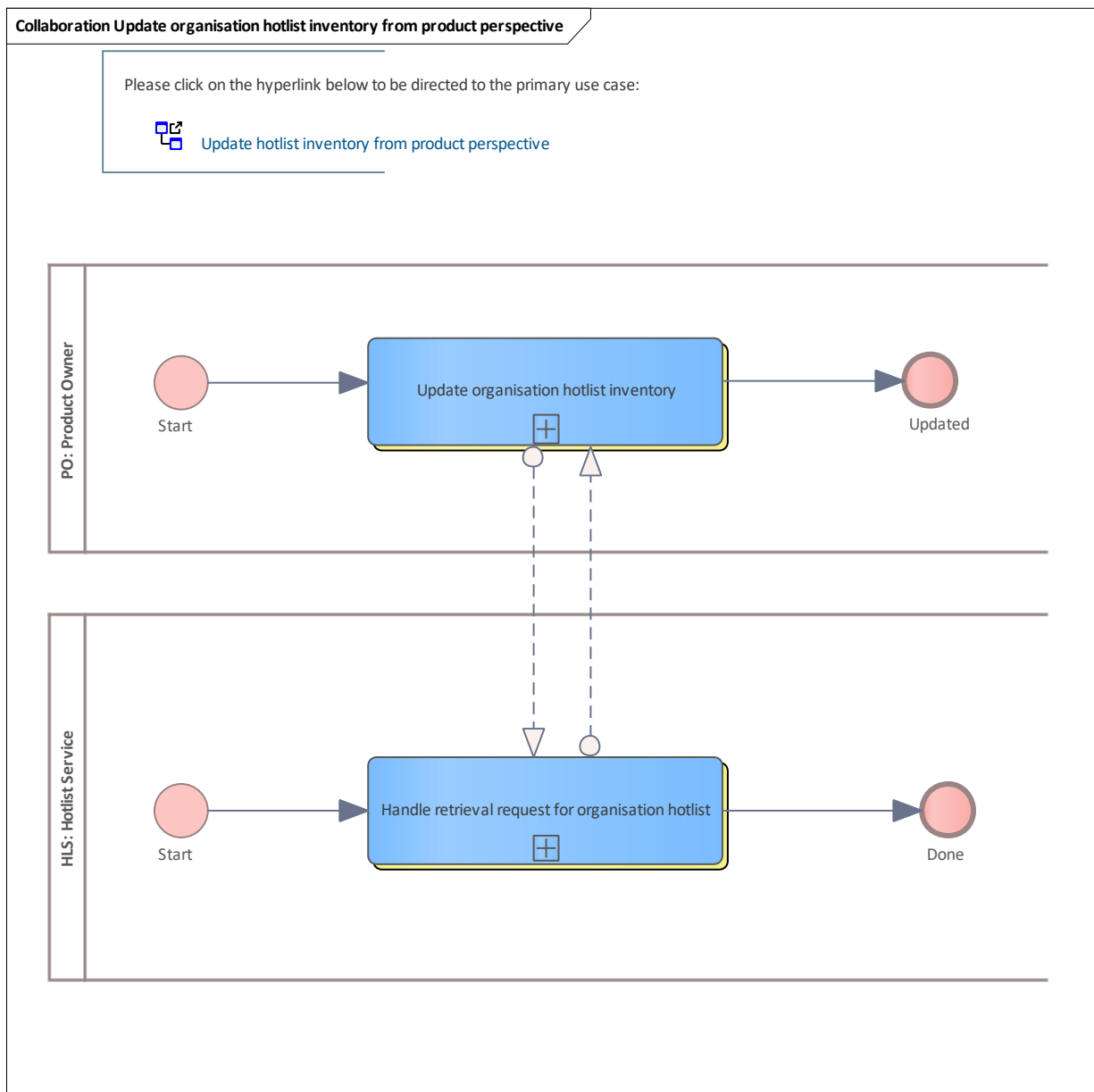


Figure 117: Update organisation hotlist inventory from product perspective

1.1.1.1.1.109 PO

See [Product Owner](#)

1.1.1.1.1.109.1 Update organisation hotlist inventory

See Update organisation hotlist inventory

1.1.1.1.1.110 HLS

See [Hotlist Service](#)

1.1.1.1.110.1 Handle retrieval request for organisation hotlist

See [Handle retrieval request for organisation hotlist](#).

9.2.22.19.13 Update organisation hotlist inventory from scheme manager perspective

Basic process of the retrieval and distribution of an organisation hotlist as the Scheme Manager. The ESH is responsible for this process.

After retrieving the hotlist, the internal hotlist inventory of the ESH system is updated with the new hotlist information for monitoring purposes. Furthermore, the ESH uses the information to provide it for the users (staff of the public transport companies).

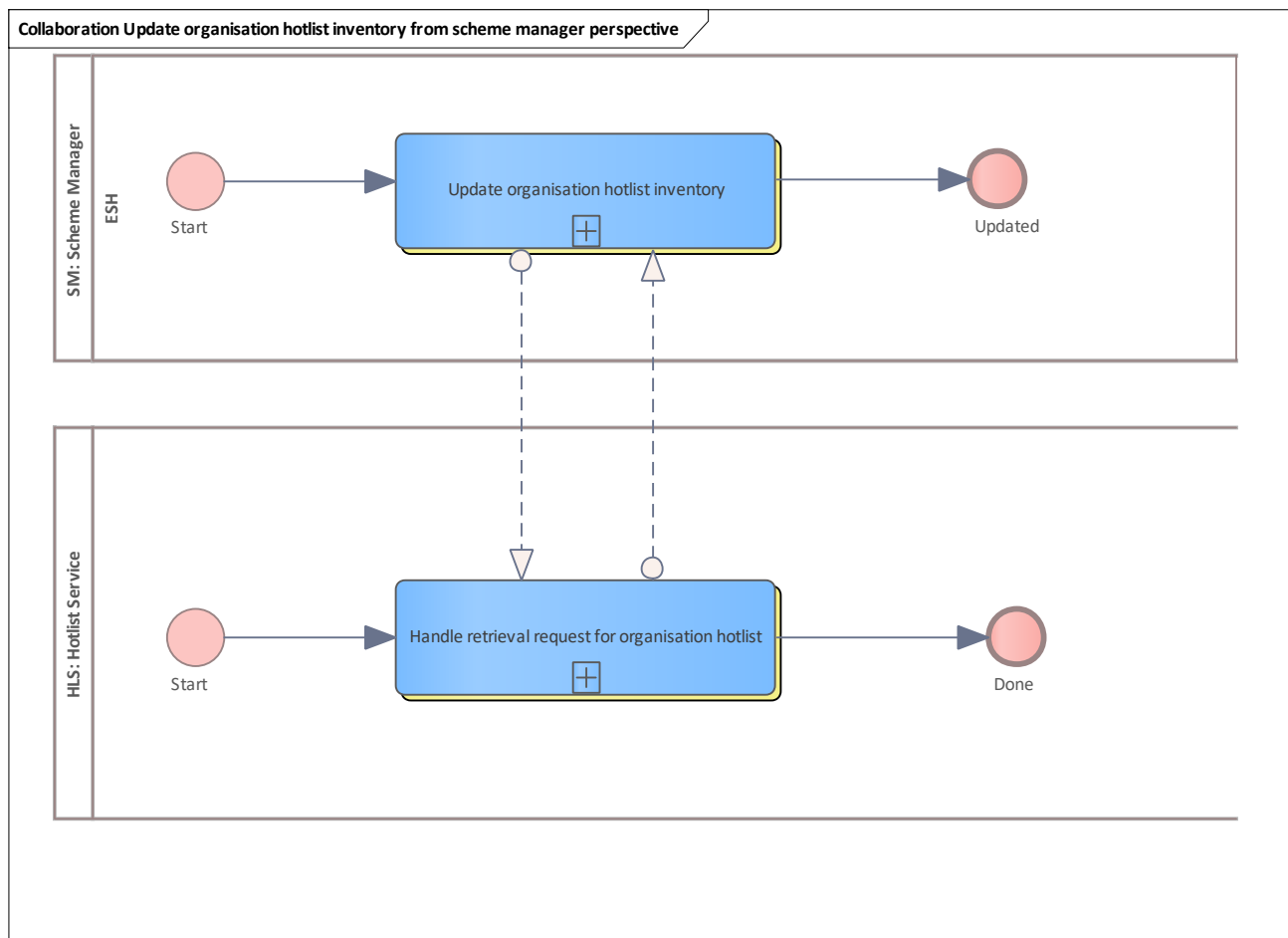


Figure 118: Update organisation hotlist inventory from scheme manager perspective

1.1.1.1.111 SM

See [Scheme Manager](#)

1.1.1.1.111.1 ESH

Lane for ESH



1.1.1.1.1.111.1.1 Update organisation hotlist inventory

See [Update organisation hotlist inventory](#)

1.1.1.1.1.112 HLS

See [Hotlist Service](#)

1.1.1.1.1.112.1 Handle retrieval request for organisation hotlist

See [Handle retrieval request for organisation hotlist](#).

9.2.22.19.14 Authentication key

Retrieve the authentication key hotlist from various perspectives.

CCP, SO and the Scheme Manager need the authentication key hotlist.

The authentication key hotlist is always delivered as a full hotlist since the size is expected to be very small.

The authentication key is used in the SAM and user medium to establish a secure session in which data exchange and transactions can be performed. Due to security reasons, it might become necessary the switch to the authentication key of the next generation. This is done by hotlisting the involved key, the switch to the next authentication key is done automatically in SAM and user medium.

9.2.22.19.15 Update authentication key hotlist inventory from operational perspective

Basic process of the retrieval and distribution of authentication key hotlist for a terminal operator (CCP or SO).

The operational perspective only considers the CCP and SO as terminal operators. The perspective of the scheme manager is described in [Update authentication key hotlist inventory from scheme manager perspective](#).

After retrieving the hotlist, the internal hotlist inventory is updated with the new hotlist information and the authentication key hotlist is distributed to the terminals.

Note: for process and data efficiency, the distribution of the hotlists to the terminals takes place when all new hotlists are available in the back-office system of the terminal operator.

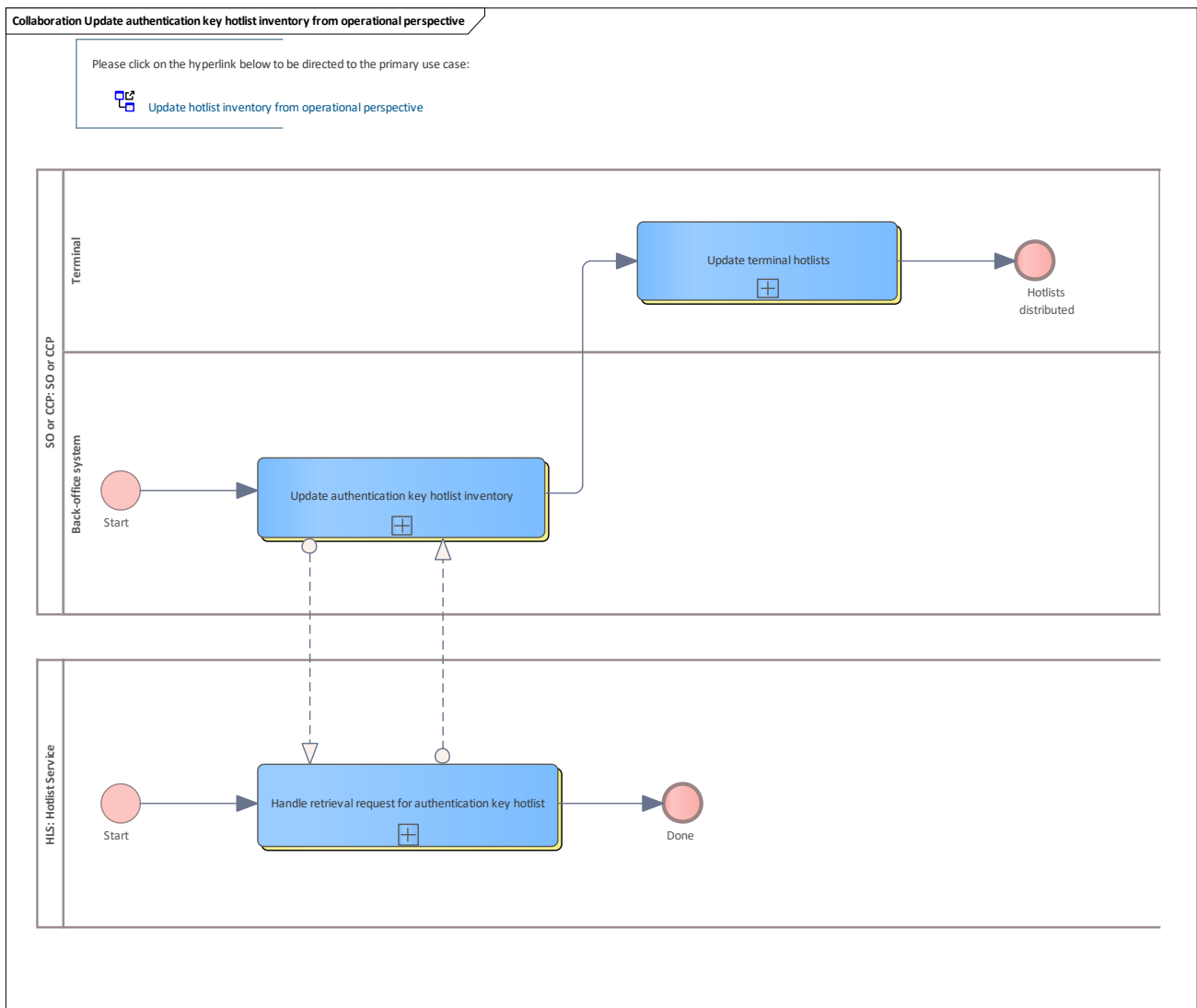


Figure 119: Update authentication key hotlist inventory from operational perspective

1.1.1.1.1.113 SO or CCP

See [SO or CCP](#)

1.1.1.1.1.113.1 Back-office system

Lane for a back-office system

1.1.1.1.1.113.1.1 Update authentication key hotlist inventory

See [Update authentication key hotlist inventory](#)

1.1.1.1.1.113.2 Terminal

Lane for terminal

1.1.1.1.113.2.1 Update terminal hotlists

See [Update terminal hotlists](#)

1.1.1.1.114 HLS

See [Hotlist Service](#)

1.1.1.1.114.1 Handle retrieval request for authentication key hotlist

See [Handle retrieval request for authentication key hotlist](#).

9.2.22.19.16 Update authentication key hotlist inventory from scheme manager perspective

Basic process of the retrieval of authentication key hotlist for the [Scheme Manager](#).

The ESH is responsible for this process.

After retrieving the hotlist, the internal hotlist inventory is updated with the new hotlist information.

Furthermore, the ESH uses the information to provide it for the users (staff of the public transport companies).

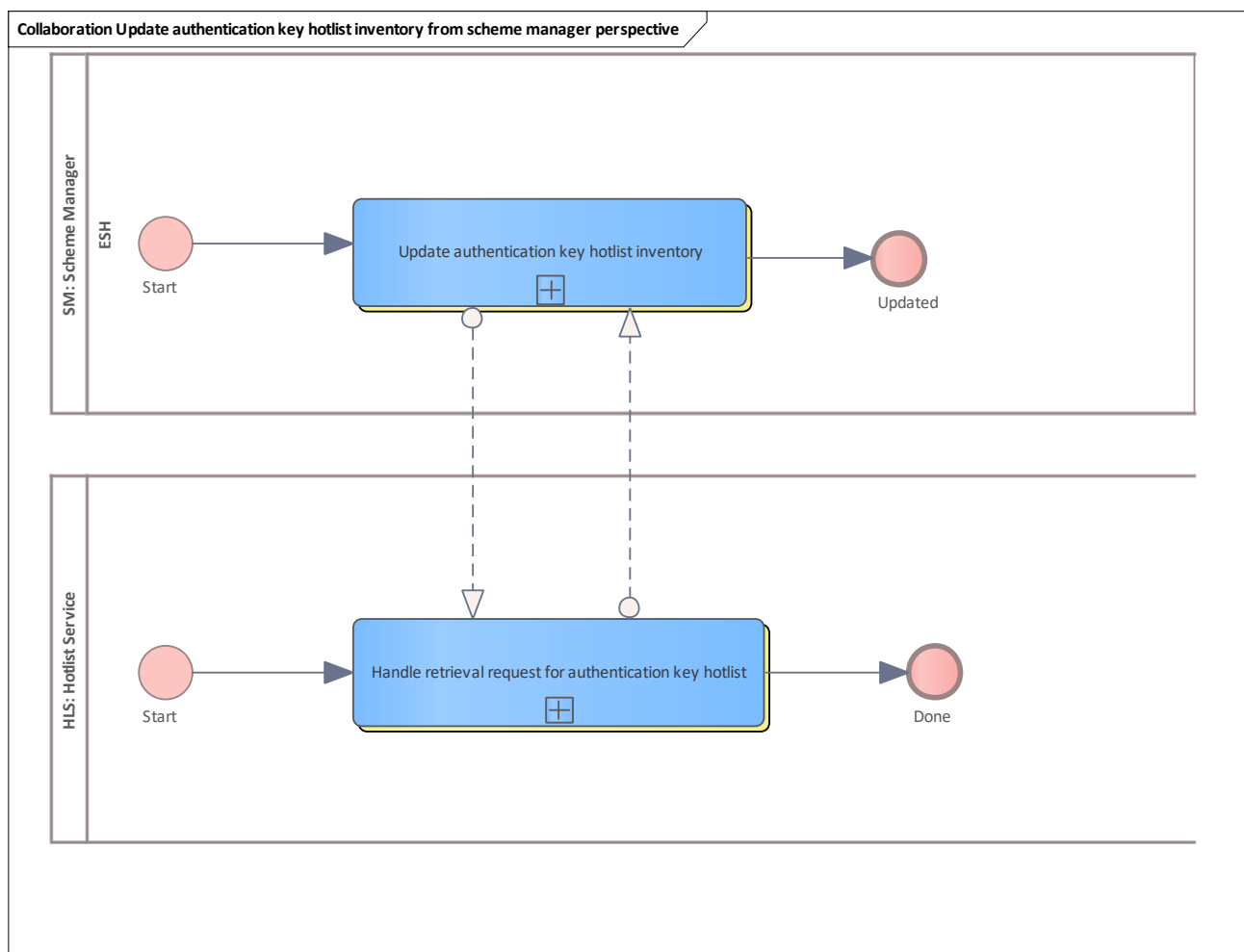


Figure 120: Update authentication key hotlist inventory from scheme manager perspective



1.1.1.1.1.115 SM

See [Scheme Manager](#)

1.1.1.1.1.115.1 ESH

Lane for ESH

1.1.1.1.1.115.1.1 Update authentication key hotlist inventory

See [Update authentication key hotlist inventory](#)

1.1.1.1.1.116 HLS

See [Hotlist Service](#)

1.1.1.1.1.116.1 Handle retrieval request for authentication key hotlist

See [Handle retrieval request for authentication key hotlist](#).

9.2.22.20 Retrieve unclaimed list information

Retrieve information of unclaimed hotlists as PO from the hotlist service.

9.2.22.21 Retrieve unclaimed list information

Basic process for the PO as a public transport association to gain information about the collection behaviour of member companies concerning hotlists. Furthermore, the scheme manager can access this information too. Since the PO is also required to update its own hotlists, the scheme manager can verify this.

Unclaimed list information includes information regarding which hotlist was not retrieved in which cycle by which organisation.

The product owner can analyse the list and communicate the results with its accepting organisations. A PO only receives information about the organisations that accept its products. The hotlist service registers each request for a hotlist. The request of the PO contains a reference cycle as a starting point. The hotlist service will collect all information about non-fetched hotlists until the current cycle.

By using this process regularly, monitoring can be established for detecting the usage of old hotlists.

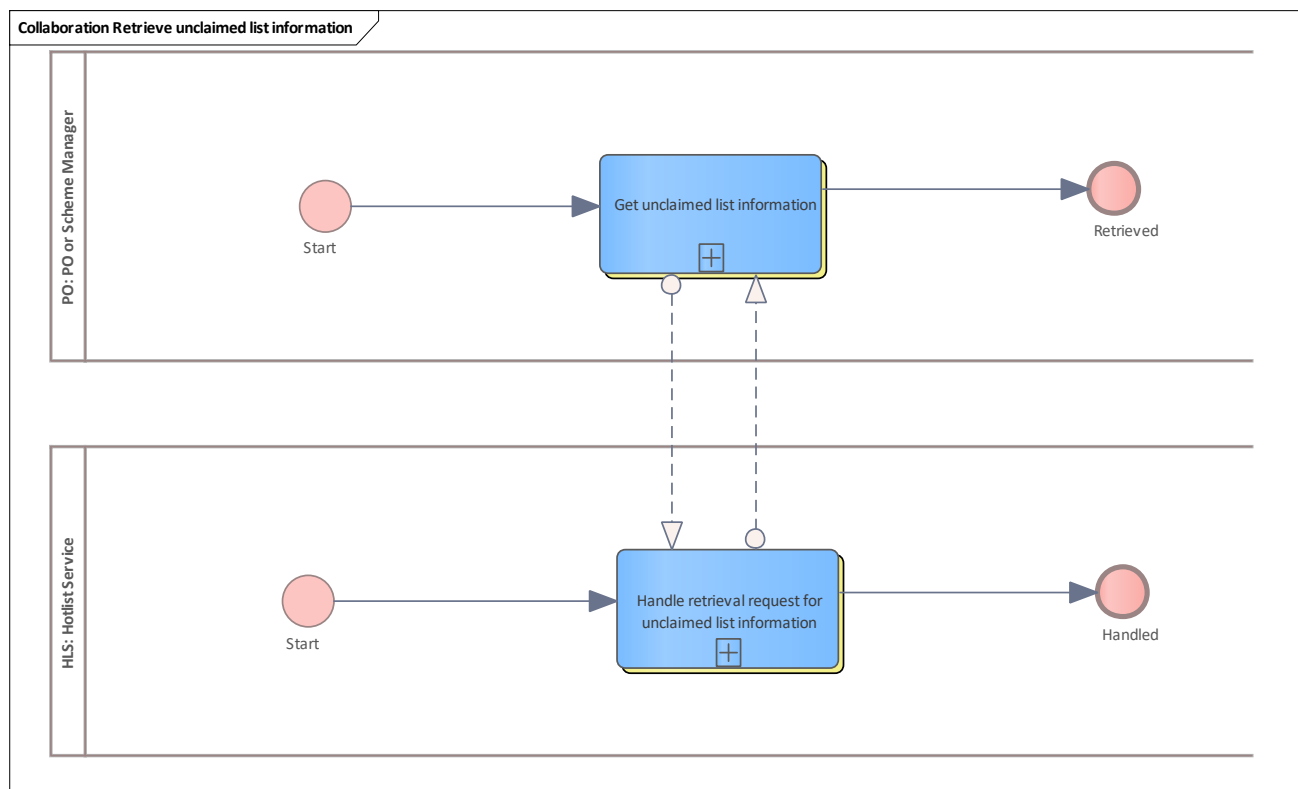


Figure 121: Retrieve unclaimed list information

9.2.22.21.1 PO

See [Product Owner](#)



1.1.1.1.1.117 Get unclaimed list information

See [Get unclaimed list information](#)

9.2.22.21.2 HLS

See [Hotlist Service](#)

1.1.1.1.1.118 Handle retrieval request for unclaimed list information

See [Handle retrieval request for unclaimed list information](#)

9.2.22.22 Revoke hotlisting

This chapter describes the participants and the activities within the basic processes for revocation of hotlisting demands for applications and entitlements.

For SAM, organisations and authentication keys, no defined revocation demands exist. Instead, the related entry can be removed directly by the [Scheme Manager](#) as an authorised instance. BPMN Collaboration diagrams are used to describe these processes.

9.2.22.23 Revoke application hotlisting

This basic process describes the revocation of a previous hotlist demand. This process is rarely used. The revocation can start by a third party (SO, sCCP or PO) where the application instance was not issued.

If the pCCP handles the revocation internally, only the communication with the hotlist service remains.

The revocation might be triggered if all blocking reasons have been cancelled. The aim is to remove the application instance from the application hotlist either to relieve the hotlist or to prevent an unwanted blocking of the application.

Note: this process has another scope as the [Block owned hotlisted application](#) or [Block non-owned hotlisted application](#). In these processes, the last step is removing the application instance from the hotlist done by the pCCP after a blocking process.

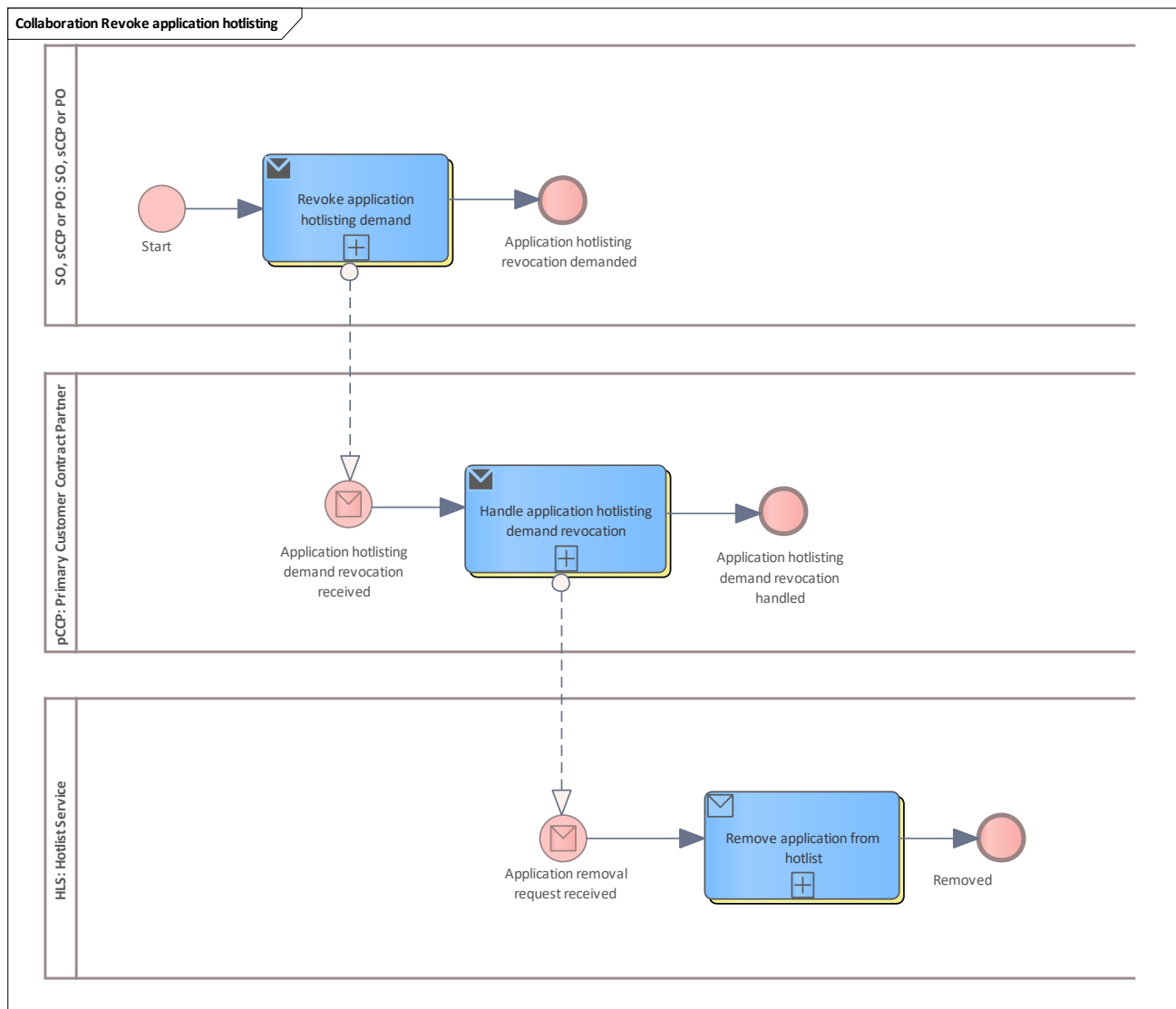


Figure 122: Revoke application hotlisting

9.2.22.23.1 SO, sCCP or PO

See [SO, sCCP or PO](#)

1.1.1.1.1.119 Revoke application hotlisting demand

See [Revoke application hotlisting demand](#)

9.2.22.23.2 pCCP

See [Primary Customer Contract Partner](#)

1.1.1.1.1.120 Handle application hotlisting demand revocation

See [Handle revocation for application hotlisting demand](#)

9.2.22.23.3 HLS

See [Hotlist Service](#)

1.1.1.1.1.121 Remove application from hotlist

See [Remove application from hotlist](#)

9.2.22.24 Revoke entitlement hotlisting

This basic process describes the revocation of a previous hotlist demand. This process is rarely used. The revocation can start by a third party (SO, sCCP or PO) where the entitlement was not issued.

If the pCCP handles the revocation internally, only the communication with the hotlist service remains.

The revocation might be triggered if all blocking reasons have been cancelled. The aim is to remove the entitlement from the entitlement hotlist either to relieve the hotlist or to prevent an unwanted blocking of the entitlement.

Note: this process has another scope as the [Block non-owned entitlement](#) or [Block owned entitlement](#). In these processes, the last step is removing the entitlement from the hotlist done by the pCCP after a blocking process.

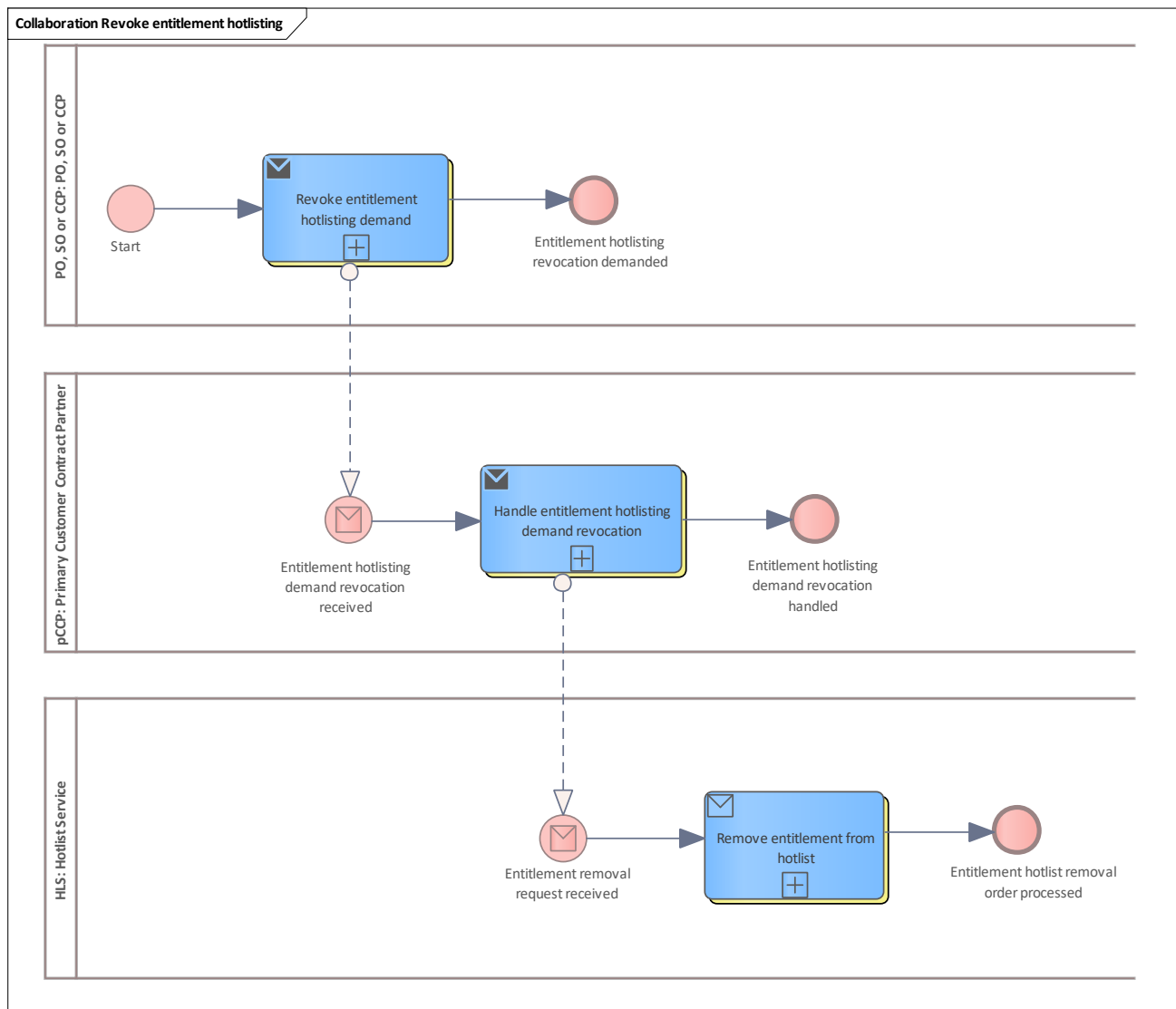


Figure 123: Revoke entitlement hotlisting

9.2.22.24.1 PO, SO or CCP

See [PO, SO or CCP](#)

1.1.1.1.1.122 Revoke entitlement hotlisting demand

See [Revoke entitlement hotlisting demand](#).

9.2.22.24.2 pCCP

See [Primary Customer Contract Partner](#)

1.1.1.1.1.123 Handle entitlement hotlisting demand revocation

See [Handle revocation for entitlement hotlisting demand](#).

9.2.22.24.3 HLS

See [Hotlist Service](#)

1.1.1.1.1.124 Remove entitlement from hotlist

See [Remove entitlement from hotlist](#).

9.2.22.25 Remove SAM from hotlist

This basic process removes a SAM from the SAM hotlist. This process is rarely used and is done between the Scheme Manager and the Hotlist Service.

Normally, SAMs remain on the hotlist. After a long time (> 10 years) or with proof of the SAM scrapping, hotlist entries can be removed to relieve the hotlist.

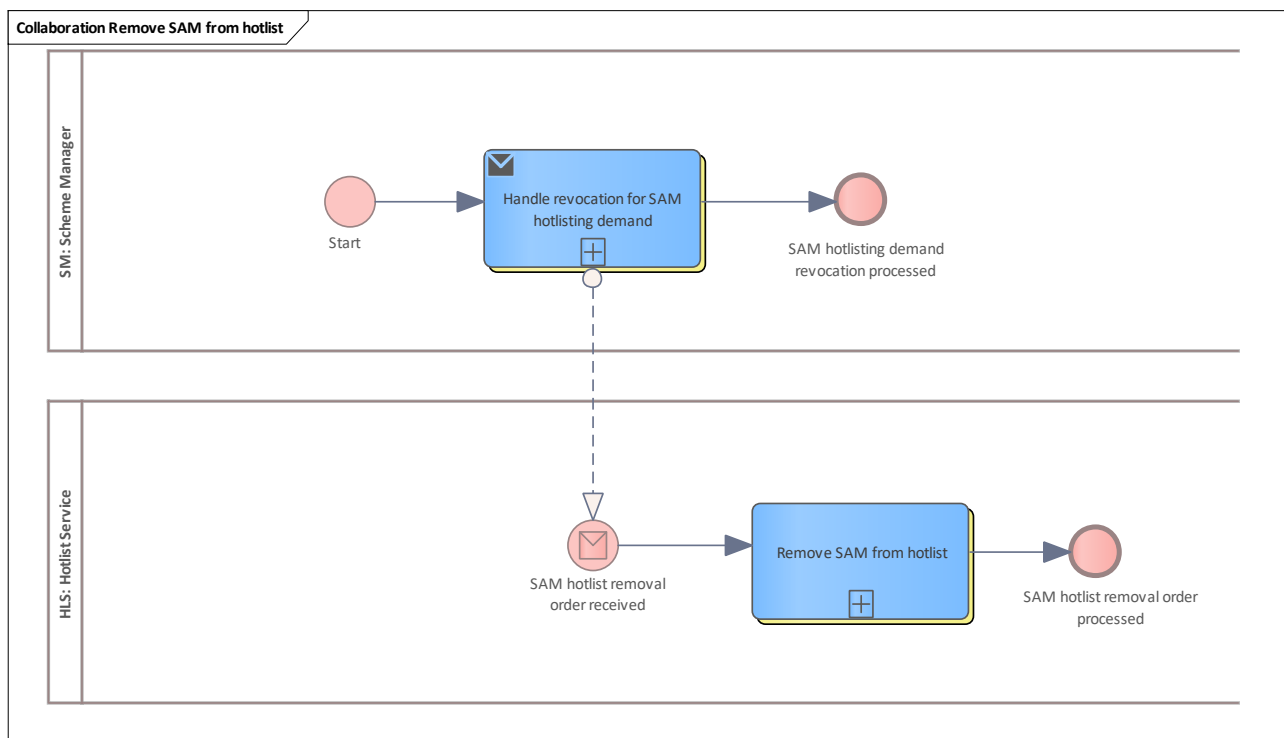


Figure 124: Remove SAM from hotlist

9.2.22.25.1 SM

See [Scheme Manager](#)

1.1.1.1.1.125 Handle revocation for SAM hotlisting demand

See [Handle revocation for SAM hotlisting demand](#).

9.2.22.25.2 HLS

See [Hotlist Service](#)

1.1.1.1.1.126 Remove SAM from hotlist

See [Remove SAM from hotlist](#).

9.2.22.26 Remove organisation from hotlist

This basic process removes an organisation from the organisation hotlist. This process is rarely used and is done between the Scheme Manager and the Hotlist Service. This process is normally preceded by a request via service management.

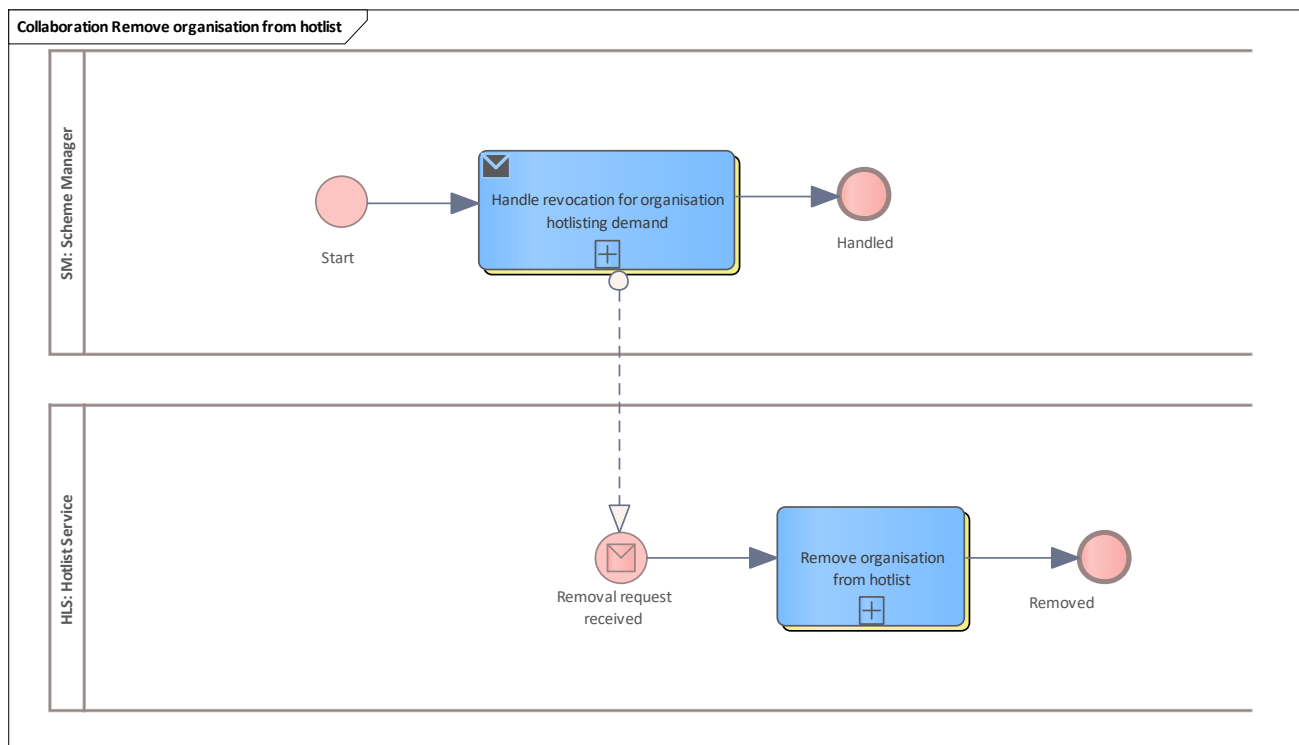


Figure 125: Remove organisation from hotlist

9.2.22.26.1 SM

See [Scheme Manager](#)

1.1.1.1.1.127 Handle revocation for organisation hotlisting demand

See [Handle revocation for organisation hotlisting demand](#)

9.2.22.26.2 HLS

See [Hotlist Service](#)

1.1.1.1.1.128 Remove organisation from hotlist

See [Remove organisation from hotlist](#)

9.2.22.27 Remove authentication key from hotlist

This basic process removes an authentication key from the authentication key hotlist. This process is rarely used and is done between the Scheme Manager and the Hotlist Service.

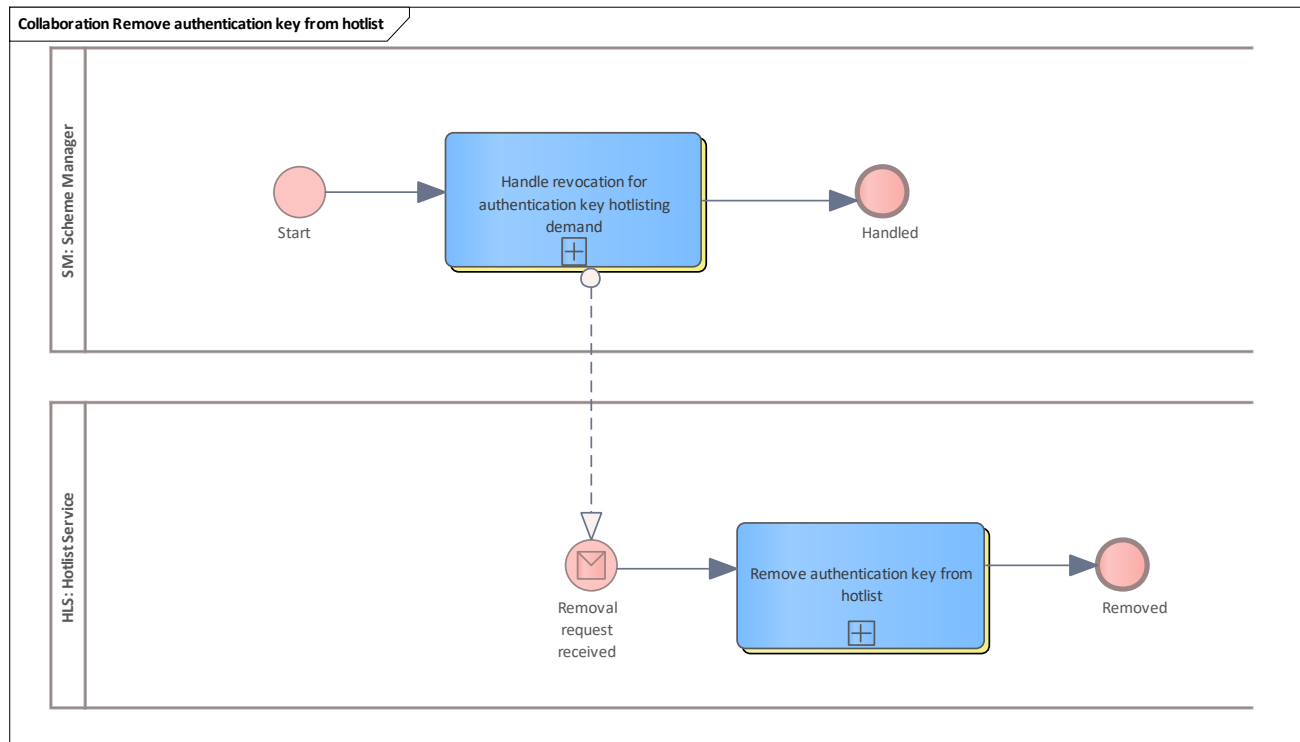


Figure 126: Remove authentication key from hotlist

9.2.22.27.1 SM

See [Scheme Manager](#)

1.1.1.1.1.129 Handle revocation for authentication key hotlisting demand

See [Handle revocation for authentication key hotlisting demand](#).

9.2.22.27.2 HLS

See [Hotlist Service](#)

1.1.1.1.1.130 Remove authentication key from hotlist

See [Remove authentication key from hotlist](#).

9.2.23 Unblocking

This chapter describes the participants and the activities within the basic processes "unblock entitlement" and "unblock application".

BPMN Collaboration is used.

9.2.24 Unblock application

This chapter describes the participants and the activities within the basic process "unblocking application".

BPMN Collaboration is used.

In this case, the typical template process for [Application owned](#) is the foundation of this basic process. This basic process can only be performed by the pCCP with the corresponding permissions.

The process starts in the terminal where these permissions are verified. The terminal notifies the pCCP back-office system. The back-office system does some operational and contractual monitoring.

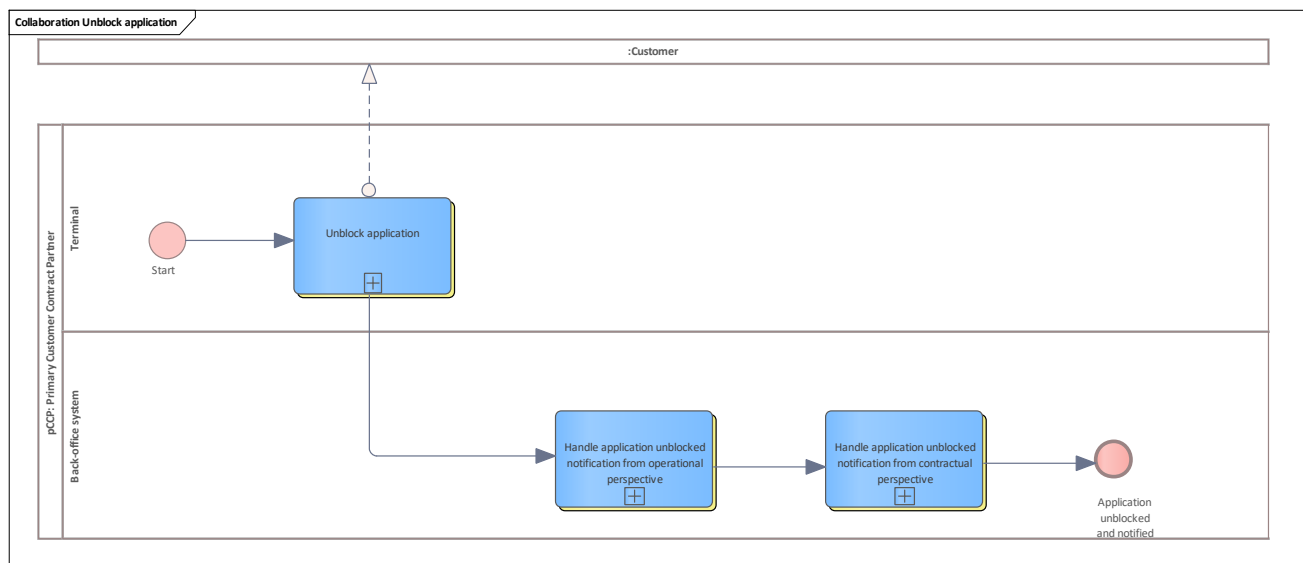


Figure 127: Unblock application

9.2.24.1 pCCP

See [Primary Customer Contract Partner](#)

9.2.24.1.1 Terminal

Lane for terminal

1.1.1.1.1.131 Unblock application

See [Unblock application](#).

9.2.24.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.132 Handle application unblocked notification from contractual perspective

See [Handle application unblocked notification from contractual perspective](#)

1.1.1.1.1.133 Handle application unblocked notification from operational perspective

See [Handle application unblocked notification from operational perspective](#)

9.2.25 Unblock entitlement

This chapter describes the participants and the activities within the basic process "unblock entitlement".

BPMN Collaboration is used.

In this case, the typical template process for [Entitlement owned](#) is the foundation of this basic process.

This basic process can only be performed by the pCCP with the corresponding permissions. The process starts in the terminal where these permissions are verified. The terminal notifies the pCCP back-office system. The back-office system does some operational and contractual monitoring, then the PO is notified about the unblocking process.

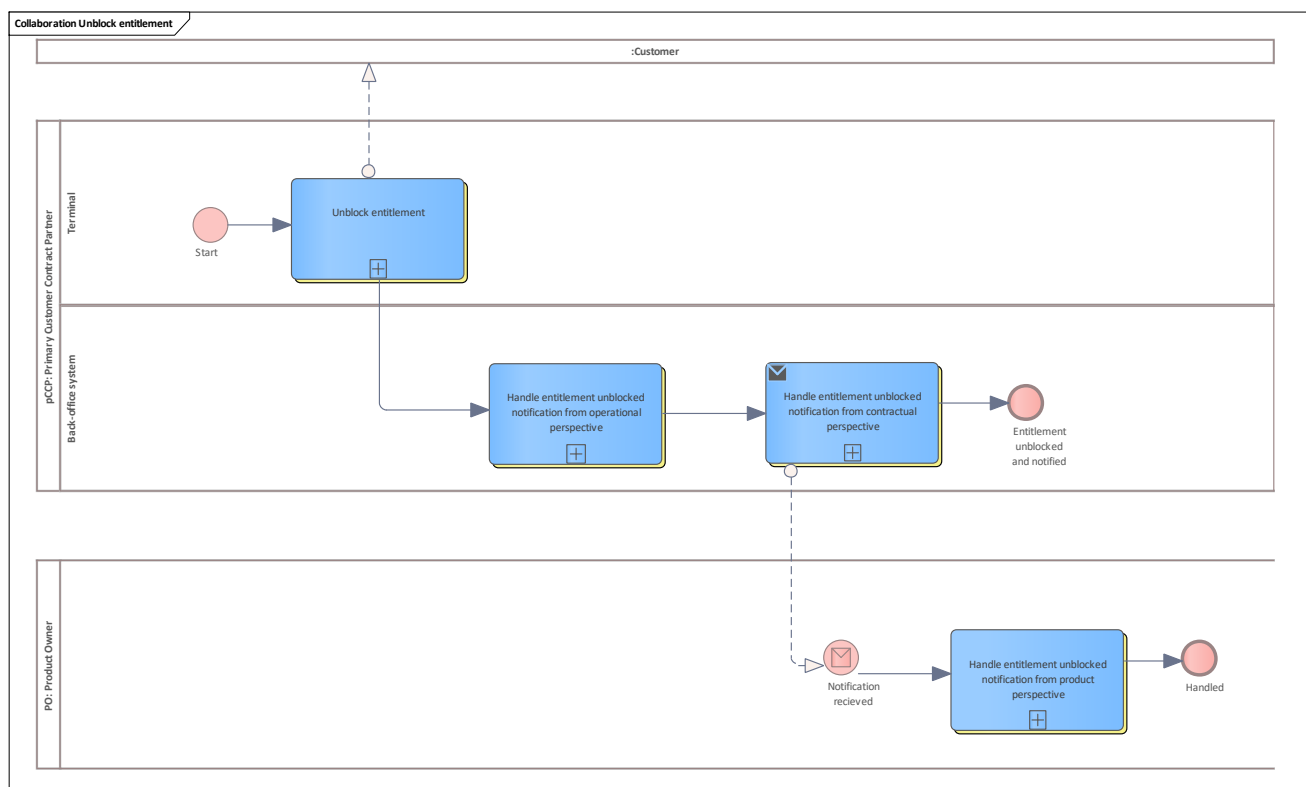


Figure 128: Unblock entitlement



9.2.25.1 pCCP

See [Primary Customer Contract Partner](#)

9.2.25.1.1 Terminal

Lane for terminal

1.1.1.1.1.134 Unblock entitlement

See [Unblock entitlement](#)

9.2.25.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.135 Handle entitlement unblocked notification from contractual perspective

See [Handle entitlement unblocked notification from contractual perspective](#)

1.1.1.1.1.136 Handle entitlement unblocked notification from operational perspective

See [Handle entitlement unblocked notification from operational perspective](#)

9.2.25.2 PO

See [Product Owner](#)

9.2.25.2.1 Handle entitlement unblocked notification from product perspective

See [Handle entitlement unblocked notification from product perspective](#)

9.2.26 Customer service

This chapter describes the participants and the activities within the basic process related to customer service.

BPMN Collaboration is used.

Note: some customer service processes (e.g. "Change entitlement") are not modelled under this package because they are a result of processes from other packages (e.g. take back and sales).

9.2.27 Process new information about customer and discounts

This basic process describes the interaction between the CCP back-office system and the CCP terminal, triggered by the back-office system.

The process starts in the back-office system due to contractual data being changed. The customer data is read from the user medium and merged with the new data to be provided. The changed data about the customer and discounts is written to the user medium. If entitlements are affected, they also have to be changed.

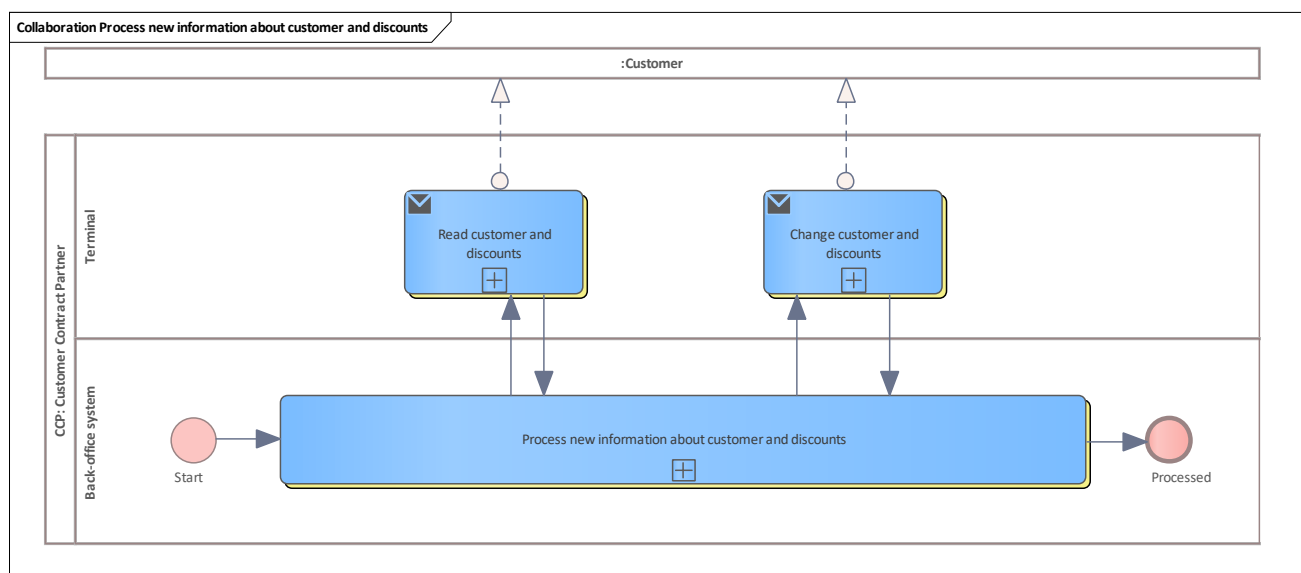


Figure 129: Process new information about customer and discounts

9.2.27.1 CCP

See [Customer Contract Partner](#)

9.2.27.1.1 Back-office system

Lane for back-office system

1.1.1.1.1.137 Process new information about customer and discounts

See [Process new information about customer and discounts](#)



9.2.27.1.2 Terminal

Lane for terminal

1.1.1.1.1.138 Change customer and discounts

See [Change customer and discounts](#)

1.1.1.1.1.139 Read customer and discounts

See [Read customer and discounts](#)

9.2.28 Take back

This chapter deals with all kinds of take-back processes including reimbursing and transferring money back to a payment method/payment means.

9.2.29 Take back application

Basic process for taking back an application (instance). The application (and in most cases the user medium) is taken back from the customer. This can only be done by the pCCP. The application/user medium can either be terminated (scrapping the application/user medium) or blocked for later re-use (not shown here).

In both cases, the underlying entitlements have to be taken back. Depending on the entitlement type, additional reimbursement might be necessary. This is not shown here. See [Take back owned entitlement](#).

Furthermore, the personal data has to be deleted (not shown here, see [De-personalise application](#)). At a minimum, the application is terminated which is done by setting it to the final and irreversible state "*terminated*".

If a hotlist entry exists for this application instance, it has to be removed from the hotlist.

If any orders in an action management system of the PO exist for this application instance, the order cancellation has to be triggered. This is only relevant, if the current pCCP is also an [Ordering Customer Contract Partner](#).

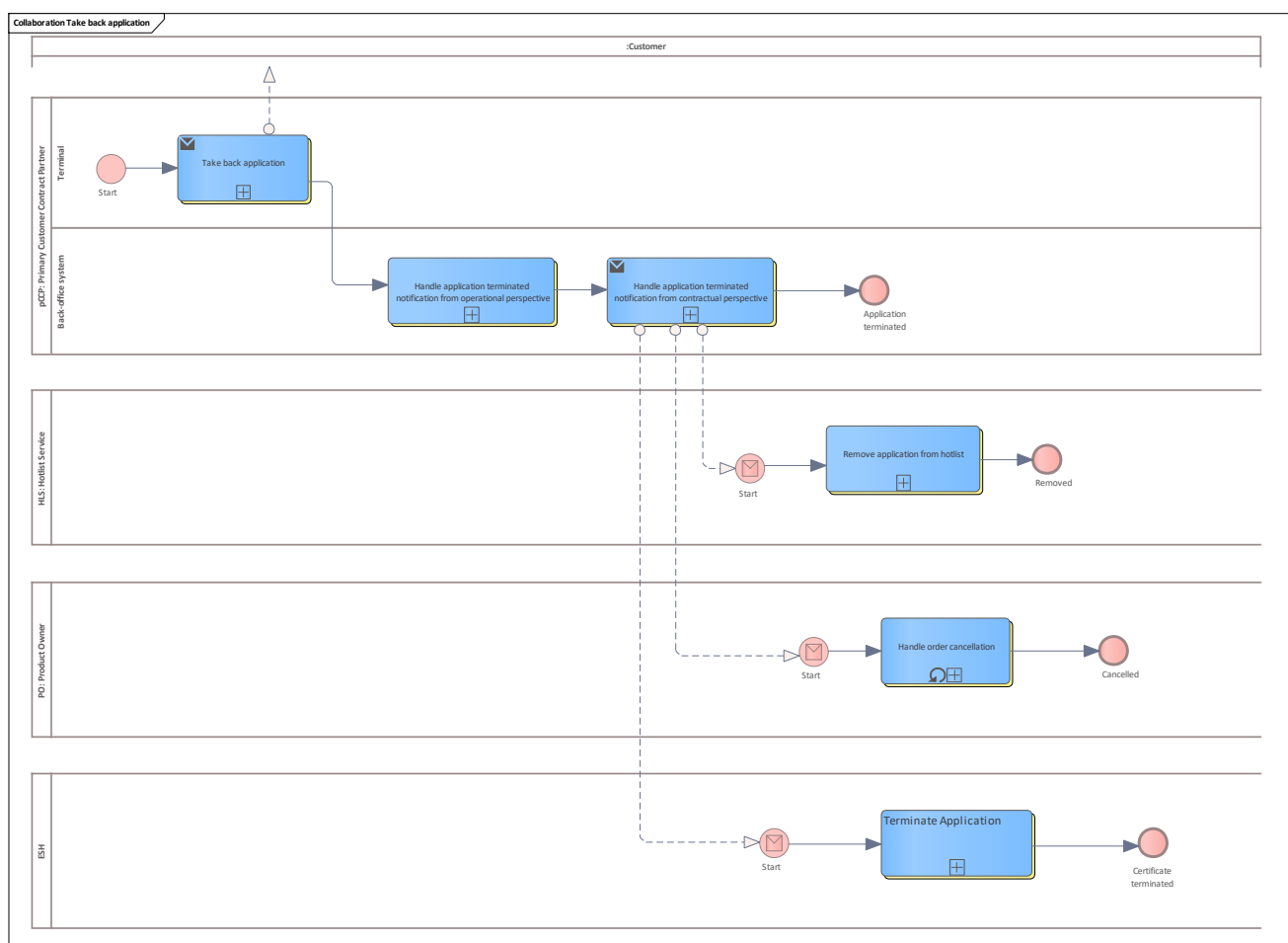


Figure 130: Take back application

9.2.29.1 pCCP

See [Primary Customer Contract Partner](#)

9.2.29.1.1 Terminal

Lane for terminal

1.1.1.1.1.140 Take back application

See [Take back application](#).

9.2.29.1.2 Back-office system

Lane for a back-office system

1.1.1.1.1.141 Handle application terminated notification from contractual perspective

See [Handle application terminated notification from contractual perspective](#).

1.1.1.1.1.142 Handle application terminated notification from operational perspective

See [Handle application terminated notification from operational perspective](#).

9.2.29.2 HLS

See [Hotlist Service](#)

9.2.29.2.1 Remove application from hotlist

If the application is still on the hotlist, it has to be removed, as the hotlist entry is now obsolete.

See [Remove application from hotlist](#).

9.2.29.3 PO

See [Product Owner](#)

9.2.29.3.1 Handle order cancellation

Any active orders relating to this application and issued by the owner of the user medium (pCCP) are to be cancelled.

See [Handle order cancellation](#).

9.2.30 De-personalise application

Basic process to remove personal data from the application on the user medium. Except for the terminal of the pCCP, no further participants are involved.

If any terminated or expired entitlements exist, they will be deleted.

After this process, no personal data is left in the application.

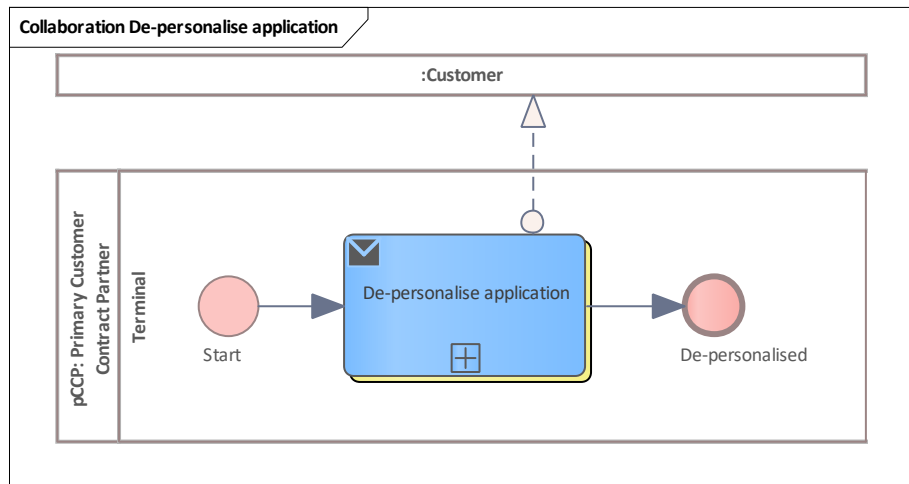


Figure 131: De-personalise application

9.2.30.1 pCCP

See [Primary Customer Contract Partner](#)

9.2.30.1.1 Terminal

Lane for terminal

1.1.1.1.1.143 De-personalise application

See [De-personalise application](#)

9.2.31 Take back non-owned entitlement

Basic process to take back an entitlement. In this case, the CCP is not the issuer of the entitlement, but the sCCP.

In this case, the typical template process for [Entitlement non-owned](#) is the foundation of this basic process.

The process starts in the CCP terminal where the common checks are performed. Depending on the entitlement type, additional reimbursement might be necessary before the entitlement is terminated by changing its state physically. Consider that reimbursing becomes complex, if the current participant is not the owner of the entitlement.

Then the termination is performed (after the termination, the entitlement is deleted, triggered by the terminal) and notified to the CCP back-office system. After some common checks, the notification is forwarded to the PO.

The PO does its monitoring and forwards the notification to the pCCP.

The pCCP does its final monitoring and updates its entitlement management.

For the termination itself, the pCCP checks if the terminated entitlement is still on the hotlist. If so, the pCCP triggers the removal of the hotlist entry.

If reimbursing, depending on the type of reimbursement, clearing may become necessary. This has to be done when the reimbursing/termination messages arrive.

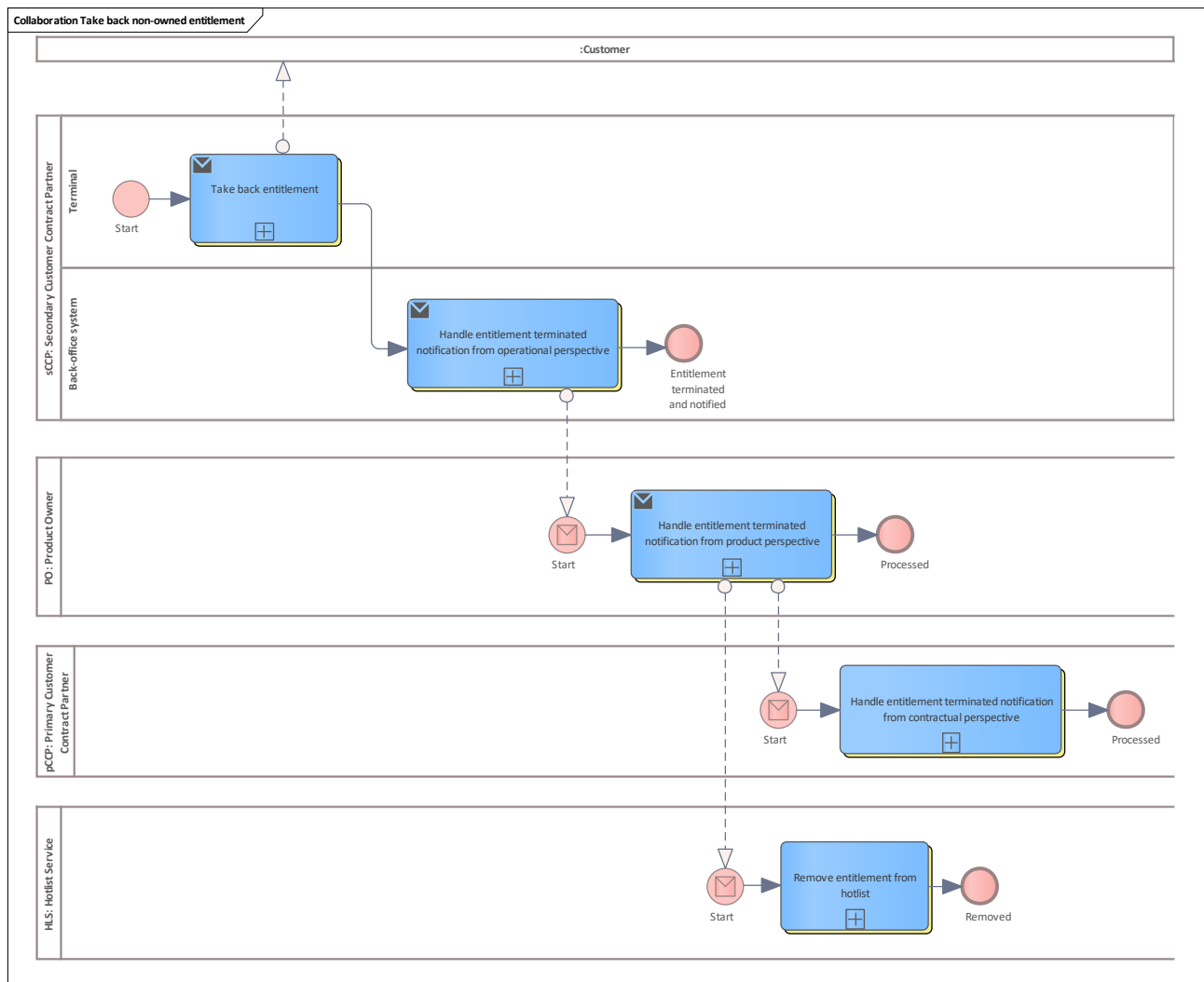


Figure 132: Take back non-owned entitlement

9.2.31.1 sCCP

See [Secondary Customer Contract Partner](#)

9.2.31.1.1 Terminal

Lane for terminal

1.1.1.1.1.144 Take back entitlement

See [Take back entitlement](#).

9.2.31.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.145 Handle entitlement terminated notification from operational perspective

See [Handle entitlement terminated notification from operational perspective](#).

9.2.31.2 PO

See [Product Owner](#)

9.2.31.2.1 Handle entitlement terminated notification from product perspective

See [Handle entitlement terminated notification from product perspective](#).

9.2.31.3 pCCP

See [Primary Customer Contract Partner](#)

9.2.31.3.1 Handle entitlement terminated notification from contractual perspective

See [Handle entitlement terminated notification from contractual perspective](#).

9.2.31.4 HLS

See [Hotlist Service](#)

9.2.31.4.1 Remove entitlement from hotlist

See [Remove entitlement from hotlist](#)

9.2.32 Take back owned entitlement

Basic process to take back an entitlement. In this case, the CCP is the issuer of the entitlement, the pCCP.

In this case, the typical template process for [Entitlement owned](#) is the foundation of this basic process.

The process starts in the pCCP terminal where the common checks are performed. Depending on the entitlement type, additional reimbursement might be necessary before the entitlement is terminated by changing its state physically.

Then the termination is performed (after the termination, the entitlement is deleted, triggered by the terminal) and notified to the pCCP back-office system.

After operational and contractual monitoring checks, the termination is registered and the notification is forwarded to the PO.

The PO does its monitoring.

For the termination itself, the pCCP checks if the terminated entitlement is still on the hotlist. If so, the pCCP triggers the removal of the hotlist entry.

If reimbursing, depending on the type of reimbursement, booking may become necessary. This has to be done when the reimbursing/termination messages arrive.

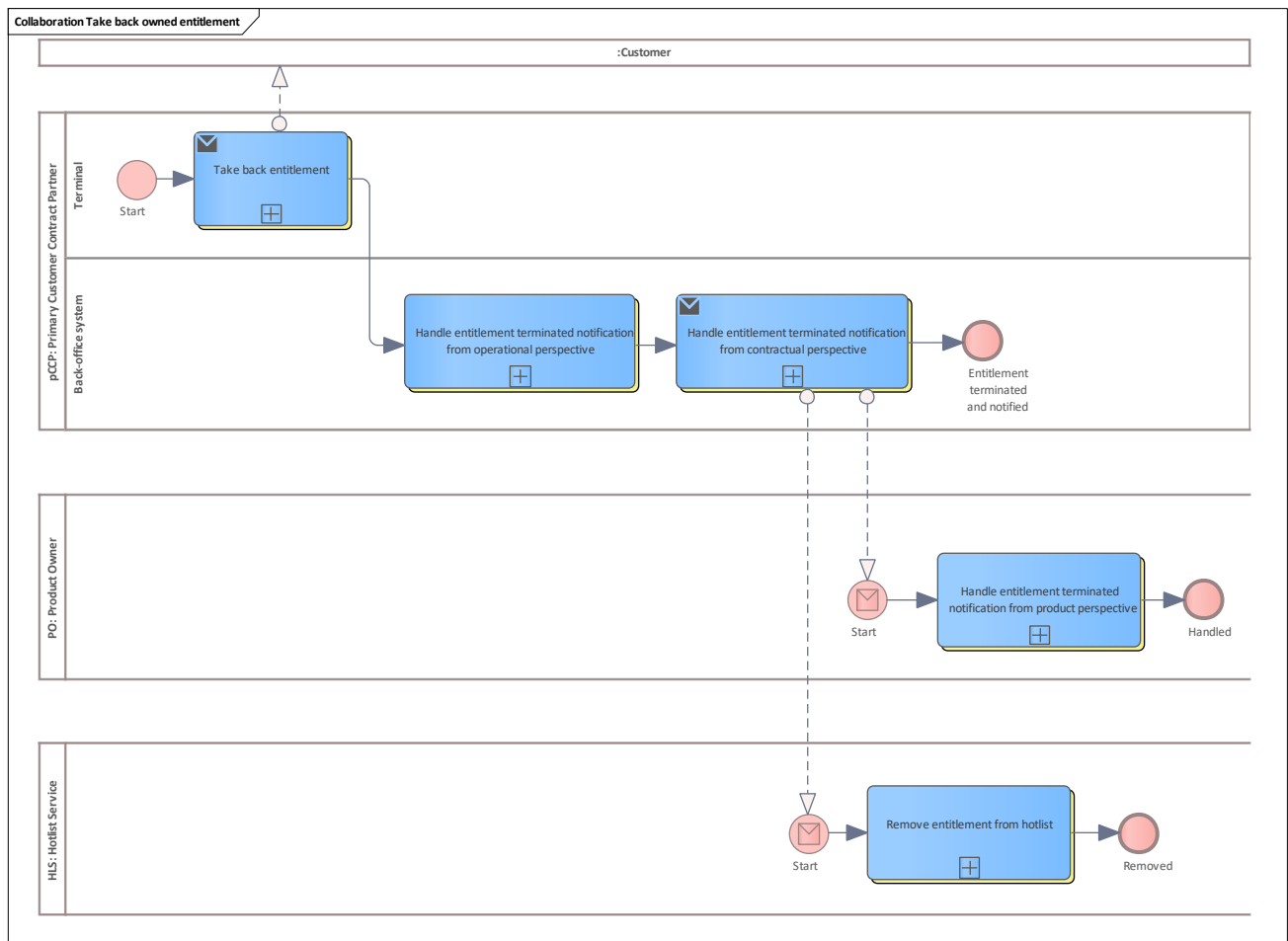


Figure 133: Take back owned entitlement

9.2.32.1 pCCP

See [Primary Customer Contract Partner](#)

9.2.32.1.1 Terminal

Lane for terminal

1.1.1.1.1.146 Take back entitlement

See [Take back entitlement](#).

9.2.32.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.147 Handle entitlement terminated notification from contractual perspective

See [Handle entitlement terminated notification from contractual perspective](#).

1.1.1.1.1.148 Handle entitlement terminated notification from operational perspective

See [Handle entitlement terminated notification from operational perspective](#).

9.2.32.2 PO

See [Product Owner](#)

9.2.32.2.1 Handle entitlement terminated notification from product perspective

See [Handle entitlement terminated notification from product perspective](#).

9.2.32.3 HLS

See [Hotlist Service](#)

9.2.32.3.1 Remove entitlement from hotlist

See [Remove entitlement from hotlist](#)

9.2.33 Take back owned static entitlement

Basic process to take back a static entitlement. In this case, the CCP is the issuer of the entitlement, the pCCP.

In this case, the typical template process for [Entitlement owned](#) is the foundation of this basic process.

The process starts in the pCCP terminal where the common checks are performed. Depending on the electronic ticket contained in the static entitlement, additional reimbursement might be necessary before the entitlement is terminated. If the static entitlement was purchased by an (((etiCORE payment method, the amount will be credited to this method.

Then the termination notification is sent to the pCCP back-office system.

After operational and contractual monitoring checks, the termination is registered and the notification is forwarded to the PO.

The PO does its monitoring.

The pCCP triggers the addition of the static entitlement to the hotlist to prevent further usage of the contained electronic ticket.

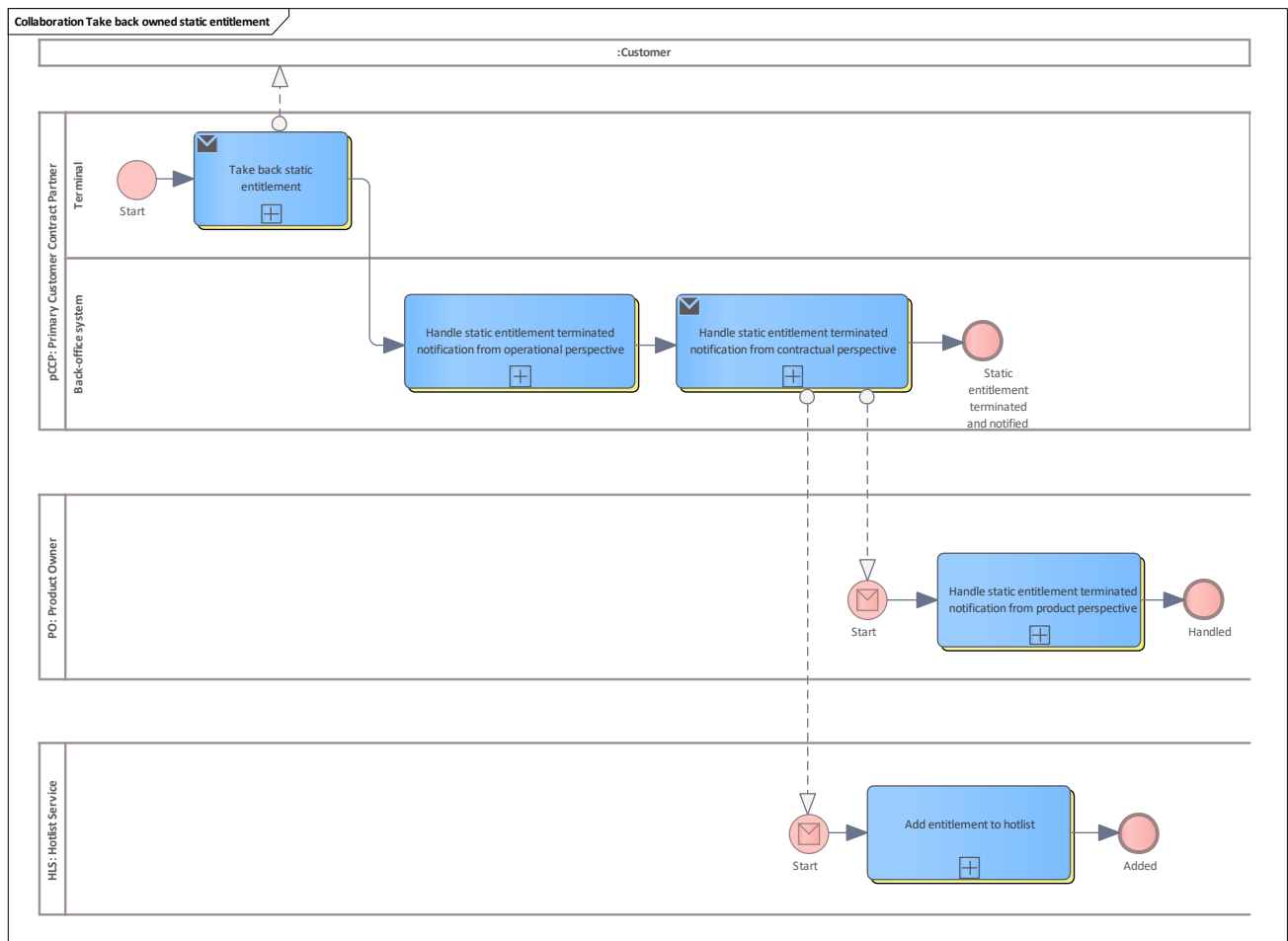


Figure 134: Take back owned static entitlement

9.2.33.1 pCCP

See [Primary Customer Contract Partner](#)

9.2.33.1.1 Terminal

Lane for terminal

1.1.1.1.1.149 Take back static entitlement

See [Take back static entitlement](#)

9.2.33.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.150 Handle static entitlement terminated notification from contractual perspective

See [Handle static entitlement terminated notification from contractual perspective](#)

1.1.1.1.1.151 Handle static entitlement terminated notification from operational perspective

See [Handle static entitlement terminated notification from operational perspective](#)

9.2.33.2 PO

See [Product Owner](#)

9.2.33.2.1 Handle static entitlement terminated notification from product perspective

See [Handle static entitlement terminated notification from product perspective](#)

9.2.33.3 HLS

See [Hotlist Service](#)

9.2.33.3.1 Add entitlement to hotlist

See [Add entitlement to hotlist](#)

9.2.34 Credit non-owned stored-value payment method

This basic process describes the crediting of an amount to a stored-value payment method. The CCP which performs the crediting is not the owner of the payment method, but the sCCP. In this case, the typical template process for [Entitlement non-owned](#) is the foundation of this basic process.

The process starts in a CCP terminal where crediting is possible. The amount is credited to the determined store-value payment method. The crediting is announced to the back-office system of the sCCP. The system does operational monitoring checks and registers the crediting attestation for potential later clearing purposes.

Then the crediting notification is forwarded to the PO that does its monitoring and registration. The PO forwards the notification to the pCCP.

The pCCP does its contractual monitoring, registers the crediting notification and triggers booking and clearing actions if necessary.

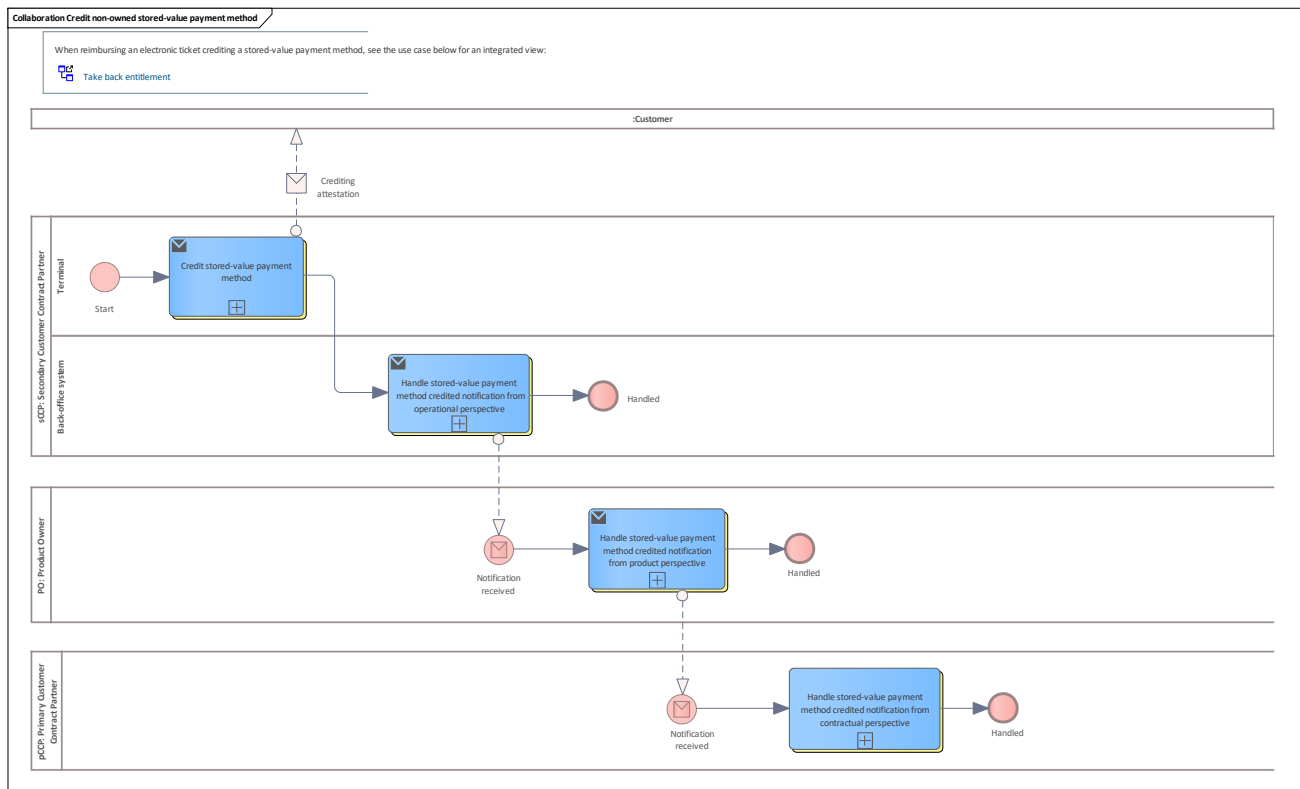


Figure 135: Credit non-owned stored-value payment method

9.2.34.1 sCCP

See [Secondary Customer Contract Partner](#)

9.2.34.1.1 Terminal

Lane for terminal

1.1.1.1.1.152 Credit stored-value payment method

See [Credit stored-value payment method](#).

9.2.34.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.153 Handle stored-value payment method credited notification from operational perspective

See [Handle stored-value payment method credited notification from operational perspective](#).

9.2.34.2 PO

See [Product Owner](#)

9.2.34.2.1 Handle stored-value payment method credited notification from product perspective

See [Handle stored-value payment method credited notification from product perspective](#).

9.2.34.3 pCCP

See [Primary Customer Contract Partner](#)

9.2.34.3.1 Handle stored-value payment method credited notification from contractual perspective

See [Handle stored-value payment method credited notification from contractual perspective](#).

9.2.35 Credit owned stored-value payment method

This basic process describes the crediting of an amount to a stored-value payment method. The CCP which performs the crediting is the owner of the payment method, the pCCP.

In this case, the typical template process for [Entitlement owned](#) is the foundation of this basic process.

The process starts in a CCP terminal where crediting is possible. The amount is credited to the determined store-value payment method. The crediting is announced to the back-office system of the pCCP. The system does operational monitoring checks and registers the crediting attestation. The pCCP does its contractual monitoring notification and triggers booking and clearing actions if necessary.

Then the crediting notification is forwarded to the PO that does its monitoring and registration.

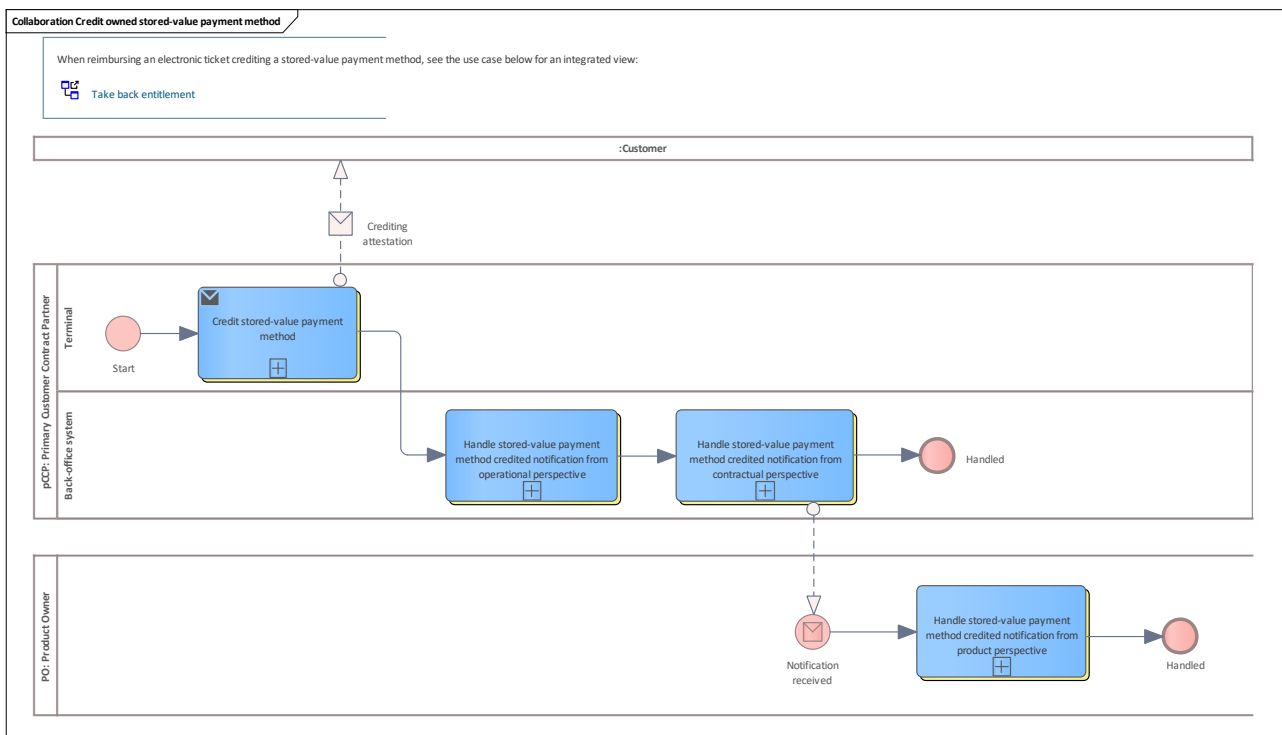


Figure 136: Credit owned stored-value payment method

9.2.35.1 pCCP

See [Primary Customer Contract Partner](#)

9.2.35.1.1 Terminal

Lane for terminal

1.1.1.1.1.154 Credit stored-value payment method

See [Credit stored-value payment method](#).

9.2.35.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.155 Handle stored-value payment method credited notification from contractual perspective

See [Handle stored-value payment method credited notification from contractual perspective](#).

1.1.1.1.1.156 Handle stored-value payment method credited notification from operational perspective

See [Handle stored-value payment method credited notification from operational perspective](#).

9.2.35.2 PO

See [Product Owner](#)

9.2.35.2.1 Handle stored-value payment method credited notification from product perspective

See [Handle stored-value payment method credited notification from product perspective](#).

9.2.36 Credit non-owned account-based payment method

This basic process describes the crediting of an amount to an account-based payment method. The CCP which performs the crediting is not the owner of the payment method, but the sCCP. In this case, the typical template process for [Entitlement non-owned](#) is the foundation of this basic process.

The process starts in a CCP terminal where crediting is possible. The amount is credited to the determined account-based payment method. The crediting is announced to the back-office system of the sCCP. The system does operational monitoring checks and registers the crediting attestation for potential later clearing purposes.

Then the crediting notification is forwarded to the PO that does its monitoring and registration. The PO forwards the notification to the pCCP.

The pCCP does its contractual monitoring, registers the crediting notification and triggers the booking action and further clearing actions, if necessary.

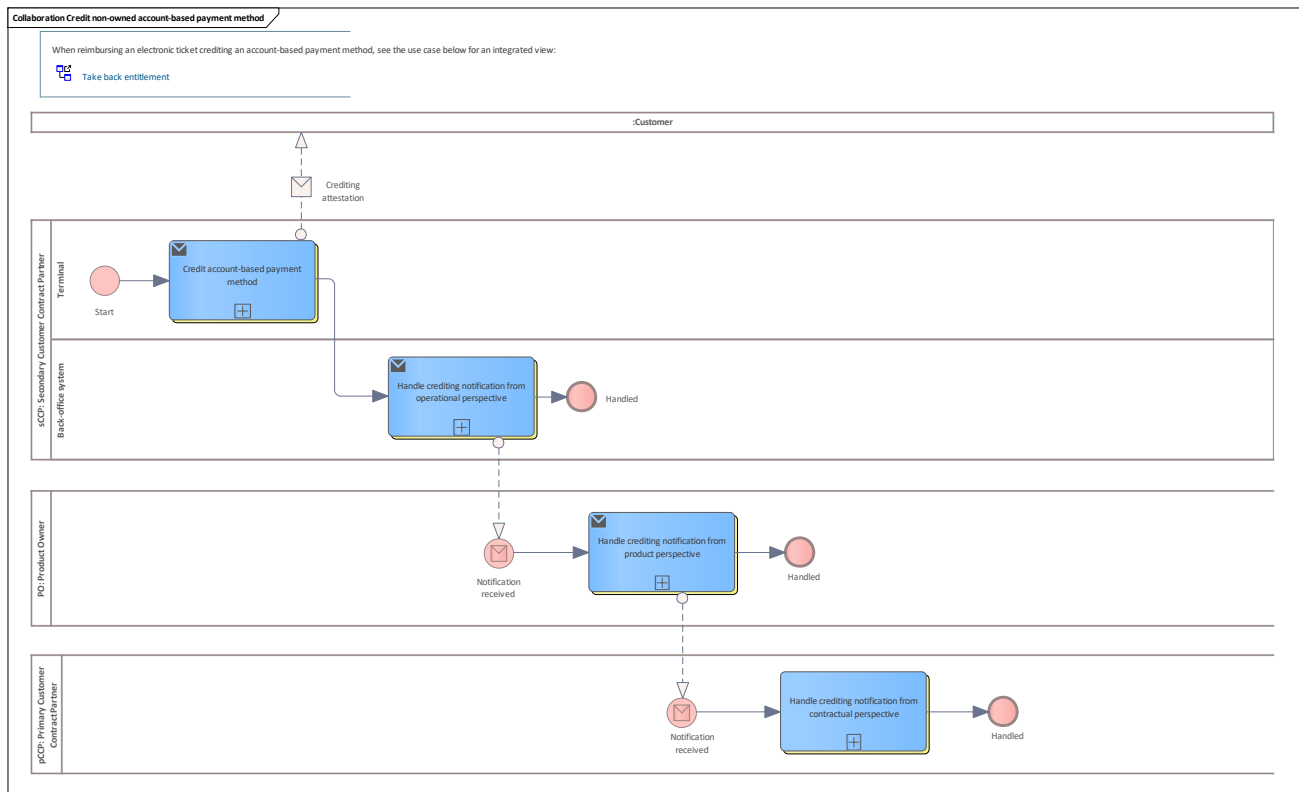


Figure 137: Credit non-owned account-based payment method

9.2.36.1 sCCP

See [Secondary Customer Contract Partner](#)

9.2.36.1.1 Terminal

Lane for terminal

1.1.1.1.1.157 Credit account-based payment method

See [Credit account-based payment method](#).

9.2.36.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.158 Handle crediting notification from operational perspective

See [Handle account-based payment method credited notification from operational perspective](#).

9.2.36.2 PO

See [Product Owner](#)

9.2.36.2.1 Handle crediting notification from product perspective

See [Handle account-based payment method credited notification from product perspective](#).

9.2.36.3 pCCP

See [Primary Customer Contract Partner](#)

9.2.36.3.1 Handle crediting notification from contractual perspective

See [Handle account-based payment method credited notification from contractual perspective](#).

9.2.37 Credit owned account-based payment method

This basic process describes the crediting of an amount to an account-based payment method. The CCP which performs the crediting is the owner of the payment method, the pCCP. In this case, the typical template process for [Entitlement owned](#) is the foundation of this basic process.

The process starts in a CCP terminal where crediting is possible. The amount is credited to the determined account-based payment method. The crediting is announced to the back-office system of the pCCP. The system does operational monitoring checks and registers the crediting attestation. The pCCP does its contractual monitoring notification and triggers the booking action and further clearing actions, if necessary.

Then the crediting notification is forwarded to the PO that does its monitoring and registration.

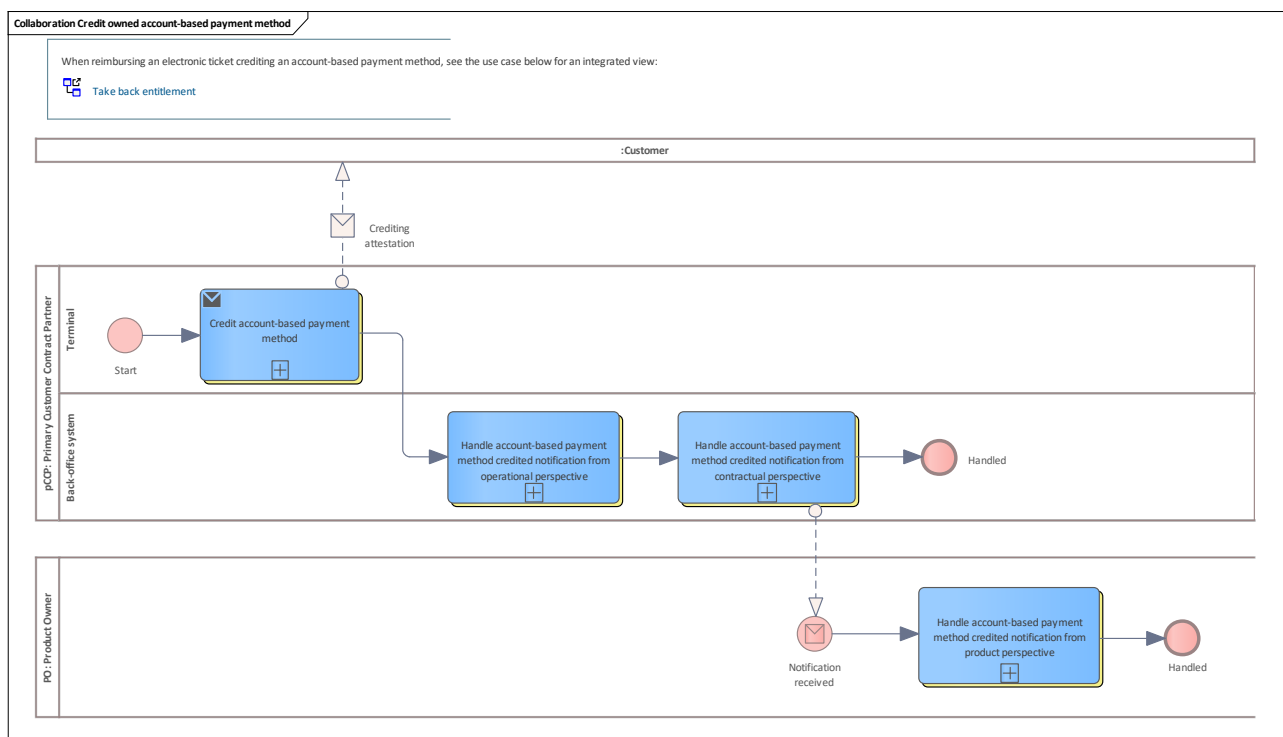


Figure 138: Credit owned account-based payment method

9.2.37.1 pCCP

See [Primary Customer Contract Partner](#)

9.2.37.1.1 Terminal

Lane for terminal

1.1.1.1.1.159 Credit account-based payment method

See [Credit account-based payment method](#)

9.2.37.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.160 Handle account-based payment method credited notification from contractual perspective

See [Handle account-based payment method credited notification from contractual perspective](#).

1.1.1.1.1.161 Handle account-based payment method credited notification from operational perspective

See [Handle account-based payment method credited notification from operational perspective](#).

9.2.37.2 PO

See [Product Owner](#)

9.2.37.2.1 Handle account-based payment method credited notification from product perspective

See [Handle account-based payment method credited notification from product perspective](#).

9.2.38 Reimburse owned stored-value payment method

This basic process describes a reimbursing of a stored-value payment method.

The CCP is the owner of the stored-value payment method, the pCCP.

In this case, the typical template process for [Entitlement owned](#) is the foundation of this basic process.

This basic process may occur if the customer wants to have his credit or a part of it on the user medium (available in the stored-value payment method) paid out.

The process starts in a suitable pCCP terminal. The desired amount is determined and deducted. The final payment to the customer is always done in legal tender.

The terminal notifies the pCCP back-office system about the reimbursement. The pCCP system does the operational monitoring and registers the reimbursement. Subsequently, the contractual monitoring checks are done.

Finally, the notification is forwarded to the PO system which does its monitoring and registers the reimbursement action.

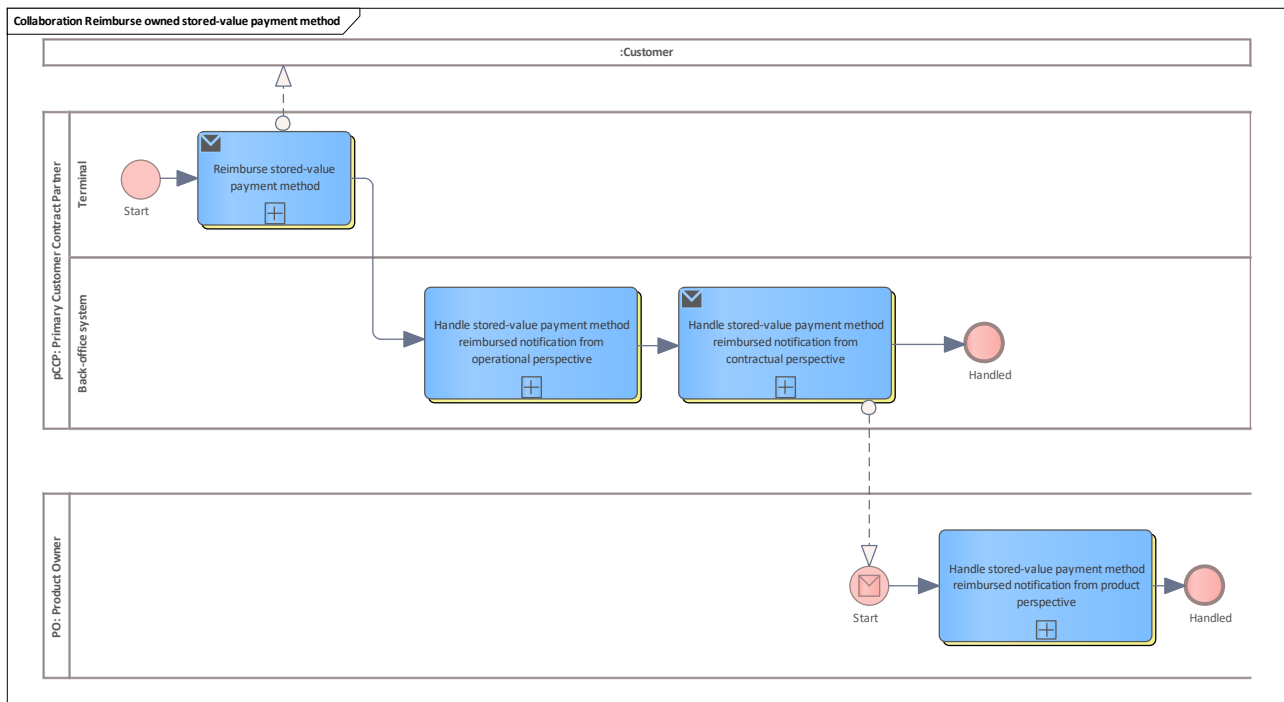


Figure 139: Reimburse owned stored-value payment method

9.2.38.1 pCCP

See [Primary Customer Contract Partner](#)

9.2.38.1.1 Terminal

Lane for terminal

1.1.1.1.1.162 Reimburse stored-value payment method

See [Reimburse stored-value payment method](#).

9.2.38.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.163 Handle stored-value payment method reimbursed notification from contractual perspective

See [Handle stored-value payment method reimbursed notification from contractual perspective](#).

1.1.1.1.1.164 Handle stored-value payment method reimbursed notification from operational perspective

See [Handle stored-value payment method reimbursed notification from operational perspective](#).

9.2.38.2 PO

See [Product Owner](#)

9.2.38.2.1 Handle stored-value payment method reimbursed notification from product perspective

See [Handle stored-value payment method reimbursed notification from product perspective](#).

9.2.39 Reimburse non-owned stored-value payment method

This basic process describes a reimbursing of a stored-value payment method.

The CCP is not the owner of the stored-value payment method, but the sCCP.

In this case, the typical template process for [Entitlement non-owned](#) is the foundation of this basic process.

This basic process may occur if the customer wants to have his credit or a part of it on the user medium (available in the stored-value payment method) paid out on the terminal of another CCP.

The process starts in a suitable sCCP terminal. The desired amount is determined and deducted. The final payment to the customer is always done in legal tender.

The terminal notifies the sCCP back-office system about the reimbursement. The sCCP system does the operational monitoring and registers the reimbursement for later clearing purposes. Subsequently, the notification is forwarded to the PO system which does its monitoring and registers the reimbursement action. The PO system forwards the notification to the pCCP system.

Finally, the pCCP system does its contractual monitoring checks and triggers the clearing process between itself and the involved sCCP.

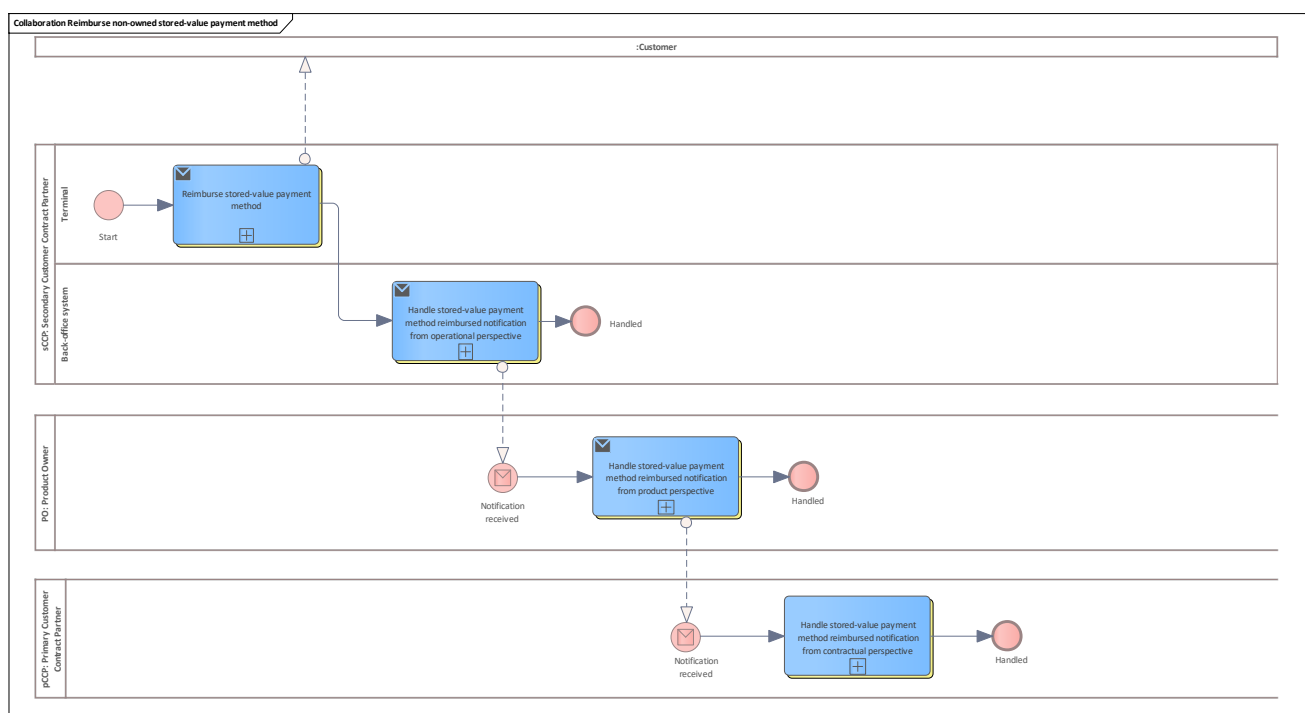


Figure 140: Reimburse non-owned stored-value payment method



9.2.39.1 sCCP

See [Secondary Customer Contract Partner](#)

9.2.39.1.1 Terminal

Lane for terminal

1.1.1.1.1.165 Reimburse stored-value payment method

See [Reimburse stored-value payment method](#).

9.2.39.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.166 Handle stored-value payment method reimbursed notification from operational perspective

See [Handle stored-value payment method reimbursed notification from operational perspective](#).

9.2.39.2 PO

See [Product Owner](#)

9.2.39.2.1 Handle stored-value payment method reimbursed notification from product perspective

See [Handle stored-value payment method reimbursed notification from product perspective](#).

9.2.39.3 pCCP

See [Primary Customer Contract Partner](#)

9.2.39.3.1 Handle stored-value payment method reimbursed notification from contractual perspective

See [Handle stored-value payment method reimbursed notification from contractual perspective](#).

9.2.40 Handle defective or lost user medium

This chapter describes the participants and the activities within the basic process related to the defective user medium process.

BPMN Collaboration is used.

9.2.41 Get entitlements of a lost user medium

Basic process between a terminal operator (SO or CCP) back-office system and the PO.

The CCP with the dedicated rights (which does not have to be the primary CCP) asks the PO for valid entitlements. Note that if the CCP is not the pCCP, the CCP does not have any information if the application of the lost user medium is currently blocked.

The product owner filters the entitlements for the passed application instance ID and considers potential hotlisted or blocked entitlements. Note that the PO does not have information about hotlisted or blocked applications.

The PO returns a list of valid entitlements (or an empty list, if no valid entitlements exist) to the requesting CCP that can use this information to reissue a new user medium with these entitlements.

The second purpose of this basic process is to find out if one or more valid entitlements are or were available for a certain timestamp. This can become important if a user medium was not readable at the time of the inspection for any reason.

A further process after a penalty fare notice can verify whether the penalty fare notice is justified. For this purpose, the list of valid entitlements for a certain timestamp is checked for the inspection parameters.

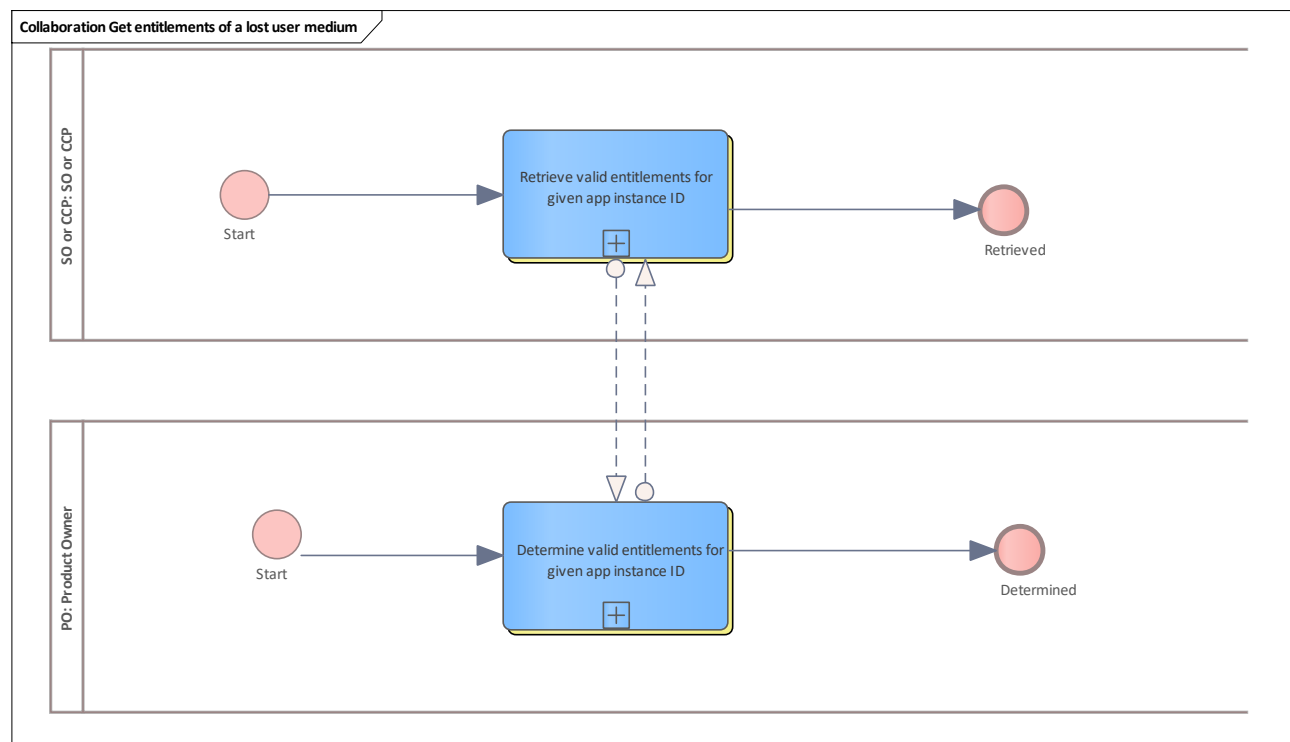


Figure 141: Get entitlements of a lost user medium

9.2.41.1 SO or CCP

See [SO or CCP](#)

9.2.41.1.1 Retrieve valid entitlements for given app instance ID

See [Retrieve valid entitlements for given app instance ID](#).

9.2.41.2 PO

See [Product Owner](#)

9.2.41.2.1 Determine valid entitlements for given app instance ID

See [Determine valid entitlements for given app instance ID](#).

9.2.42 Handle defective user medium

Basic process between a terminal and the back-office system of a terminal operator that can be a CCP or SO.

The terminal is the first instance which contacts the user medium and can determine a defective (non-readable) user medium.

The terminal logs the time and the location. The medium ID has to be scanned or read off (usually printed on the chip card in the form of a barcode). The notification with this information is transferred to the back-office system.

The back-office system registers this notification and the use case [Handle defective user medium with application](#) triggers the next basic processes

- [Look up application instance ID](#) and
- [Hotlist non-owned application](#) (if the terminal operator is not the owner of the application instance)
- or [Hotlist owned application](#) (if it is the owner)

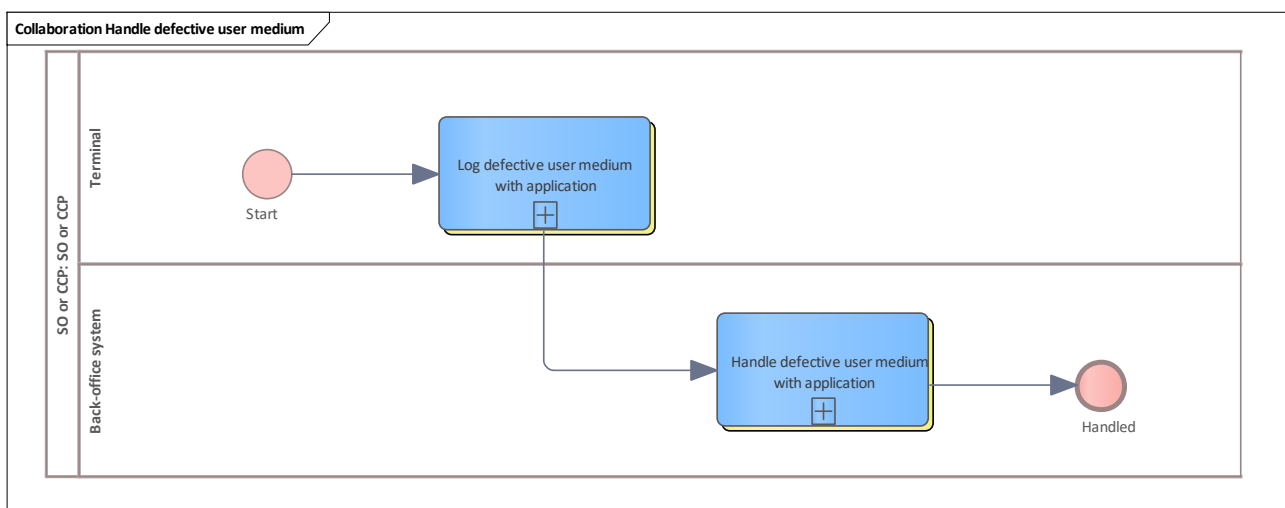


Figure 142: Handle defective user medium

9.2.42.1 SO or CCP

See [SO or CCP](#).

9.2.42.1.1 Terminal

Lane for terminal

1.1.1.1.1.167 Log defective user medium with application

See [Log defective user medium with application](#).

9.2.42.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.168 Handle defective user medium with application

See [Handle defective user medium with application](#).

9.2.43 Look up application instance ID

Basic process between the terminal operator (SO or CCP) and the scheme manager. The terminal operator requests the application instance ID for a certain medium ID. The basic process will be rather rare since, in most cases, the medium ID ((usually printed on the chip card)) will be the application instance ID.

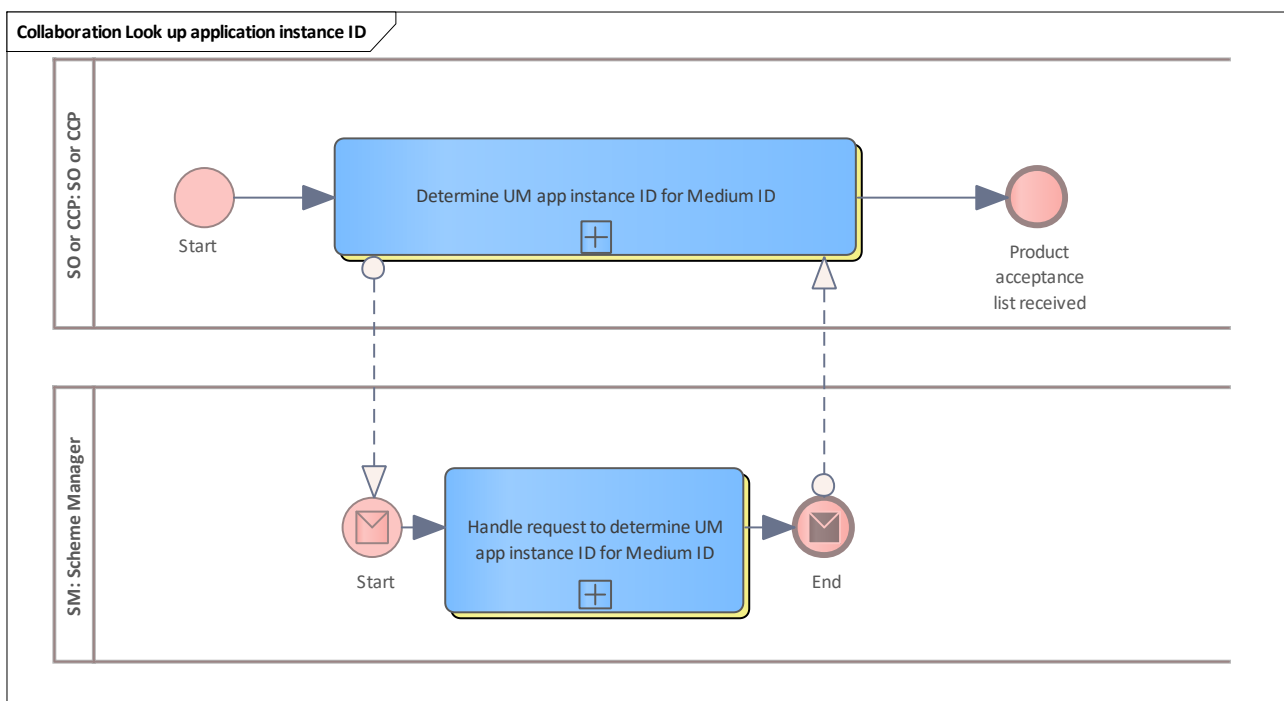


Figure 143: Look up application instance ID

9.2.43.1 SO or CCP

See [SO or CCP](#)

9.2.43.1.1 Determine UM app instance ID for Medium ID

See [Determine UM app instance ID for Medium ID](#).

9.2.43.2 SM

See [Scheme Manager](#)

9.2.43.2.1 Handle request to determine UM app instance ID for Medium ID

See [Handle request to determine UM app instance ID for Medium ID](#).

9.2.44 Ordered action management

This chapter describes the basic processes within the scope of action management. Actions can be ordered remotely for a user medium respective application instance ID and executed later if a suitable terminal contacts the involved user medium. Each action is related to an order that is unique for an [Ordering Customer Contract Partner](#).

9.2.45 Distribute action list retrieval configuration

Basic process that takes place between the [Product Owner](#) (its management module for ordered actions) and the [Executing Customer Contract Partner](#) that must regularly fetch the action lists. To perform this in an efficiently way, the product owner offers a configuration item to the executing customer contract partner. This configuration determines the provided time for the retrieval request and the interval until a new action list is offered.

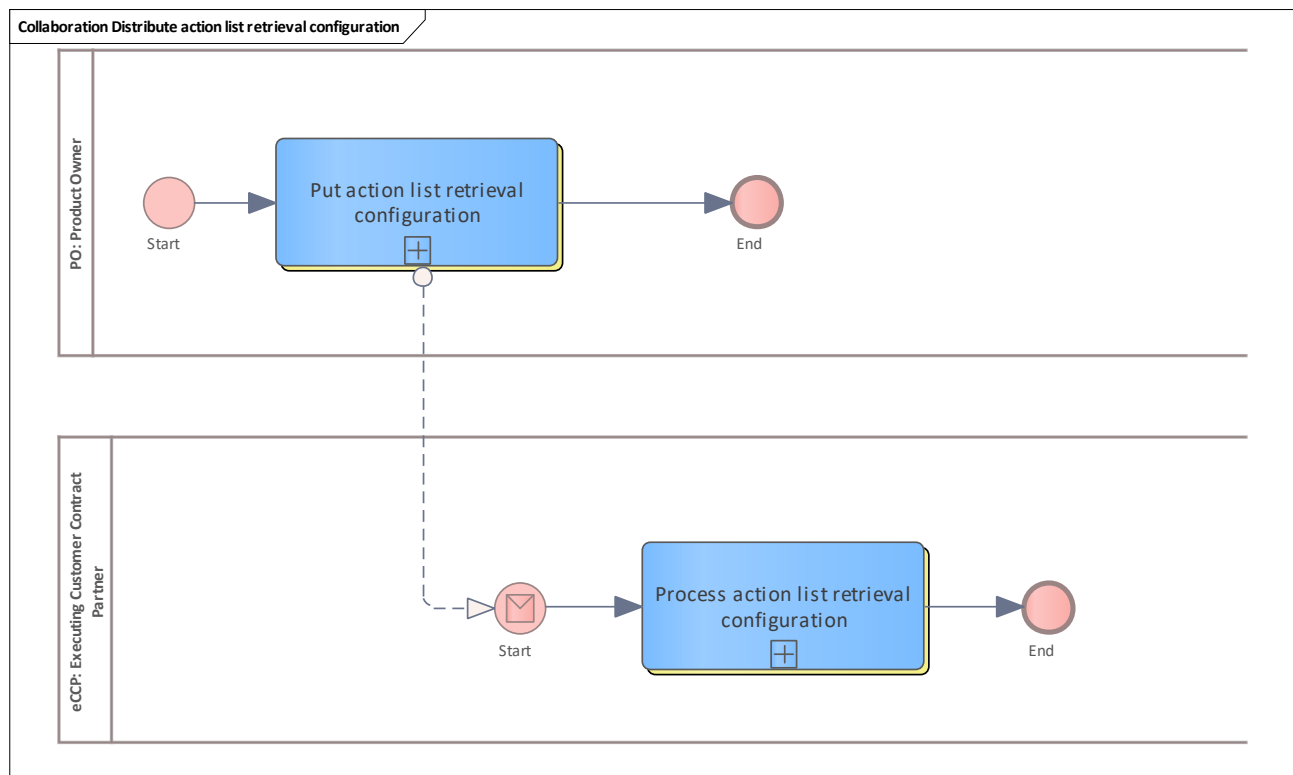


Figure 144: Distribute action list retrieval configuration

9.2.46 Order entitlement issuance

Basic process that takes place between the [Ordering Customer Contract Partner](#) and the [Product Owner](#) (its action management system module).

The ordering customer contract partner orders the issuance of a new entitlement with a unique order ID and the related application instance ID. The product owner registers this new order and puts it into its action inventory.

Note: the entitlement ID of the new entitlement does not yet exist. It will be created on the fly by the SAM in the acting terminal when the entitlement is issued locally and linked to the order ID to prevent a further attempt to issue the entitlement by another terminal.

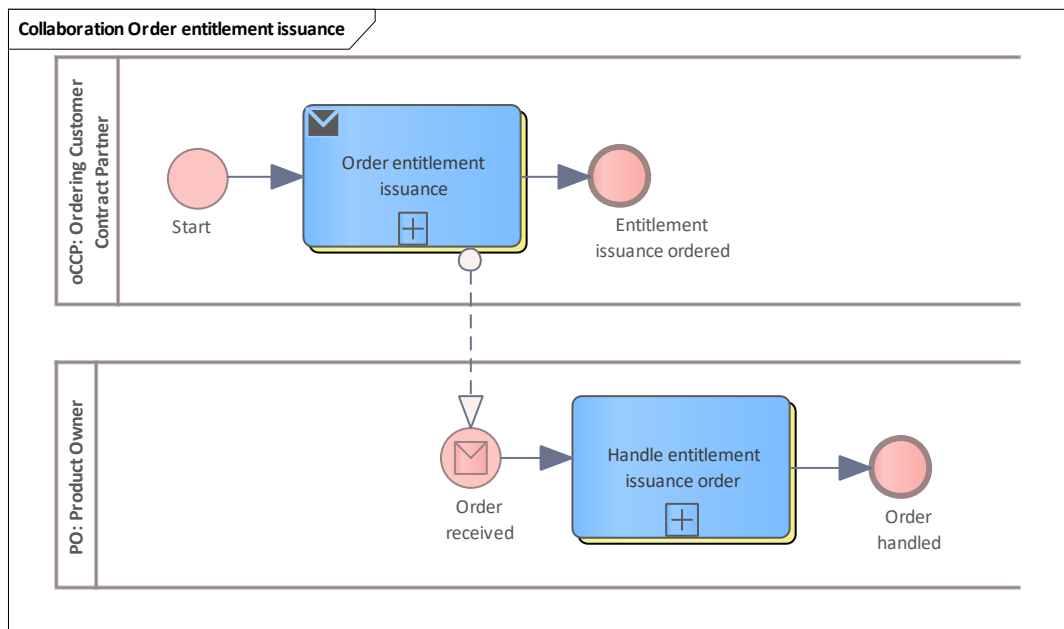


Figure 145: Order entitlement issuance

9.2.46.1 oCCP

See [Ordering Customer Contract Partner](#).

9.2.46.1.1 Order entitlement issuance

See [Order entitlement issuance](#).

9.2.46.2 PO

See [Product Owner](#)

9.2.46.2.1 Handle entitlement issuance order

See [Handle entitlement issuance order](#).

9.2.47 Order entitlement termination

Basic process that takes place between the [Ordering Customer Contract Partner](#) and the [Product Owner](#) (its action management system module).

The ordering customer contract partner orders the termination of an existing entitlement with a unique order ID together with the entitlement ID and the application instance ID. The product owner registers this new order and puts it into its action inventory.

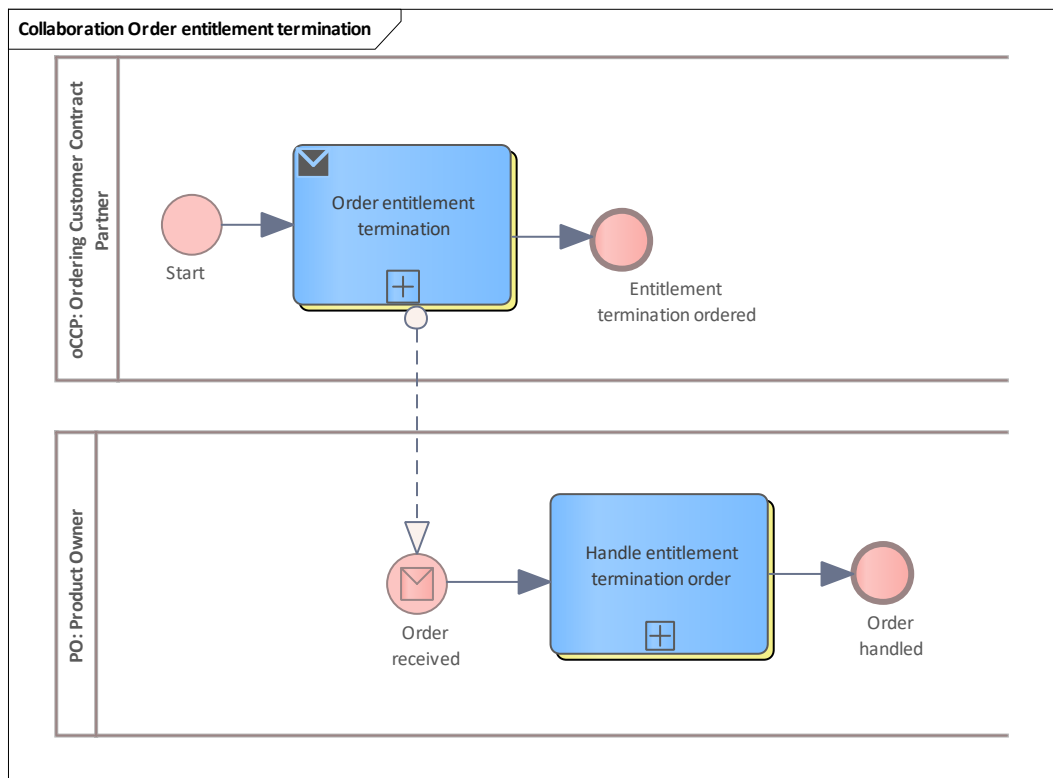


Figure 146: Order entitlement termination

9.2.47.1 oCCP

See [Ordering Customer Contract Partner](#)

9.2.47.1.1 Order entitlement termination

See [Order entitlement termination](#).

9.2.47.2 PO

See [Product Owner](#)

9.2.47.2.1 Handle entitlement termination order

See [Handle entitlement termination order](#).

9.2.48 Order entitlement blocking

Basic process that takes place between the [Ordering Customer Contract Partner](#) and the [Product Owner](#) (its action management system module).

The ordering customer contract partner orders the blocking of a previous issuance order. This process is intended for the exceptional situation where the order for an entitlement issuance has been placed and is now in circulation, but this is to be revised by blocking the entitlement using the order ID.

As no entitlement ID is known without the entitlement issuance attestation, the blocking must therefore take place based on the order ID. Accordingly, the regular hotlist mechanism cannot be used.

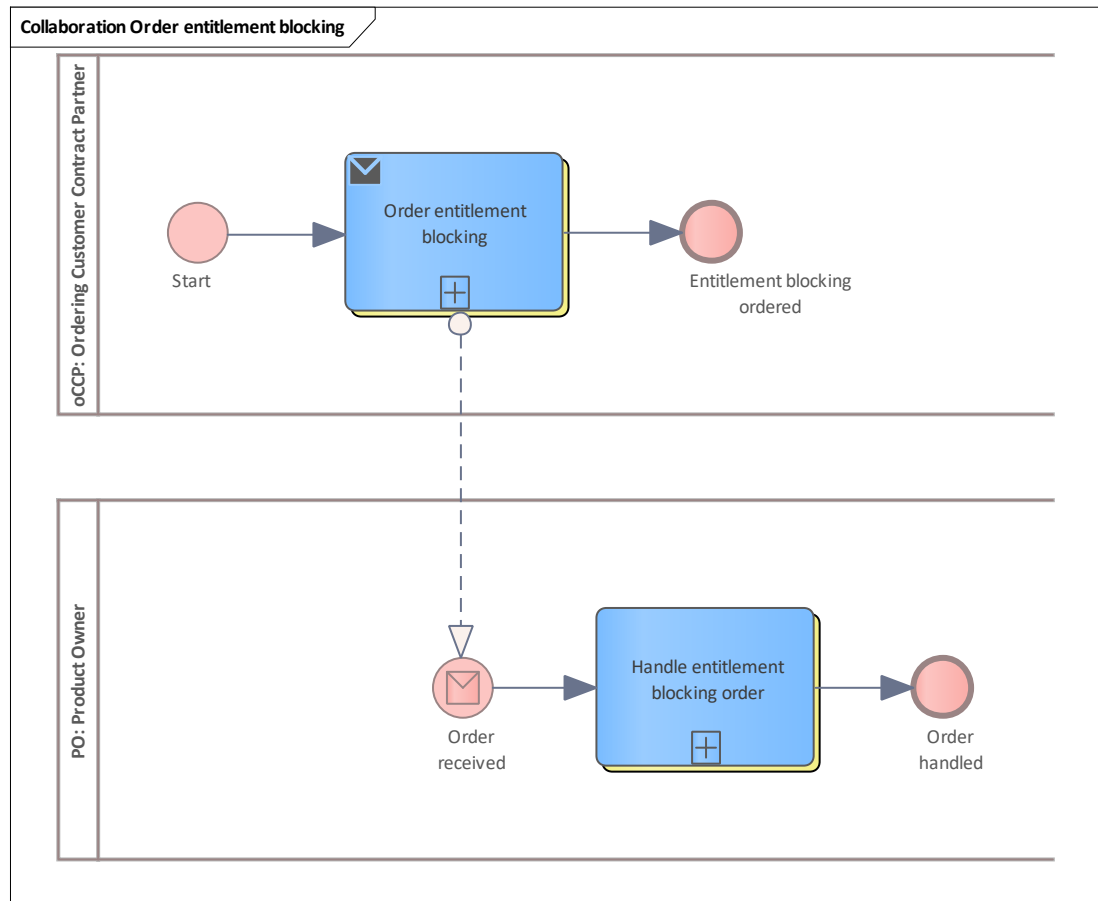


Figure 147: Order entitlement blocking

9.2.48.1 oCCP

See [Ordering Customer Contract Partner](#)

9.2.48.1.1 Order entitlement blocking

See [Order entitlement blocking](#)

9.2.48.2 PO

See [Product Owner](#)

9.2.48.2.1 Handle entitlement blocking order

See [Handle entitlement blocking order](#)

9.2.49 Order entitlement unblocking

Basic process that takes place between the [Ordering Customer Contract Partner](#) and the [Product Owner](#) (its action management system module).

The ordering customer contract partner orders the unblocking of an existing entitlement with a unique order ID together with the entitlement ID and the application instance ID. The product owner registers this new order and puts it into its action inventory.

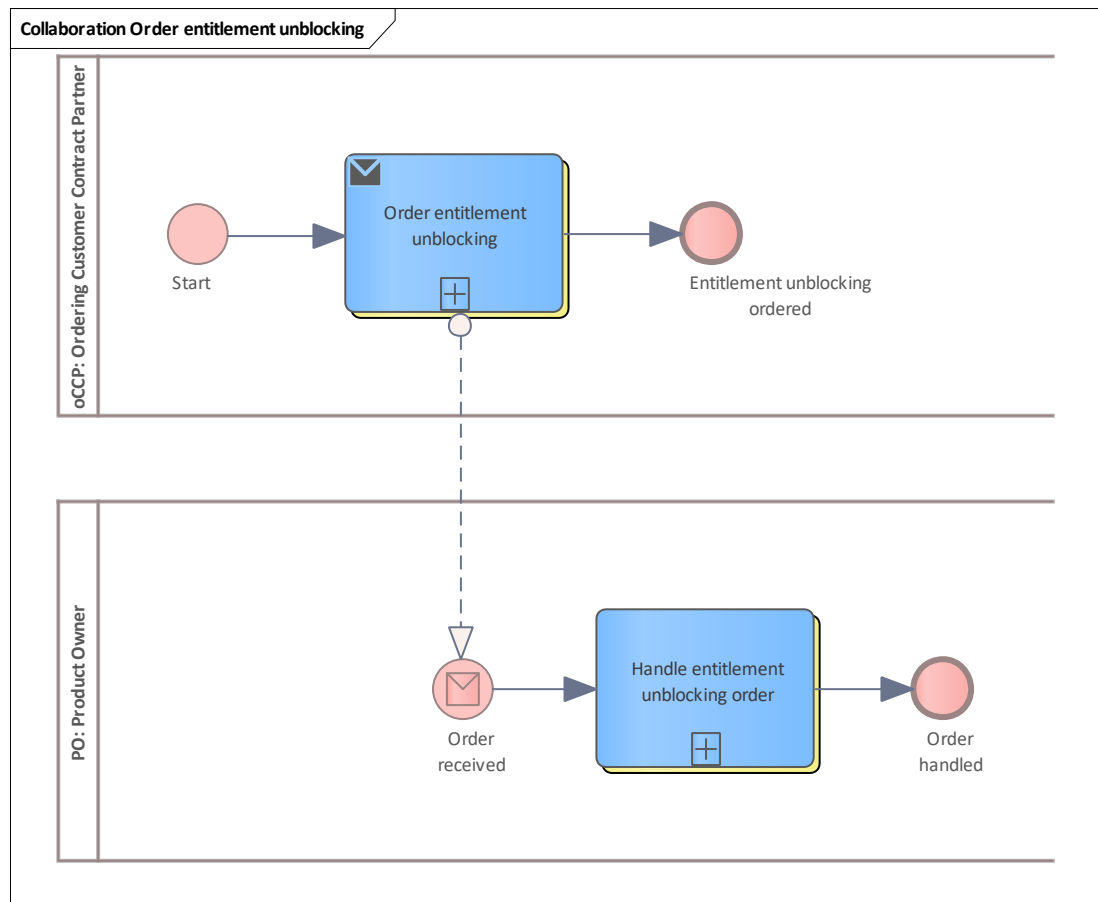


Figure 148: Order entitlement unblocking

9.2.49.1 oCCP

See [Ordering Customer Contract Partner](#)

9.2.49.1.1 Order entitlement unblocking

See [Order entitlement unblocking](#).

9.2.49.2 PO

See [Product Owner](#)

9.2.49.2.1 Handle entitlement unblocking order

See [Handle entitlement unblocking order](#).

9.2.50 Order group

Basic process that takes place between the [Ordering Customer Contract Partner](#) and the [Product Owner](#) (its action management system module).

The ordering customer contract partner places a group of orders to perform the contained actions together. The order group contains at least one order. The most common scenario will be two orders, one for the termination of an entitlement and one for the issuance of a new entitlement.

The product owner registers this new order group and puts it into its action inventory.

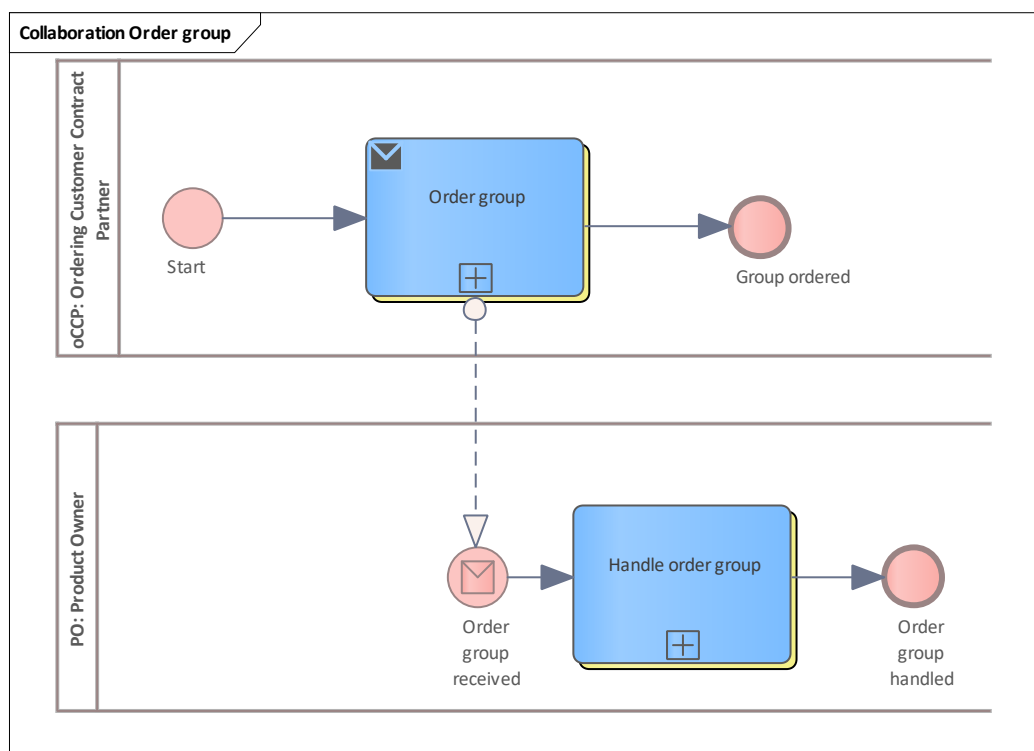


Figure 149: Order group

9.2.50.1 oCCP

See [Ordering Customer Contract Partner](#)

9.2.50.1.1 Order group

See [Order group](#)

9.2.50.2 PO

See [Product Owner](#)

9.2.50.2.1 Handle order group

See [Handle order group](#)

9.2.51 Update action list inventory

Basic process of the retrieval and distribution of action lists.

Only [Executing Customer Contract Partners](#) need the action lists.

Due to the size of the list, it is also possible to work with incremental action lists. Either the full action list or the incremental action list has to be retrieved regularly.

Optionally, if using the incremental action list, verification can be triggered by the [Executing Customer Contract Partner](#) system. The system updates its inventory with incremental elements and calculates a checksum. This checksum is sent to the action list service of the PO that calculates the checksum over the full action list the requestor should have and compares the two checksum values.

Finally, the internal action list inventory is updated with the new action information and the action list is distributed to the appropriate terminals.

See [Update action list inventory from operational perspective](#).

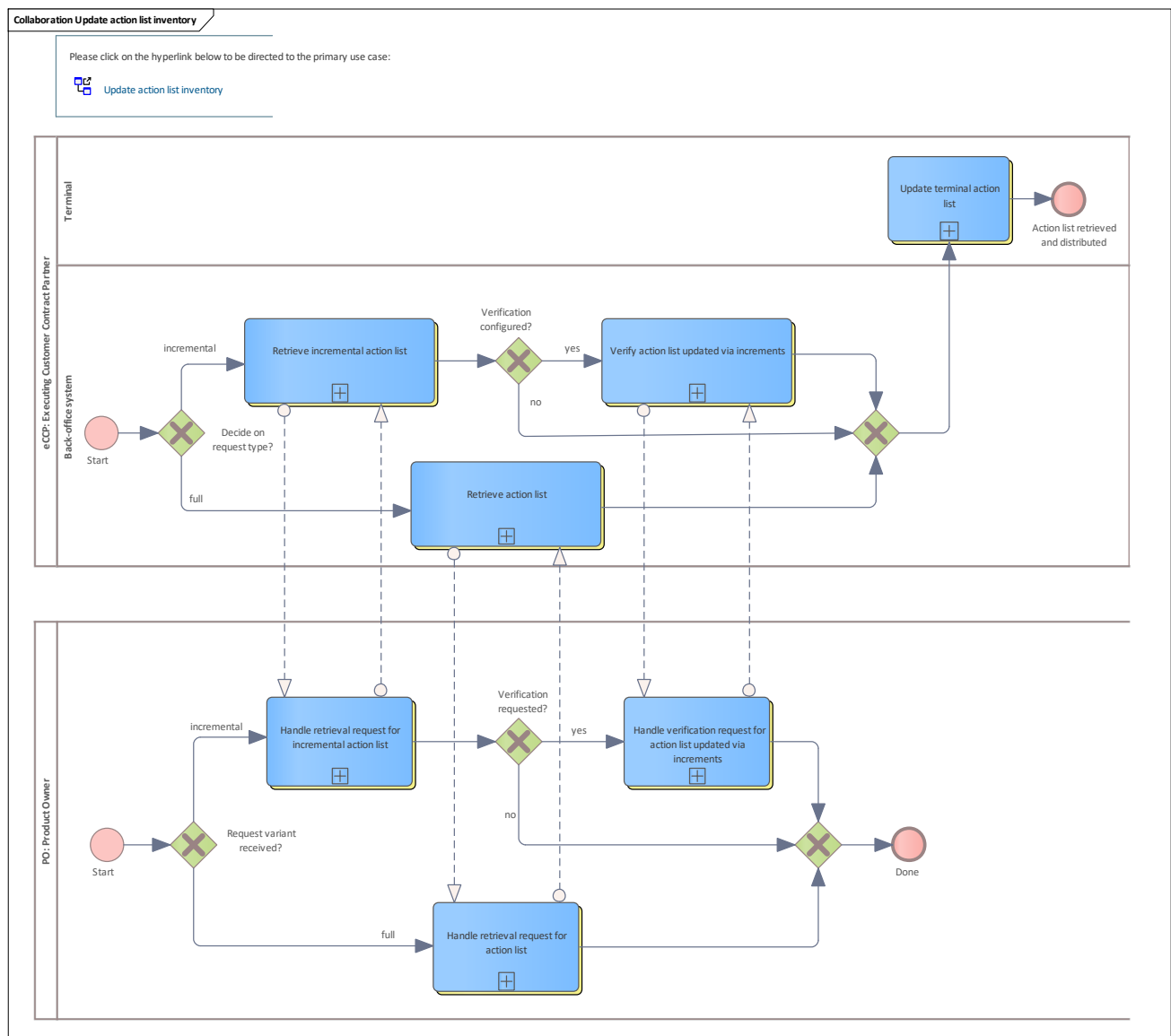


Figure 150: Update action list inventory



9.2.51.1 eCCP

See [Executing Customer Contract Partner](#)

9.2.51.1.1 Back-office system

Lane for a back-office system

1.1.1.1.1.169 Retrieve action list

See [Retrieve action list](#).

1.1.1.1.1.170 Verify action list updated via increments

See [Verify action list updated via increments](#).

1.1.1.1.1.171 Retrieve incremental action list

See [Retrieve incremental action list](#).

9.2.51.1.2 Terminal

Lane for terminal

1.1.1.1.1.172 Update terminal action list

See [Update terminal action list](#)

9.2.51.2 PO

See [Product Owner](#)

9.2.51.2.1 Handle retrieval request for action list

See [Handle retrieval request for action list](#).

9.2.51.2.2 Handle retrieval request for incremental action list

See [Handle retrieval request for incremental action list](#).

9.2.51.2.3 Handle verification request for action list updated via increments

See [Handle verification request for action list updated via increments](#).

9.2.52 Execute ordered entitlement issuance

Basic process between the [Executing Customer Contract Partner](#) and the action management of the PO and, in a second step, between the PO and the [Ordering Customer Contract Partner](#). The terminal of the executing CCP issues a new entitlement due to an action list entry with the order for the current application instance on the user medium.

This transaction is embedded in a signed attestation and notified to the back-office system of the executing CCP. This system does some operational checks including the verification of the signature. Then, the notification is forwarded to the PO that does its monitoring and removes the action entry from its action inventory. Then the PO finally forwards the notification to the [Ordering Customer Contract Partner](#) that does its contractual monitoring, registers the notification and updates the state of its order to "executed".

Note: even if the [Ordering Customer Contract Partner](#) and the [Executing Customer Contract Partner](#) have the same organisation ID, the functionality is often split between two different systems.

The scenario is comparable with [Entitlement non-owned](#).

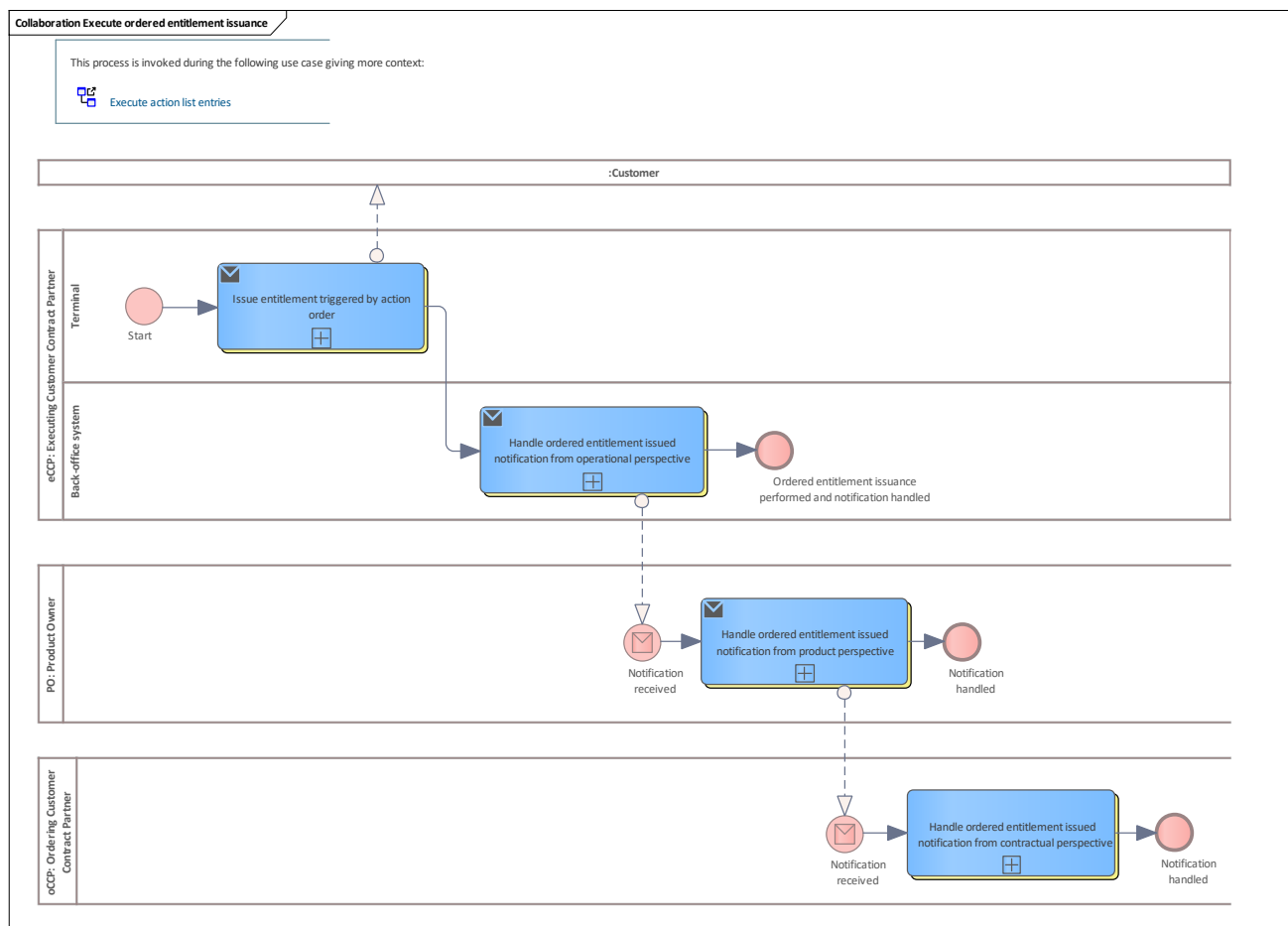


Figure 151: Execute ordered entitlement issuance

9.2.52.1 eCCP

See [Executing Customer Contract Partner](#)

9.2.52.1.1 Terminal

Lane for terminal

1.1.1.1.1.173 Issue entitlement triggered by action order

See [Issue entitlement triggered by action order](#).

9.2.52.1.2 Back-office system

Lane for a back-office system

1.1.1.1.1.174 Handle ordered entitlement issued notification from operational perspective

See [Handle ordered entitlement issued notification from operational perspective](#).

9.2.52.2 PO

See [Product Owner](#)

9.2.52.2.1 Handle ordered entitlement issued notification from product perspective

See [Handle ordered entitlement issued notification from product perspective](#).

9.2.52.3 oCCP

See [Ordering Customer Contract Partner](#)

9.2.52.3.1 Handle ordered entitlement issued notification from contractual perspective

See [Handle ordered entitlement issued notification from contractual perspective](#).

9.2.53 Execute ordered entitlement termination

Basic process between the [Executing Customer Contract Partner](#) and the action management of the PO and, in a second step, between the PO and the [Ordering Customer Contract Partner](#). The terminal of the executing CCP terminates an entitlement due to an action list entry with the order for this entitlement and the application instance on the involved user medium.

This transaction is embedded in a signed attestation and notified to the back-office system of the executing CCP. This system does some operational checks including the verification of the signature. Then, the notification is forwarded to the PO that does its monitoring and removes the action entry from its action inventory. Then the PO finally forwards the notification to the [Ordering Customer Contract Partner](#) that does its contractual monitoring, registers the notification and updates the state of its order to "executed".

The [Ordering Customer Contract Partner](#) checks if the entitlement is still on the hotlist. If so, the [Ordering Customer Contract Partner](#) requests the removal of the hotlist entry for this entitlement.

Note: even if the [Ordering Customer Contract Partner](#) and the [Executing Customer Contract Partner](#) have the same organisation ID, the functionality is often split between two different systems.

The scenario is comparable with [Entitlement non-owned](#).

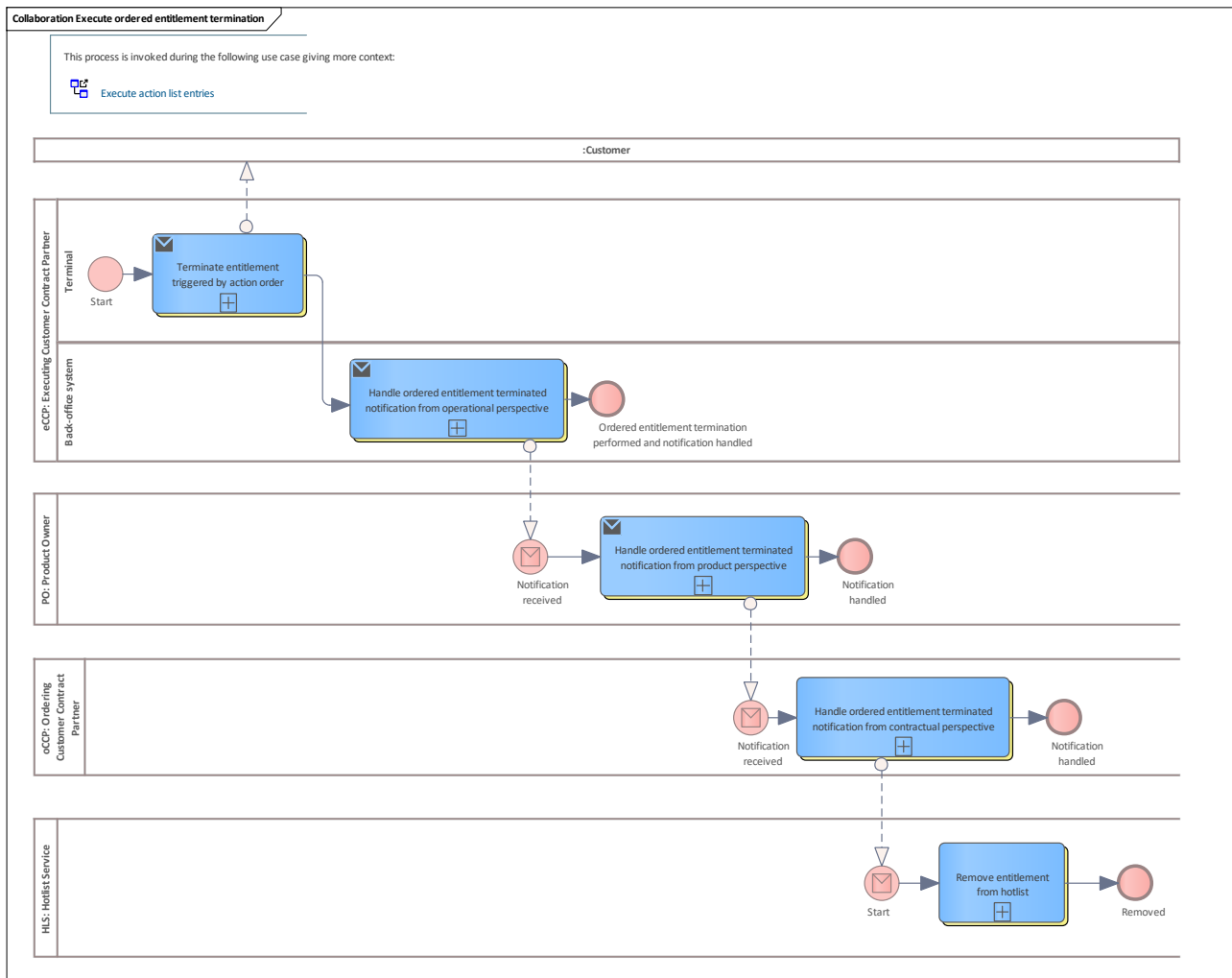


Figure 152: Execute ordered entitlement termination

9.2.53.1 eCCP

See [Executing Customer Contract Partner](#)

9.2.53.1.1 Terminal

Lane for terminal

1.1.1.1.1.175 Terminate entitlement triggered by action order

See [Terminate entitlement triggered by action order](#).

9.2.53.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.176 Handle ordered entitlement terminated notification from operational perspective

See [Handle ordered entitlement terminated notification from operational perspective](#).

9.2.53.2 PO

See [Product Owner](#)

9.2.53.2.1 Handle ordered entitlement terminated notification from product perspective

See [Handle ordered entitlement terminated notification from product perspective](#).

9.2.53.3 oCCP

See [Ordering Customer Contract Partner](#)

9.2.53.3.1 Handle ordered entitlement terminated notification from contractual perspective

See [Handle ordered entitlement terminated notification from contractual perspective](#).

9.2.53.4 HLS

See [Hotlist Service](#)

9.2.53.4.1 Remove entitlement from hotlist

See [Remove entitlement from hotlist](#)

9.2.54 Execute ordered entitlement blocking

Basic process between the [Executing Customer Contract Partner](#) and the action management of the PO and, in a second step, between the PO and the [Ordering Customer Contract Partner](#). The terminal of the executing CCP blocks an entitlement due to an action list entry with the application instance on the involved user medium. If the entitlement has to be blocked without knowing the entitlement ID, the terminal has to determine the order ID which is stored additionally in the entitlement. If this order ID matches the order ID in the action list entry, the related entitlement will be blocked.

This transaction is embedded in a signed attestation and notified to the back-office system of the executing CCP. This system does some operational checks including the verification of the signature. Then, the notification is forwarded to the PO that does its monitoring and removes the action entry from its action inventory. Then the PO finally forwards the notification to the [Ordering Customer Contract Partner](#) that does its contractual monitoring, registers the notification and updates the state of its order to "executed". Since the entitlement was never on the hotlist, no removal of the hotlist entry is needed.

Note: even if the [Ordering Customer Contract Partner](#) and the [Executing Customer Contract Partner](#) have the same organisation ID, the functionality is often split between two different systems.

The scenario is comparable with [Entitlement non-owned](#).

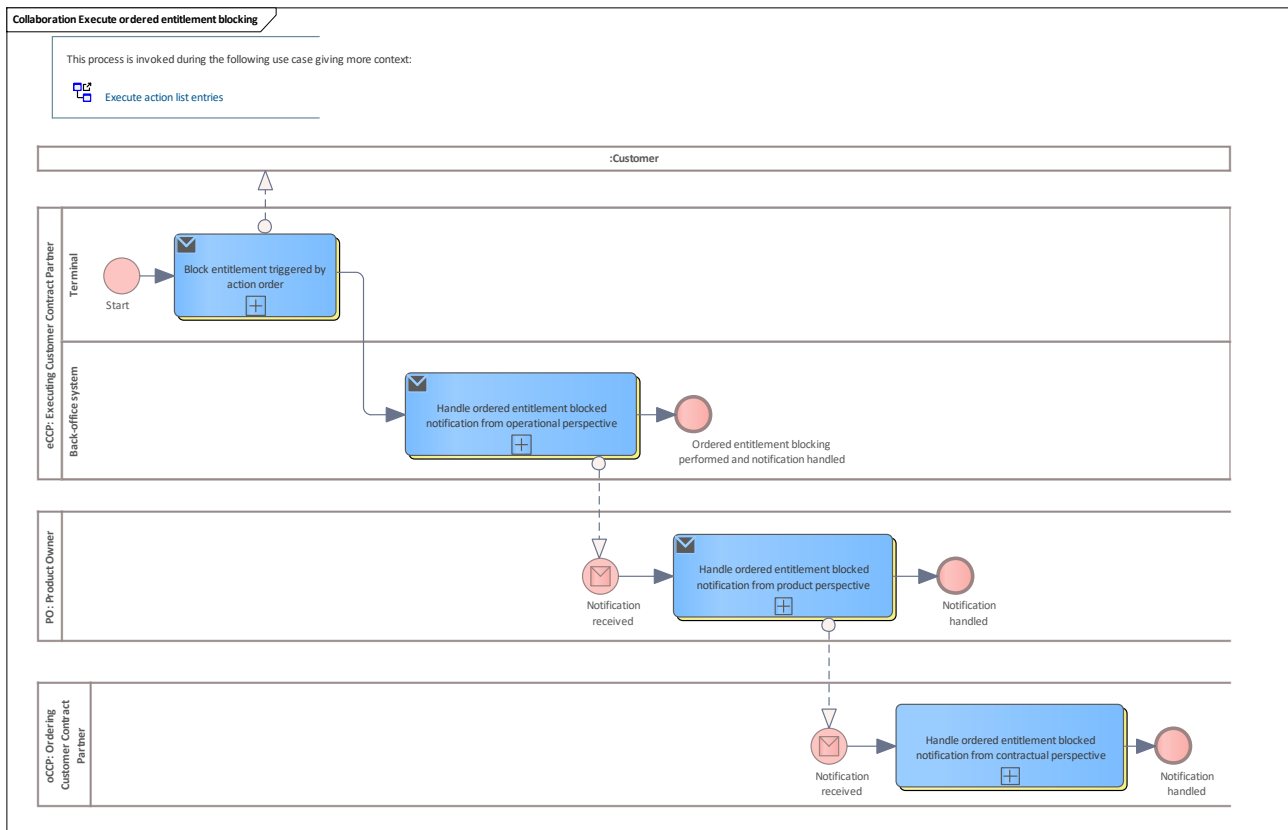


Figure 153: Execute ordered entitlement blocking

9.2.54.1 eCCP

See [Executing Customer Contract Partner](#)

9.2.54.1.1 Terminal

Lane for terminal

1.1.1.1.1.177 Block entitlement triggered by action order

See [Block entitlement triggered by action order](#).

9.2.54.1.2 Back-office system

Lane for a back-office system

1.1.1.1.1.178 Handle ordered entitlement blocked notification from operational perspective

See [Handle ordered entitlement blocked notification from operational perspective](#).

9.2.54.2 PO

See [Product Owner](#)

9.2.54.2.1 Handle ordered entitlement blocked notification from product perspective

See [Handle ordered entitlement blocked notification from product perspective](#).

9.2.54.3 oCCP

See [Ordering Customer Contract Partner](#)

9.2.54.3.1 Handle ordered entitlement blocked notification from contractual perspective

See [Handle ordered entitlement blocked notification from contractual perspective](#).

9.2.55 Execute ordered entitlement unblocking

Basic process between the [Executing Customer Contract Partner](#) and the action management of the PO and, in a second step, between the PO and the [Ordering Customer Contract Partner](#).

The terminal of the executing CCP unblocks an entitlement due to an action list entry with the order for this entitlement and the application instance on the involved user medium.

This transaction is embedded in a signed attestation and notified to the back-office system of the executing CCP. This system does some operational checks including the verification of the signature. Then, the notification is forwarded to the PO that does its monitoring and removes the action entry from its action inventory. Then the PO finally forwards the notification to the [Ordering Customer Contract Partner](#) that does its contractual monitoring, registers the notification and updates the state of its order to "executed".

Note: even if the [Ordering Customer Contract Partner](#) and the [Executing Customer Contract Partner](#) have the same organisation ID, the functionality is often split between two different systems.

The scenario is comparable with [Entitlement non-owned](#).

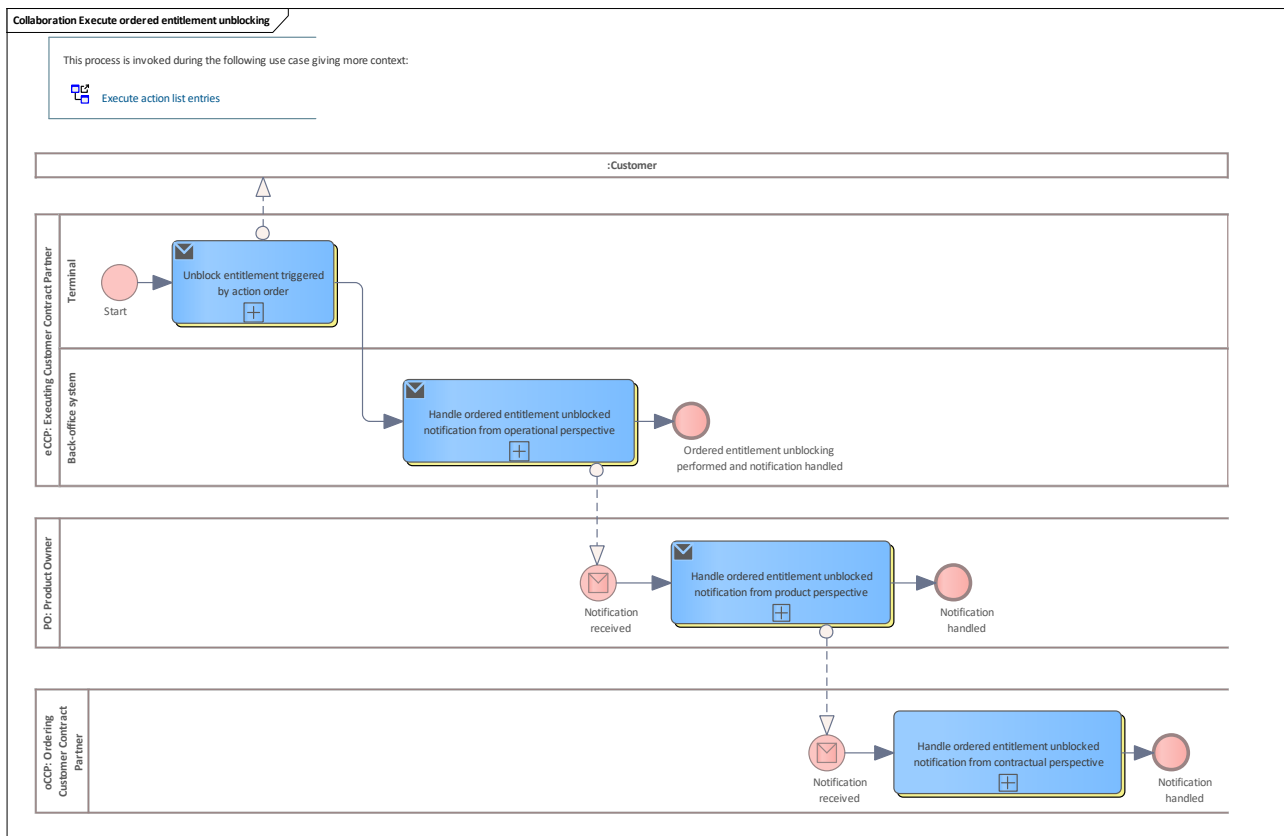


Figure 154: Execute ordered entitlement unblocking

9.2.55.1 eCCP

See [Executing Customer Contract Partner](#)

9.2.55.1.1 Terminal

Lane for terminal

1.1.1.1.1.179 Unblock entitlement triggered by action order

See [Unblock entitlement triggered by action order](#).

9.2.55.1.2 Back-office system

Lane for a back-office system.

1.1.1.1.1.180 Handle ordered entitlement unblocked notification from operational perspective

See [Handle ordered entitlement unblocked notification from operational perspective](#).



9.2.55.2 PO

See [Product Owner](#)

9.2.55.2.1 Handle ordered entitlement unblocked notification from product perspective

See [Handle ordered entitlement unblocked notification from product perspective](#).

9.2.55.3 oCCP

See [Ordering Customer Contract Partner](#)

9.2.55.3.1 Handle ordered entitlement unblocked notification from contractual perspective

See [Handle ordered entitlement unblocked notification from contractual perspective](#).

9.2.56 Cancel order

Basic process between the [Ordering Customer Contract Partner](#) and the action management of the PO.

The [Ordering Customer Contract Partner](#) wants to cancel an existing order. This can be done for each possible order type (issue/ terminate, unblock/block entitlement). An order group cannot be cancelled. In this case, each contained order in the order group has to be individually cancelled.

The PO action management system receives the order cancellation and removes the order from its action inventory and, therefore, from all future action lists.

Note: due to possible race conditions, the premature execution of the order might not be prevented by this cancellation process.

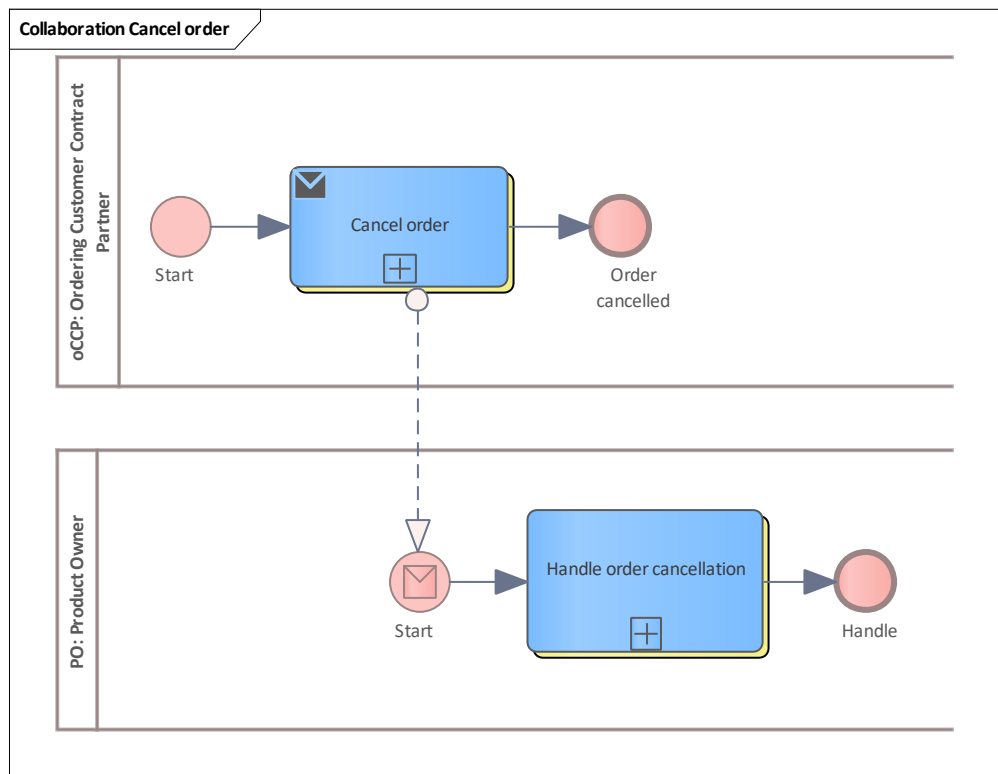


Figure 155: Cancel order

9.2.56.1 oCCP

See [Ordering Customer Contract Partner](#)

9.2.56.1.1 Cancel order

See [Cancel order](#).

9.2.56.2 PO

See [Product Owner](#)

9.2.56.2.1 Handle order cancellation

See [Handle order cancellation](#).

9.2.57 Handle obsolete order

This basic process is started whenever an entitlement notification is handled that makes an active order superfluous. Note that this is only possible for unblocking orders for technical reasons.

The result is a cleared action list.

Furthermore, the [Ordering Customer Contract Partner](#) is informed about the obsolete order.

The [Executing Customer Contract Partner](#) is informed implicitly since the order is removed from future action lists.

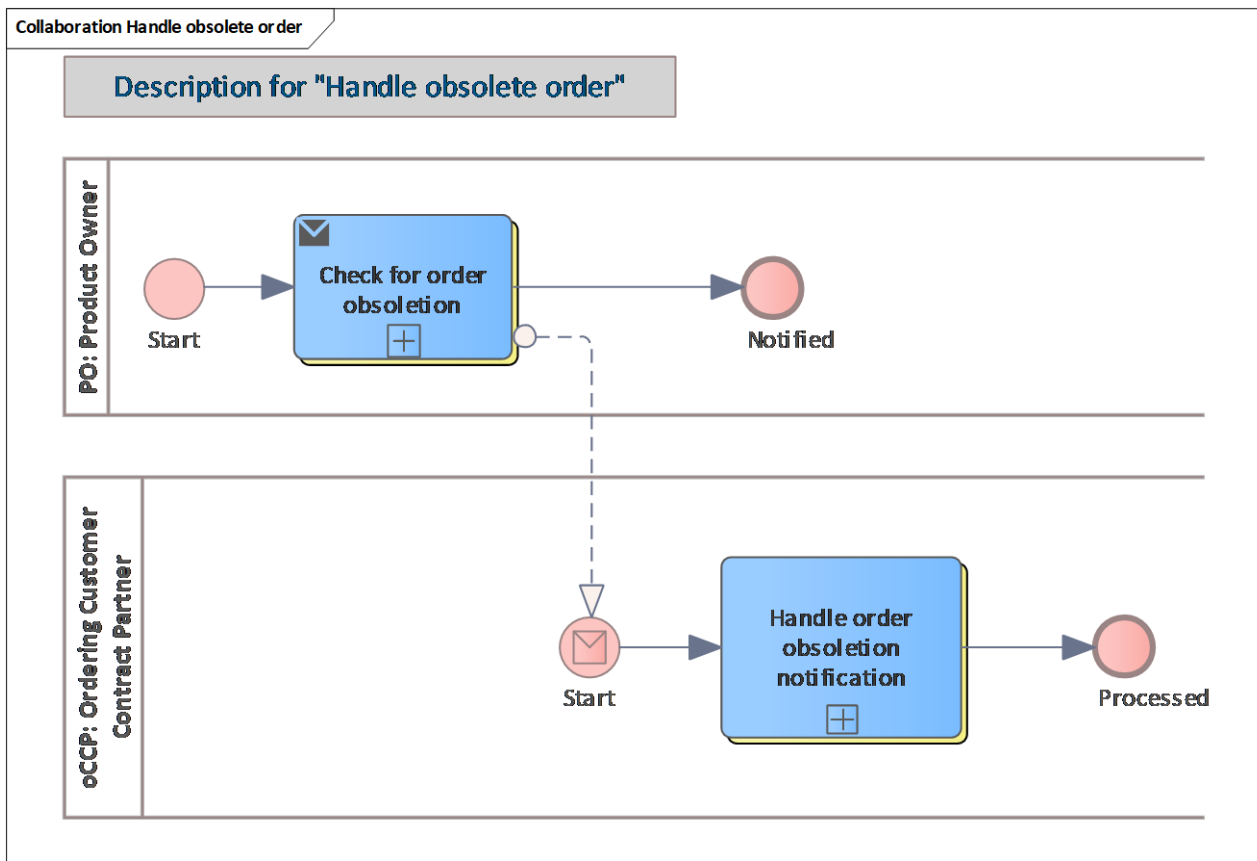


Figure 156: Handle obsolete order

9.2.57.1 oCCP

See [Ordering Customer Contract Partner](#)

9.2.57.1.1 Handle order obsolescence notification

See [Handle order obsolescence notification](#).

9.2.57.2 PO

See [Product Owner](#)

9.2.57.2.1 Check for order obsolescence

This Use Case is triggered by the monitoring processes of the PO if the action management is used.

See [Check for order obsolescence](#).

9.2.58 CICO

This chapter describes the participants and the activities within the basic process related to the check-in/check-out process.

BPMN Collaboration is used.

9.2.59 Change user tariff parameters of a non-owned entitlement

Basic process to change the user's tariff parameters temporarily for the next trip, journey or even longer. The changed parameters are passed through the check-in and check-out transactions until the customer resets these parameters back to his defaults.

The user/customer changes his parameters concerning class, accompaniment of further persons/things/etc. using a CCP terminal. The used payment method (store-value or account-based) does not belong to the CCP that operates the terminal. The CCP is a sCCP.

In this case, the typical template process for [Entitlement non-owned](#) is the foundation of this basic process.

The process starts in the CCP terminal. The customer changes his parameters. These changes are stored on the user medium and will be considered in the next check-in procedure and then forwarded until revoked.

The change is notified from the terminal to the sCCP back-office system. The sCCP back-office does its operational checks and monitoring and forwards the notification to the responsible PO. The PO does its monitoring and registers the notification. In the case of an account-based payment method with post-pricing, the change must be considered. Note: in case of a stored-value payment method, an adapted amount will be booked on the user medium with the next check-in. Finally, the PO forwards the notification to the responsible pCCP.

The pCCP does its contractual checks and monitoring and registers the notification.

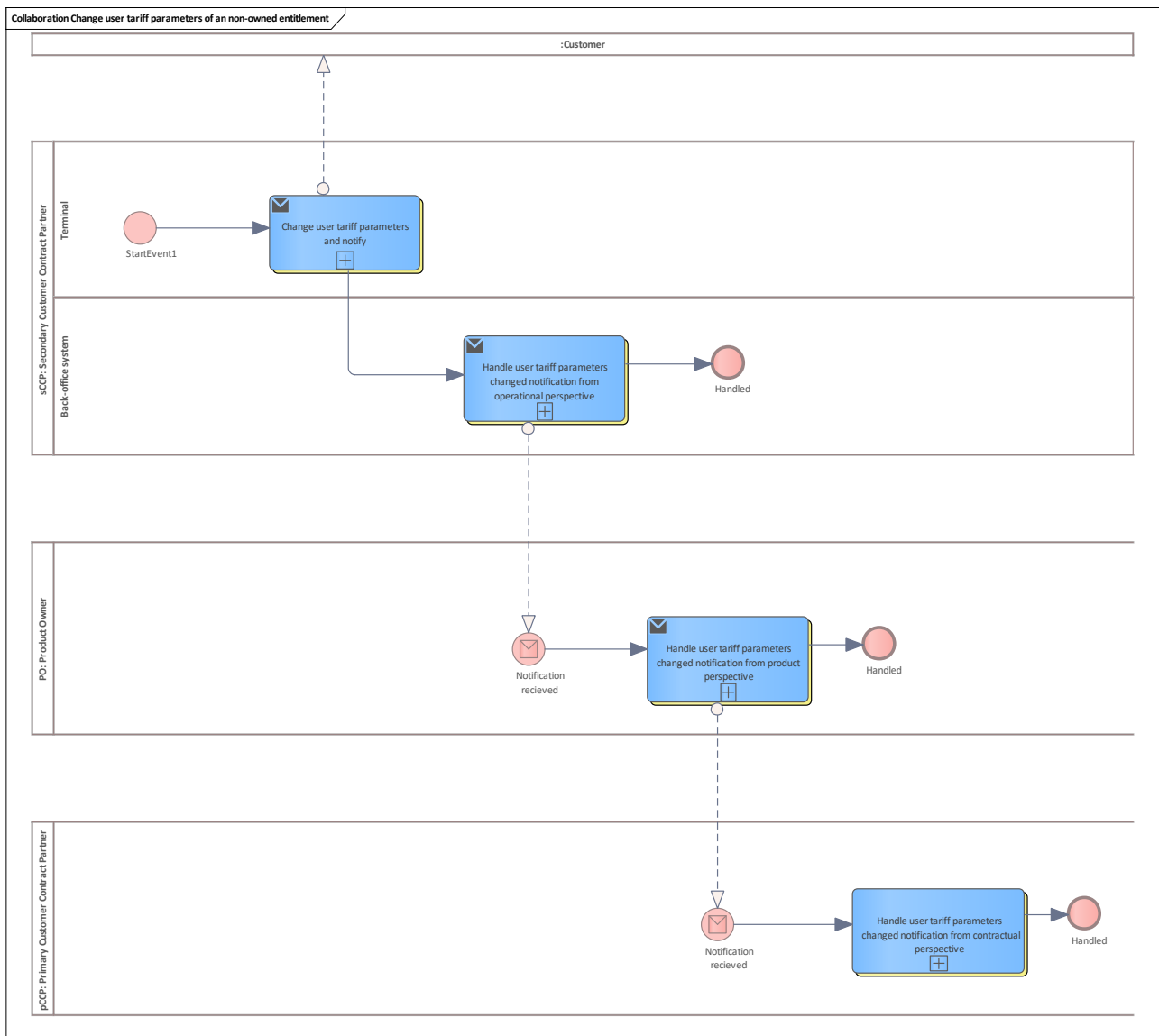


Figure 157: Change user tariff parameters of a non-owned entitlement

9.2.59.1 sCCP

See [Secondary Customer Contract Partner](#)

9.2.59.1.1 Terminal

Lane for terminal

1.1.1.1.1.181 Change user tariff parameters and notify

See [Change user tariff parameters and notify](#).

9.2.59.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.182 Handle user tariff parameters changed notification from operational perspective

See [Handle user tariff parameters changed notification from operational perspective.](#)

9.2.59.2 PO

See [Product Owner](#)

9.2.59.2.1 Handle user tariff parameters changed notification from product perspective

See [Handle user tariff parameters changed notification from product perspective.](#)

9.2.59.3 pCCP

See [Primary Customer Contract Partner](#)

9.2.59.3.1 Handle user tariff parameters changed notification from contractual perspective

See [Handle user tariff parameters changed notification from contractual perspective.](#)

9.2.60 Change user tariff parameters of an owned entitlement

Basic process to change the user's tariff parameters temporarily for the next trip, journey or even longer. The changed parameters are passed through the check-in and check-out transactions until the customer resets these parameters back to his defaults.

The user/customer changes his parameters concerning class, accompaniment of further persons/things/etc. using a CCP terminal. The used payment method (store-value or account-based) belongs to the CCP that operates the terminal. The CCP is the pCCP.

In this case, the typical template process for [Entitlement owned](#) is the foundation of this basic process.

The process starts in the CCP terminal. The customer changes his parameters. These changes are stored on the user medium and will be considered in the next check-in procedure and then forwarded until revoked.

The change is notified from the terminal to the pCCP back-office system. The pCCP back-office does its operational checks and monitoring and then its contractual checks and monitoring and registers the notification. Then the notification is forwarded to the responsible PO.

The PO does its monitoring and registers the notification. In the case of an account-based payment method with post-pricing, the change must be considered.

Note: in case of a stored-value payment method, an adapted amount will be booked on the user medium with the next check-in.

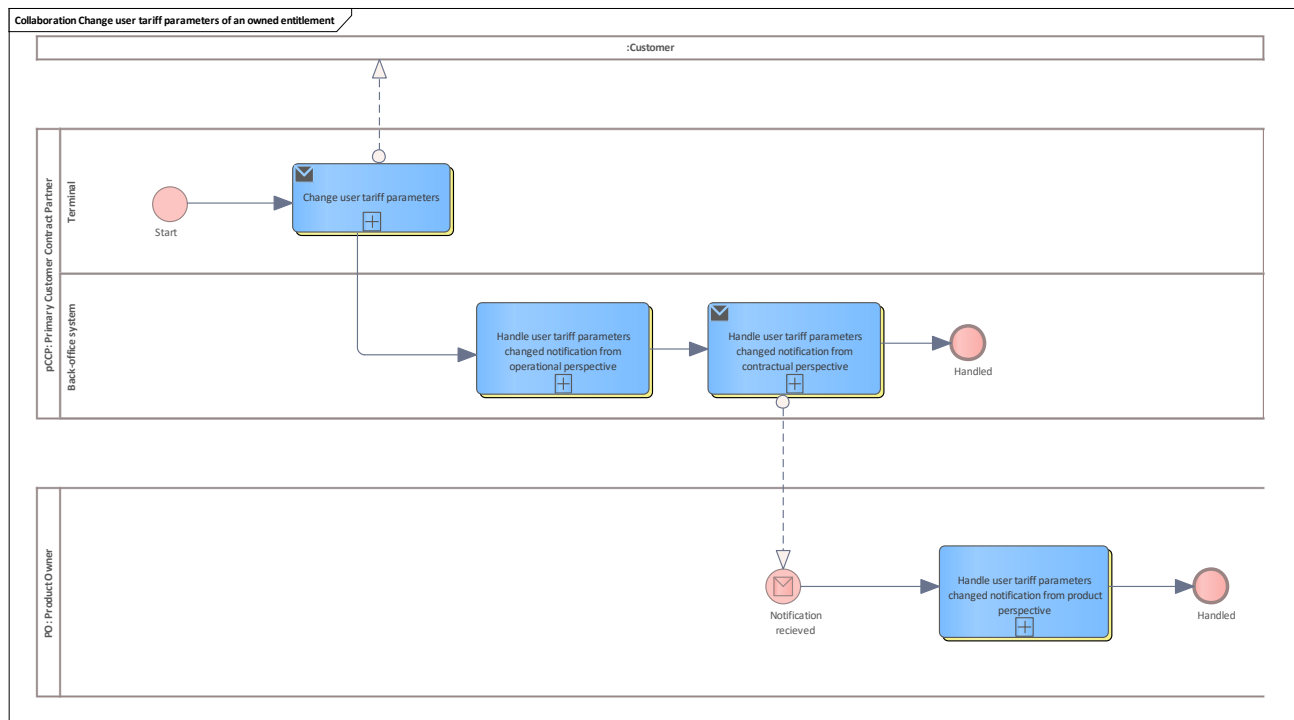


Figure 158: Change user tariff parameters of an owned entitlement

9.2.60.1 pCCP

See [Primary Customer Contract Partner](#)

9.2.60.1.1 Terminal

Lane for terminal

1.1.1.1.1.183 Change user tariff parameters

See [Change user tariff parameters](#).

9.2.60.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.184 Handle user tariff parameters changed notification from operational perspective

See [Handle user tariff parameters changed notification from operational perspective](#).

1.1.1.1.1.185 Handle user tariff parameters changed notification from contractual perspective

See [Handle user tariff parameters changed notification from contractual perspective](#).

9.2.60.2 PO

See [Product Owner](#)

9.2.60.2.1 Handle user tariff parameters changed notification from product perspective

See [Handle user tariff parameters changed notification from product perspective](#).

9.2.61 Charge account-based payment method

Basic process between the PO and the pCCP. The PO starts a process to rate all registered journeys paid by account-based payment methods, e.g. once a month. This is done for each pCCP. All journeys for all account-based payment methods of a pCCP are condensed to a list which is sent to the related pCCP.

The pCCP receives the list and stores the entries for customer information purposes and booking.

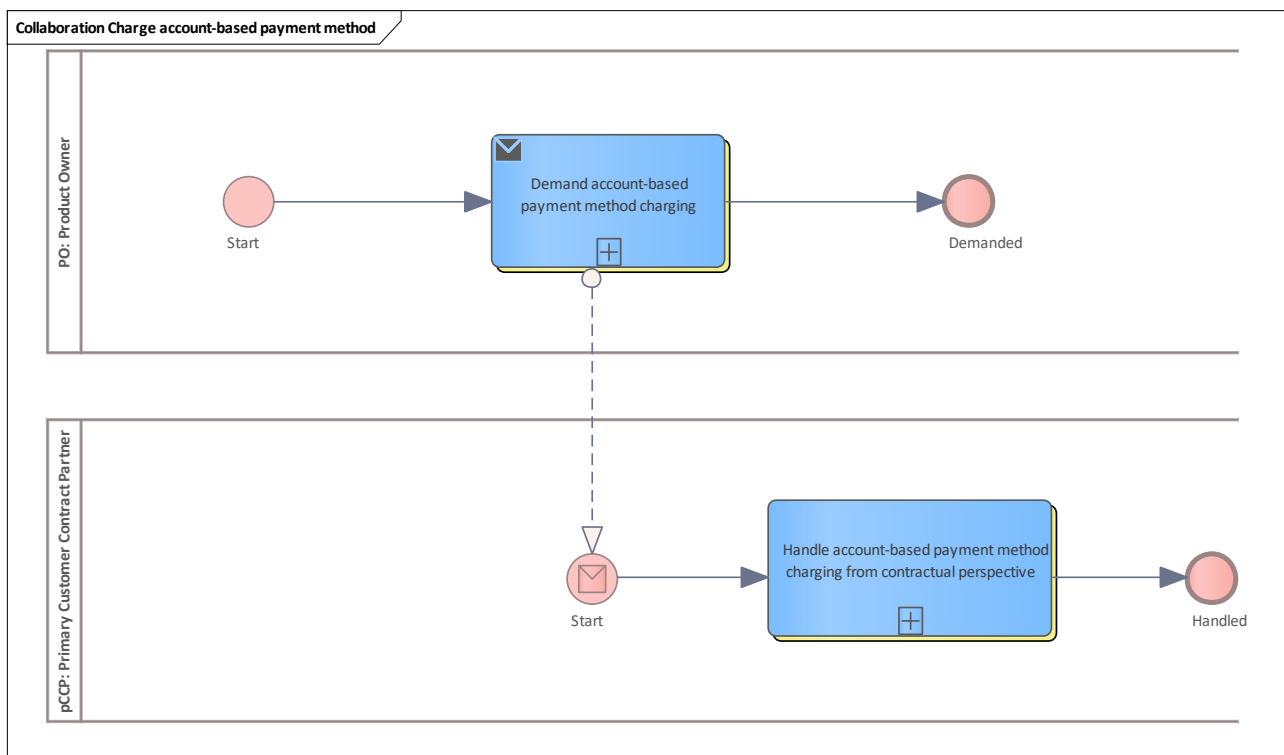


Figure 159: Charge account-based payment method

9.2.61.1 PO

See [Product Owner](#)

9.2.61.1.1 Demand account-based payment method charging

See [Demand account-based payment method charging](#)

9.2.61.2 pCCP

See [Primary Customer Contract Partner](#)

9.2.61.2.1 Handle account-based payment method charging from contractual perspective

[Handle account-based payment method charging from contractual perspective](#)

9.2.62 Record entitlement within the check-in process

Basic process to record an entitlement (etiCORE payment method) and perform a check-in. In this case, the typical template process for [Entitlement non-owned](#) is the foundation of this basic process.

The process starts in the terminal of an SO.

The terminal detects an etiCORE payment method and performs a check-in action. If any user tariff parameters were changed before, these parameters have to be transferred to the current check-in attestation that is stored on the user medium.

The check-in action is notified to the back-office system of the SO.

The SO does its operational checks and monitoring and registers the notification. Then, the notification is forwarded to the responsible PO.

The PO does its checks and monitoring and registers the notification. Especially for an account-based payment method with post pricing, the check-in notification is needed for rating.

The PO forwards the notification to the pCCP that does its contractual checks and monitoring.

The check-in notification is registered. Especially for an account-based payment method with post pricing, the subsequent itemised bill of the PO has to be verified using these notifications.

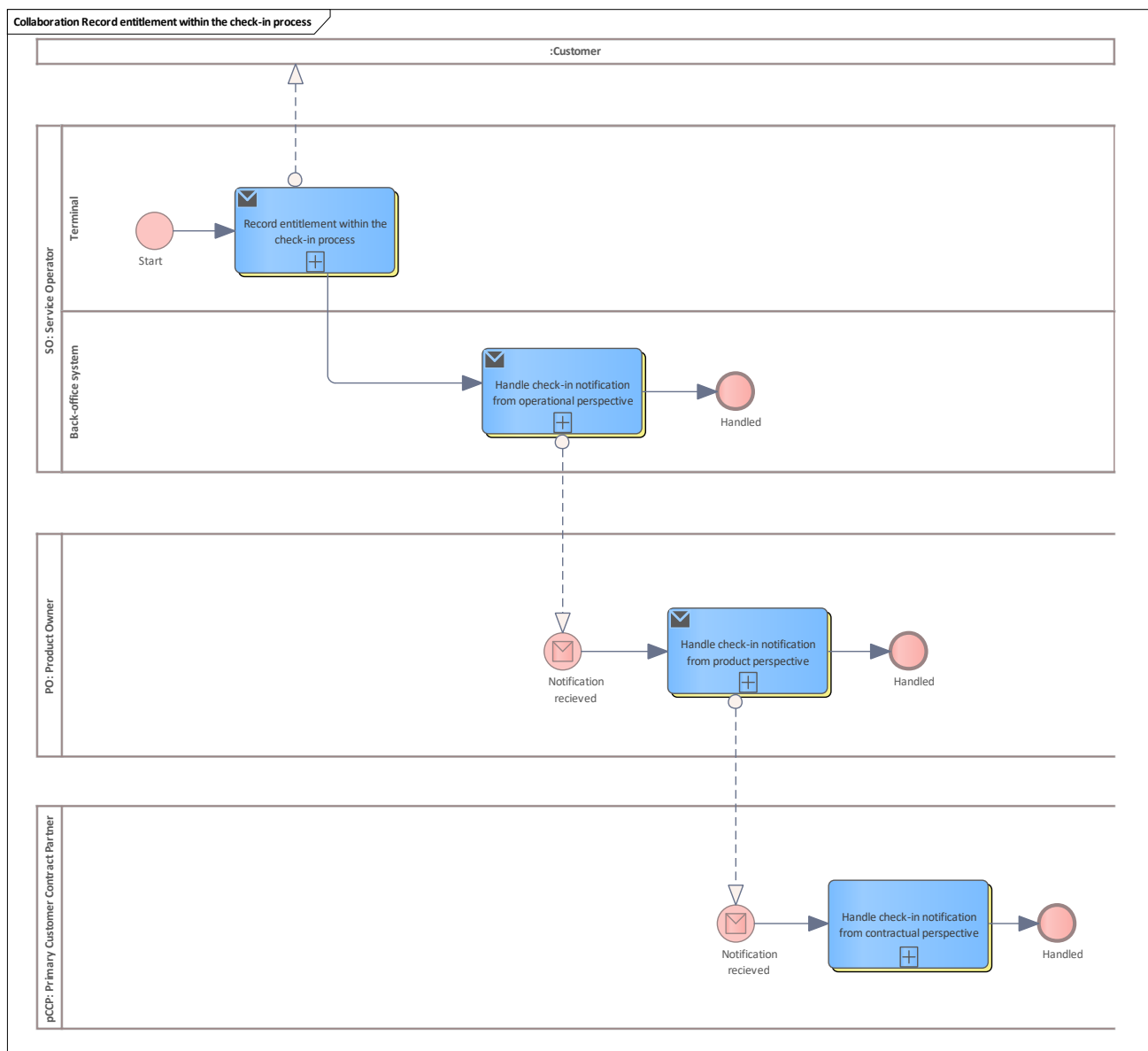


Figure 160: Record entitlement within the check-in process

9.2.62.1 SO

See [Service Operator](#)

9.2.62.1.1 Terminal

Lane for terminal

1.1.1.1.1.186 Record entitlement within the check-in process

See [Record entitlement within CICO system](#).

Please note that the decision for executing a check-in or check-out is taken in the middle of the CICO process.



For the sake of simplicity, the collaboration model is designed in a way as if the check-in decision is taken at the beginning of the process.

9.2.62.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.187 Handle check-in notification from operational perspective

See [Handle check-in notification from operational perspective](#).

9.2.62.2 PO

See [Product Owner](#)

9.2.62.2.1 Handle check-in notification from product perspective

See [Handle check-in notification from product perspective](#).

9.2.62.3 pCCP

See [Primary Customer Contract Partner](#)

9.2.62.3.1 Handle check-in notification from contractual perspective

See [Handle check-in notification from contractual perspective](#).

9.2.63 Record entitlement within the check-out process

Basic process to record an entitlement (etiCORE payment method) and perform a check-out. In this case, the typical template process for [Entitlement non-owned](#) is the foundation of this basic process.

The process starts in the terminal of an SO.

The terminal detects an etiCORE payment method and performs a check-out transaction that is stored on the user medium. If changed user tariff parameters exist from earlier recordings, these parameters must be adopted.

The check-out transaction is notified to the back-office system of the SO.

The SO does its operational checks and monitoring and registers the notification. Then, the notification is forwarded to the responsible PO.

The PO does its checks and monitoring and registers the notification. Especially for an account-based payment method with post-pricing, the check-out notification is needed for rating.

The PO forwards the notification to the pCCP that does its contractual checks and monitoring. The check-out notification is registered. Especially for an account-based payment method with post pricing the subsequent itemised bill of the PO has to be verified using these notifications.

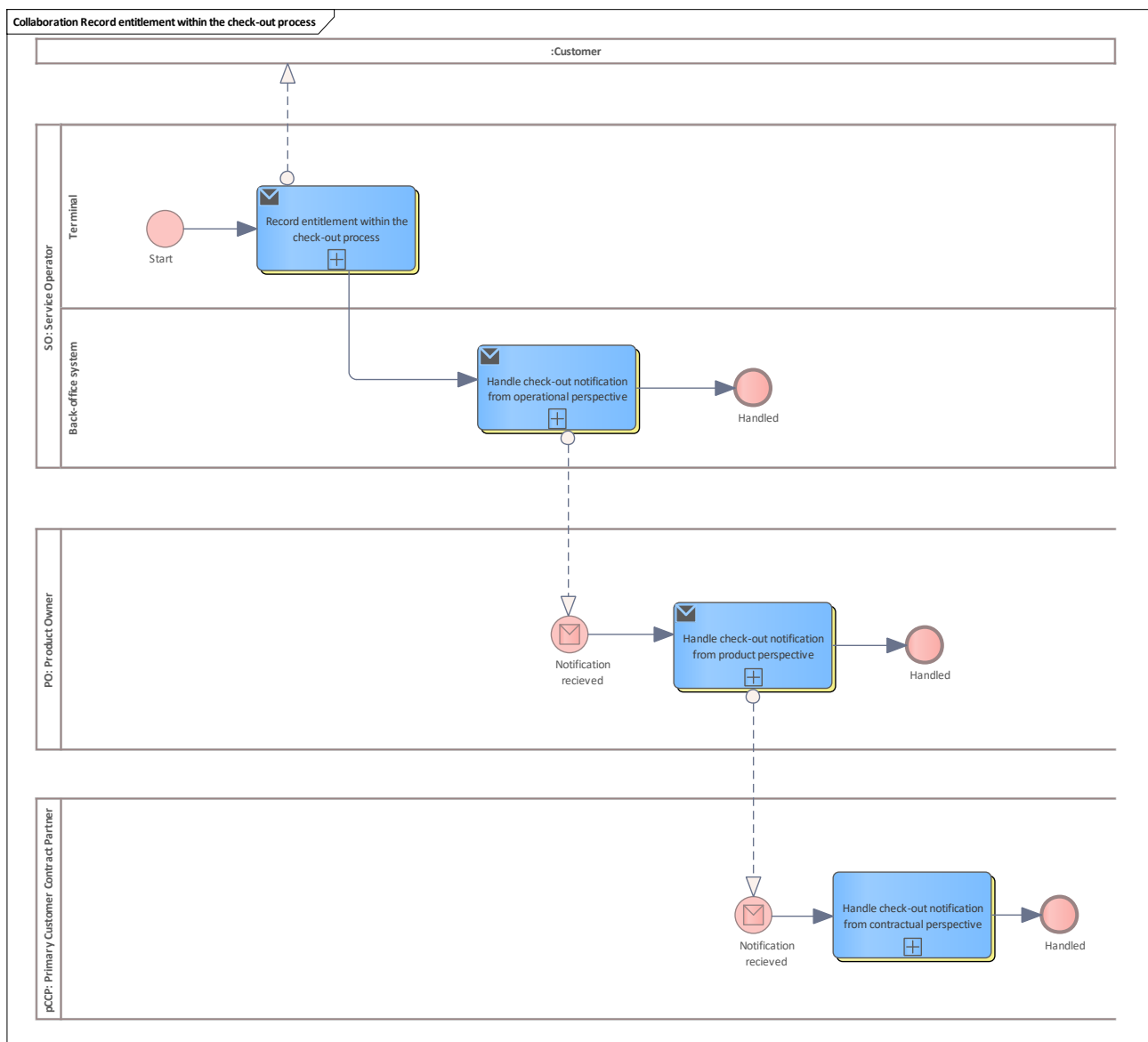


Figure 161: Record entitlement within the check-out process

9.2.63.1 SO

See [Service Operator](#)

9.2.63.1.1 Terminal

Lane for terminal

1.1.1.1.1.188 Record entitlement within the check-out process

See [Record entitlement within CICO system](#).

Please note that the decision for executing a check-in or check-out is taken in the middle of CICO process.



For the sake of simplicity, the collaboration model is designed in a way as if the check-out decision (found out by tariff regulations) is taken at the beginning of the process.

9.2.63.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.189 Handle check-out notification from operational perspective

See [Handle check-out notification from operational perspective](#).

9.2.63.2 PO

See [Product Owner](#)

9.2.63.2.1 Handle check-out notification from product perspective

See [Handle check-out notification from product perspective](#).

9.2.63.3 pCCP

See [Primary Customer Contract Partner](#)

9.2.63.3.1 Handle check-out notification from contractual perspective

See [Handle check-out notification from contractual perspective](#).

9.2.64 Validation

This chapter describes the participants and the activities within the basic process "validation". BPMN Collaboration is used.

9.2.65 Validate electronic ticket

Basic process for special products that allow electronic tickets that have to be validated before they are used.

In this case, the typical template process for [Entitlement non-owned](#) is the foundation of this basic process.

The validation is comparable with a check-in, see [Record entitlement within the check-in process](#).

The process starts in a SO terminal where the terminal writes a validation record to the user medium. The electronic ticket entitlement is only valid together with this validation record. The terminal notifies the validation action to the back-office system of the SO. The SO does its operational checks and monitoring and forwards the notification to the responsible PO.

The PO does its checks and monitoring, registers the validation action and forwards the notification to the responsible pCCP of the entitlement.

The pCCP does its contractual checks and monitoring and registers the validation action.

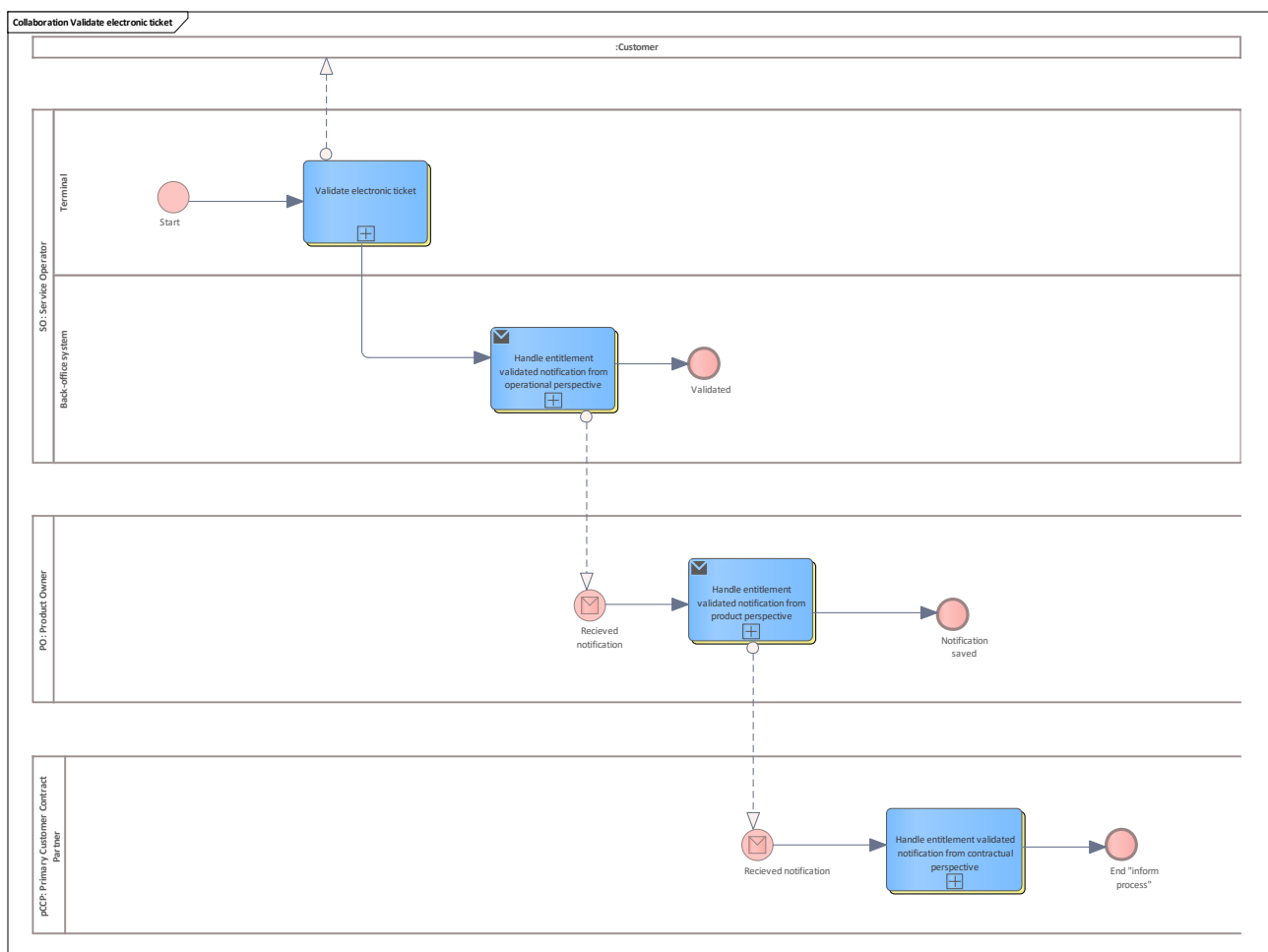


Figure 162: Validate electronic ticket

9.2.65.1 SO

See [Service Operator](#)

9.2.65.1.1 Terminal

Lane for terminal

1.1.1.1.1.190 Validate electronic ticket

See [Validate electronic ticket](#).

9.2.65.1.2 Back-office system

Lane for back-office system

1.1.1.1.1.191 Handle entitlement validated notification from operational perspective

See [Handle entitlement validated notification from operational perspective](#).

9.2.65.2 PO

See [Product Owner](#)

9.2.65.2.1 Handle entitlement validated notification from product perspective

See [Handle entitlement validated notification from product perspective](#).

9.2.65.3 pCCP

See [Customer Contract Partner](#)

9.2.65.3.1 Handle entitlement validated notification from contractual perspective

See [Handle entitlement validated notification from contractual perspective](#).

9.2.66 Monitoring and notification

This chapter contains certain processes that support the monitoring of the system components.

9.2.66.1 Extended logging

This chapter describes the participants and the activities within the basic process related to extended logging for an application or entitlement process in case that an invalid or expired entitlement/application is found.

BPMN Collaboration is used.

9.2.66.2 Extended logging for an entitlement

Small basic process to log an invalid entitlement.

Done by the terminal operator (SO or CCP).

The invalid entitlement is detected in the terminal, the reason may be

- entitlement expired or not yet valid
- entitlement blocked or terminated
- static entitlement hotlisted or expired or related SAM hotlisted
- etc.

Together with the current terminal data (ID, time, location, etc.) the entitlement and the reason are notified to the terminal operator's back-office system.

The back-office system registers the invalid entitlement for potential analysis purposes.

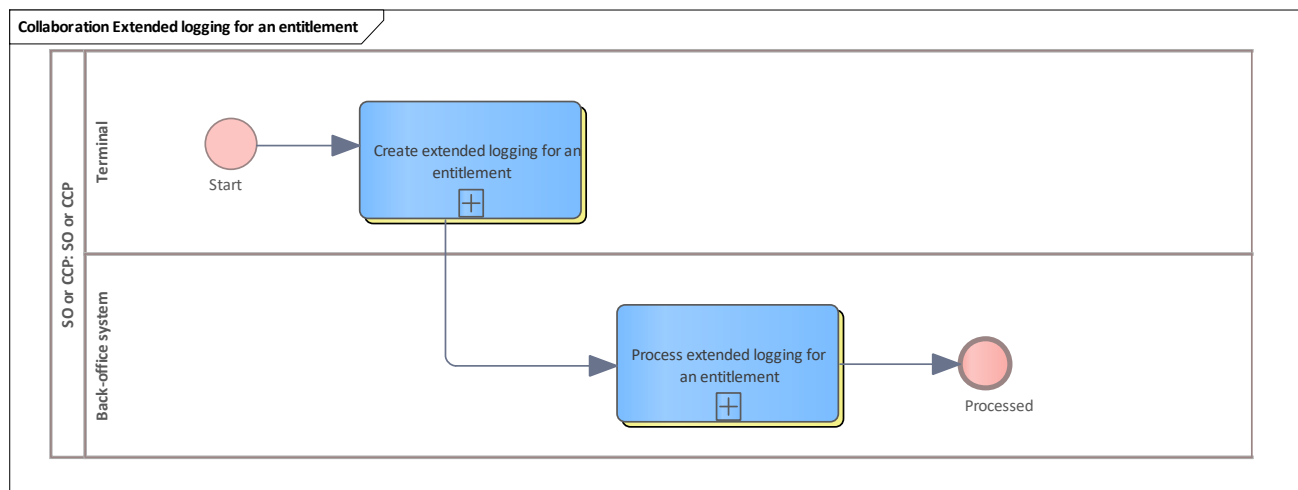


Figure 163: Extended logging for an entitlement

9.2.66.2.1 SO or CCP

See [SO or CCP](#)

1.1.1.1.1.192 Terminal

Lane for terminal

1.1.1.1.1.192.1 Create extended logging for an entitlement

See [Create extended logging for an entitlement](#).

1.1.1.1.1.193 Back-office system

Lane for back-office system

1.1.1.1.1.193.1 Process extended logging for an entitlement

See [Process extended logging for an entitlement](#).

9.2.66.3 Extended logging for an application

Small basic process to log an invalid application.
Done by the terminal operator (SO or CCP).

The invalid application is detected in the terminal, the reason may be

- application expired or not yet valid
- application blocked or terminated

Together with the current terminal data (ID, time, location, etc.) the application and the reason are notified to the terminal operator's back-office system.

The back-office system registers the invalid application for potential analysis purposes.

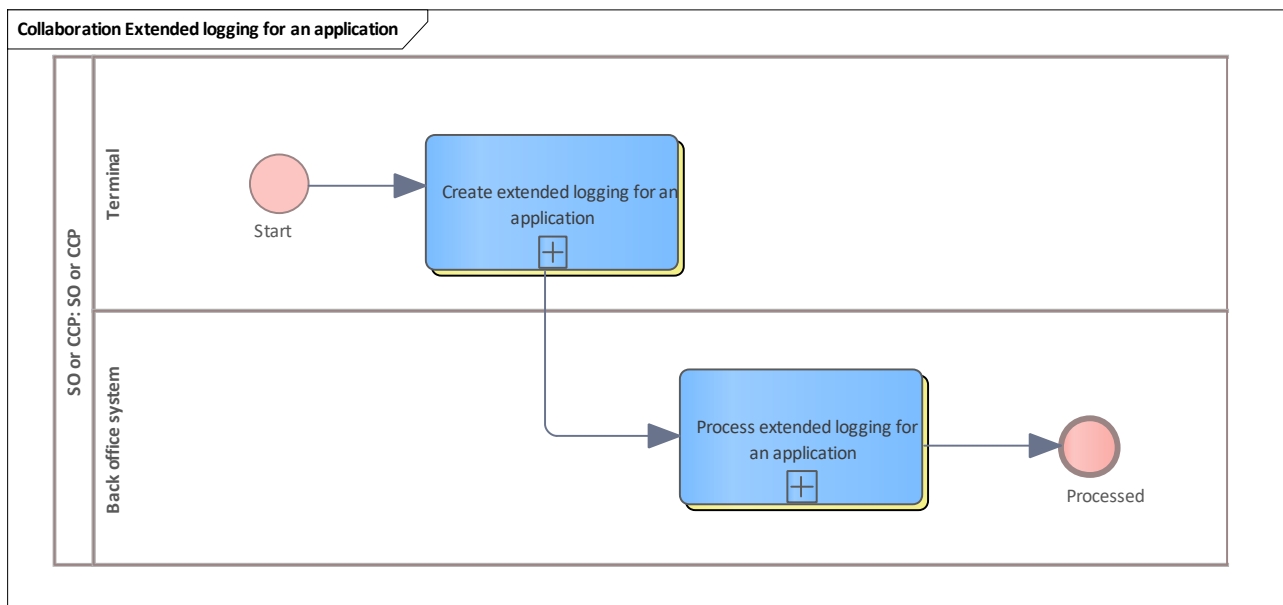


Figure 164: Extended logging for an application

9.2.66.3.1 SO or CCP

See [SO or CCP](#)

1.1.1.1.1.194 Terminal

Lane for terminal

1.1.1.1.1.194.1 Create extended logging for an application

See [Create extended logging for an application](#).

1.1.1.1.1.195 Back office system

Lane for back-office system

1.1.1.1.1.195.1 Process extended logging for an application

See [Process extended logging for an application](#).

9.2.67 Discarded messages

Basic process to notify discarded messages to the original sender.

This is done by the scheme manager's CRE.

The CRE gathers information about messages which ultimately could not be delivered to the intended receivers from the delivery queue.

The back-office system of the CCP, SO or PO receives the message information, registers it and supports the responsible staff by finding out the reason for the discarded messages.

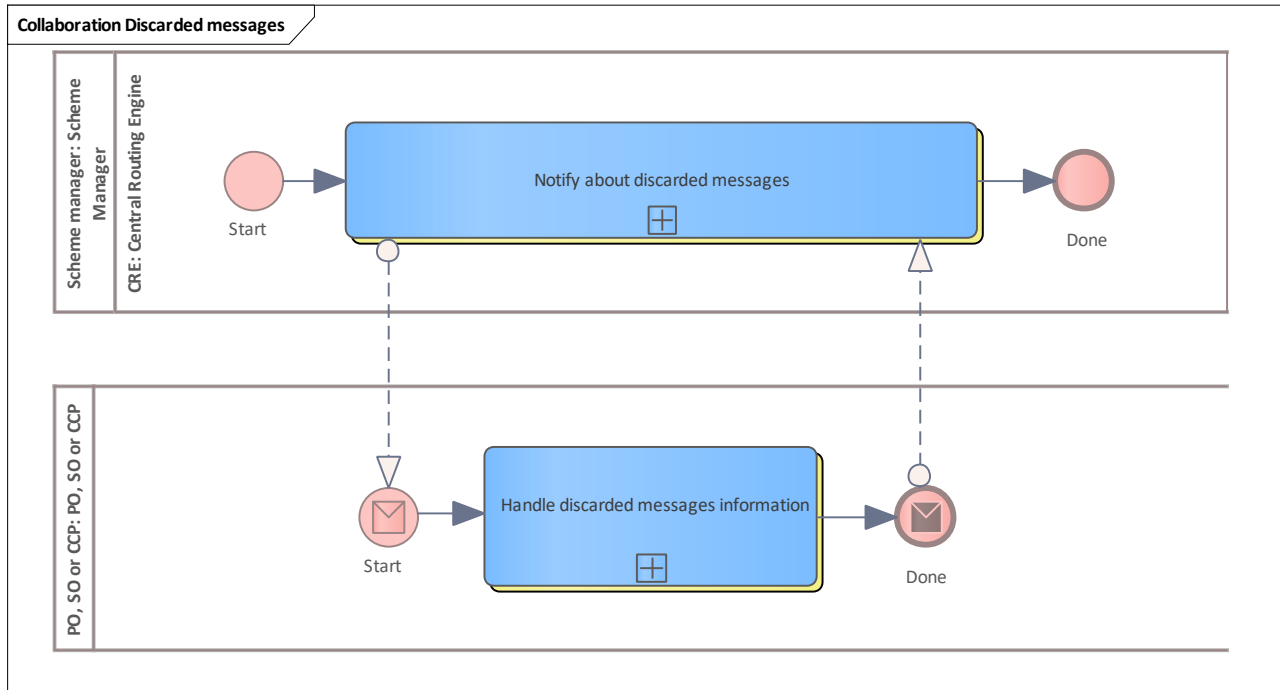


Figure 165: Discarded messages

9.2.67.1 PO, SO or CCP

See [PO, SO or CCP](#)

9.2.67.1.1 Handle discarded messages information

See [Handle discarded messages information](#).

9.2.67.2 Scheme manager

See [Scheme Manager](#) as CRE owner.

9.2.67.2.1 CRE

Lane for CRE

1.1.1.1.1.196 Notify about discarded messages

See [Notify about discarded messages](#).

9.2.68 Notify events

Small basic process which can be found at the end of a downstream monitoring process. The downstream monitoring detects a problem and the underlying back-office system of the CCP, SO or PO sends a warning plus optional information about the involved messages.

This information is sent to the CCP, SO or PO back-office system of the organisation that is potentially the originator of the problem. The back-office system registers the monitoring event and informs the responsible staff for clarification purposes.

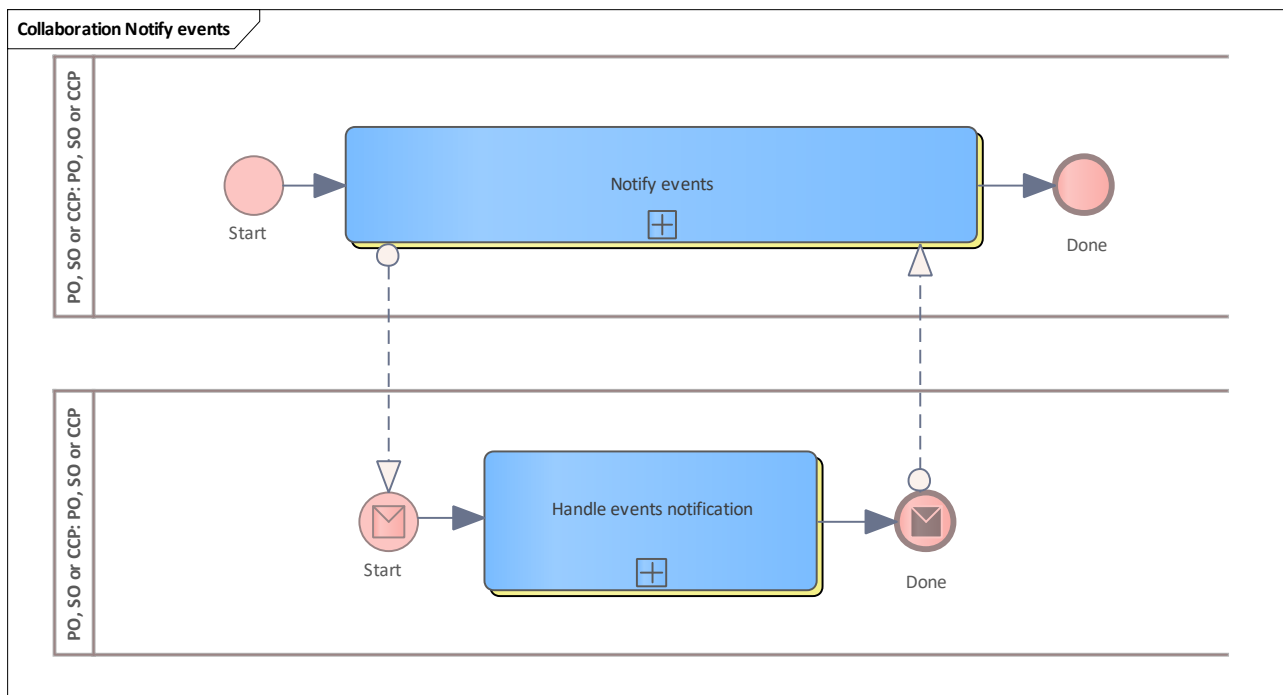


Figure 166: Notify events



9.2.68.1 PO, SO or CCP

See [PO, SO or CCP](#)

9.2.68.1.1 Notify events

See [Notify events](#).

9.2.68.2 PO, SO or CCP

See [PO, SO or CCP](#)

9.2.68.2.1 Handle events notification

See [Handle events notification](#).

9.2.69 Configurations

The following chapter contains all basic processes concerning the configuration of different system components.

9.2.69.1 Hotlist service and PO

The following chapter contains all basic processes concerning the external configuration of the hotlist service performed by the [Product Owner](#).

9.2.69.2 Add acceptance entry to hotlist configuration

Basic process between the PO and the hotlist service.

The PO configures the products which have to be accepted by a CCP or SO. This impacts the filtering of the entitlement hotlist for these organisations.

A CCP/SO gets only hotlist entries for products that must be supported.

The hotlist service receives the request and updates its configuration. Since the hotlists are organised in cycles, the new configuration impacts the hotlist filtering of the next cycle, not the current cycle.

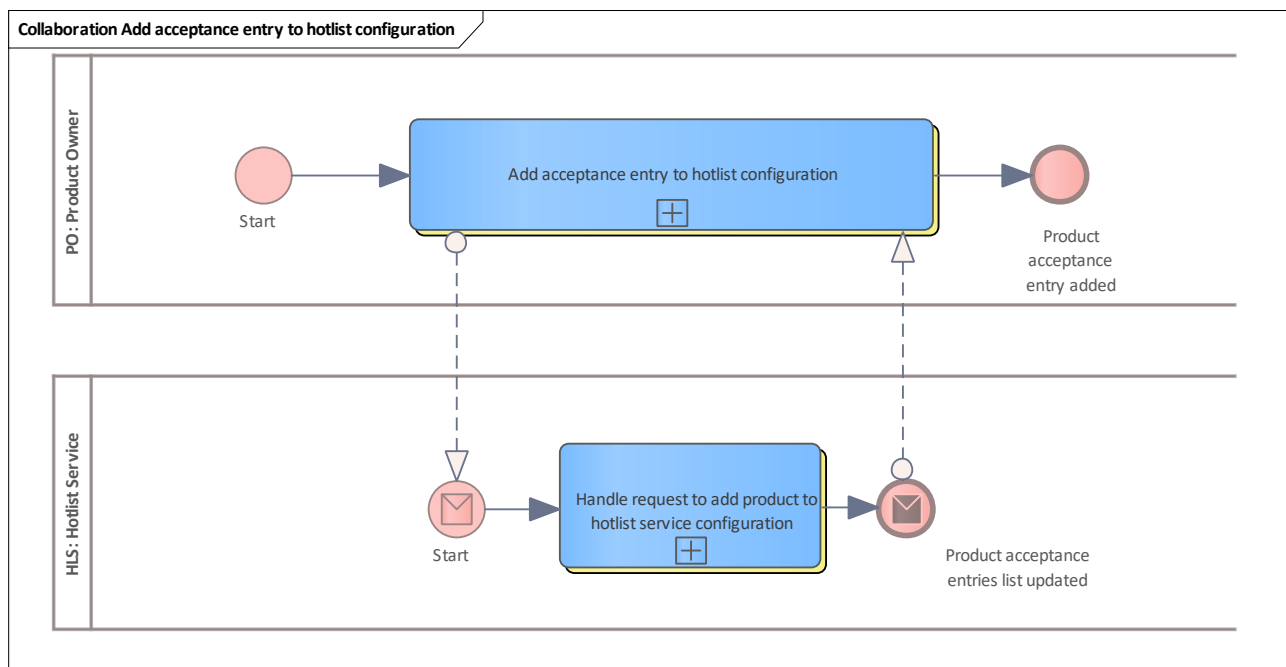


Figure 167: Add acceptance entry to hotlist configuration

9.2.69.2.1 PO

See [Product Owner](#)

1.1.1.1.1.197 Add acceptance entry to hotlist configuration

See [Add product acceptance entry to hotlist configuration](#).

9.2.69.2.2 HLS

See [Hotlist Service](#)

1.1.1.1.1.198 Handle request to add product to hotlist service configuration

See [Handle request to add product acceptance entry to hotlist service configuration](#).

9.2.69.3 Remove acceptance entry from hotlist configuration

Basic process between the PO and the hotlist service.

The PO configures the products which have to be removed from the acceptance list for a CCP or SO. This impacts the filtering of the entitlement hotlist for these organisations.

A CCP/SO gets only hotlist entries for products that must be supported.

The hotlist service receives the request and updates its configuration. Since the hotlists are organised in cycles, the new configuration impacts the hotlist filtering of the next cycle, not the current cycle.

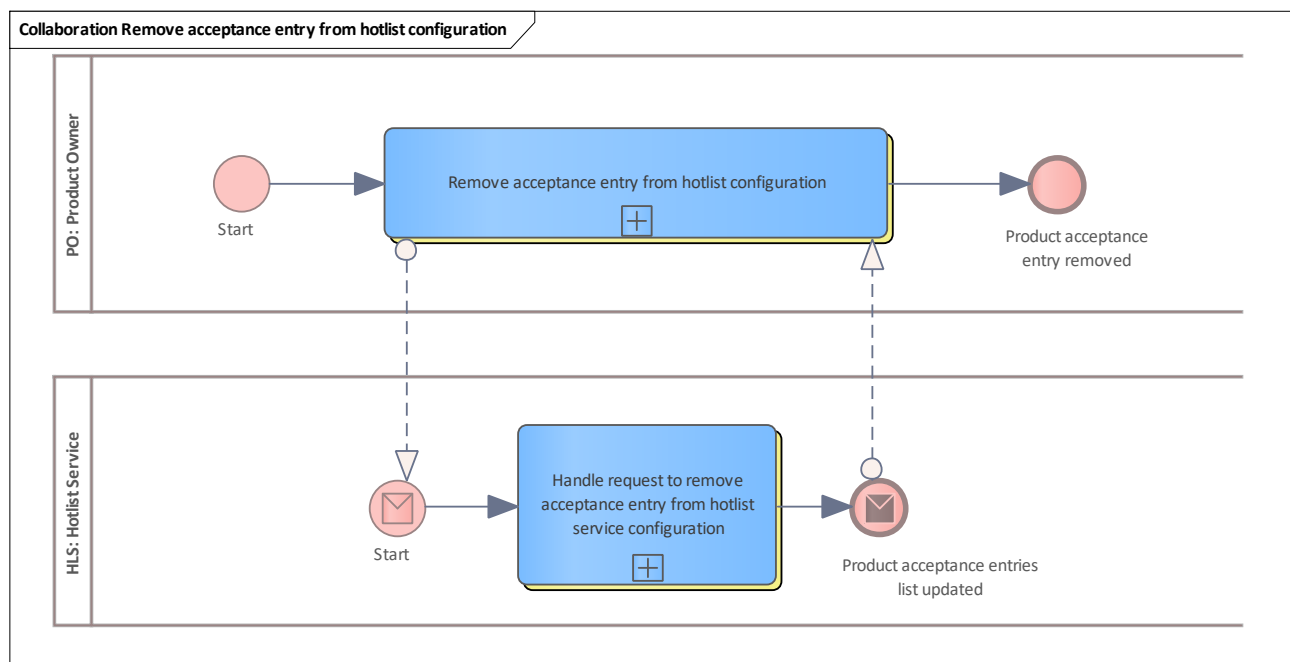


Figure 168: Remove acceptance entry from hotlist configuration

9.2.69.3.1 PO

See [Product Owner](#)

1.1.1.1.1.199 Remove acceptance entry from hotlist configuration

See [Remove product acceptance entry from hotlist configuration](#).

9.2.69.3.2 HLS

See [Hotlist Service](#)

1.1.1.1.1.200 Handle request to remove acceptance entry from hotlist service configuration

See [Handle request to remove product acceptance entry from hotlist service configuration](#).

9.2.69.4 Remove product acceptance from participants

Basic process between the PO and the hotlist service.

The PO configures the products which have to be accepted by a CCP or SO. If a product is no longer valid or supported, this product can be removed from all accepting organisations with one request. This impacts the filtering of the entitlement hotlist for these organisations.

A CCP/SO gets only hotlist entries for products that must be supported.

The hotlist service receives the request and removes the product from all organisations in the PO's acceptance list. Since the hotlists are organised in cycles, the new configuration impacts the hotlist filtering of the next cycle, not the current cycle.

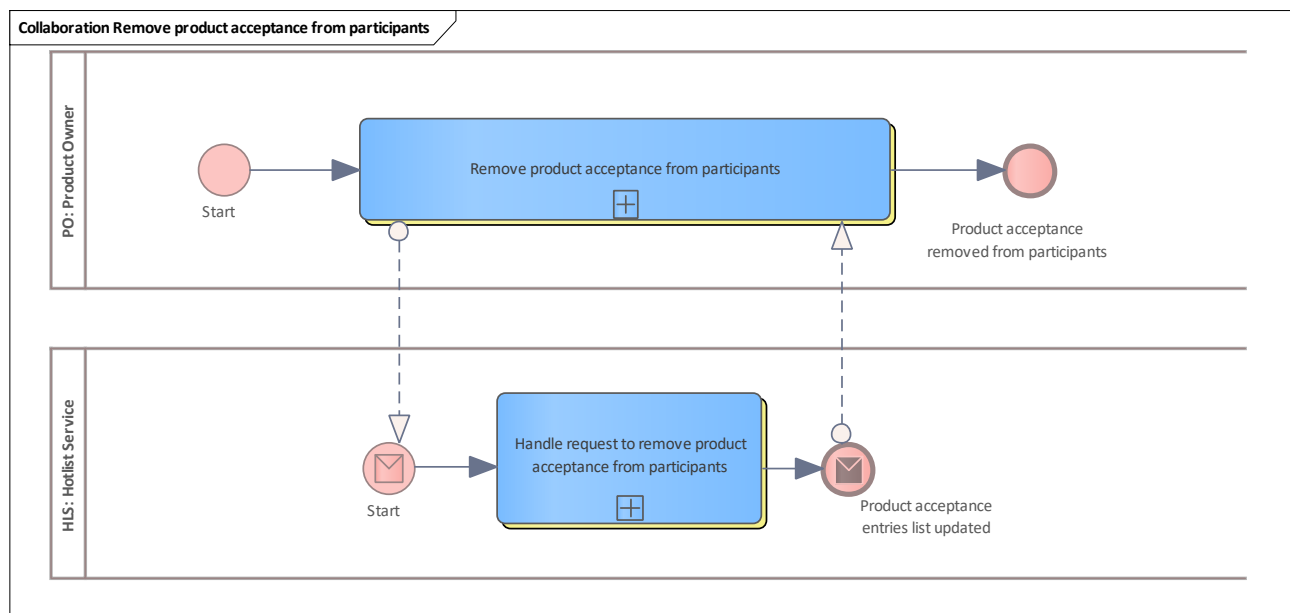


Figure 169: Remove product acceptance from participants

9.2.69.4.1 PO

See [Product Owner](#)

1.1.1.1.1.201 Remove product acceptance from participants

See [Remove product acceptance from participants](#).

9.2.69.4.2 HLS

See [Hotlist Service](#)

1.1.1.1.1.202 Handle request to remove product acceptance from participants

See [Handle request to remove product acceptance from participants](#).

9.2.69.5 Retrieve product acceptance configuration list

Basic process between the PO and the hotlist service.

The PO configures the products which have to be accepted by a CCP or SO.

This configuration list can be requested for verification purposes.

The hotlist service receives the request and returns the list of acceptance candidates together with the configured products to the PO.

Note: this list is valid for the current hotlist cycle. Pending configuration changes for the next hotlist cycle will not be contained in the configuration list.

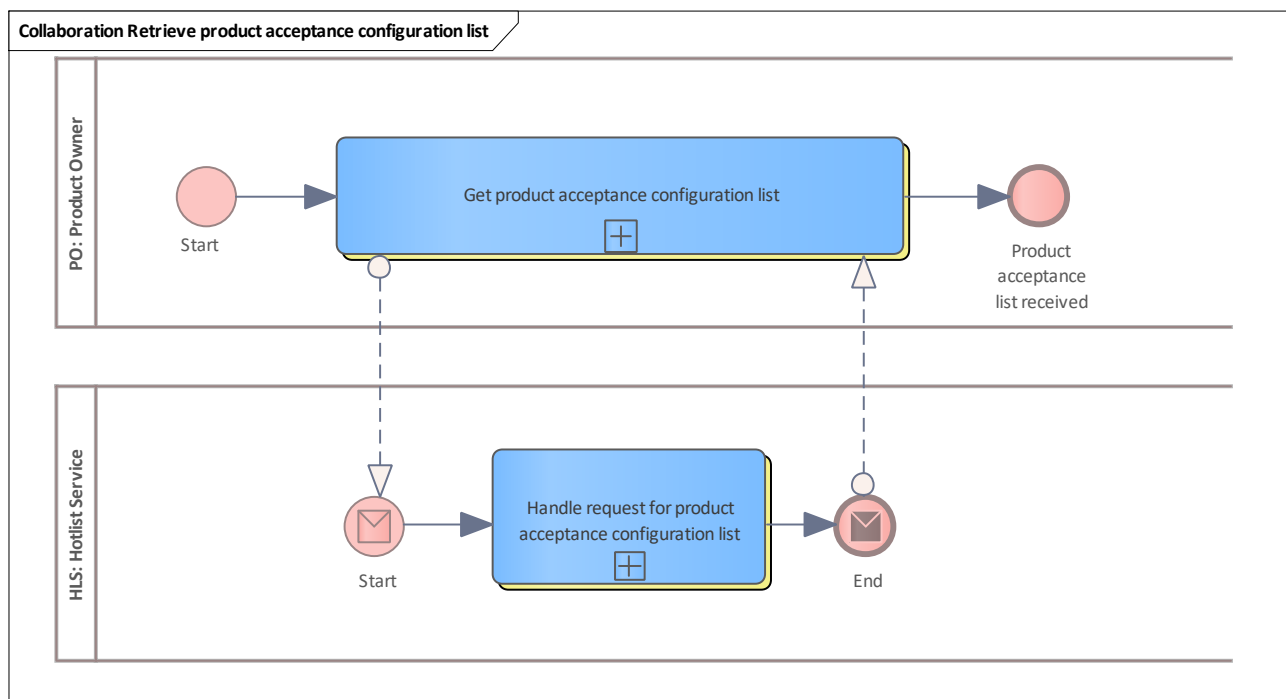


Figure 170: Retrieve product acceptance configuration list

9.2.69.5.1 PO

See [Product Owner](#)

1.1.1.1.1.203 Get product acceptance configuration list

See [Get product acceptance configuration list](#).

9.2.69.5.2 HLS

See [Hotlist Service](#)

1.1.1.1.1.204 Handle request for product acceptance configuration list

See [Handle request for product acceptance configuration list](#).

9.2.70 Distribute tariff modules

This basic process describes the distribution of tariff modules. These are defined by the product owner. The form is optional and can, for example, be in the format of the tariff modules according to PKM. The type of distribution from the product owner to the service operator or customer contract partner, as well as the location of provision, are also not part of the (((etiCORE specification.

However, the distribution of tariff data from the back-office systems to the terminals is specified in (((etiCORE.

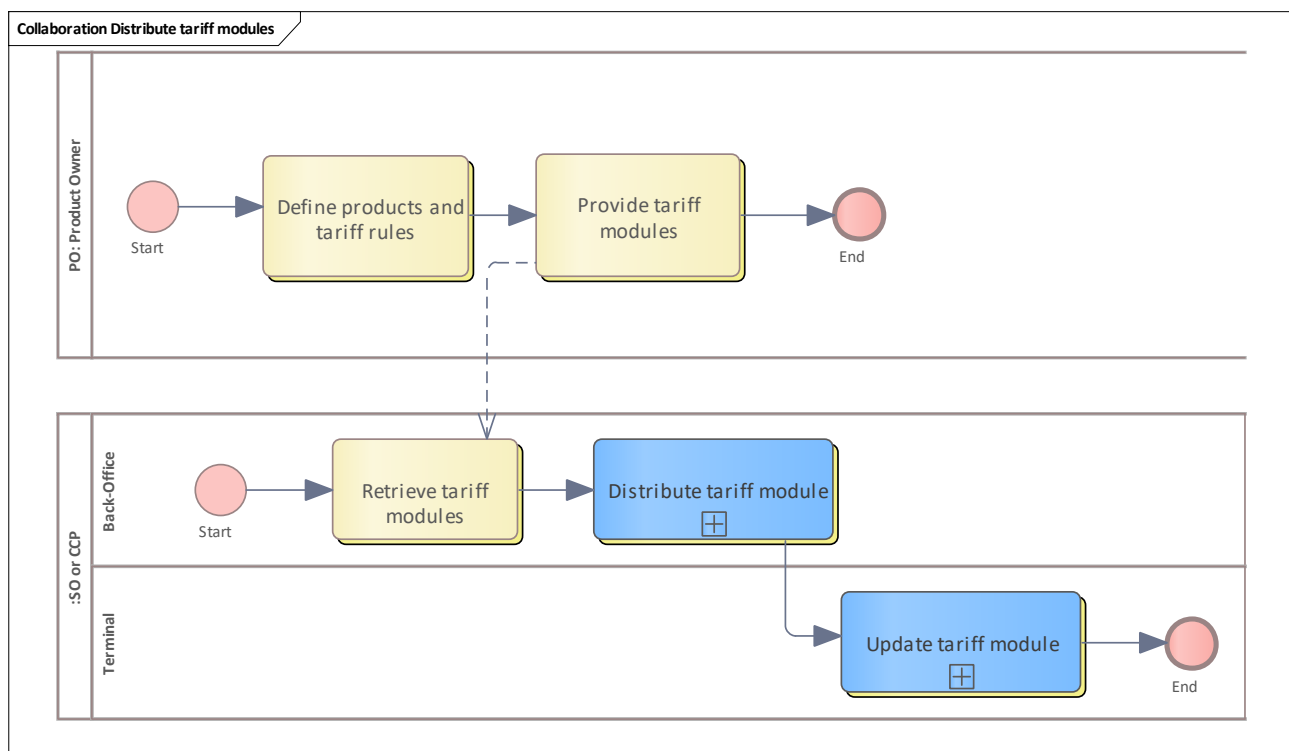


Figure 171: Distribute tariff modules

9.2.71 Set a service as available for a participant

Basic process between the scheme manager's CRE and all technical participants that support asynchronous use cases (CCP, SO and PO).

Each participant has one or more organisational units (organisation + role), and each organisational unit is split into different services.

For each service (organisation, role, service name), the participant has to announce whether the service is available or not.

Per default, the service is configured as "unavailable" and has to be changed to "available" using this basic process.

The CRE receives the request and switches the passed service to "available" which allows messages to be sent to this service from now on.

Note: participants must configure their services via the ESH in a preparatory step

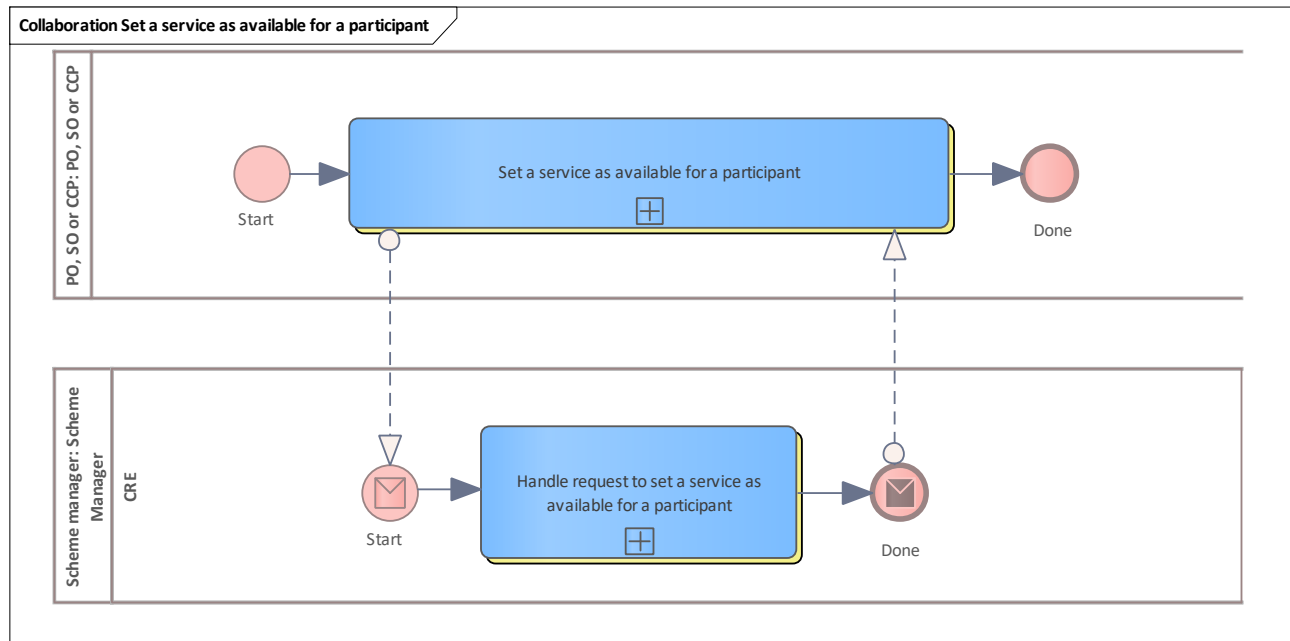


Figure 172: Set a service as available for a participant

9.2.71.1 PO, SO or CCP

See [PO, SO or CCP](#)

9.2.71.1.1 Set a service as available for a participant

See [Set a service as available for participant](#).

9.2.71.2 Scheme manager

See [Scheme Manager](#)

9.2.71.2.1 CRE

Lane for CRE

1.1.1.1.1.205 Handle request to set a service as available for a participant

See [Handle request to set a service as available for a participant](#).

9.2.72 Set a service as unavailable for a participant

Basic process between the scheme manager's CRE and all technical participants that support asynchronous use cases (CCP, SO and PO).

Each participant has one or more organisational units (organisation + role), and each organisational unit is split into different services.

For each service (organisation, role, service name), the participant has to announce whether the service is available or not.

For maintenance or other reasons, a service can be notified as being "*unavailable*" temporarily using this basic process.

The CRE receives the request and switches the passed service to "*unavailable*", which causes the queuing of messages for this service from now on.

Note: participants must configure their services via the ESH in a preparatory step.

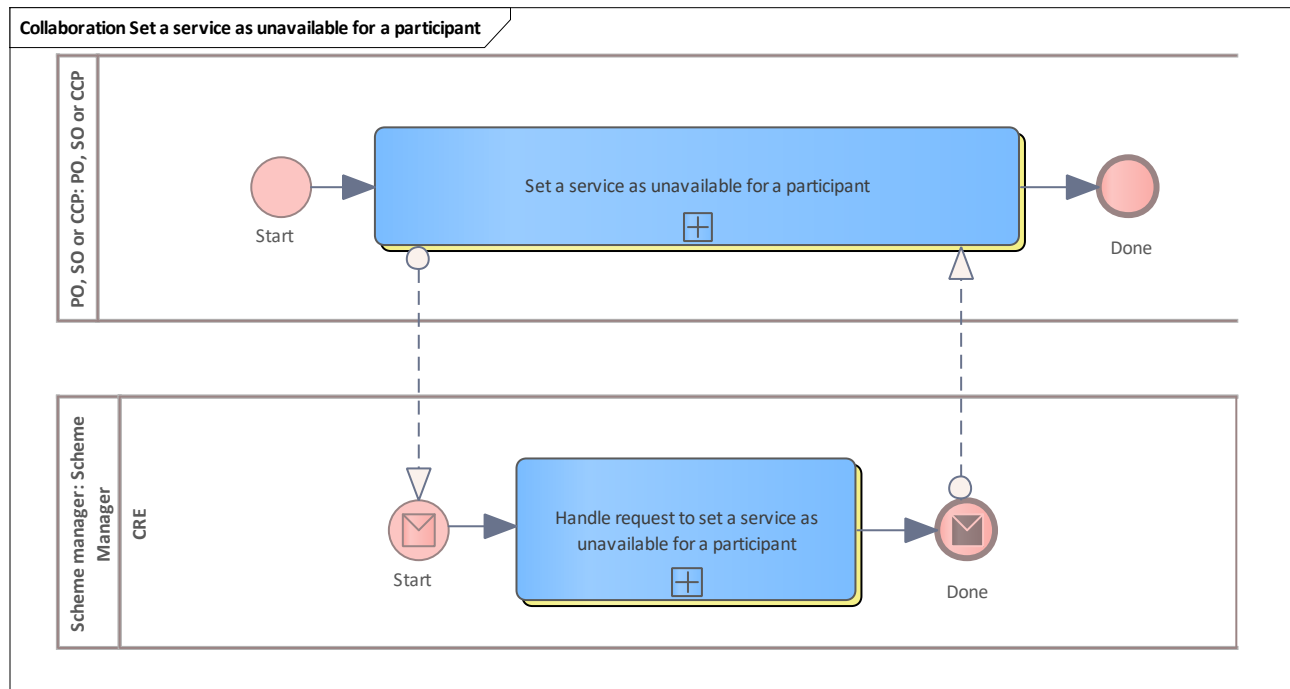


Figure 173: Set a service as unavailable for a participant

9.2.72.1 PO, SO or CCP

See [PO, SO or CCP](#)

9.2.72.1.1 Set a service as unavailable for a participant

See [Set a service as unavailable for a participant](#).

9.2.72.2 Scheme manager

See [Scheme Manager](#)

9.2.72.2.1 CRE

Lane for CRE

1.1.1.1.1.206 Handle request to set a service as unavailable for a participant

See [Handle request to set a service as unavailable for a participant](#).

9.2.73 Retrieve organisation list

Basic process between the scheme manager and CCP, SO, PO and hotlist service.

The list contains further information about the organisations, their IDs, names and related roles. This list is not organised in cycles. If a new organisation is configured, the next list request contains this information.

The back-office system of a CCP, SO, PO or hotlist service requests the list.

The scheme manager composes and returns the current list. This list has to be stored in the back-office system and, in case of a terminal operator (SO or CCP), distributed to the terminals. This list has to be fetched regularly once per day.

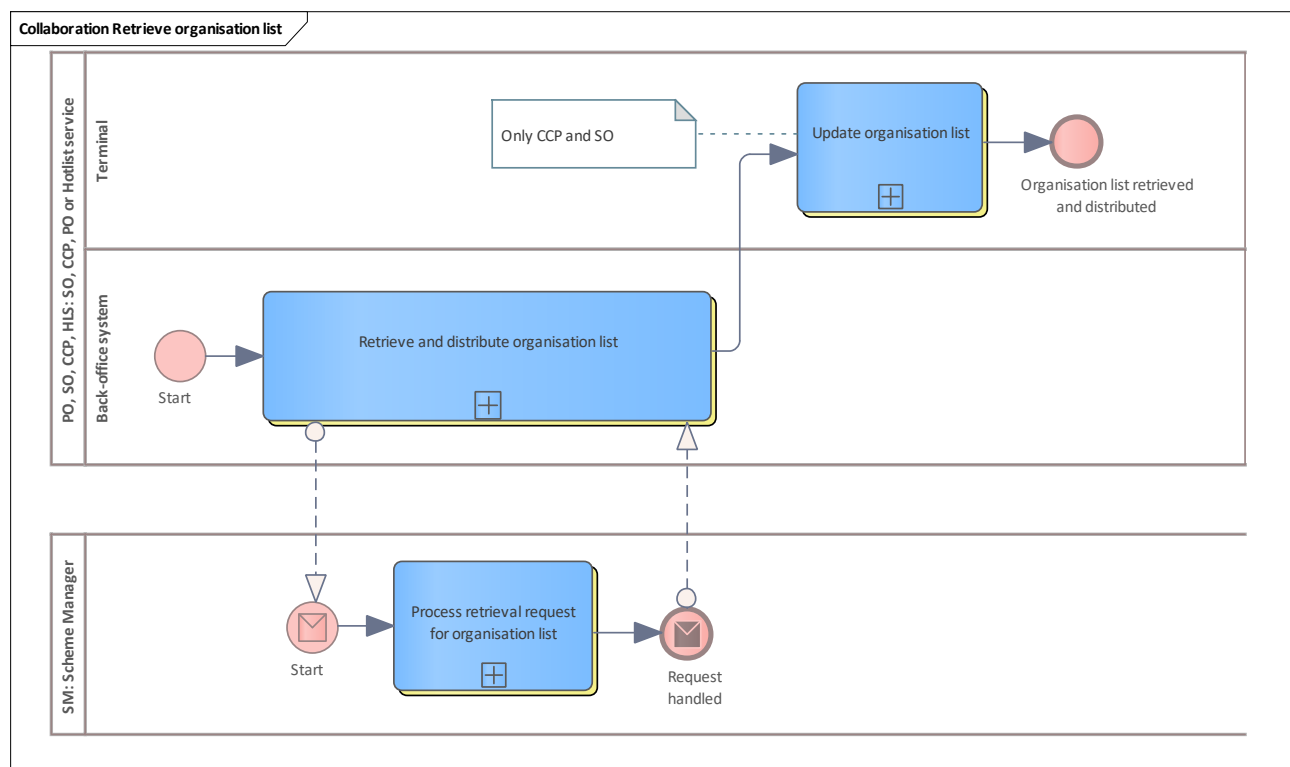


Figure 174: Retrieve organisation list

9.2.73.1 PO, SO, CCP, HLS

See [SO, CCP, PO or Hotlist service](#)

9.2.73.1.1 Back-office system

Lane for back-office system

1.1.1.1.1.207 Retrieve and distribute organisation list

See [Retrieve and distribute organisation list](#).

9.2.73.1.2 Terminal

Lane for terminal

1.1.1.1.1.208 Update organisation list

See [Update organisation list](#).

9.2.74 Retrieve the CA certificate repository

Basic process between the scheme manager's media PKI and CCP, SO and PO.

In order to verify the certification chain e.g. during a signature check, the CA certificate repository is needed, containing Root and Sub-CA(s).

The back-office system of a CCP, SO or PO forms an appropriate LDAP search request and sends it to the media PKI.

The received CA certificate repository has to be stored in the back-office system and, in case of a terminal operator (SO or CCP), distributed to the terminals.

This repository has to be fetched and update regularly once per day.

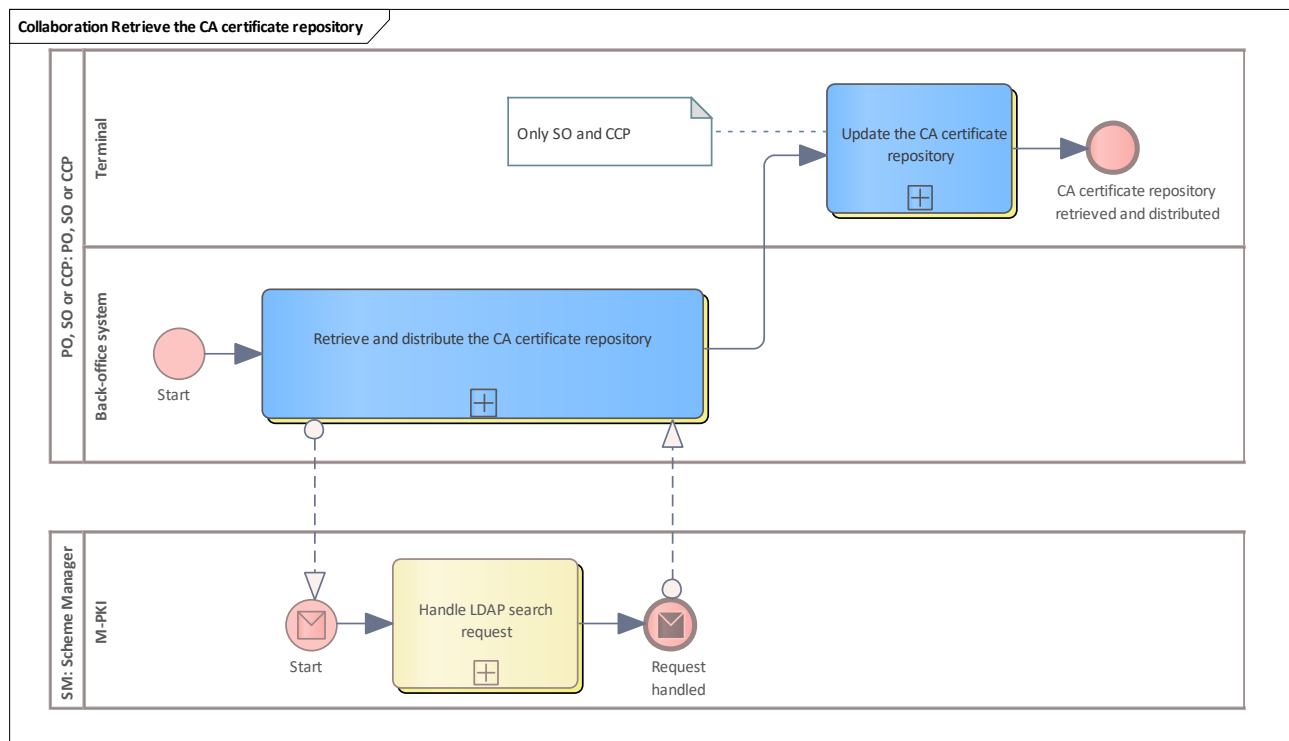


Figure 175: Retrieve the CA certificate repository

9.2.74.1 PO, SO or CCP

See [PO, SO or CCP](#)

9.2.74.1.1 Back-office system

Lane for back-office system

1.1.1.1.1.209 Retrieve and distribute the CA certificate repository

See [Retrieve and distribute the CA certificate repository](#).

9.2.74.1.2 Terminal

Lane for terminal

1.1.1.1.1.210 Update the CA certificate repository

See [Update the CA certificate repository](#).

9.2.75 Retrieve the CV certificate revocation list

Basic process between the scheme manager's media PKI and CCP, SO and PO.

In order to find out, if a Root- or Sub-CA or SAM activation certificate is on the certificate revocation list (CRL), this list has to be received and distributed.

The back-office system of a CCP, SO or PO forms a appropriate LDAP search request and sends it to the media PKI.

The received CRL has to be stored in the back-office system and, in the case of a terminal operator (SO or CCP), distributed to the terminals.

This list has to be fetched and updated regularly once per day.

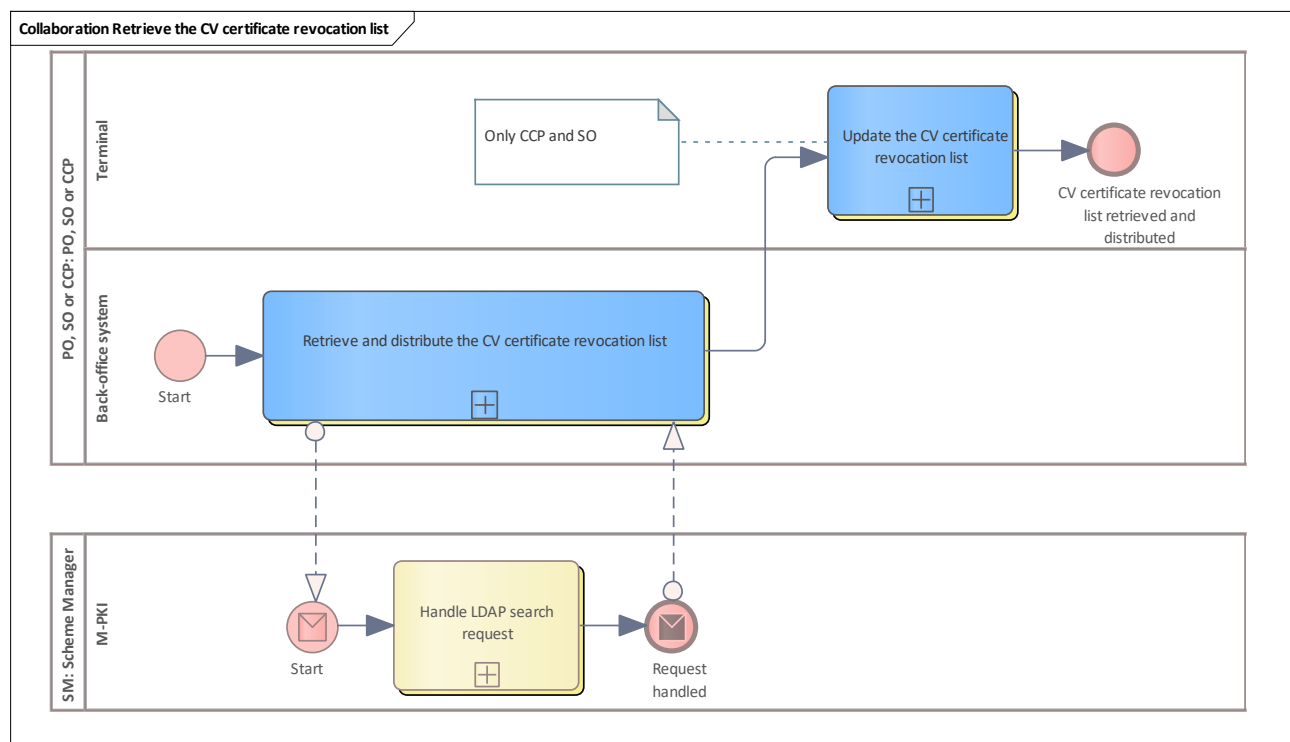


Figure 176: Retrieve the CV certificate revocation list



9.2.75.1 PO, SO or CCP

See [PO, SO or CCP](#)

9.2.75.1.1 Back-office system

Lane for back-office system

1.1.1.1.1.211 Retrieve and distribute the CV certificate revocation list

See [Retrieve and distribute the CV certificate revocation list](#).

9.2.75.1.2 Terminal

Lane for terminal

1.1.1.1.1.212 Update the CV certificate revocation list

See [Update the CV certificate revocation list](#).

9.2.76 SAM

The following chapter contains basic processes for the SAM administration.

9.2.76.1 Configuration

This chapter contains basic processes for the SAM configuration.

9.2.76.2 Individualise SAM

Basic process that shows the interaction between the [Scheme Manager](#) and the card manufacturer concerning SAM individualisation. The card manufacturer follows the [etiCORE Secure Application Module Individualisation Specification](#).

The Scheme Manager orders SAMs and their individualisation. The card manufacturer individualises SAMs and ships them to the Scheme Manager.

The Scheme Manager stores them in its warehouse and registers the received SAMs in the ESH.

The MMS system orders SAM individualisation per bulk or single processing.

The MMS checks the response data sent by the card manufacturer.

Note: individualisation means that the VDV-ETS SAM application is transferred to the SAM chip. The SAM application gets its instance ID that identifies the SAM. The individualisation is the prerequisite for a subsequent SAM configuration for a public transport company.

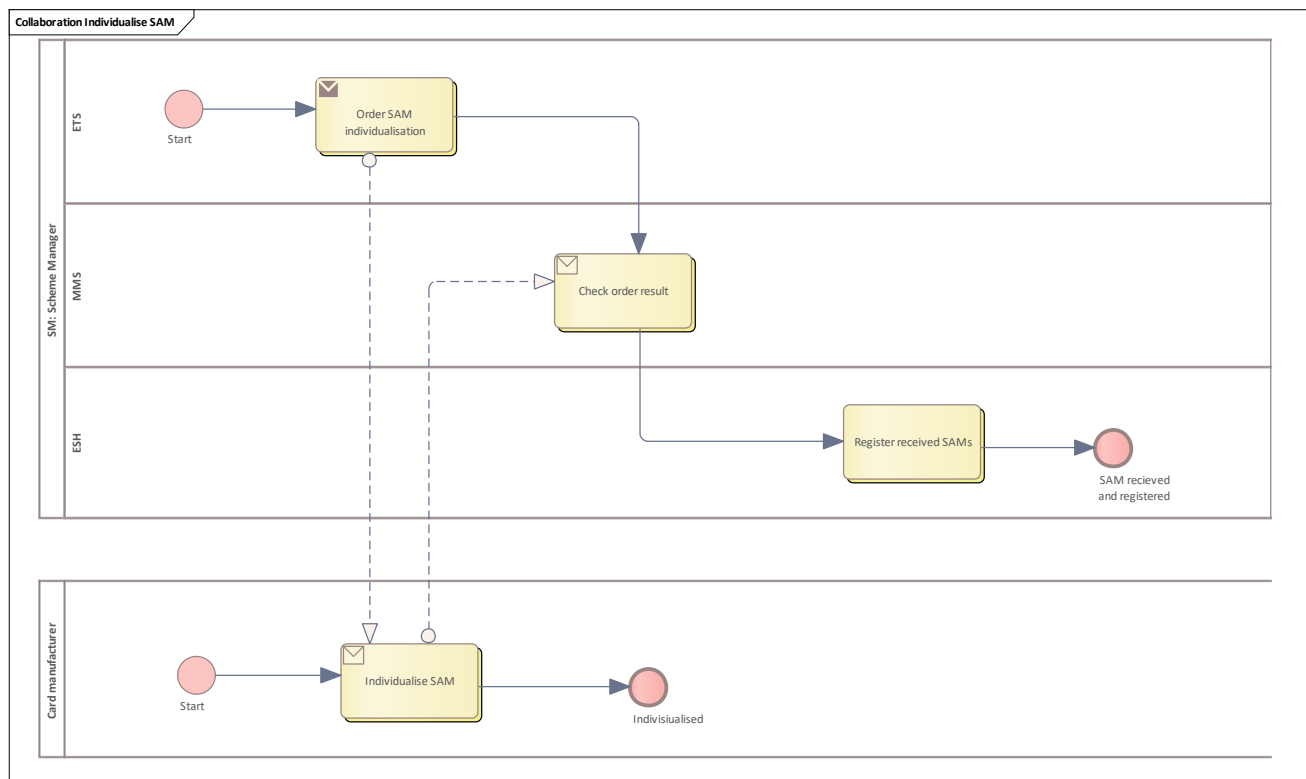


Figure 177: Individualise SAM

9.2.76.2.1 SM

See [Scheme Manager](#).



1.1.1.1.1.213 ETS

Lane for ETS

1.1.1.1.1.213.1 Order SAM individualisation

The SAM individualisation is ordered.

1.1.1.1.1.214 MMS

Lane for MMS

1.1.1.1.1.214.1 Check order result

The request and response files for ordering SAM individualisation are checked.

If the check is positive, the MMS informs the ESH about the SAM.

If the check is negative, a clarification case is open for that SAM.

1.1.1.1.1.215 ESH

Lane for ESH

1.1.1.1.1.215.1 Register received SAMs

The ESH receives information about the SAM ordered.

After the SAMs are physically located in the warehouse, they can be marked as "*in the warehouse*" in the system by submitting the SAM's app instance IDs (SAM-IDs).

9.2.76.2.2 Card manufacturer

The organisation which creates the SAMs for etiCORE.

1.1.1.1.1.216 Individualise SAM

SAMs are individualised by the card manufacturer:

- Configure Java card platform
- Load, install and make a selectable SAM applet
- Store individualisation data
- Check individualisation data

Individualised SAMs are shipped to the scheme manager.

9.2.76.3 Demand SAM configuration script

Basic process that shows the interaction between the later SAM owners (SO and CCP), the scheme manager system components and the PO for the product rights.

An SO or CCP orders SAMs. After receiving SAMs shipped by the Scheme Manager, they confirm the received SAMs in the ESH.

If required, an SO or CCP can submit a request for the update of SAM configuration data such as product issuance rights and SAM activation rights. The PO has to confirm the requested product rights.

The CCP or SO submits a request for creating a bulk configuration script. After that, the script can be downloaded in the ESH.

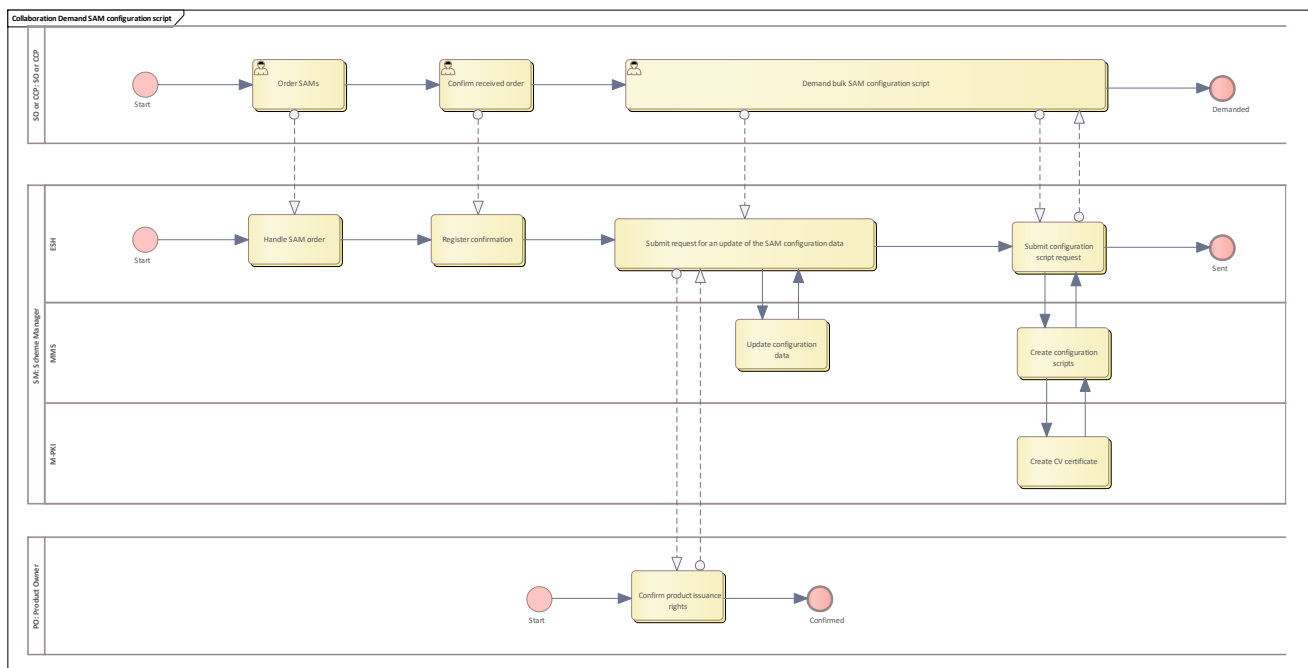


Figure 178: Demand SAM configuration script

9.2.76.3.1 SO or CCP

See [SO or CCP](#)

1.1.1.1.1.217 Order SAMs

SAMs are ordered by the SAM owner with the role of SO or CCP. Ordering a single SAM or SAMs in bulk does not any difference in the process flow.

1.1.1.1.1.218 Confirm received order

SAMs are received and the SAM owner confirms the receipt of SAMs.

1.1.1.1.1.219 Demand bulk SAM configuration script

In this process the following steps are performed:

- Step 1 - optional: CCP or SO can submit a request for an update of SAM configuration data. This should be executed only if it is required to update SAM configuration data such as product issuance rights and SAM activation rights. When submitting, the role and organisation for which the configuration is requested must be entered for each SAM.
- Step 2: CCP or SO submits a request for creating a configuration script.

9.2.76.3.2 SM

See [Scheme Manager](#)

1.1.1.1.1.220 ESH

Lane for ESH



1.1.1.1.1.220.1 Handle SAM order

SAM order is handled.

SAMs are shipped to the SAM owner. The status of SAM is changed to "sent to SAM owner"

1.1.1.1.1.220.2 Register confirmation

Register confirmation for SAM-IDs.

1.1.1.1.1.220.3 Submit request for an update of the SAM configuration data

Using the user interface of the ESH, it is possible to have product issuance rights and/or SAM activation rights updated with the submitted organisation and role for each SAM.

This can be requested by an organisation with the role of CCP or SO.

For the product issuance rights, approval is required from each product owner (that comes in question due to different products)

The CCP and SO submit a certification effective and expiry time for each SAM to be configured.

After checking the request, an acknowledgement or exception will be sent as a response.

1.1.1.1.1.220.4 Submit configuration script request

The request for creating configuration scripts is submitted by the CCP/SO in the ESH.

The ESH has the scripts created in the MMS. The ESH receives information from the MMS that the scripts for the SAM(s) have been created.

The ESH informs the contact person of SO/CCP that scripts can be downloaded. The contact person downloads the script.

1.1.1.1.1.221 MMS

Lane for MMS

1.1.1.1.1.221.1 Update configuration data

One or both of the data structures SAMActivationRights or ProductIssuanceRights will be updated for one or multiple SAMs in the Media Database.

1.1.1.1.1.221.2 Create configuration scripts

For each SAM-AppInstanceID an APDU command script is created in a file.

The MMS receives from PKI the certificate chain for SAM configuration and attaches it to the script.

1.1.1.1.1.222 M-PKI

Lane for media PKI

1.1.1.1.1.222.1 Create CV certificate

Create max 3 certificates of a certificate chain for SAM configuration.

9.2.76.3.3 PO

See [Product Owner](#).

1.1.1.1.1.223 Confirm product issuance rights

The product owner confirms the product issuance rights for the given organisation and SAMs.

9.2.76.4 Distribute and update SAM

Basic process that shows the script distribution and applying the configuration script to the SAMs.

The script that was ordered aided by [Demand SAM configuration script](#) and configures the SAMs for a certain public transport company as SAM owner by adding the organisation's certificates (that determine the validity period of the SAM), etc to each SAM.

The back-office system distributes the SAM configuration script to the terminals with SAMs requiring configuration updates. The SAM configuration is updated in the terminals.

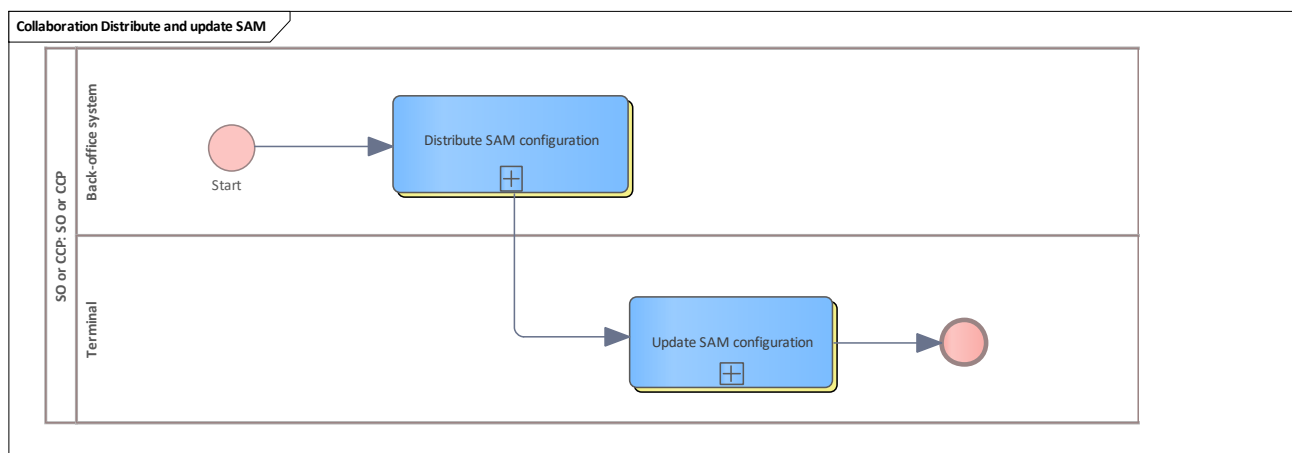


Figure 179: Distribute and update SAM

9.2.76.4.1 SO or CCP

See [SO or CCP](#)

1.1.1.1.1.224 Back-office system

Lane for a back-office system

1.1.1.1.1.224.1 Distribute SAM configuration

See [Distribute SAM configuration script](#).



1.1.1.1.1.225 Terminal

Lane for terminal

1.1.1.1.1.225.1 Update SAM configuration

See [Update SAM configuration](#).

9.2.76.5 Reset

This chapter contains basic processes for resetting a SAM.

9.2.76.6 Demand SAM reset script

Basic process that shows the interaction between the SAM owners (SO and CCP) and the scheme manager system components.

An SO or CCP demands a bulk SAM reset script for some of its SAMs.

The MMS creates the reset script. After that, the script can be downloaded in the ESH.

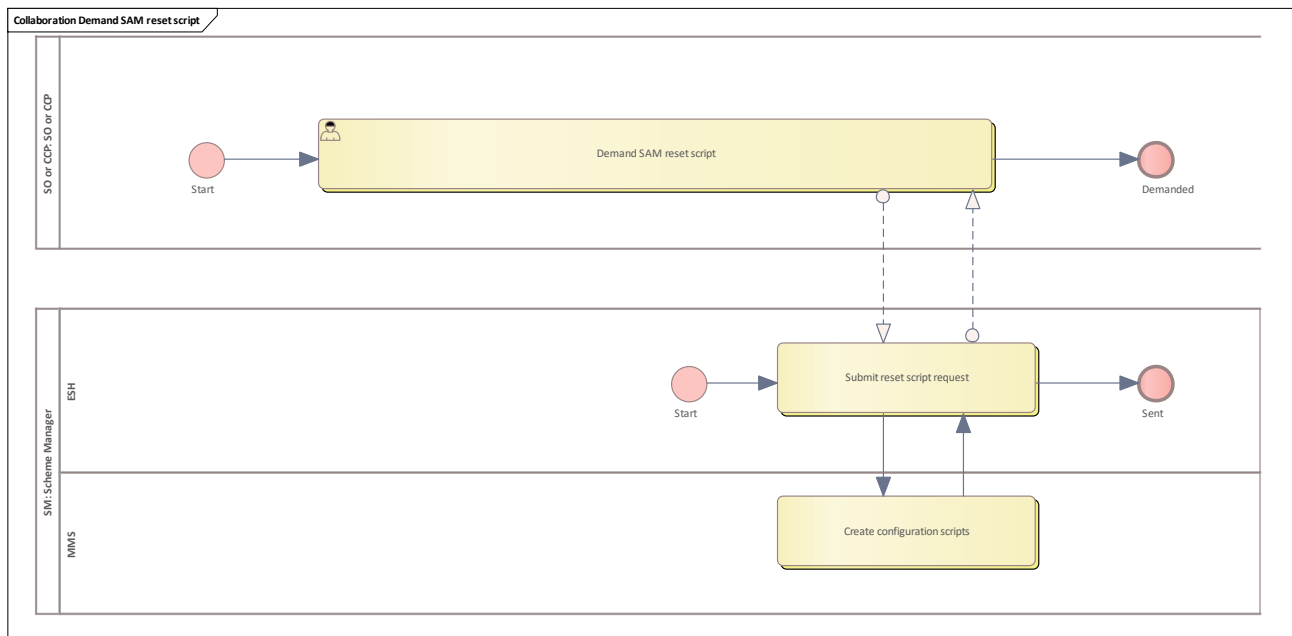


Figure 180: Demand SAM reset script

9.2.76.6.1 SO or CCP

see [SO or CCP](#)

1.1.1.1.1.226 Demand SAM reset script

The CCP or SO submits a request for creating a reset script for the selected SAMs.

9.2.76.6.2 SM

See [Scheme Manager](#)

1.1.1.1.1.227 ESH

Lane for ESH

1.1.1.1.1.227.1 Submit reset script request

The request for creating a reset script is submitted by the CCP/SO in the ESH.

The ESH has the scripts created in the MMS. The ESH receives information from the MMS that the scripts for SAM(s) have been created.

The ESH informs the contact person of the SO/CCP that scripts can be downloaded. The contact person downloads the scripts.

1.1.1.1.1.228 MMS

Lane for MMS

1.1.1.1.1.228.1 Create configuration scripts

For each selected SAM-AppInstanceID an APDU command script to reset is created in a file.

9.2.76.7 Distribute and reset SAM

Basic process that shows the script distribution and applying the reset script to the SAMs.

The script that was ordered aided by [Demand SAM reset script](#) and resets the SAMs for a certain public transport company as SAM owner by removing the organisation's certificates (that determine the validity period of the SAM), etc from each SAM. Thus, the SAM returns from the state "Configured" to the state "Individualised".

The back-office system distributes the SAM configuration script to the terminals with SAMs requiring a reset. The SAM configuration is updated in the terminals.

Note: the reset can be used also if problems with a re-configuration occur. In this case, an upstream reset might help.

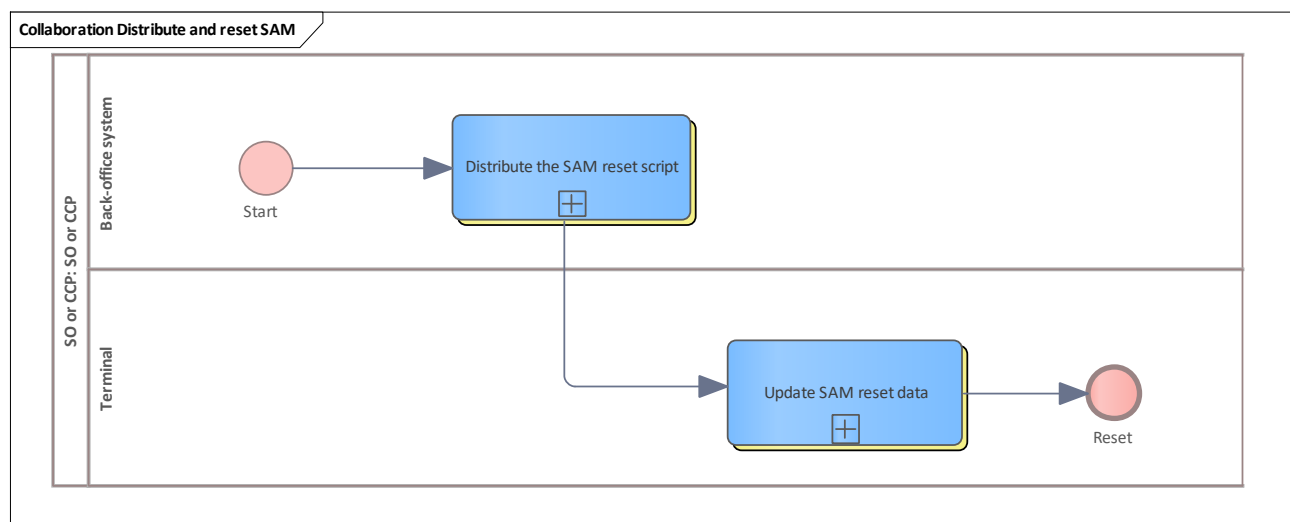


Figure 181: Distribute and reset SAM

9.2.76.7.1 SO or CCP

See [SO or CCP](#)

1.1.1.1.1.229 Back-office system

Lane for back-office system



1.1.1.1.1.229.1 Distribute the SAM reset script

See [Distribute the SAM reset script](#).

1.1.1.1.1.230 Terminal

Lane for terminal

1.1.1.1.1.230.1 Update SAM reset data

See [Update SAM reset data](#).

9.2.76.8 Look up SAM owner

This chapter contains a basic process for the lookup of a SAM owner.

9.2.76.9 Look up SAM owner

Small basic process between the ESH of the scheme manager and the CCP, SO, PO or hotlist service.

The ESH provides information about a SAM, especially the SAM owner.

This becomes important when the monitoring results in a hotlist demand for a (non-owned) SAM that must be sent to the responsible SAM owner.

The hotlist service needs this information to verify that a certain sender is authorised to add a SAM to the hotlist.

One of the participants above requests the ESH for the SAM owner by sending the SAM ID.

The ESH handles this request and does an internal lookup for the SAM owner. The information about the SAM's associated organisation ID and role is returned.

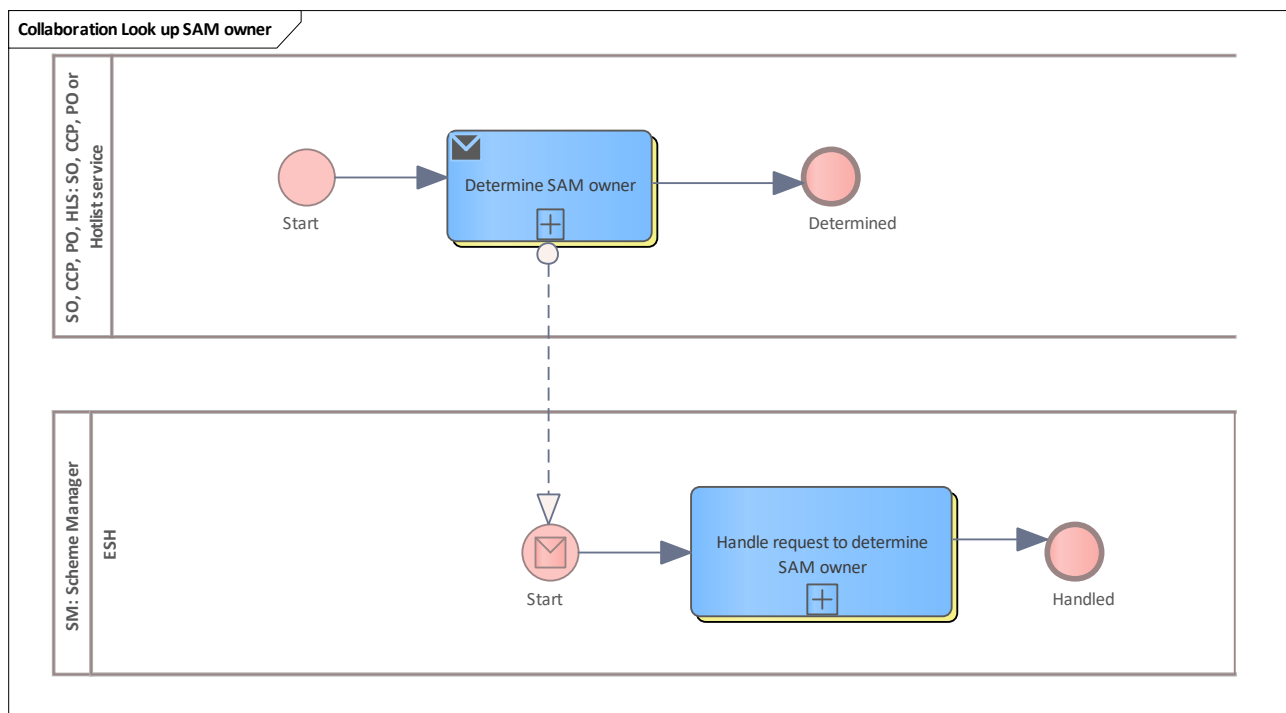


Figure 182: Look up SAM owner

9.2.76.9.1 SO, CCP, PO, HLS

See [SO, CCP, PO or Hotlist service](#)

1.1.1.1.1.231 Determine SAM owner

See [Determine SAM owner](#).

9.2.76.9.2 SM

See [Scheme Manager](#)



1.1.1.1.1.232 ESH

Lane for ESH

1.1.1.1.1.232.1 Handle request to determine SAM owner

See [Handle request to determine SAM owner](#).

9.2.77 User Medium

The following chapter contains basic processes for the user medium (UM) administration.

9.2.78 Individualise UM

Basic process that shows the interaction between the CCP, the scheme manager and the card manufacturer concerning UM individualisation.

If the CCP works with individualised UMs (depending on the card manufacturer it might also be possible to order configured UMs directly), the CCP orders these individualised UMs. The card manufacturer individualises the UMs and ships them to the CCP. During individualisation, the scheme manager's MMS (check data) and ESH (import medium IDs) are involved.

The MMS checks the individualisation request and response data sent by the card manufacturer.

Note: individualisation means, that the VDV-ETS UM application is transferred to the UM chip.

The UM application gets its instance ID that uniquely identifies the UM for etiCORE. The individualisation is the prerequisite for a subsequent UM configuration for a public transport company.

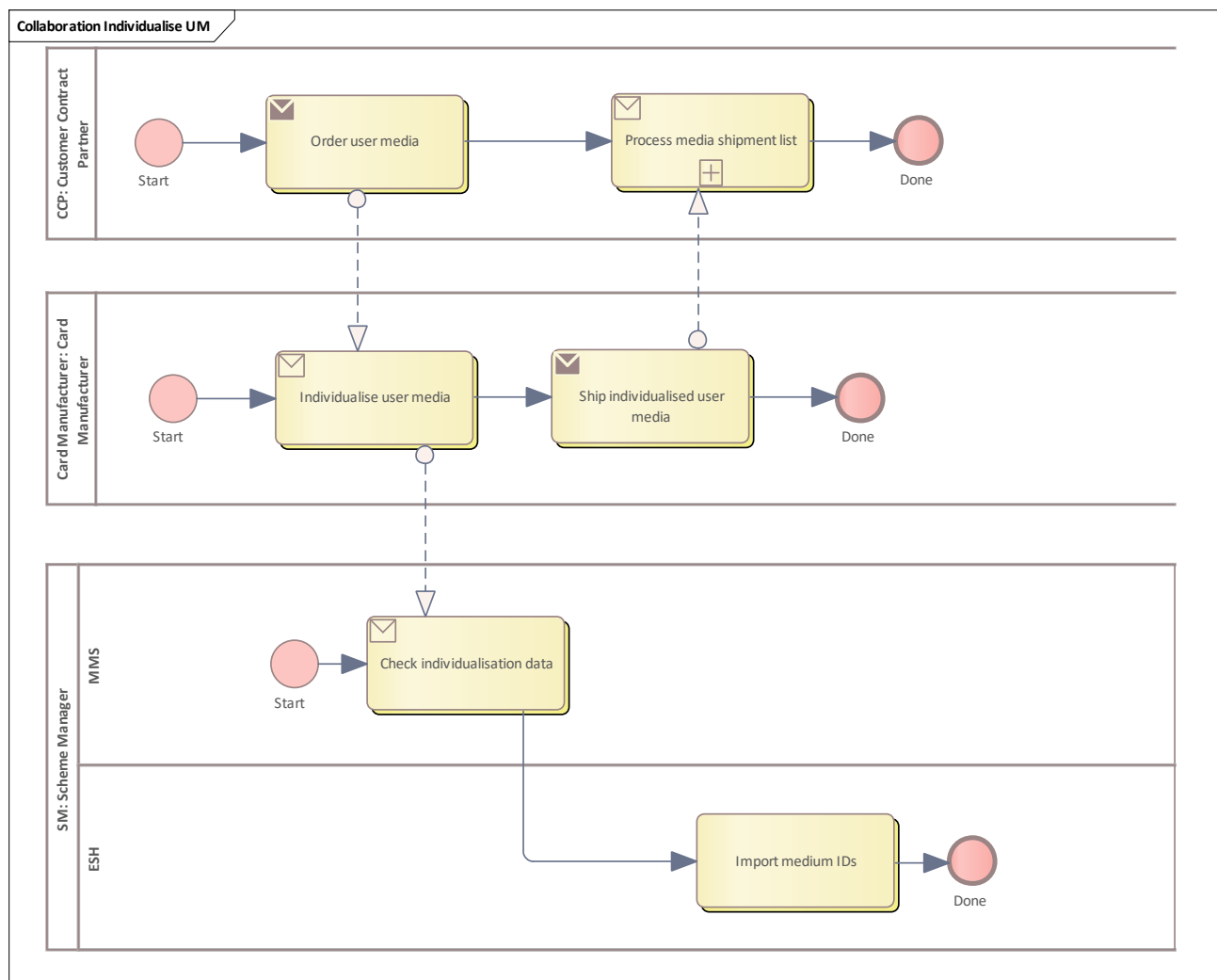


Figure 183: Individualise UM



9.2.78.1 CCP

See [Customer Contract Partner](#)

9.2.78.1.1 Order user media

The Customer Contract Partner orders the individualisation of user media.

9.2.78.1.2 Process media shipment list

The Customer Contract Partner processes the media shipment list.

9.2.78.2 Card Manufacturer

The organisation which creates the user media with the etiCORE application for chip cards.

9.2.78.2.1 Individualise user media

The card manufacturer individualises the user media.

9.2.78.2.2 Ship individualised user media

The card manufacturer ships the individualised user media to the customer contract partner. The card manufacturer transmits an accompanying media shipment list to the customer contract partner electronically.

9.2.78.3 SM

See [Scheme Manager](#)

9.2.78.3.1 MMS

Lane for MMS

1.1.1.1.1.233 Check individualisation data

The MMS checks and imports the user medium individualisation data.

9.2.78.3.2 ESH

Lane for ESH

1.1.1.1.1.234 Import medium IDs

The ESH imports the medium IDs and app instance IDs for the positively checked user media. This information is required for the use case [Determine UM app instance ID for Medium ID](#).

9.2.79 Configure UM

Basic process that describes the configuration of user media in the case that a mass personalisation is involved. Mass personalisation can be done by the card manufacturer or another trusted third party (as shown in the diagram).

The process starts with the CCP that orders configured user media from the mass personaliser. The mass personaliser orders the configuration scripts containing the matching certificates from the MMS (via the ESH) which requests the certificates from the media-PKI (M-PKI). The mass personaliser applies these scripts to its user media and ships the finished media to the CCP.

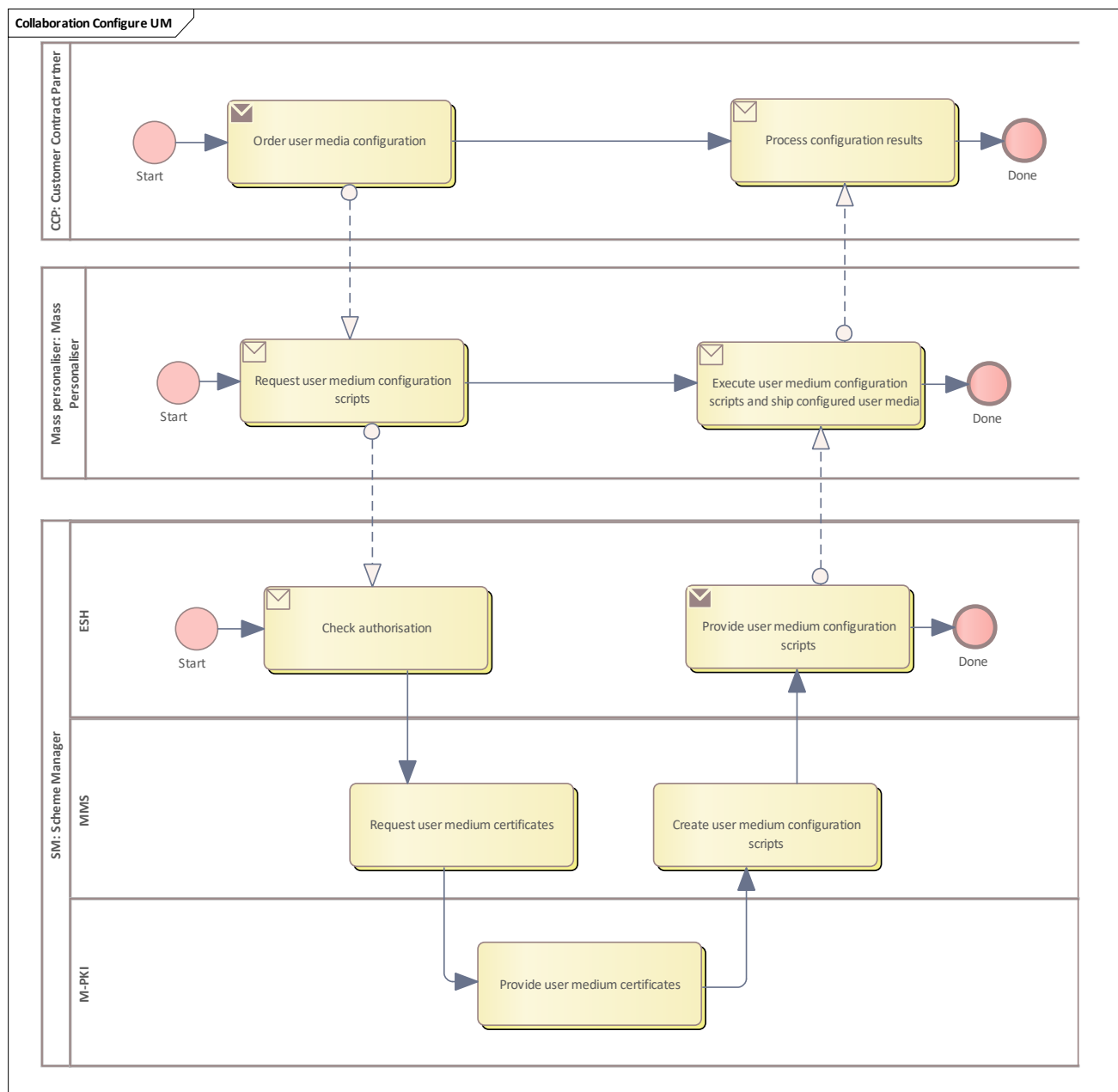


Figure 184: Configure UM

9.2.79.1 CCP

See [Customer Contract Partner](#)

9.2.79.1.1 Order user media configuration

Order user media including the configuration of them.

9.2.79.1.2 Process configuration results

Takes the configured user media and register them in the system.

9.2.79.2 Mass personaliser

The company that provides the mass personalisation of chip-based user media. Can be the customer contract partner itself or a card manufacturer.

9.2.79.2.1 Execute user medium configuration scripts and ship configured user media

The mass personaliser executes the script it received and applies the configuration data (including the certificates) to the user media in the scope of the current order.

9.2.79.2.2 Request user medium configuration scripts

The mass personaliser must order certificates matching the user media order of the CCP.

9.2.79.3 SM

See [Scheme Manager](#).

9.2.79.3.1 ESH

Lane for ESH

1.1.1.1.1.235 Check authorisation

The ESH checks the authorisation of the mass personaliser and forwards the configuration request to the MMS.

1.1.1.1.1.236 Provide user medium configuration scripts

The ESH provides the configuration script(s) for download.

9.2.79.3.2 MMS

Lane for MMS

1.1.1.1.1.237 Create user medium configuration scripts

The MMS creates suitable scripts embedding the received certificates. The scripts can be executed in a suitable infrastructure, in this process from the mass personaliser.

1.1.1.1.1.238 Request user medium certificates

The MMS requests the user medium certificates from the media-PKI.

9.2.79.3.3 M-PKI

Lane for media PKI

1.1.1.1.1.239 Provide user medium certificates

Provide the requested user medium certificates and deliver them back to the MMS.

9.3 Supporting Choreography Models

Contains supporting BPMN choreography models to distinguish between owned and non-owned entities or further things that have to be distinguished first. Normally, these small models are placed between the layer 1 and layer 2 processes. In these cases, the underlying layer 2 processes vary due to the (chosen) supporting choreography.

9.3.1 Sale

Contains helper BPMN choreography models for sale that allow distinguishing between "owned" (primary CCP itself) and "non-owned" (another, secondary CCP) scenarios.

9.3.2 Debit account-based payment method

For the sake of simplicity, the process of debiting an owned (terminal belongs to the issuer of the payment method) account-based payment method and the process of debiting a non-owned (terminal does not belong to the issuer of the payment method) account-based payment method are shown separately.

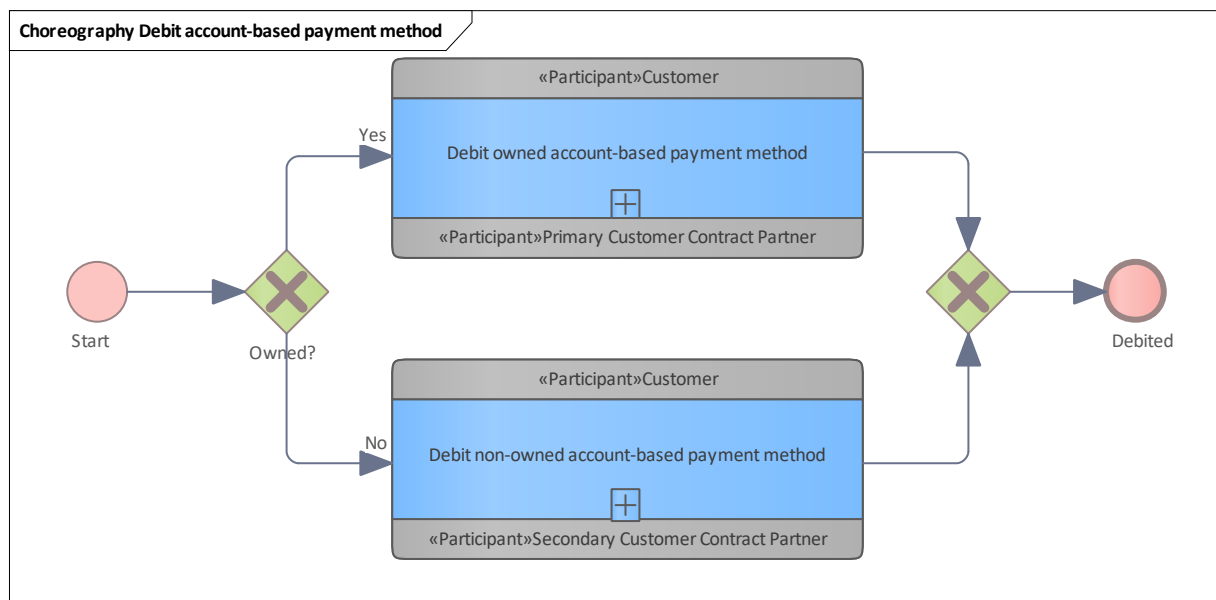


Figure 185: Debit account-based payment method

9.3.2.1 Debit non-owned account-based payment method

Debit an account-based payment method that was issued by someone other than the debiting party. See [Debit non-owned account-based payment method](#).

9.3.2.2 Debit owned account-based payment method

Debit an account-based payment method that was issued by the debiting party. See [Debit owned account-based payment method](#).

9.3.2.3 Owned?

Is the CCP the issuer of the account-based payment method?

9.3.3 Debit stored-value payment method

For the sake of simplicity, the process of debiting an owned (terminal belongs to the issuer of the payment method) stored-value payment method and the process of debiting a non-owned (terminal does not belong to the issuer of the payment method) stored-value payment method are shown separately.

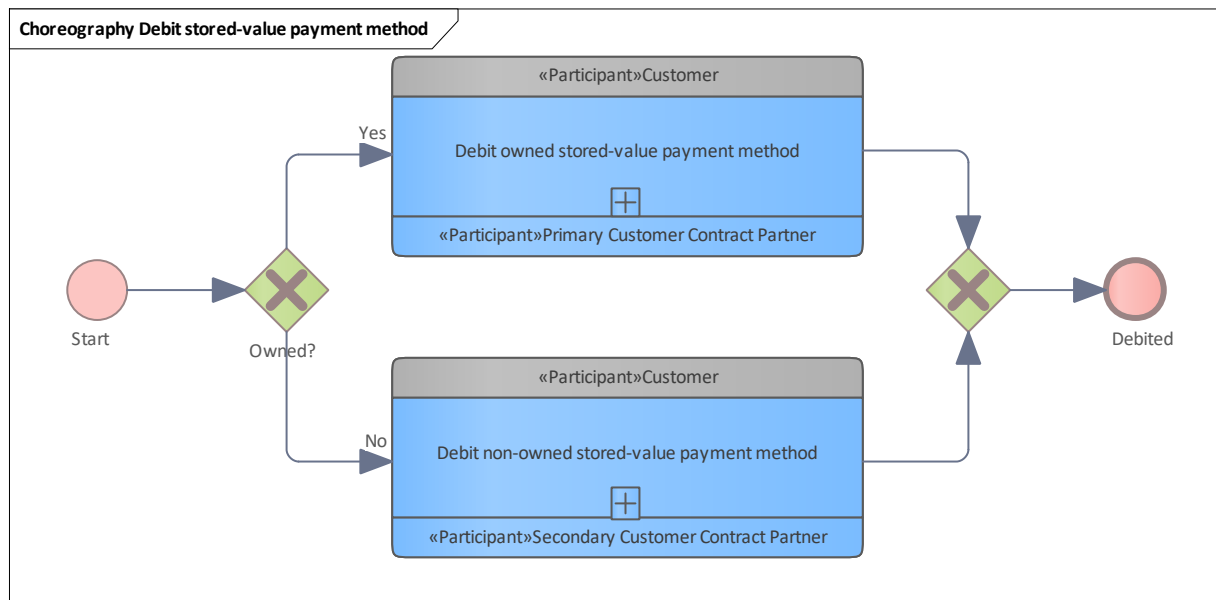


Figure 186: Debit stored-value payment method

9.3.3.1 Debit non-owned stored-value payment method

Debits an amount using a stored-value payment method that was issued by someone other than the debiting party. See [Debit non-owned stored-value payment method](#).

9.3.3.2 Debit owned stored-value payment method

Debits an amount using a stored-value payment method that was issued by the debiting party. See [Debit owned stored-value payment method](#).

9.3.3.3 Owned?

Is the CCP the issuer of the stored-value payment method?

9.3.4 Autoload stored-value payment method

For the sake of simplicity, the process of recharging an owned (terminal belongs to the issuer of the payment method) stored-value payment method and the process of recharging a non-

owned (terminal does not belong to the issuer of the payment method) stored-value payment method are shown separately.

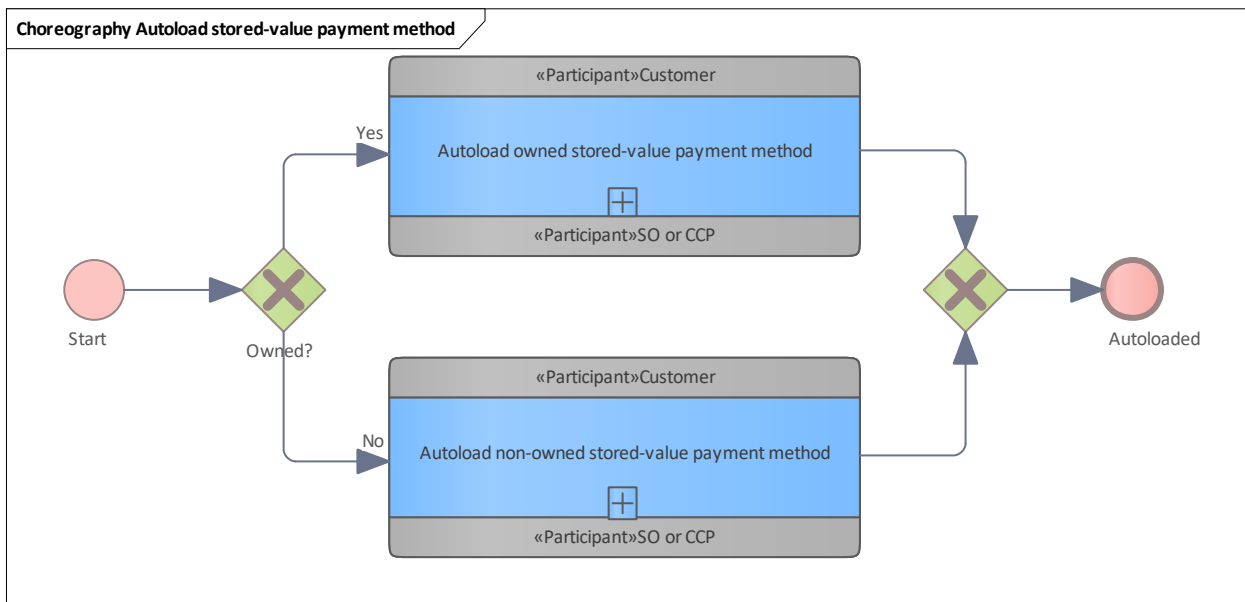


Figure 187: Autoload stored-value payment method

9.3.4.1 Autoload non-owned stored-value payment method

Autoload a stored-value payment method that was not issued by the autoloading party. See [Autoload non-owned stored-value payment method](#).

9.3.4.2 Autoload owned stored-value payment method

Autoload a stored-value payment method that was issued by the autoloading party. See [Autoload owned stored-value payment method](#).

9.3.4.3 Owned?

Is the CCP the issuer of the stored-value payment method?

9.3.5 Recharge stored-value payment method

For the sake of simplicity, the process of recharging an owned (terminal belongs to the issuer of the payment method) stored-value payment method and the process of recharging a non-owned (terminal does not belong to the issuer of the payment method) stored-value payment method are shown separately.

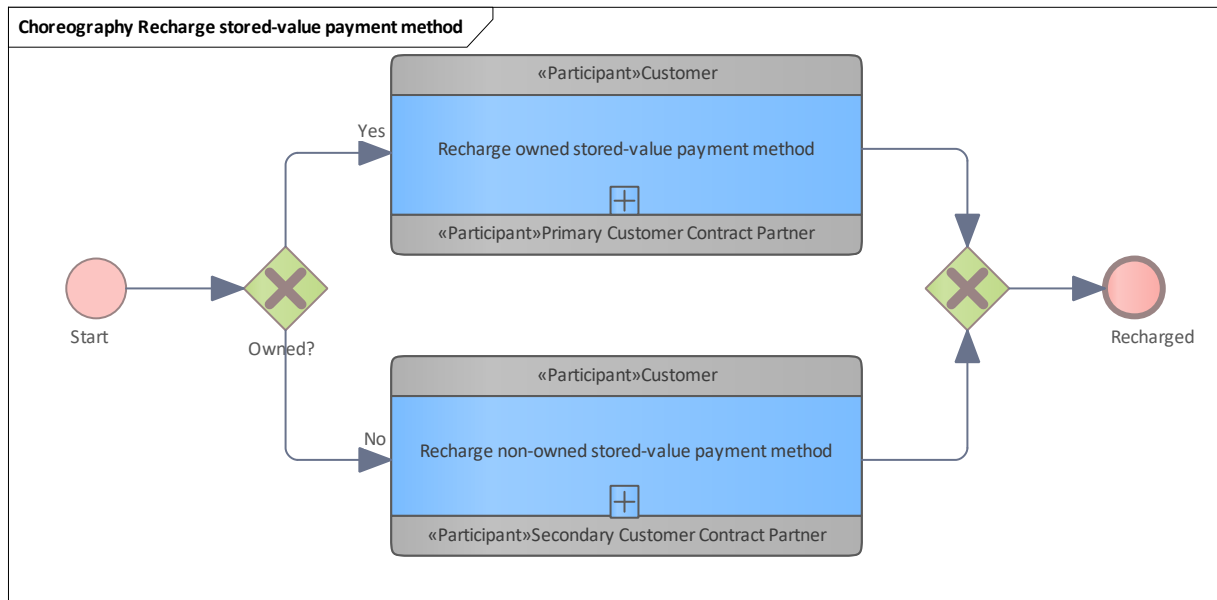


Figure 188: Recharge stored-value payment method

9.3.5.1 Recharge non-owned stored-value payment method

Recharge a stored-value payment method that was not issued by the recharging party. See [Recharge non-owned stored-value payment method](#).

9.3.5.2 Recharge owned stored-value payment method

Recharge a stored-value payment method that was issued by the recharging party. See [Recharge owned stored-value payment method](#).

9.3.5.3 Owned?

Is the CCP the issuer of the stored-value payment method?

9.3.6 Issue entitlement

For the sake of simplicity, the processes of issuing an entitlement either starting in a terminal or in a back-office system are shown separately.

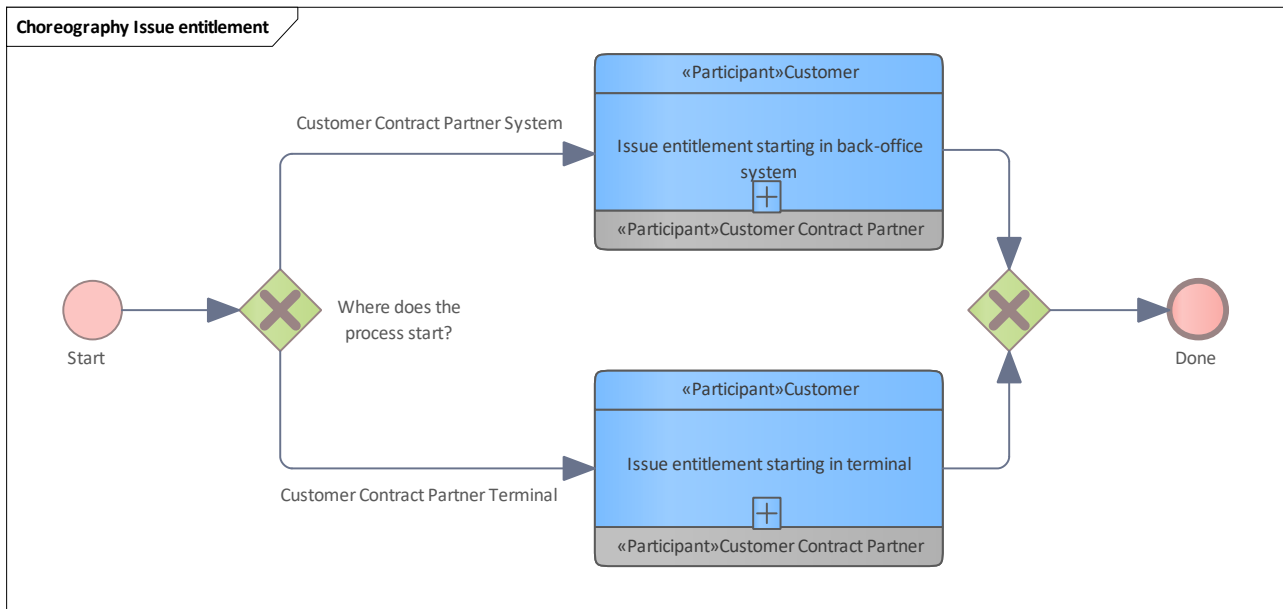


Figure 189: Issue entitlement

9.3.6.1 Issue entitlement starting in back-office system

See [Issue entitlement starting in back-office system](#).

9.3.6.2 Issue entitlement starting in terminal

See [Issue entitlement starting in terminal](#).

9.3.7 Hotlisting and blocking

Contains helper BPMN choreography models for hotlisting and blocking that allow distinguishing between "owned" (primary CCP itself) and "non-owned" (another, secondary CCP) scenarios.

9.3.8 Block hotlisted application

For the sake of simplicity, the blocking of owned and non-owned application instances are shown separately.

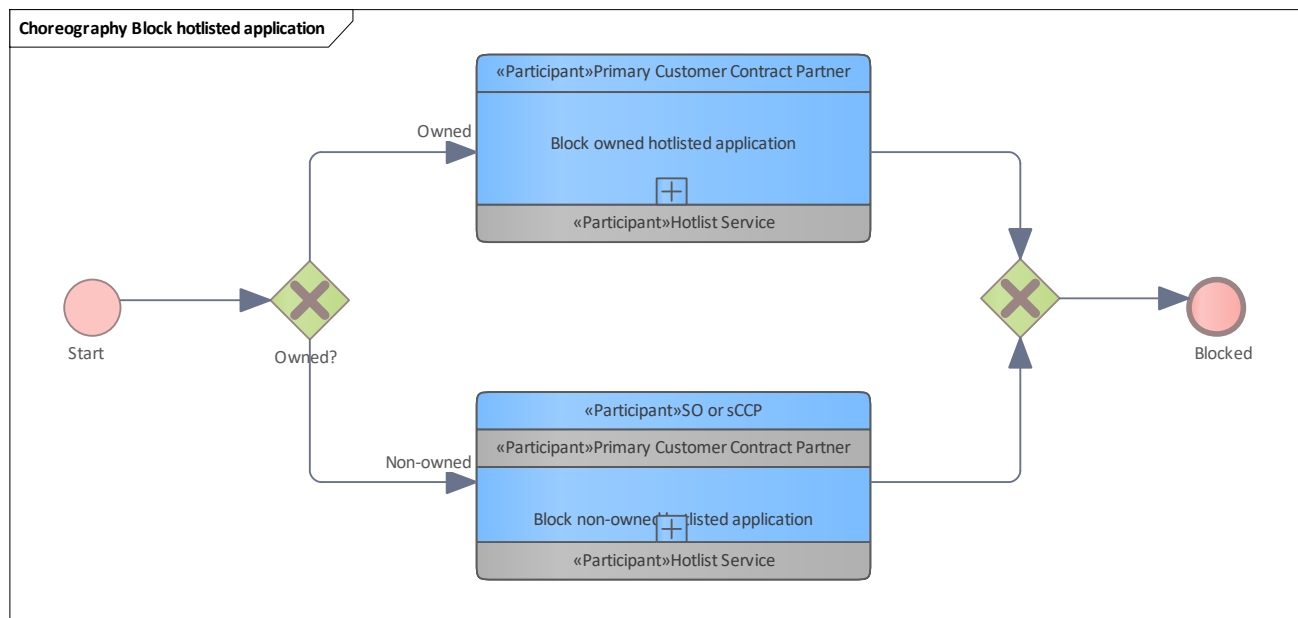


Figure 190: Block hotlisted application

9.3.8.1 Block non-owned hotlisted application

Block a hotlisted application instance that was issued by another CCP. Thus, the current operator that blocks the application instance is not the owner.

See [Block non-owned hotlisted application](#).

9.3.8.2 Block owned hotlisted application

Block an owned hotlisted application instance. The owner of the application instance is the CCP that issued the medium with the application instance to the customer.

See [Block owned hotlisted application](#).

9.3.9 Block hotlisted entitlement

For the sake of simplicity, the blocking of owned and non-owned entitlement are shown separately.

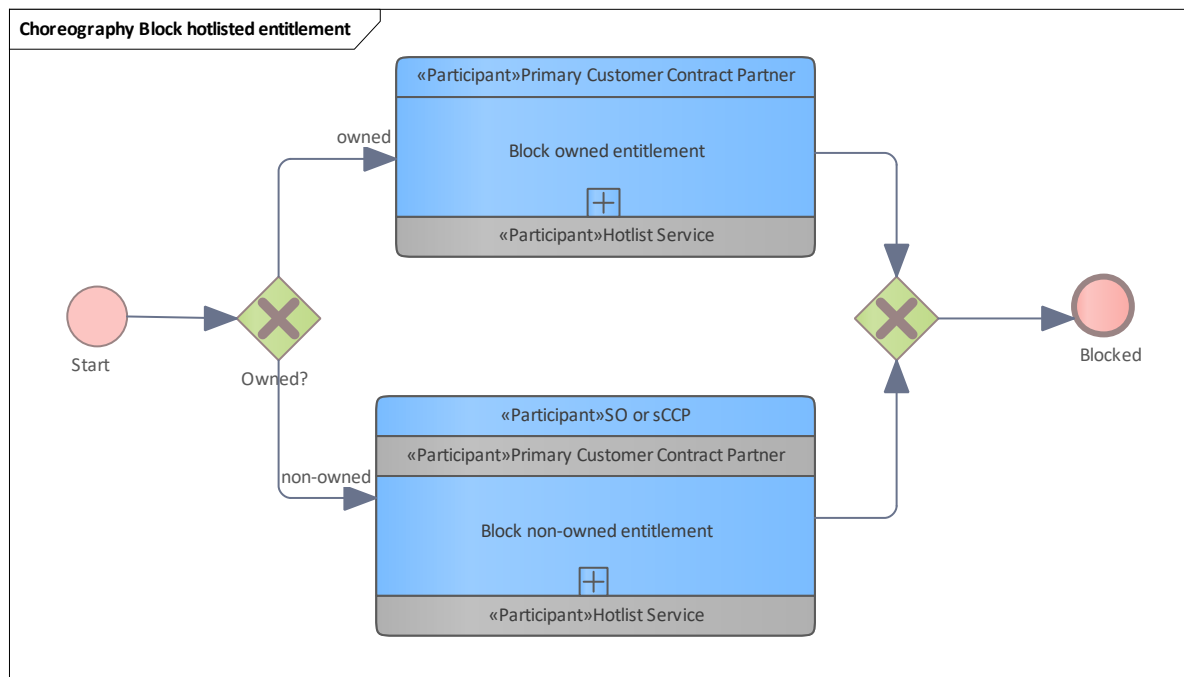


Figure 191: Block hotlisted entitlement

9.3.9.1 Block non-owned entitlement

Block a non-owned entitlement in a user medium with an application.
See [Block non-owned entitlement](#).

9.3.9.2 Block owned entitlement

Block an owned entitlement in a user medium with an application.
See [Block owned entitlement](#).

9.3.10 Hotlist entitlement

For the sake of simplicity, the hotlisting of owned and non-owned entitlements are shown separately.

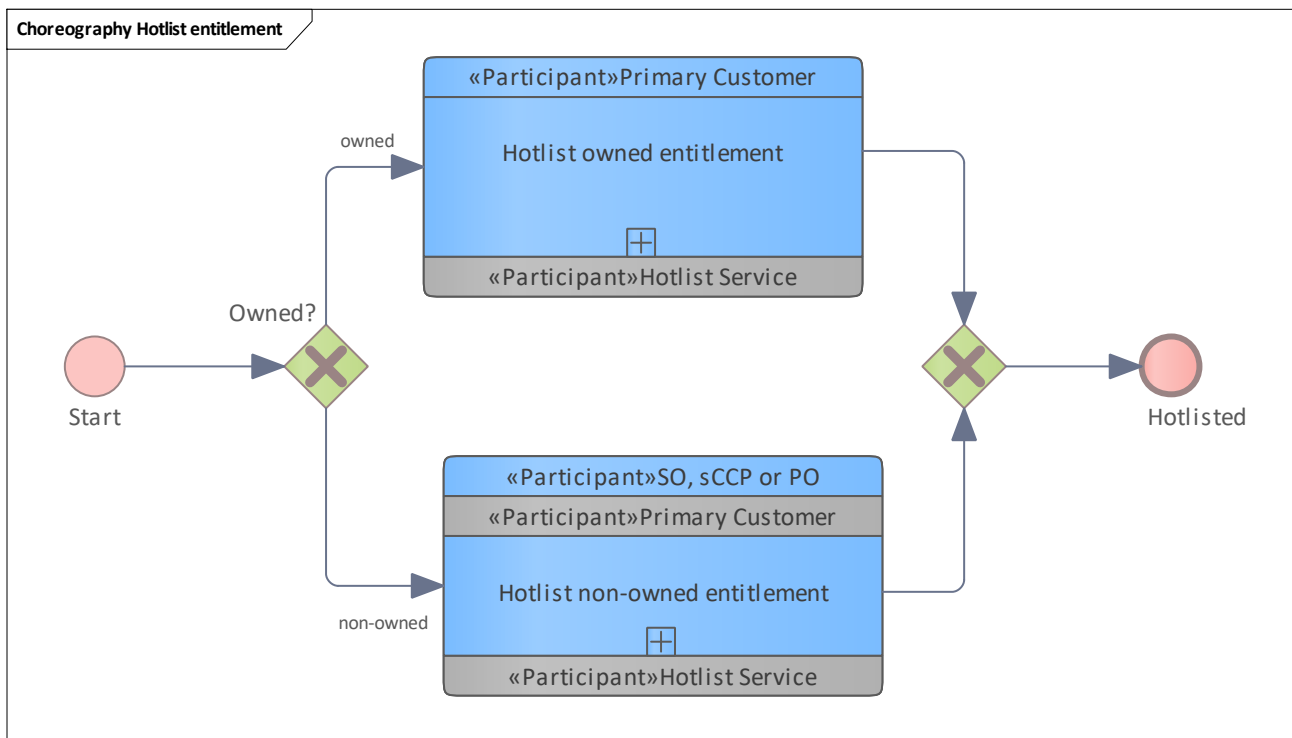


Figure 192: Hotlist entitlement

9.3.10.1 Hotlist non-owned entitlement

Hotlist an entitlement that was owned by another CCP.
See [Hotlist non-owned entitlement](#).

9.3.10.2 Hotlist owned entitlement

Hotlist an owned entitlement. See [Hotlist owned entitlement](#).

9.3.11 Hotlist application

For the sake of simplicity, the hotlisting of owned and non-owned applications are shown separately.

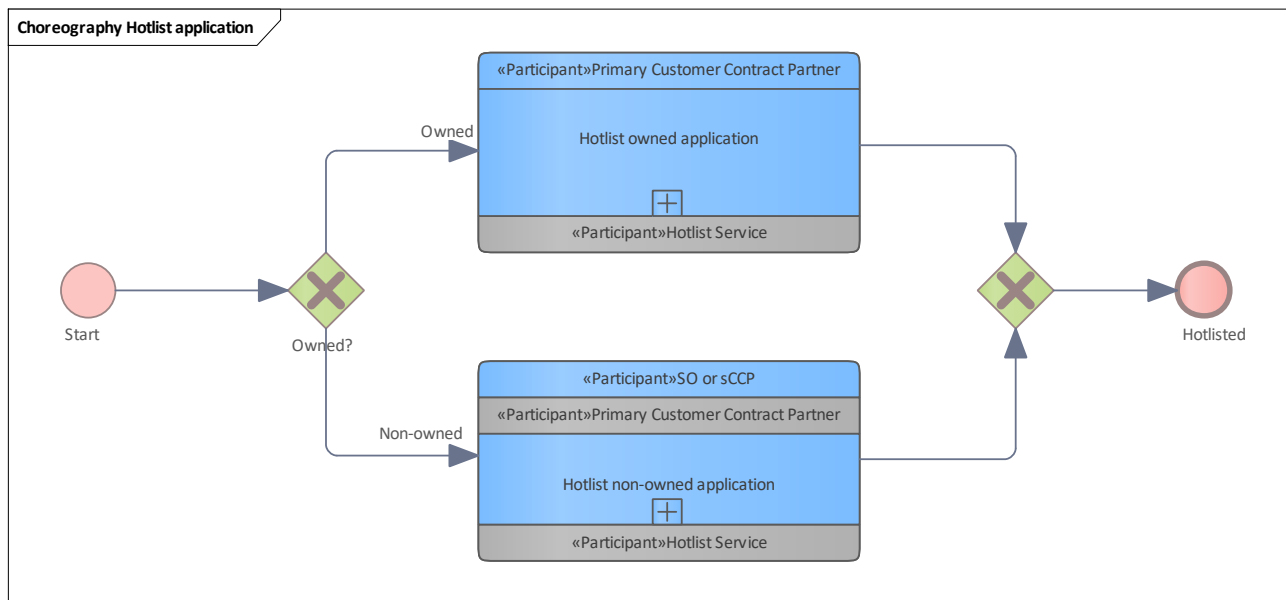


Figure 193: Hotlist application

9.3.11.1 Hotlist non-owned application

Hotlist an application that was issued by another CCP.
See [Hotlist non-owned application](#).

9.3.11.2 Hotlist owned application

Hotlist an owned application. See [Hotlist owned application](#).

9.3.12 Hotlist SAM

Helper BPMN choreography. For the sake of simplicity, the hotlisting of an owned and non-owned SAMs as well as hotlisting SAMs by the Scheme Manager are shown separately.

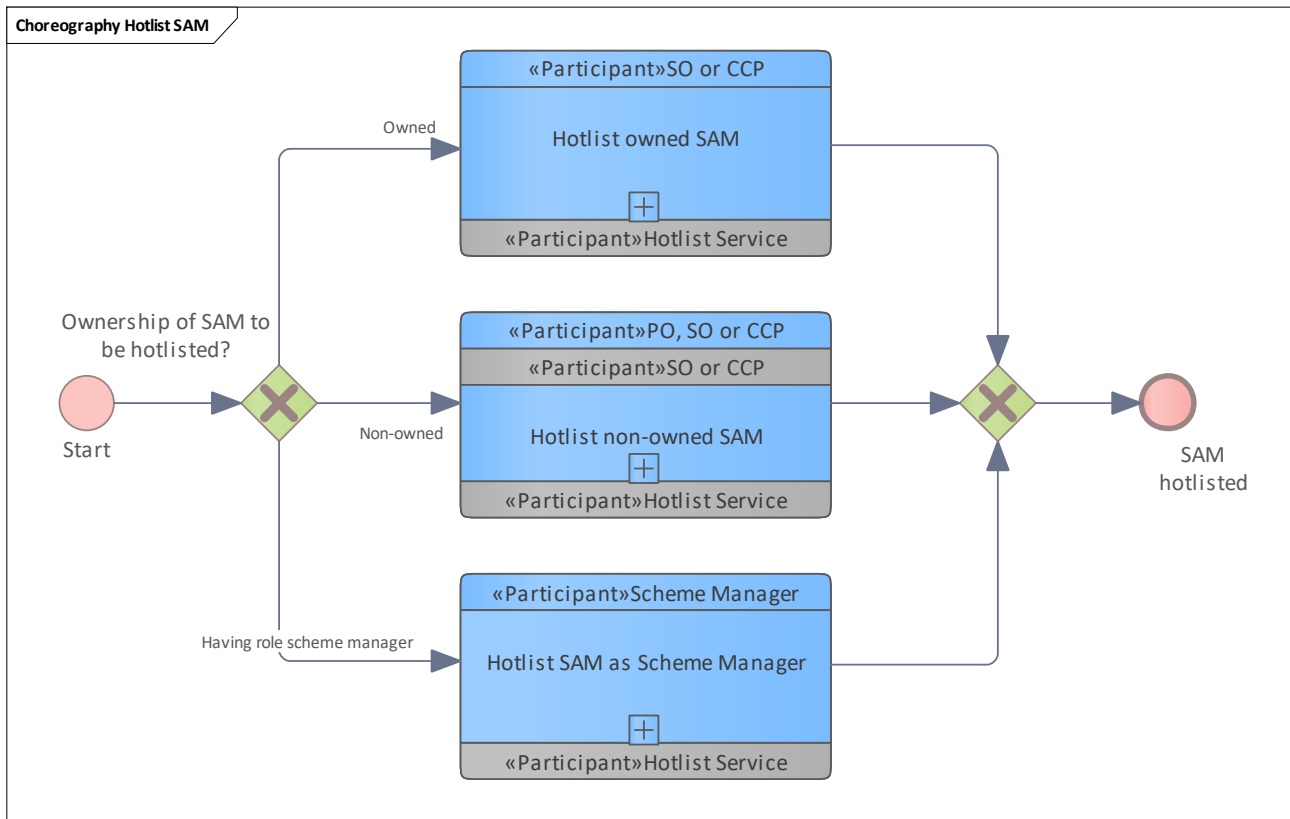


Figure 194: Hotlist SAM

9.3.12.1 Hotlist non-owned SAM

See [Hotlist non-owned SAM](#).

9.3.12.2 Hotlist owned SAM

See [Hotlist owned SAM](#).

9.3.12.3 Hotlist SAM as Scheme Manager

See [Hotlist SAM as Scheme Manager](#).

9.3.13 Take back

Contains helper BPMN choreography models for the scope of take back that allow distinguishing between "owned" (primary CCP itself) and "non-owned" (another, secondary CCP) scenarios.

9.3.14 Take back entitlement

For the sake of simplicity, the process of taking back an owned entitlement (terminal belongs to the issuer of the entitlement) and the process of taking back a non-owned entitlement (terminal does not belong to the issuer of the entitlement) are shown separately.

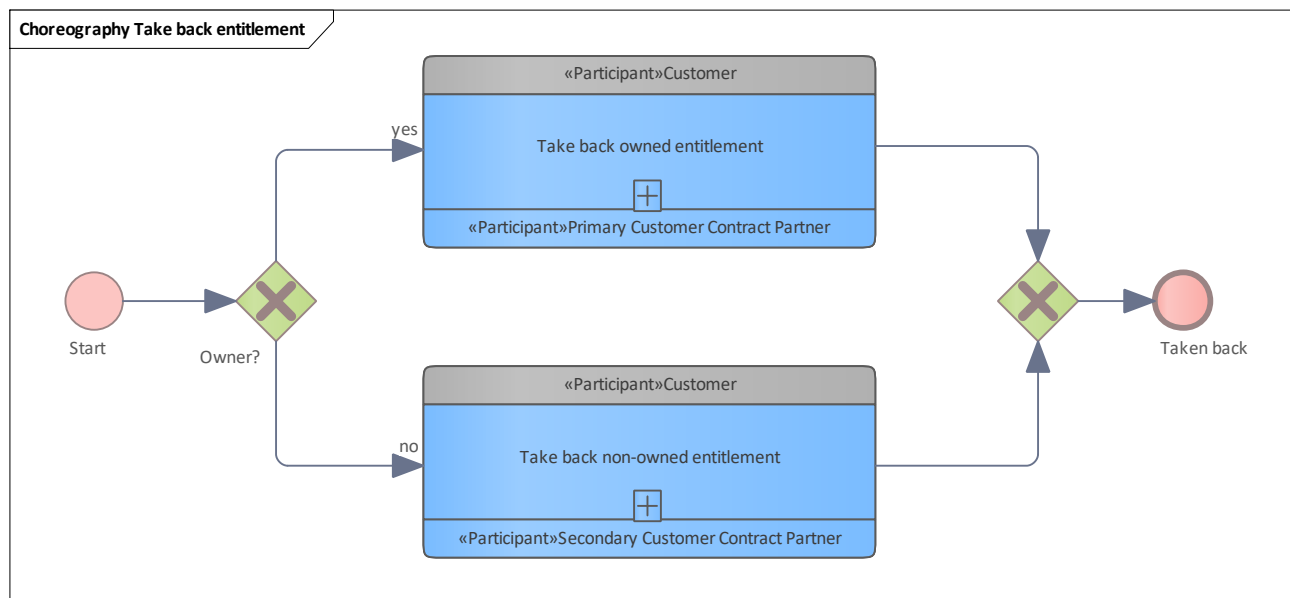


Figure 195: Take back entitlement

9.3.14.1 Owner?

Is the CCP the issuer of the entitlement?

9.3.14.2 Take back owned entitlement

Terminate an owned entitlement. See [Take back owned entitlement](#).

9.3.14.3 Take back non-owned entitlement

Terminate a non-owned entitlement. See [Take back non-owned entitlement](#).

9.3.15 Credit account-based payment method

For the sake of simplicity, the process of crediting an owned (terminal belongs to the issuer of the payment method) account-based payment method and the process of crediting a non-owned (terminal does not belong to the issuer of the payment method) account-based payment method are shown separately.

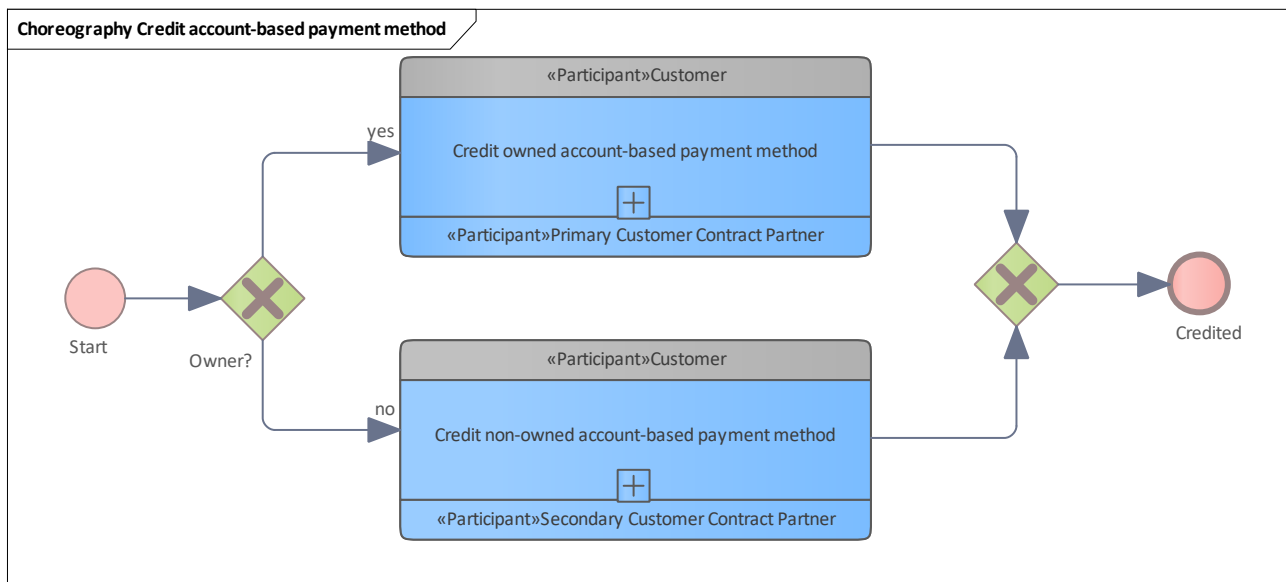


Figure 196: Credit account-based payment method

9.3.15.1 Credit non-owned account-based payment method

Credit an account-based payment method that was issued by someone other than the crediting party. See [Credit owned account-based payment method](#).

9.3.15.2 Credit owned account-based payment method

Credit an account-based payment method that was issued by the crediting party. See [Credit non-owned account-based payment method](#).

9.3.16 Credit stored-value payment method

For the sake of simplicity, the process of crediting an owned (terminal belongs to the issuer of the payment method) stored-value payment method and the process of crediting a non-owned (terminal does not belong to the issuer of the payment method) stored-value payment method are shown separately.

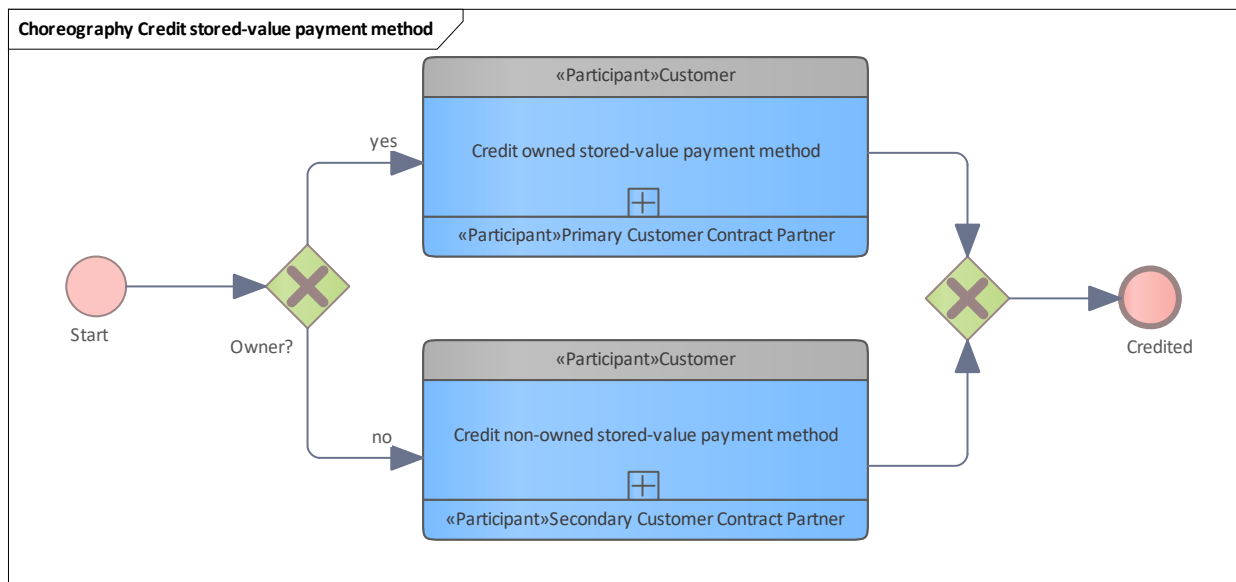


Figure 197: Credit stored-value payment method

9.3.16.1 Credit non-owned stored-value payment method

Credits an amount using a stored-value payment method that was issued by someone other than the crediting party.

See [Credit non-owned stored-value payment method](#).

9.3.16.2 Credit owned stored-value payment method

Credits an amount using a stored-value payment method that was issued by the crediting party.

See [Credit owned stored-value payment method](#).

9.3.17 Reimburse stored-value payment method

For the sake of simplicity, the process of reimbursing an owned (terminal belongs to the issuer of the payment method) stored-value payment method and the process of reimbursing a non-owned (terminal does not belong to the issuer of the payment method) stored-value payment method are shown separately.

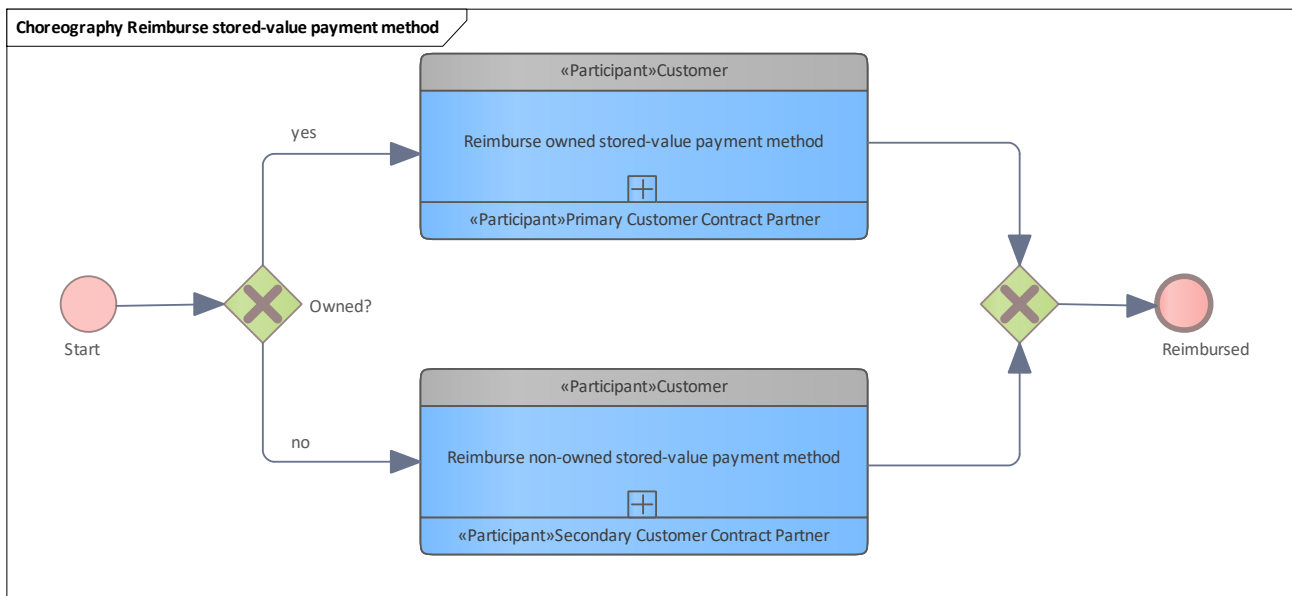


Figure 198: Reimburse stored-value payment method

9.3.17.1 Reimburse non-owned stored-value payment method

Reimburses an amount from a stored-value payment method that was issued by someone other than the reimbursing party.

See [Reimburse non-owned stored-value payment method](#).

9.3.17.2 Reimburse owned stored-value payment method

Reimburses an amount from a stored-value payment method that was issued by the reimbursing party.

See [Reimburse owned stored-value payment method](#).

9.4 Participants

This chapter lists the BPMN participants that are used in the BPMN diagrams. Some of these participants are related 1:1 to their roles in the [Role Model](#). Some participants serve as a support structure to merge several roles for a certain functionality which can be used by participants in different roles.

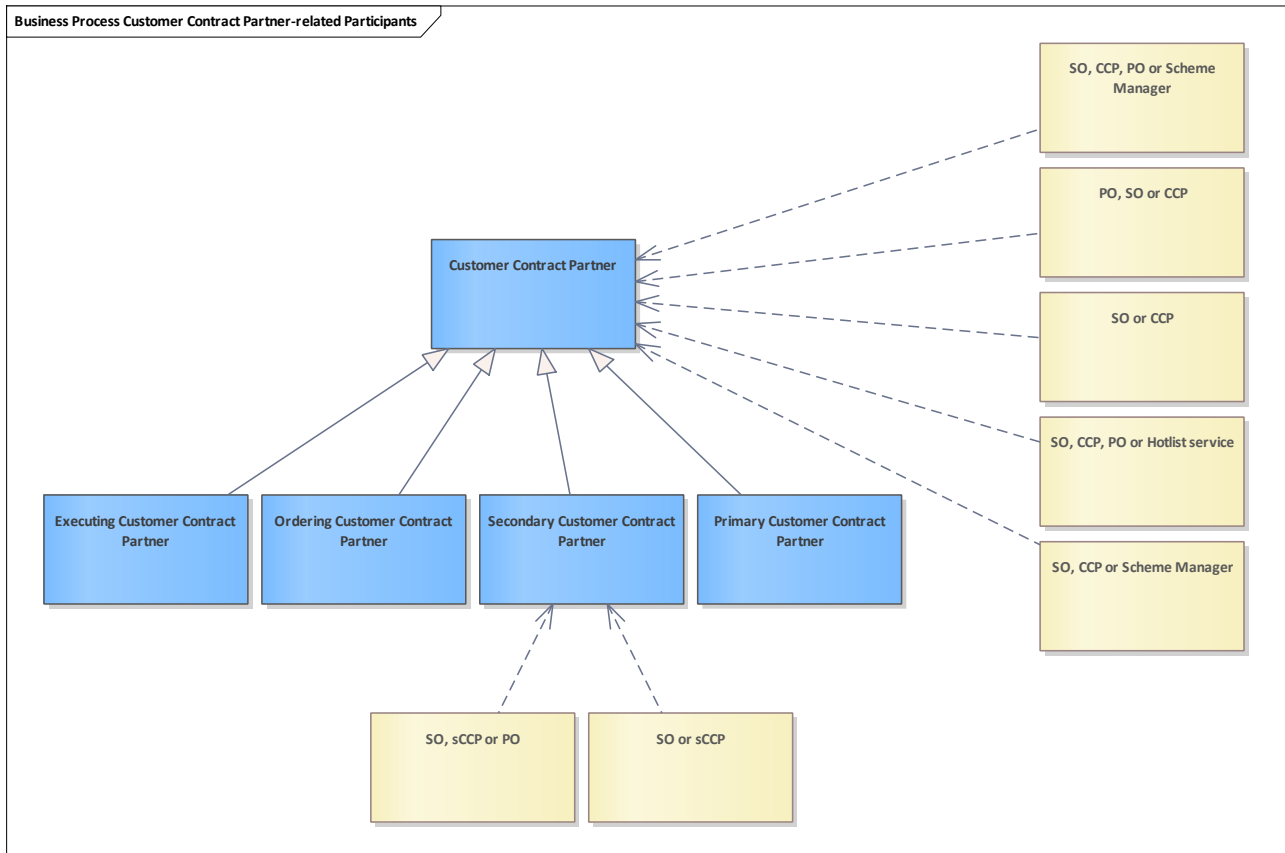


Figure 199: Customer Contract Partner-related Participants

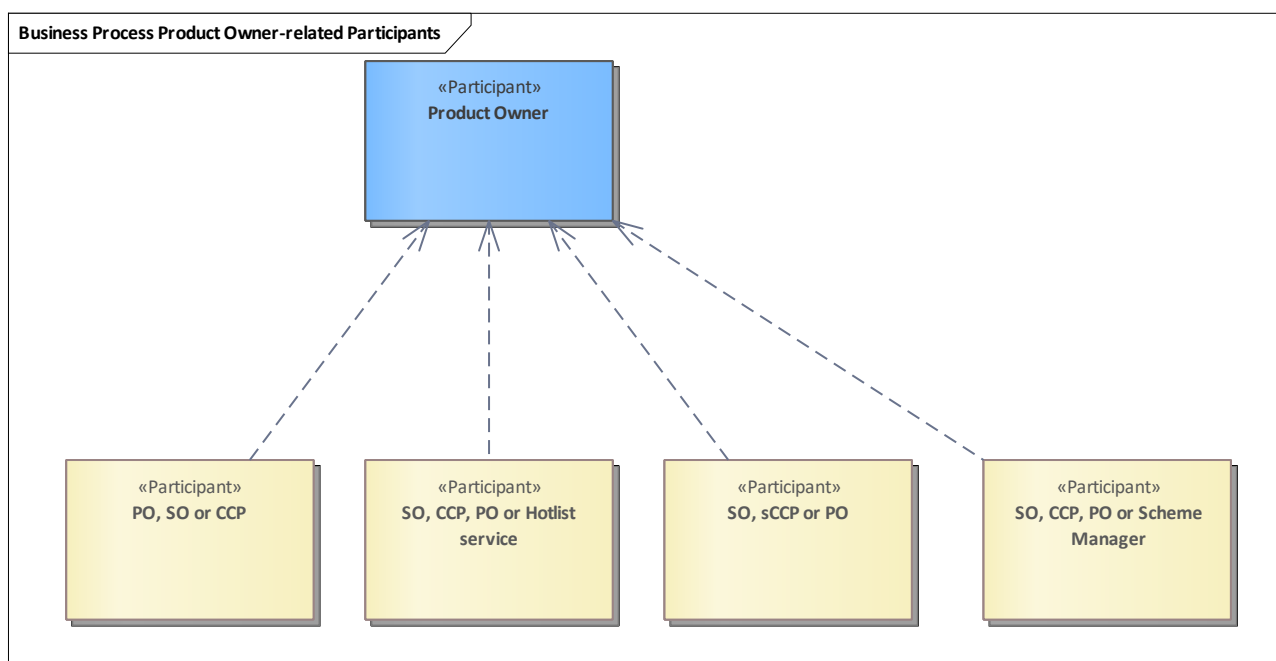


Figure 200: Product Owner-related Participants

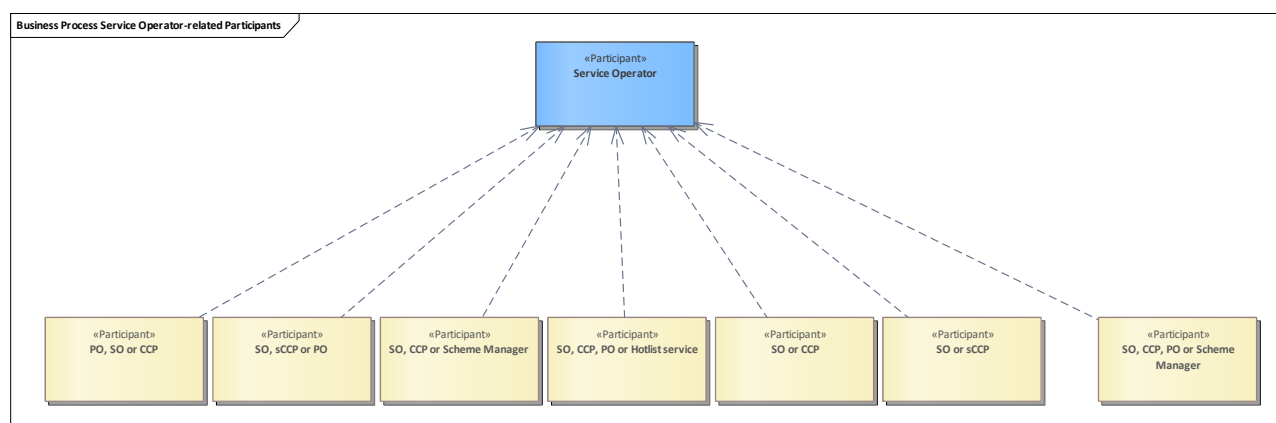


Figure 201: Service Operator-related Participants

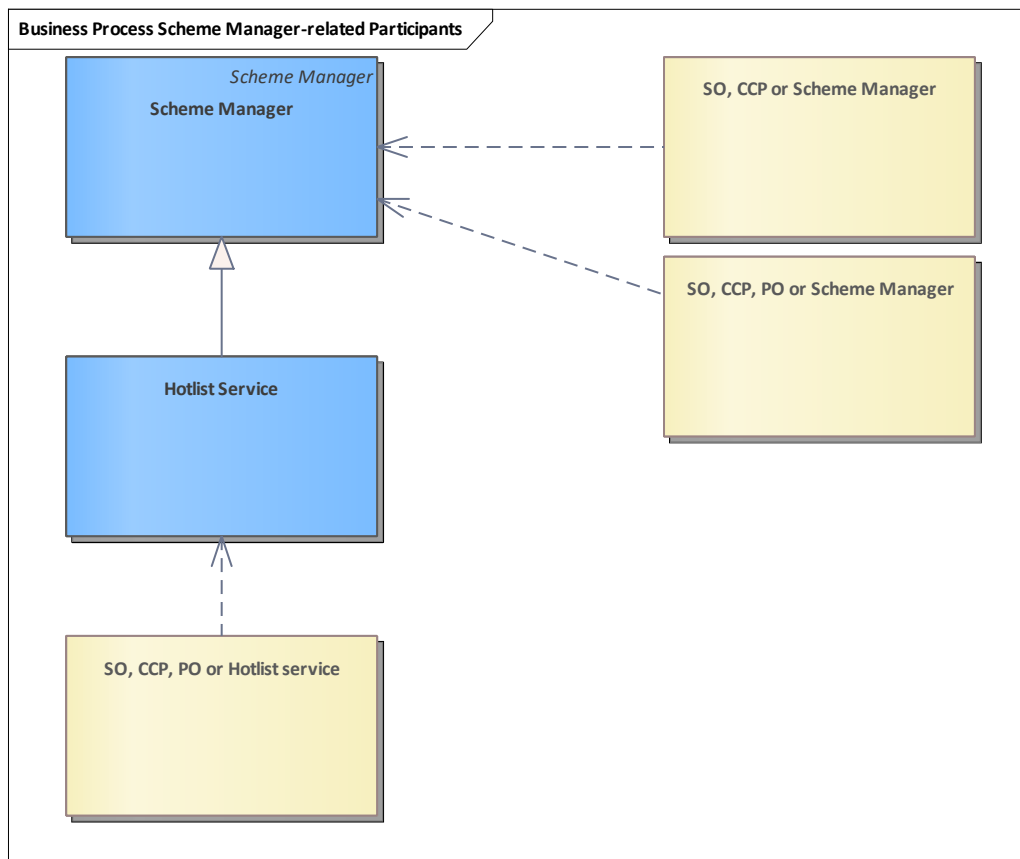


Figure 202: Scheme Manager-related Participants

9.5 Customer Contract Partner

Concrete participant in the role of a [Customer Contract Partner](#).

9.6 Primary Customer Contract Partner

Concrete participant in the role of a [Primary Customer Contract Partner](#).

9.7 Secondary Customer Contract Partner

Concrete participant in the role of a [Secondary Customer Contract Partner](#).

9.8 Ordering Customer Contract Partner

Concrete participant in the role of a [Customer Contract Partner](#) that orders actions in action management of a [Product Owner](#) to be executed on a user medium the next time the user medium contacts a suitable terminal. See also [Ordering Customer Contract Partner](#) in the role model.

9.9 Executing Customer Contract Partner

Concrete participant in the role of a [Customer Contract Partner](#) that executes actions in an action list coming from the action management of a [Product Owner](#). These actions are to be executed on a user medium the next time the user medium contacts a suitable terminal of the executing customer contract partner. See also [Executing Customer Contract Partner](#) in the role model.

9.10 Service Operator

Concrete participant in the role of a [Service Operator](#).

9.11 Product Owner

Concrete participant in the role of a [Product Owner](#).

9.12 Scheme Manager

Concrete participant in the role of the [Scheme Manager](#).

9.13 Hotlist Service

Concrete participant of the category [Hotlist Service](#) in the context of the [Scheme Manager](#).

9.14 SO, CCP or Scheme Manager

Participant that can be either a [Service Operator](#), a [Customer Contract Partner](#) or the [Scheme Manager](#).

9.15 SO, CCP, PO or Hotlist service

Participant that can be either a [Service Operator](#), a [Customer Contract Partner](#), a [Product Owner](#) or the [Hotlist Service](#) as part of the [Scheme Manager](#).

9.16 SO, CCP, PO or Scheme Manager

Participant that can be either a [Service Operator](#), a [Customer Contract Partner](#), a [Product Owner](#) or the [Scheme Manager](#).

9.17 PO, SO or CCP

Participant that can be either a [Service Operator](#), a [Customer Contract Partner](#) or a [Product Owner](#).

9.18 PO or Scheme Manager

Participant that can be either a [Product Owner](#) or the [Scheme Manager](#).

9.19 SO or CCP

Participant in the role of a [Service Operator](#) or a [Customer Contract Partner](#). This participant can be considered as a terminal operator.

9.20 SO or sCCP

Participant in the role of a [Service Operator](#) or a [Secondary Customer Contract Partner](#). This participant can be considered as a terminal operator.

9.21 SO, sCCP or PO

Participant in the role of a [Service Operator](#), a [Secondary Customer Contract Partner](#) or a [Product Owner](#).

9.22 Customer

Concrete participant in the role of a [Customer](#).

9.23 Card Manufacturer

The organisation which creates the user media with the etiCORE application for chip cards.

9.24 Mass Personaliser

The company that provides the mass personalisation of chip-based user media. Can be the customer contract partner itself or a card manufacturer.

9.25 Central Routing Engine

Actor which stands for the central routing engine (CRE).

The central routing engine routes messages from an [Initiator](#) to a [Processor](#) and back.

10 Actors

In the public transport, according to ISO 24014, different roles exist to enable interoperable fare management. These roles are described in [Role Model ISO 24014-1](#) and customized to (((etiCORE in [Role Model etiCORE](#).

For modelling purposes, these roles are considered as actors. The actors in this chapter are UML representations for the roles and participant from the chapters linked above.

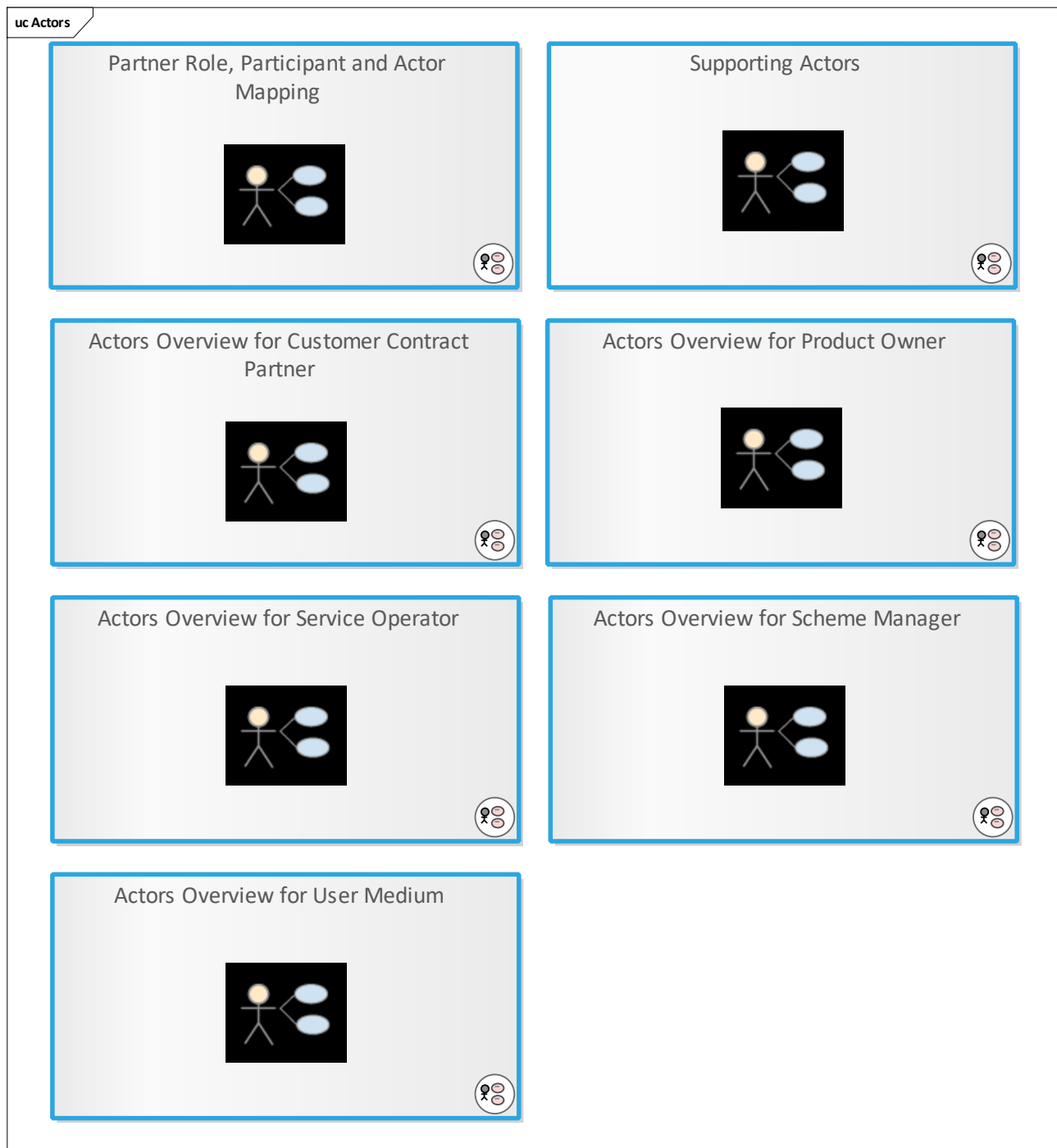


Figure 203: Actors

10.1 Explanatory Actors

Contains actors which belongs to their role and which have explanatory purposes only. The diagrams show the relation of the actors to their systems and components and between the BPMN participant and the UML actor.

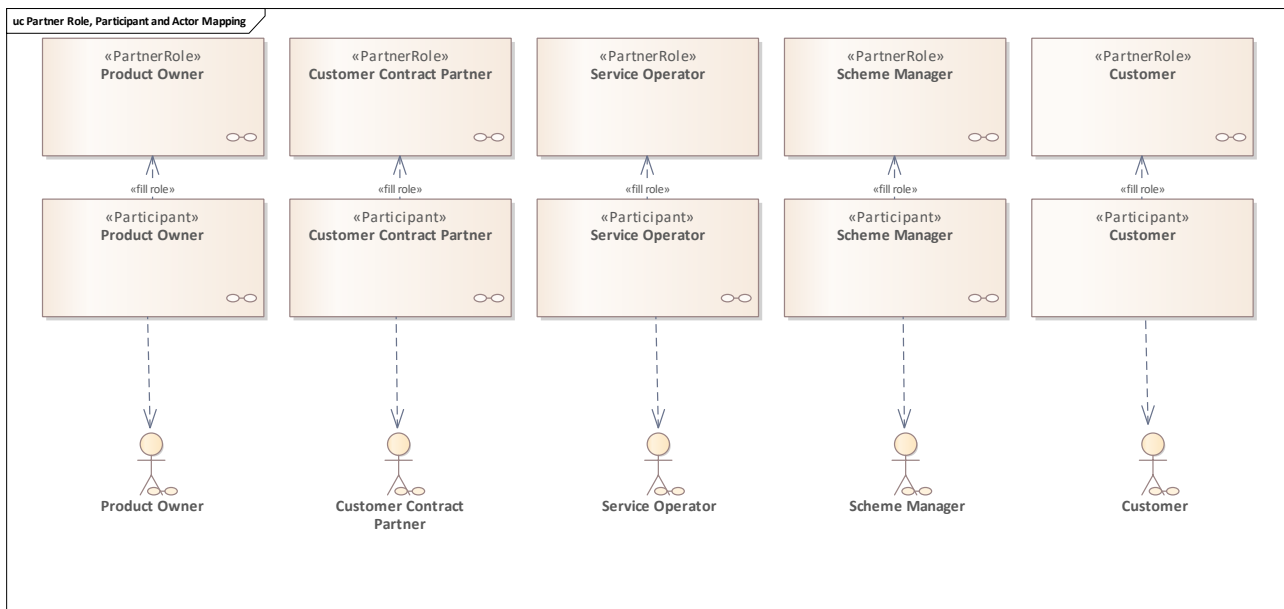


Figure 204: Partner Role, Participant and Actor Mapping

This diagram shows the relation between partner role, participant and actor.

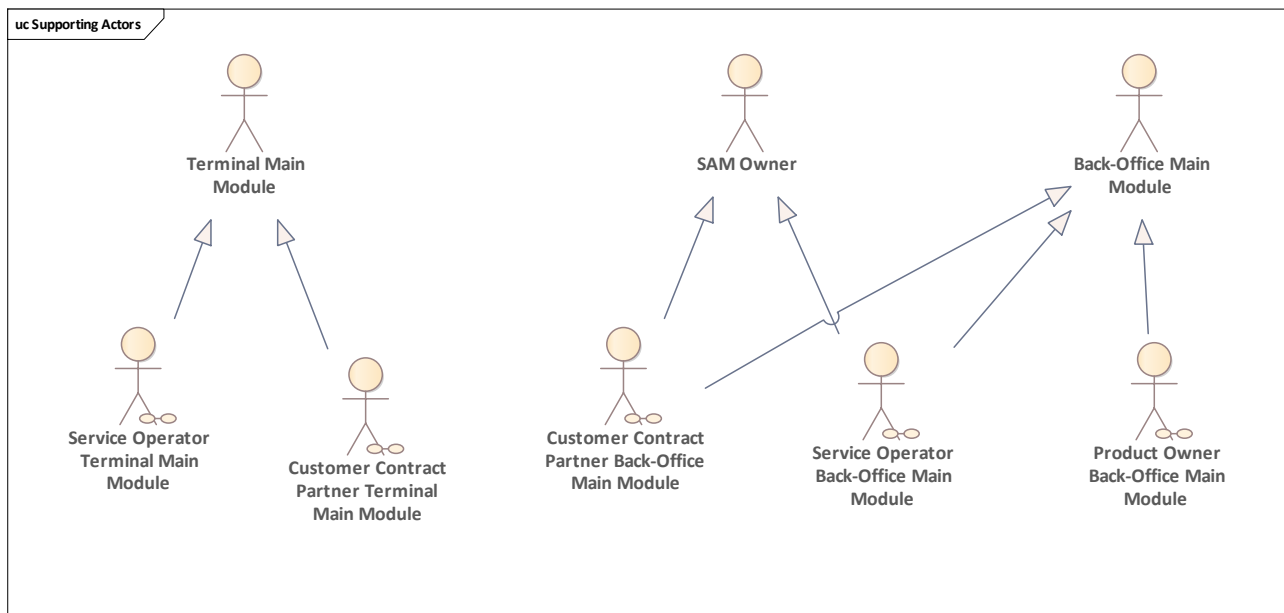


Figure 205: Supporting Actors

This diagram shows supporting actors created for the sake of simplicity during modelling.

10.1.1 Back-Office Main Module

The back-office main module is a generalisation of

- [Service Operator Back-Office Main Module](#),
- [Customer Contract Partner Back-Office Main Module](#) and
- [Product Owner Back-Office Main Module](#).

10.1.2 Customer Contract Partner System

This actor represents the back-office system of the participant [Customer Contract Partner](#).

10.1.3 Customer Contract Partner Terminal

This actor represents the terminal of the participant [Customer Contract Partner](#).

10.1.4 Product Owner System

This actor represents the back-office system of the participant [Product Owner](#).

10.1.5 Scheme Manager System

This actor represents the back-office system of the participant [Scheme Manager](#).

10.1.6 Service Operator System

This actor represents the back-office system of the participant [Service Operator](#).

10.1.7 Service Operator Terminal

This actor represents the terminal of the participant [Service Operator](#).

10.1.8 Terminal Main Module

Common actor to represent all types of terminal main modules such as

- [Customer Contract Partner Terminal Main Module](#) and
- [Service Operator Terminal Main Module](#)

10.1.9 Customer

This actor is the UML representation of a participant who takes the role [Customer](#).

10.1.10 Customer Contract Partner

This actor is the UML representation of a participant who takes the role [Customer Contract Partner](#).

10.1.11 Product Owner

This actor is the UML representation of a participant who takes the role [Product Owner](#).

10.1.12 SAM Owner

Both [Customer Contract Partner](#) and [Service Operator](#) use SAM to secure transactions with the user medium. They are combined into the role SAM Owner.

10.1.13 Scheme Manager

This actor is the UML representation of the participant who takes the role [Scheme Manager](#).

10.1.14 User Medium

This actor represents a user medium which belongs to a [Customer](#).

10.1.15 Service Operator

This actor is the UML representation of a participant who takes the role [Service Operator](#).

10.2 Customer Contract Partner

In this chapter, actors are defined under context of [Customer Contract Partner](#).

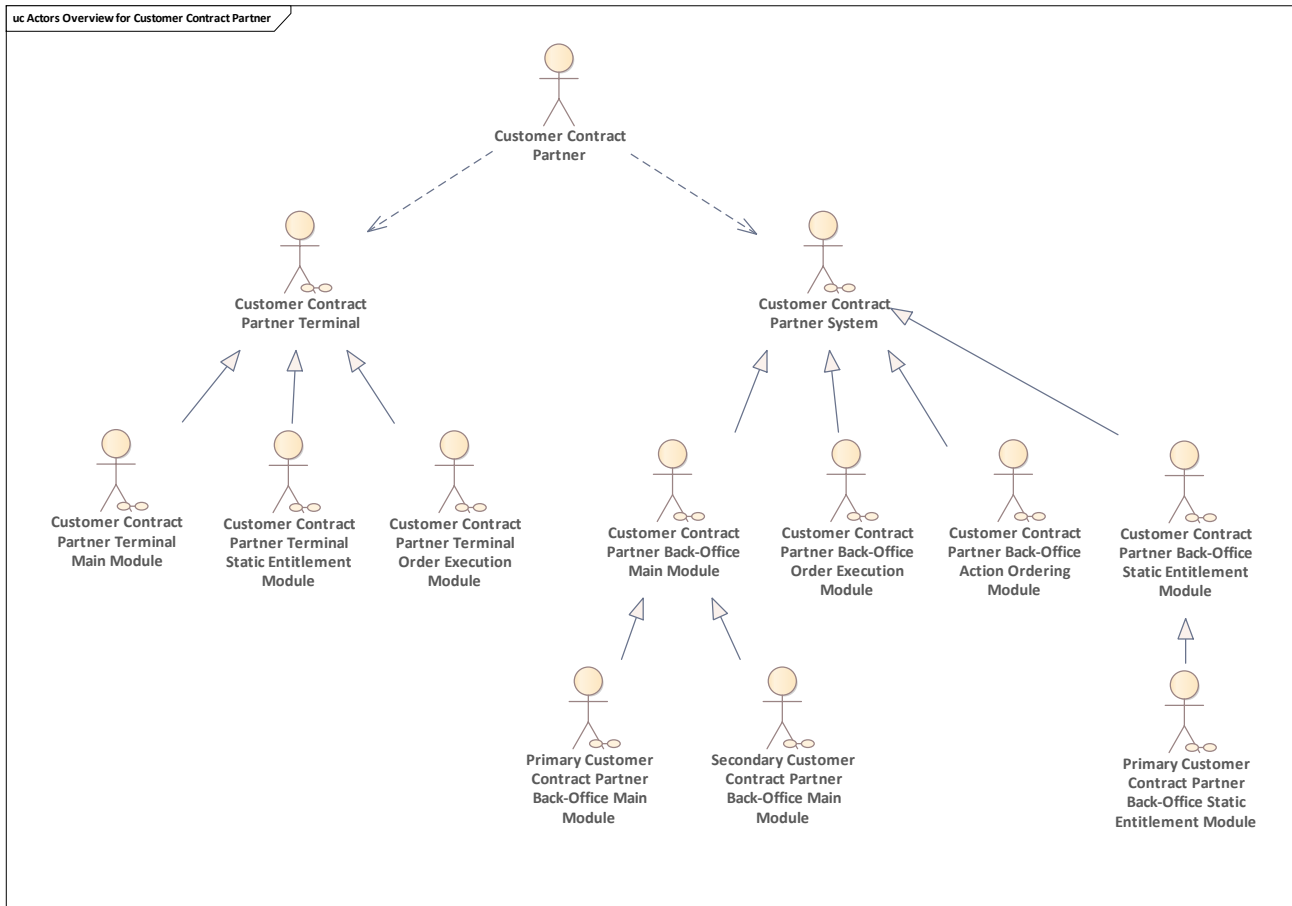


Figure 206: Actors Overview for Customer Contract Partner

This diagram provides an overview for actors and their relationships with each other under the context of the role [Customer Contract Partner](#).

10.2.1 Customer Contract Partner Terminal Main Module

This actor represents the main module of a CCP terminal and is responsible for all basic processes (for example: [Issue entitlement](#), [Unblock entitlement](#) etc.) **except the ones** related to static entitlement and order execution.

10.2.2 Customer Contract Partner Terminal Order Execution Module

This actor represents the order execution module of a CCP-Terminal and is responsible for all ordered actions processes related to order execution.



For example: [Issue entitlement triggered by action order](#), [Block entitlement triggered by action order](#) etc.).

10.2.3 Customer Contract Partner Terminal Static Entitlement Module

This actor represents the static entitlement module of a CCP-Terminal and is responsible for all basic processes related to static entitlement. (For example: [Issue static entitlement](#) etc.)

10.2.4 Customer Contract Partner Back-Office Main Module

This actor represents the main module of a CCP-back-office system and is responsible for all basic processes.

For example:

- [Handle entitlement issued notification from operational perspective](#),
- [Handle stored-value payment method recharged notification from operational perspective](#),
- etc.

except the ones related to static entitlement as well as action ordering and execution.

10.2.5 Primary Customer Contract Partner Back-Office Main Module

This actor represents a specialisation of Customer Contract Partner Back Office Main Module in respect of [Primary Customer Contract Partner](#).

10.2.6 Secondary Customer Contract Partner Back-Office Main Module

This actor represents a specialisation of Customer Contract Partner Back Office Main Module in respect of [Secondary Customer Contract Partner](#).

10.2.7 Customer Contract Partner Back-Office Action Ordering Module

This actor represents the order execution module of a CCP back-office system and is responsible for all ordered actions processes related to action execution.

For example:

- [Order entitlement blocking](#),
- [Order entitlement termination](#),
- etc.

See also role [Ordering Customer Contract Partner](#).

10.2.8 Customer Contract Partner Back-Office Order Execution Module

This actor represents the order execution module of a CCP back-office system and is responsible for all ordered actions processes related to action management.
See also role [Executing Customer Contract Partner](#).

10.2.9 Customer Contract Partner Back-Office Static Entitlement Module

This actor represents the static entitlement module of a CCP back-office system and is responsible for all basic processes related to static entitlement.

10.2.10 Primary Customer Contract Partner Back-Office Static Entitlement Module

This actor represents a specialisation of Customer Contract Partner Back Office Static Entitlement Module in respect of [Primary Customer Contract Partner](#).

10.3 Product Owner

In this chapter, actors are defined under context of [Product Owner](#).

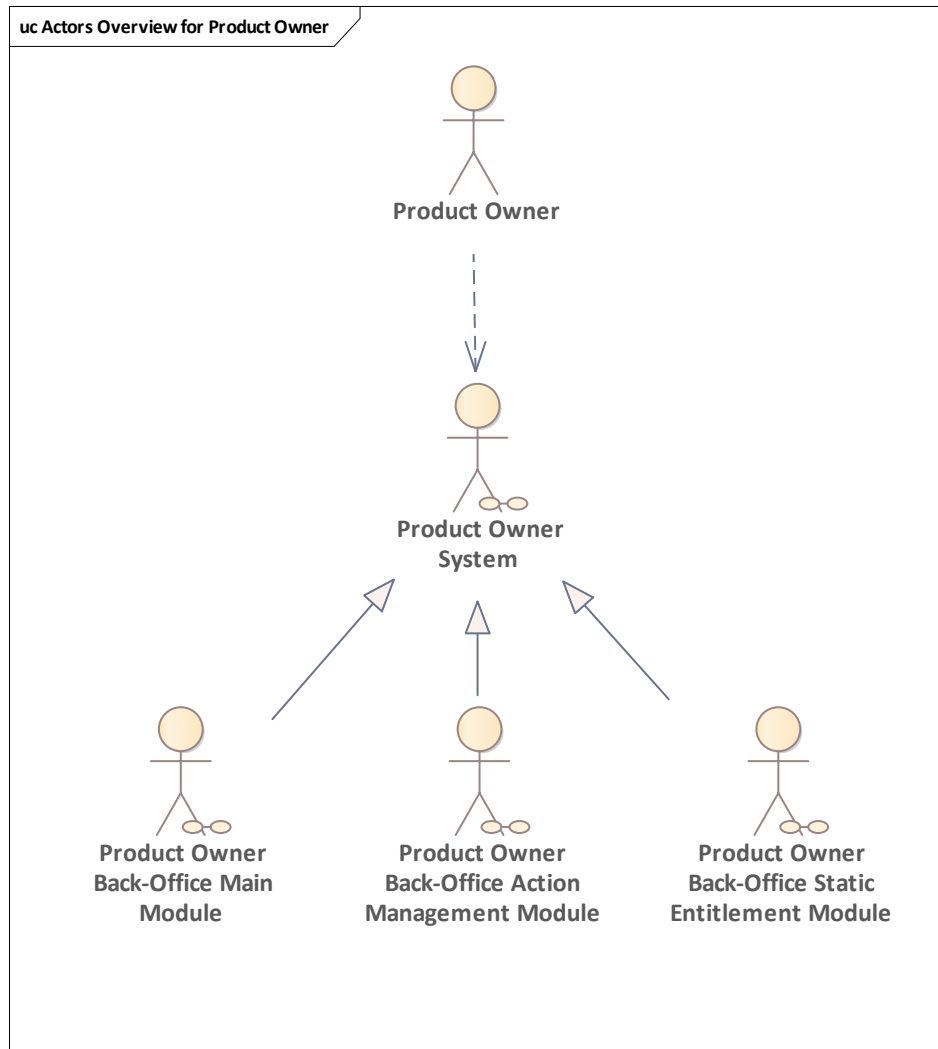


Figure 207: Actors Overview for Product Owner

10.3.1 Product Owner Back-Office Main Module

This actor represents the main module of a PO back-office system and is responsible for all basic processes.

For example:

- [Handle entitlement validated notification from product perspective](#),
- [Determine valid entitlements for given app instance ID](#)
- etc.

except the ones related to static entitlement, and action management.



10.3.2 Product Owner Back-Office Action Management Module

This actor represents the action management module of a PO back-office system and is responsible for all processes of the ordered action management.

10.3.3 Product Owner Back-Office Static Entitlement Module

This actor represents the module of a PO back-office system that is responsible for all basic processes related to static entitlements.

10.4 Service Operator

In this chapter, actors are defined under context of [Service Operator](#).

10.4.1 Service Operator Terminal Main Module

This actor represents the main module of terminal of the service operator and is responsible for all basic processes (for example: [Perform check in and notify](#)) **except the ones** related to static entitlements.

10.4.2 Service Operator Terminal Static Entitlement Module

This actor represents the main module of the back-office system of the service operator and is responsible for all basic processes related to static entitlement (For example: [Inspect user medium without application](#)).

10.4.3 Service Operator Back-Office Main Module

This actor represents the main module of the back office system of the service operator and is responsible for all basic processes **except the ones** related to static entitlement.

For example:

- [Handle stored-value payment method recharged notification from operational perspective](#),
- [Handle check-out notification from operational perspective](#)
- etc.

10.4.4 Service Operator Back-Office Static Entitlement Module

This actor represents the main module of the back office system of the service operator and is responsible for all basic processes related to static entitlement. (For example: [Handle static entitlement inspection notification from operational perspective](#))

10.5 Scheme Manager

In this chapter, actors are defined under context of [Scheme Manager](#).

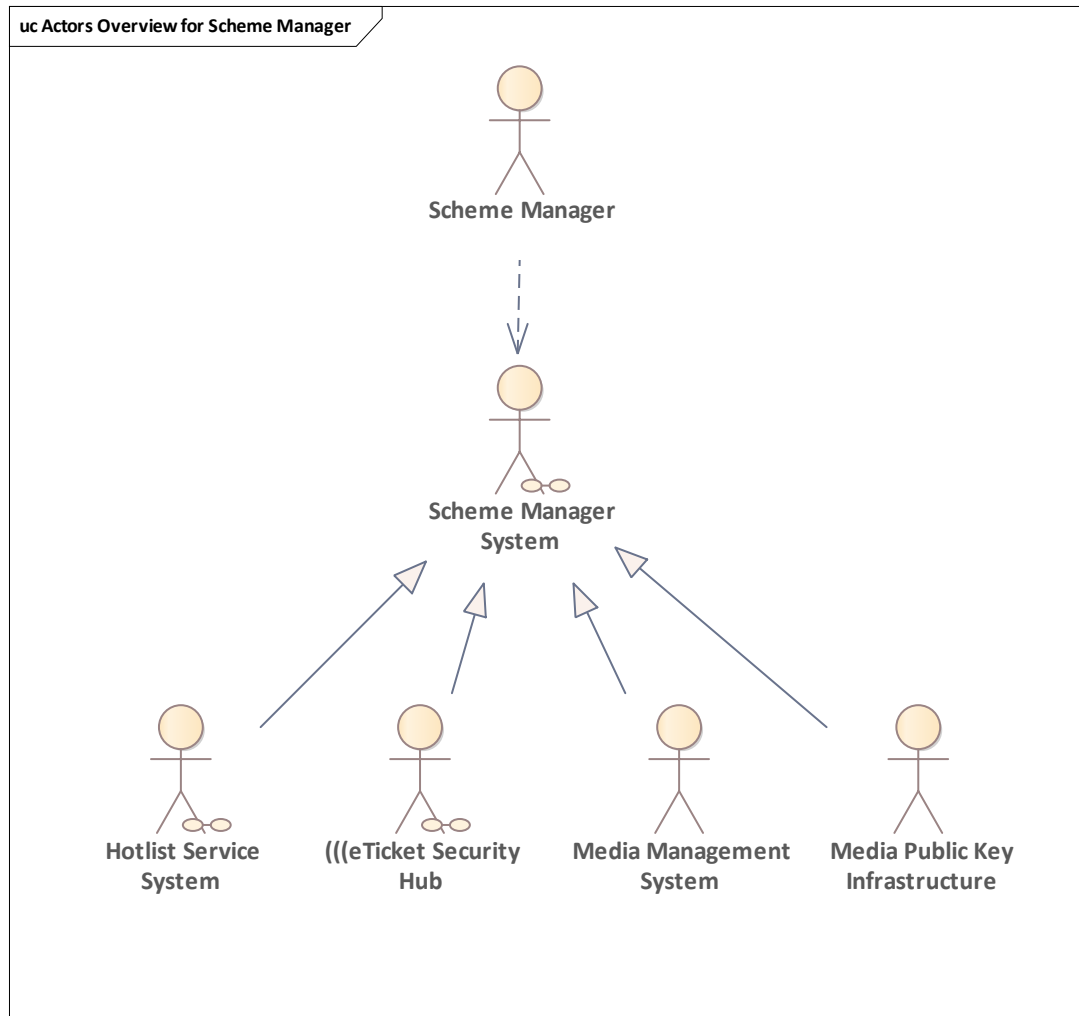


Figure 208: Actors Overview for Scheme Manager

10.5.1 Hotlist Service System

This actor is the UML system representation for the system of the [Hotlist Service](#). The Hotlist service system handles hotlisting and hotlist revocations requests for the entities of entitlement, application, SAM, organisation and authentication key. It generates hotlists and send them on participant requests.

10.5.2 (((eTicket Security Hub

The (((eTicket Security Hub is the central system for service and administration of (((eTicket Deutschland. It is responsible for administration of participants, acts as a portal for registration for (((eTicket Deutschland and as a part of security monitoring system and for order management (for example) of SAMs.



10.5.3 Media Management System

Media management system is responsible for the management of "configuration data" and their secure deployment to SAMs and User Media.

10.5.4 Media Public Key Infrastructure

This PKI is a system designed to issue, manage and store security certificates for User Media, SAMs and related entities communicating with User Media and SAMs.

10.6 User Medium

In this chapter, actors are defined under context of [Customer](#).

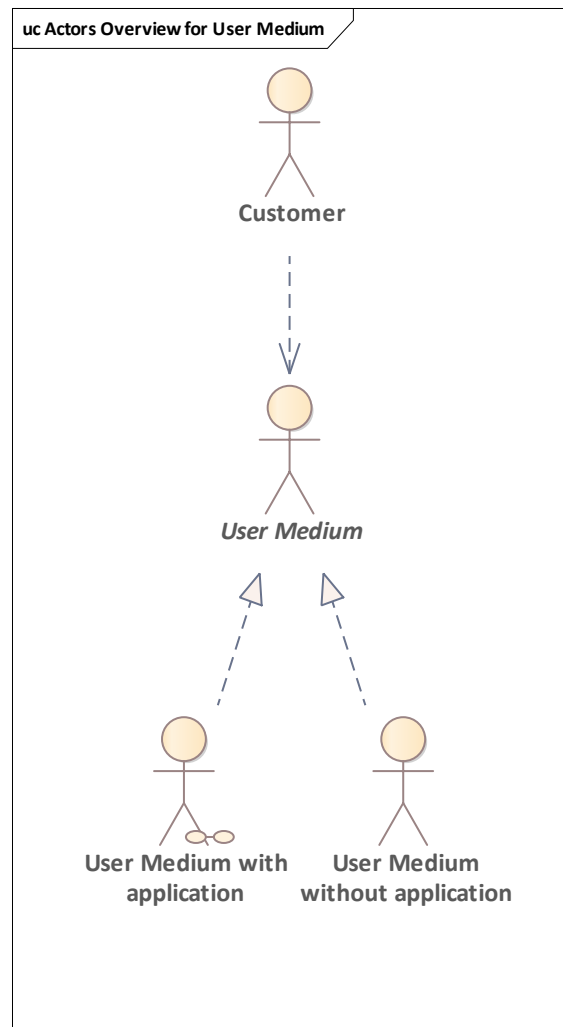


Figure 209: Actors Overview for User Medium

10.6.1 User Medium with application

This actor is specialisation of a user medium in respect of an existing chip-based application.

10.6.2 User Medium without application

This actor is a specialisation of user medium with respect to a non-existing application. Such a user medium does not require a chip-based (((etiCORE application and can be paper tickets or mobile phones processing tickets with their own application (with integrated MOTICS library).

11 List of Use Cases

The following chapter lists all use cases in alphabetical order. This is intended to list the use cases that are referenced in the processes above.

To find use cases related to the roles and functionality bundles, please refer to the reference specifications.

11.1 Activate SAM

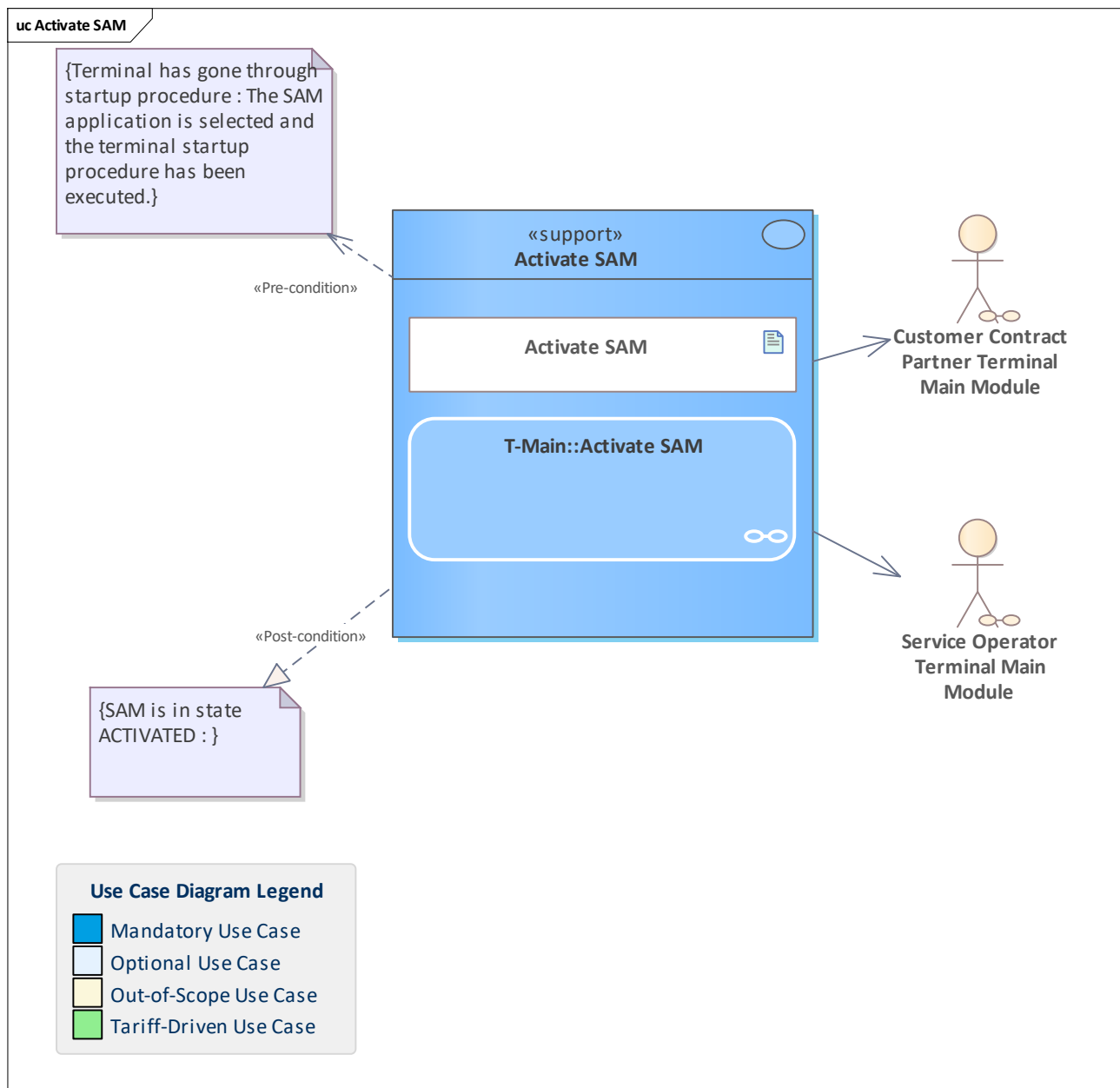


Figure 210: Activate SAM

The terminal activates the SAM to enable the use of the full feature set of the SAM.

11.2 Add application to hotlist

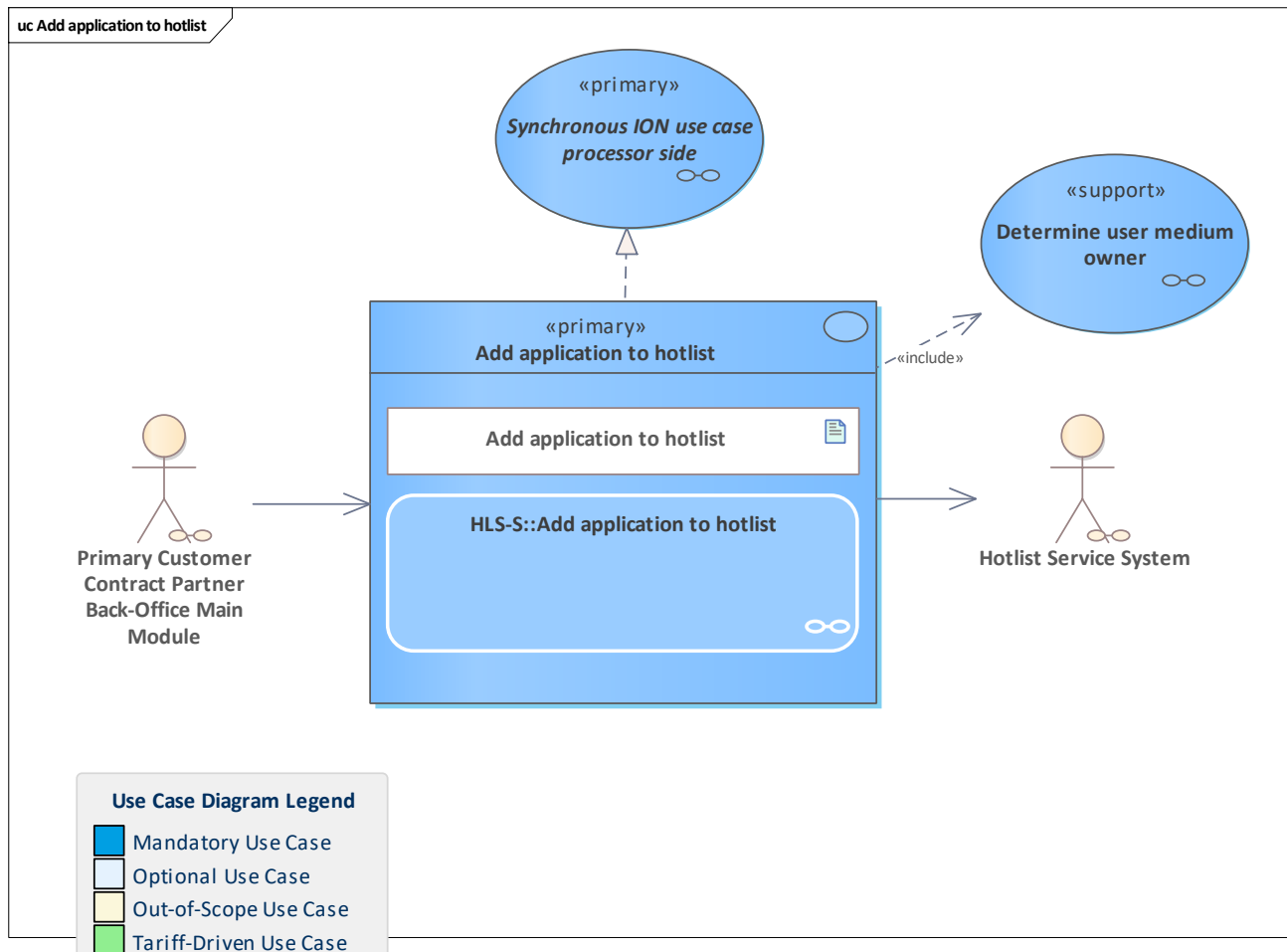


Figure 211: Add application to hotlist

The pCCP sends a hotlisting request for an application instance ID coming from a chip-based user medium or an SCE ID coming from a MOTICS app.

The Hotlist service system checks the request.

The hotlist service system determines the application/user medium owner and compares it with the sender.

The hotlist service system performs further checks.

If all checks are positive, the application is added to the application hotlist.

Note: the application hotlist that is compiled in the next hotlist cycle will contain the hotlist entry with the application instance ID or SCE ID. When the new hotlist is requested and distributed to the terminals, the new entry is taken into account when the check against the hotlist is performed.

11.3 Add authentication key to hotlist

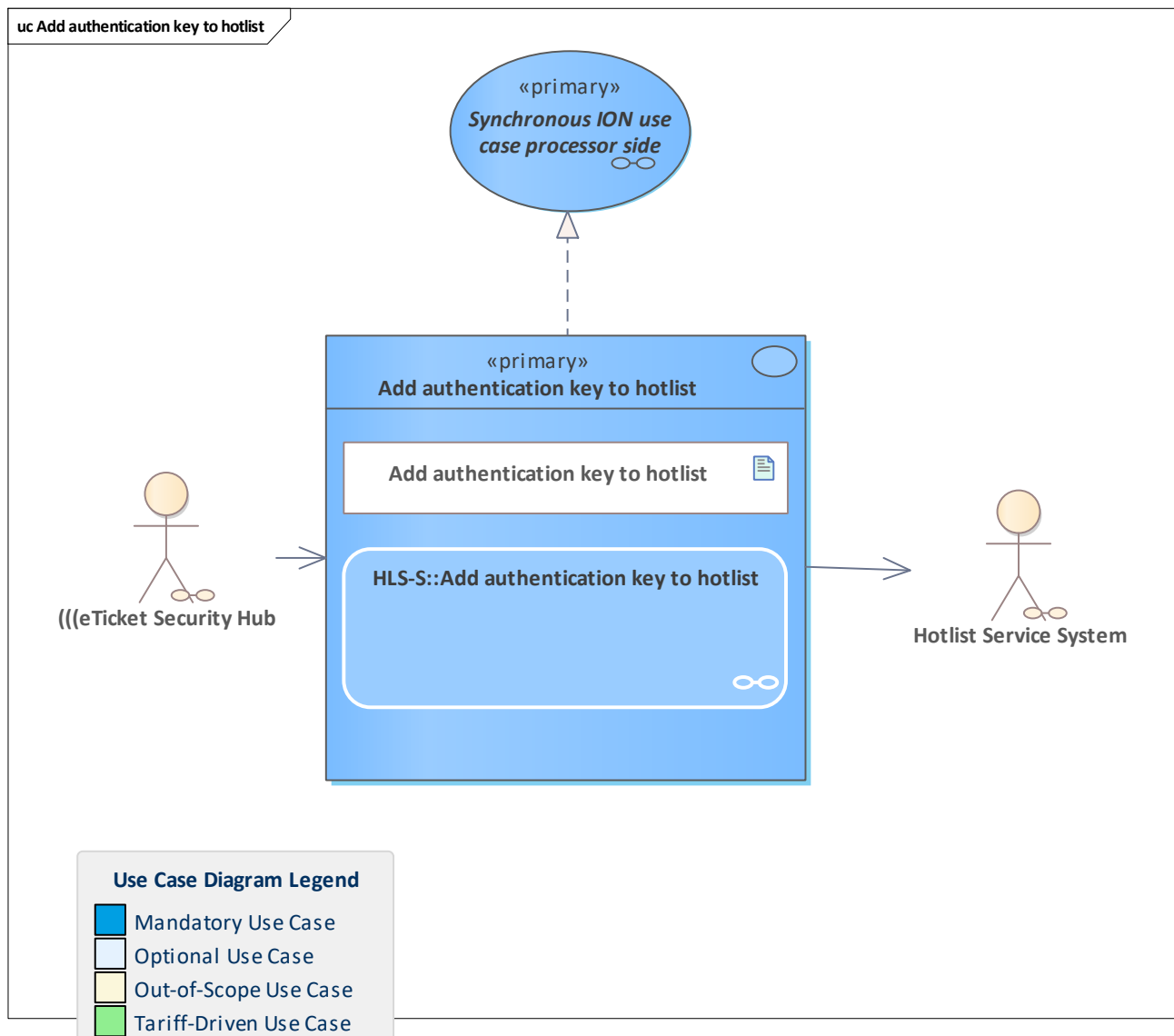


Figure 212: Add authentication key to hotlist

The authentication key is to be hotlisted.

This can only be done by the scheme manager's ESH.

Note: the authentication key hotlist that is compiled in the next hotlist cycle will contain the new hotlist entry. When the new hotlist is requested and distributed to the terminals, the new entry is taken into account when the check against the hotlist is performed.

11.4 Add entitlement to hotlist

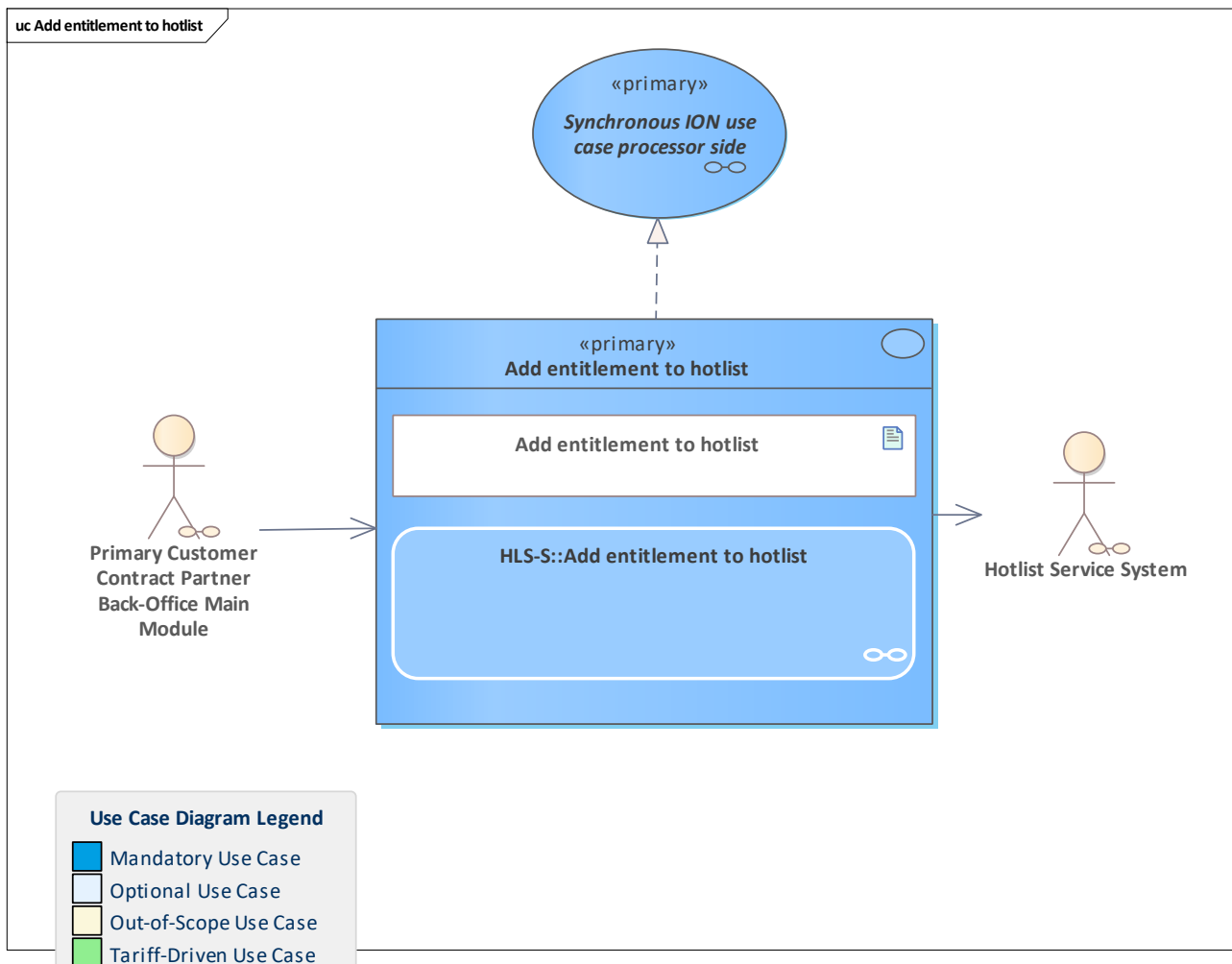


Figure 213: Add entitlement to hotlist

The pCCP sends a hotlisting request for an entitlement ID coming from a chip-based user medium or an entitlement ID from a static entitlement.

The Hotlist service system checks the request.

If all checks are positive, the entitlement is added to the entitlement hotlist.

Note: the entitlement hotlist that is compiled in the next hotlist cycle will contain the hotlist entry with the entitlement. When the new hotlist is requested and distributed to the terminals, the new entry is taken into account when the check against the hotlist is performed.

11.5 Add organisation to hotlist

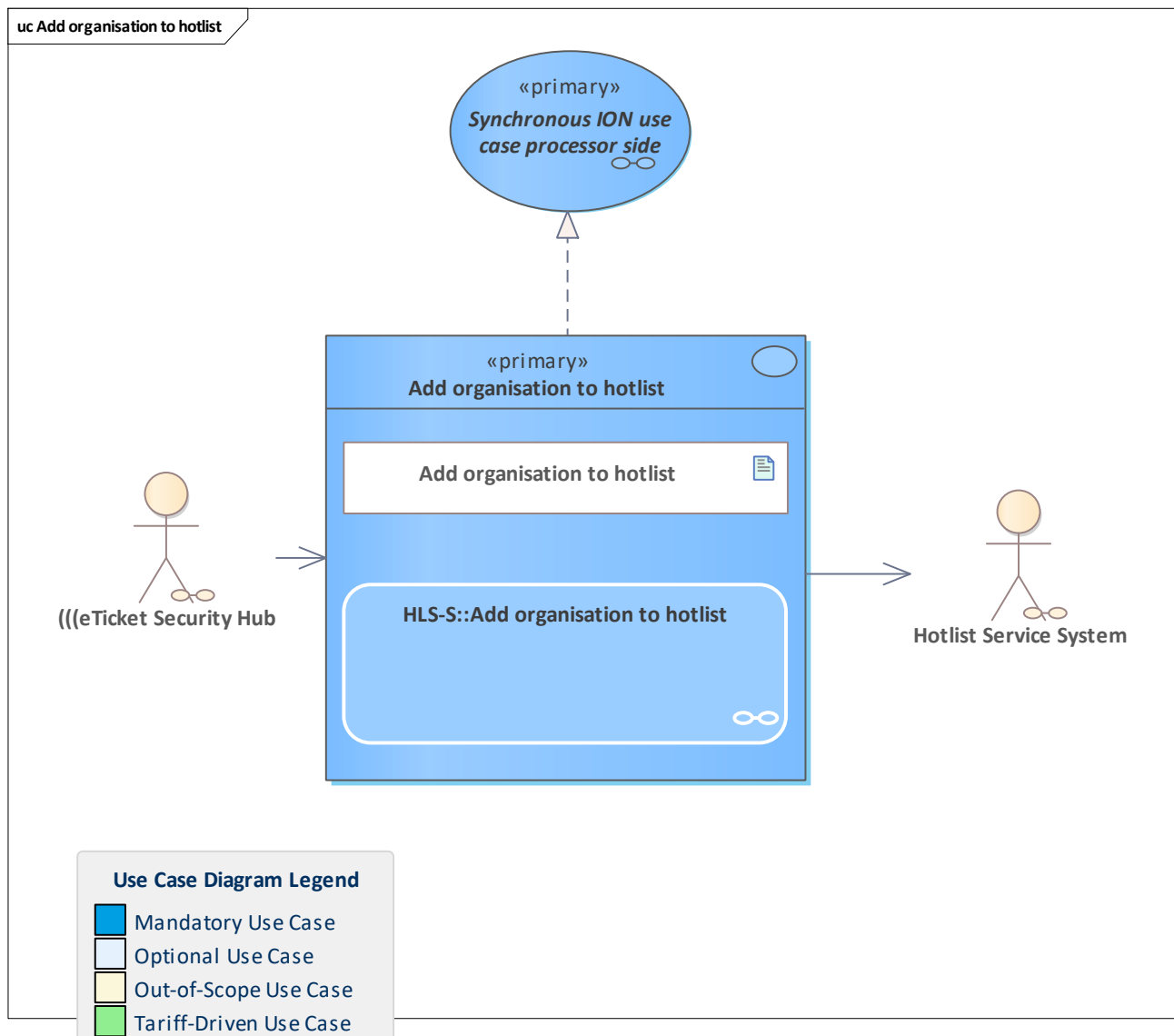


Figure 214: Add organisation to hotlist

Use case to add an organisation to the organisation hotlist.

This can only be done by the scheme manager's ESH.

Note: the organisation hotlist that is compiled in the next hotlist cycle will contain the new hotlist entry. When the new hotlist is requested and distributed to the terminals, the new entry is taken into account when the check against the hotlist is performed.

11.6 Add product acceptance entry to hotlist configuration

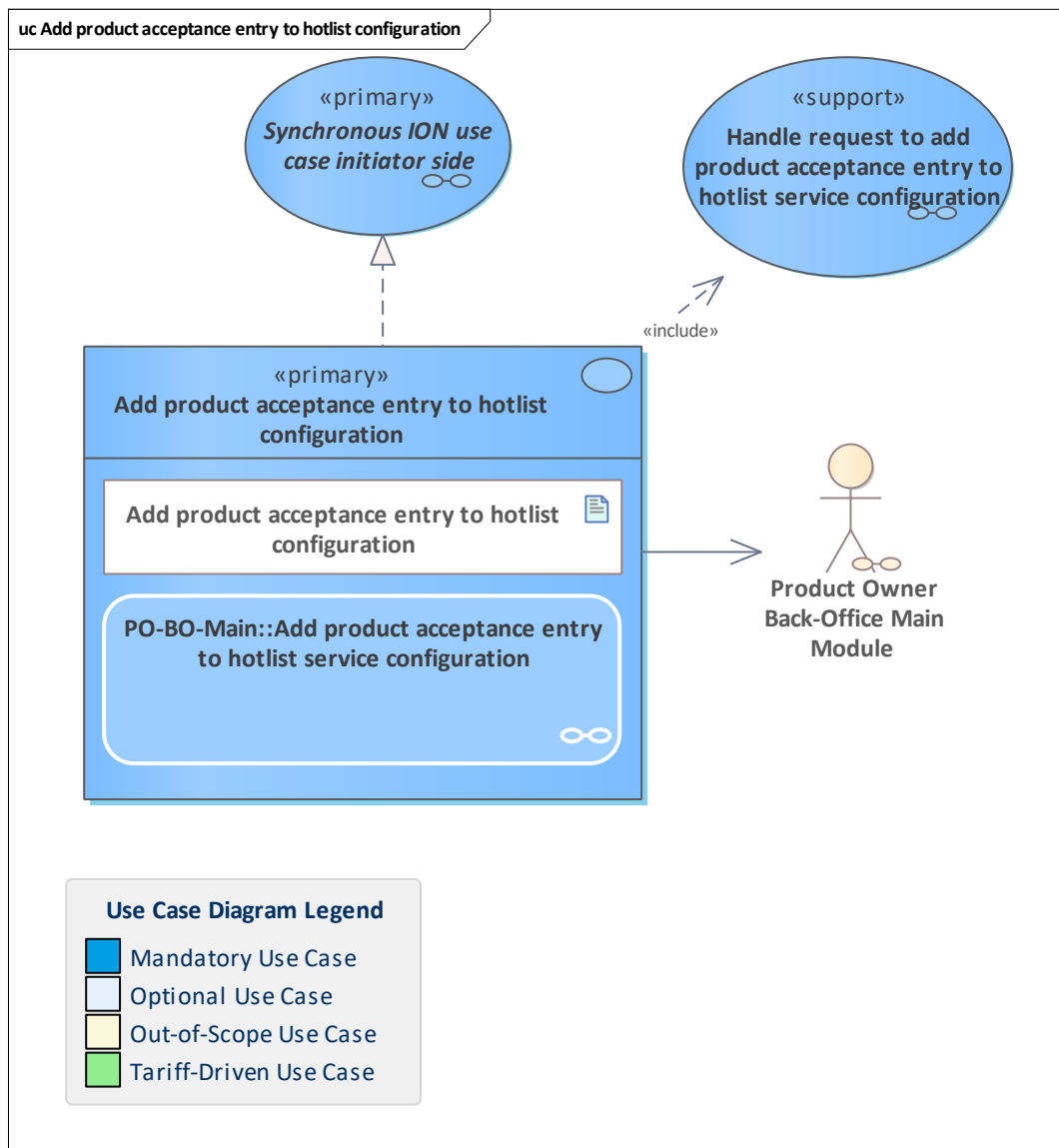


Figure 215: Add product acceptance entry to hotlist configuration

To facilitate specific generation of hotlists, the hotlist service system uses information provided by the PO about which organisation accepts which products. For each of the products accepted by a certain organisation, the PO sends an acceptance request containing the accepting organisation ID, the PO's organisation ID and a product number. If the organisation accepts all products of a PO, the product number is omitted in the request.

Please note that interoperable products are not allowed to be added to the configuration list.

11.7 Add SAM to hotlist

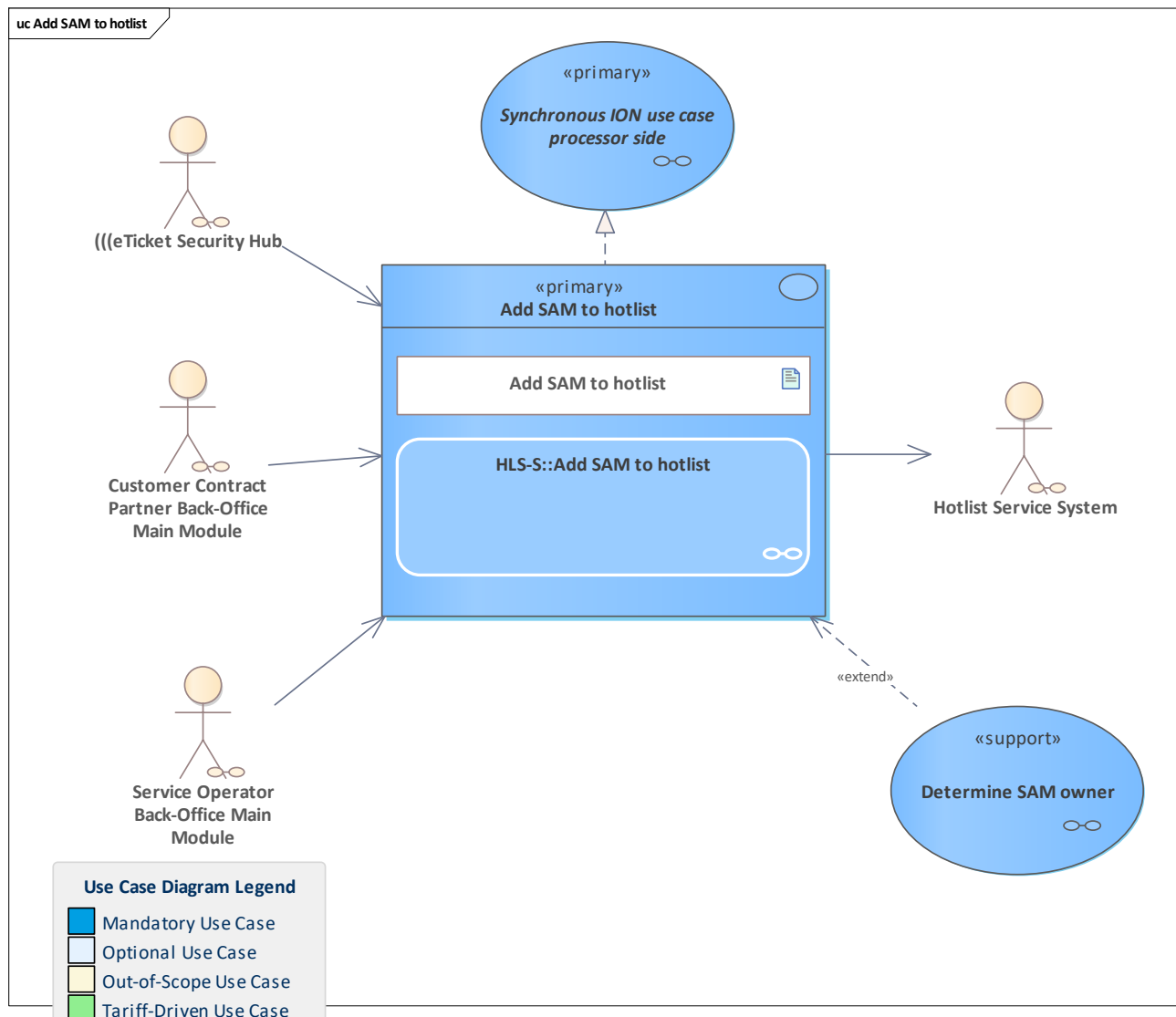


Figure 216: Add SAM to hotlist

Use case to add a SAM to the SAM hotlist either by the SAM owner (SO or CCP) or the scheme manager (ESH module).

The actor sends a request to add a SAM to the SAM hotlist.

The Hotlist service system checks the request. In the case of an SO or CCP, the hotlist service determines the SAM owner to verify the authorisation of the sender to add the SAM to the hotlist. If the sender is the scheme manager, the determination of the SAM owner is skipped. If all checks are passed, the hotlist service system will add the SAM to the hotlist.

Note: the SAM hotlist that is composed in the next hotlist cycle will contain the new hotlist entry. When the new hotlist is requested and distributed to the terminals, the new entry is taken into account when the check against the hotlist is performed.

Note: this process is irreversible since the SAM will also be terminated and its certificate revoked. This is done for security level-3 SAMs as well as for level-2 SAMs.

11.8 Analyse application history from contractual perspective

11.9 Analyse application history from contractual perspective

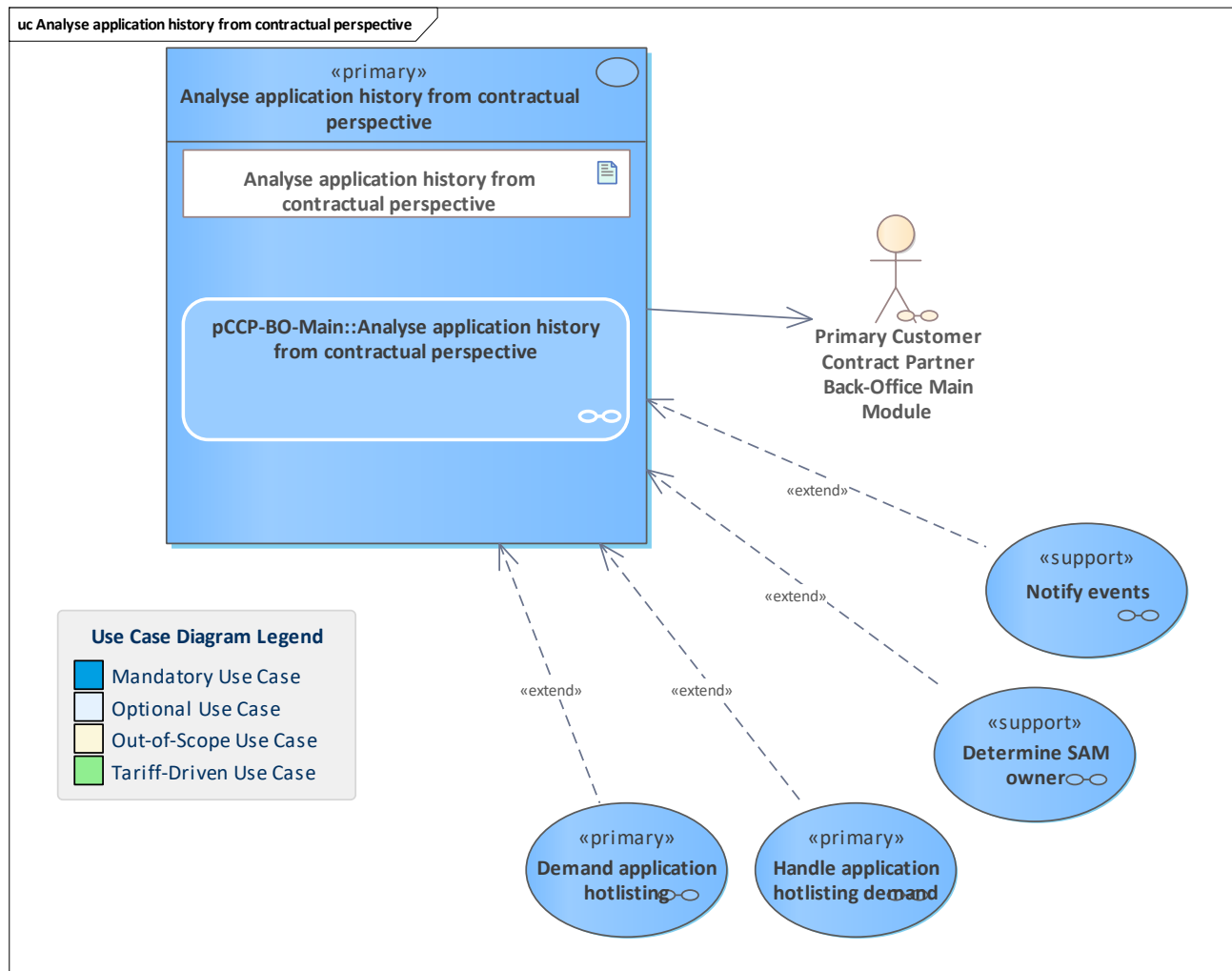


Figure 217: Analyse application history from contractual perspective

Analyse the application history for inconsistencies or gaps in their history. The pCCP needs to have all information about their applications, which means they need to have all notifications regarding them.

Every issue found shall only be reported once.

11.10 Analyse entitlement history from contractual perspective

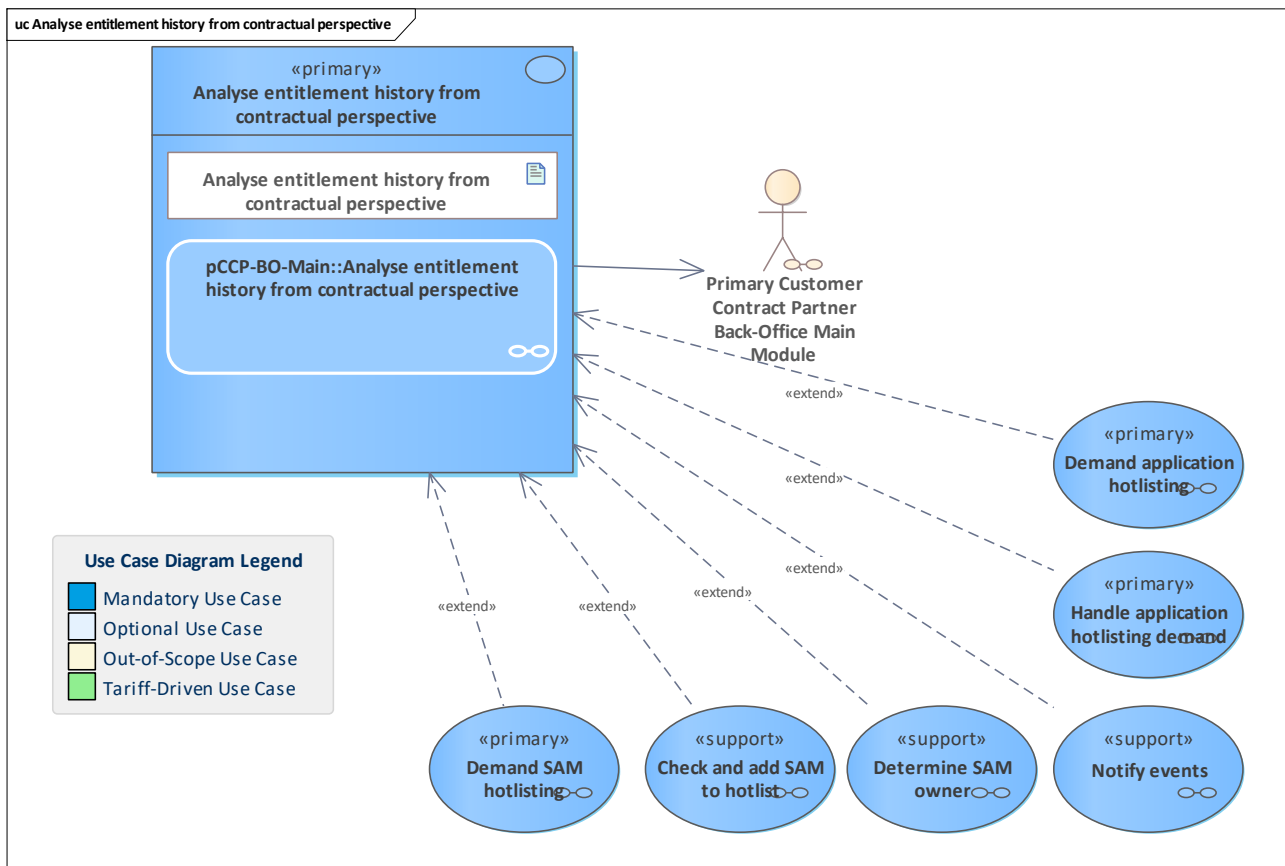


Figure 218: Analyse entitlement history from contractual perspective

Analyse the entitlement history for inconsistencies or gaps in the usage history. PO and pCCP both need to have all information about their entitlements, which means they need to have all notifications regarding them.

11.11 Analyse entitlement history from product perspective

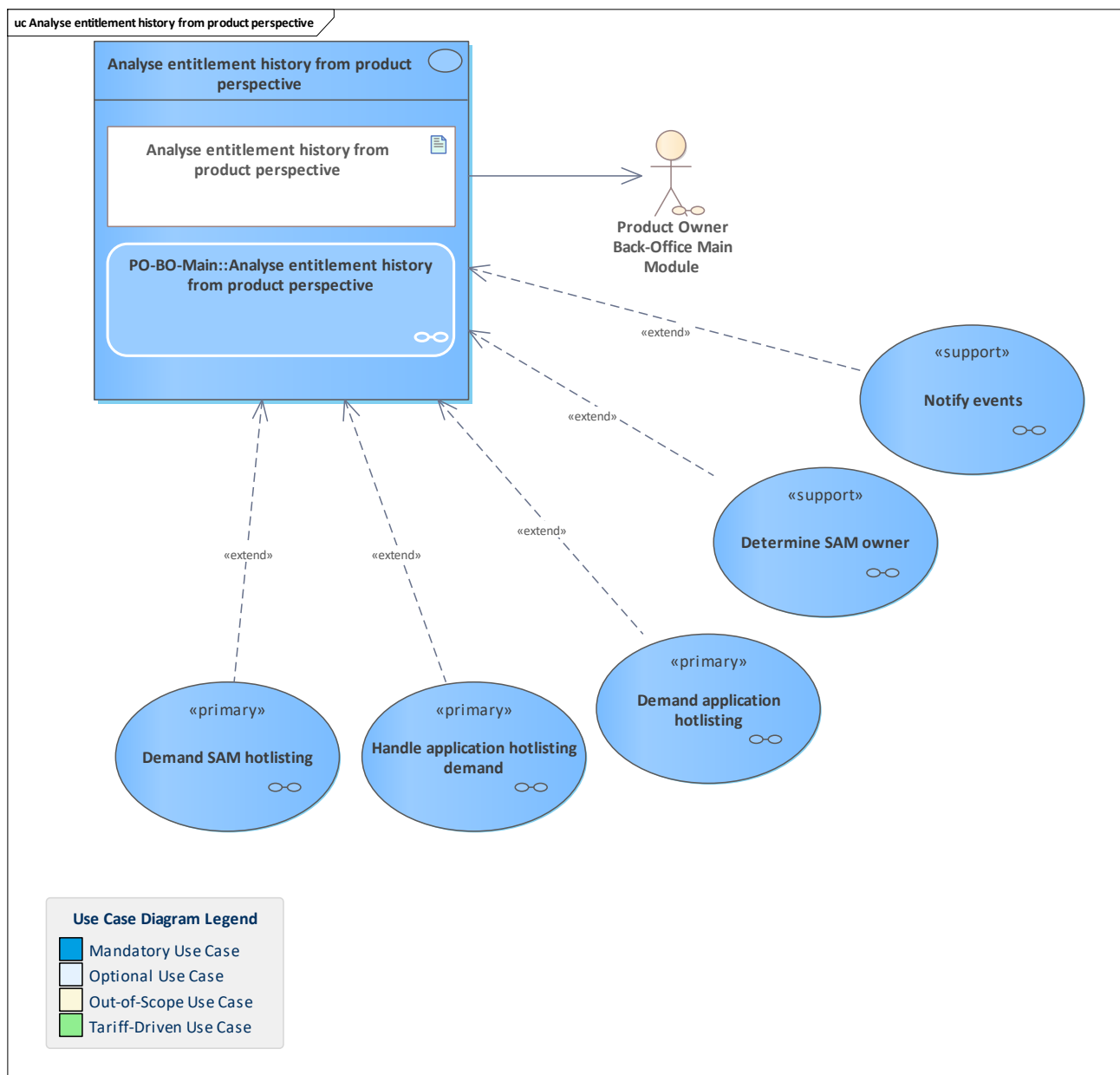


Figure 219: Analyse entitlement history from product perspective

Analyse the entitlement history for inconsistencies or gaps in the usage history. PO and pCCP both need to have all information about their entitlements, which means they need to have all notifications regarding them. Every issue found shall only be reported once.

11.12 Analyse order history from contractual perspective

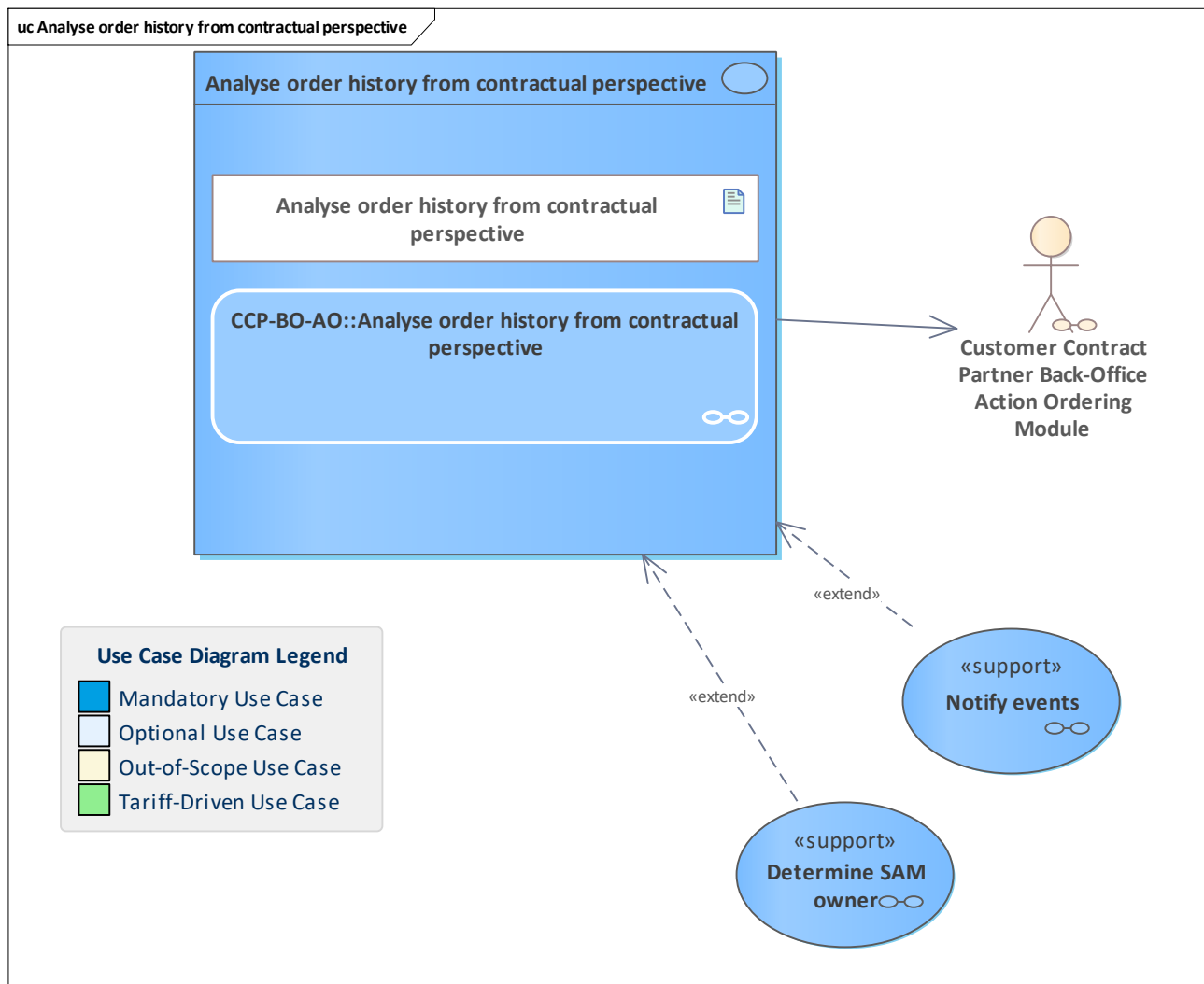


Figure 220: Analyse order history from contractual perspective

The notifications related to action orders are analysed for correctness.

11.13 Analyse order history from product perspective

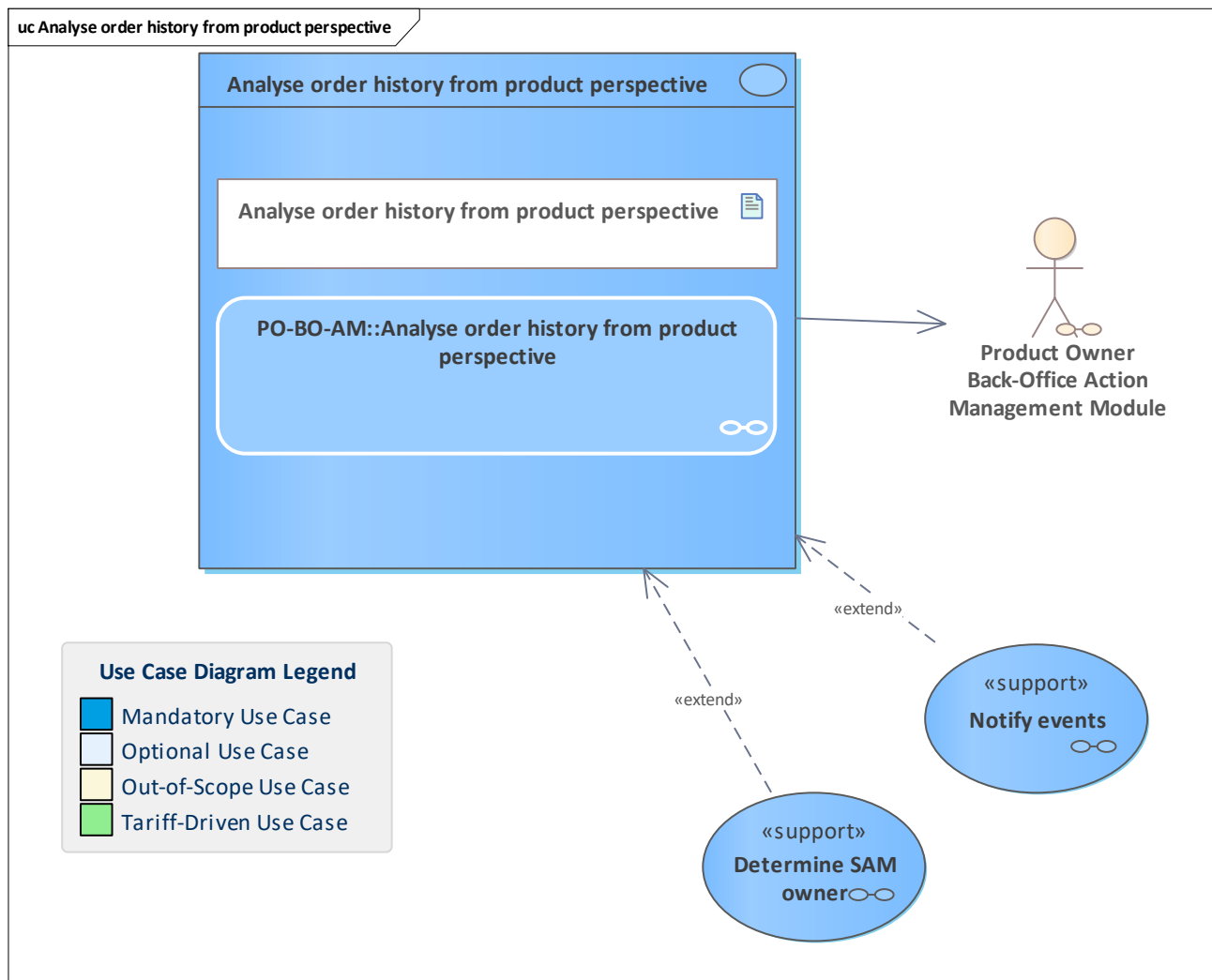


Figure 221: Analyse order history from product perspective

The notifications related to action orders are analysed for correctness.

11.14 Authorise static entitlement

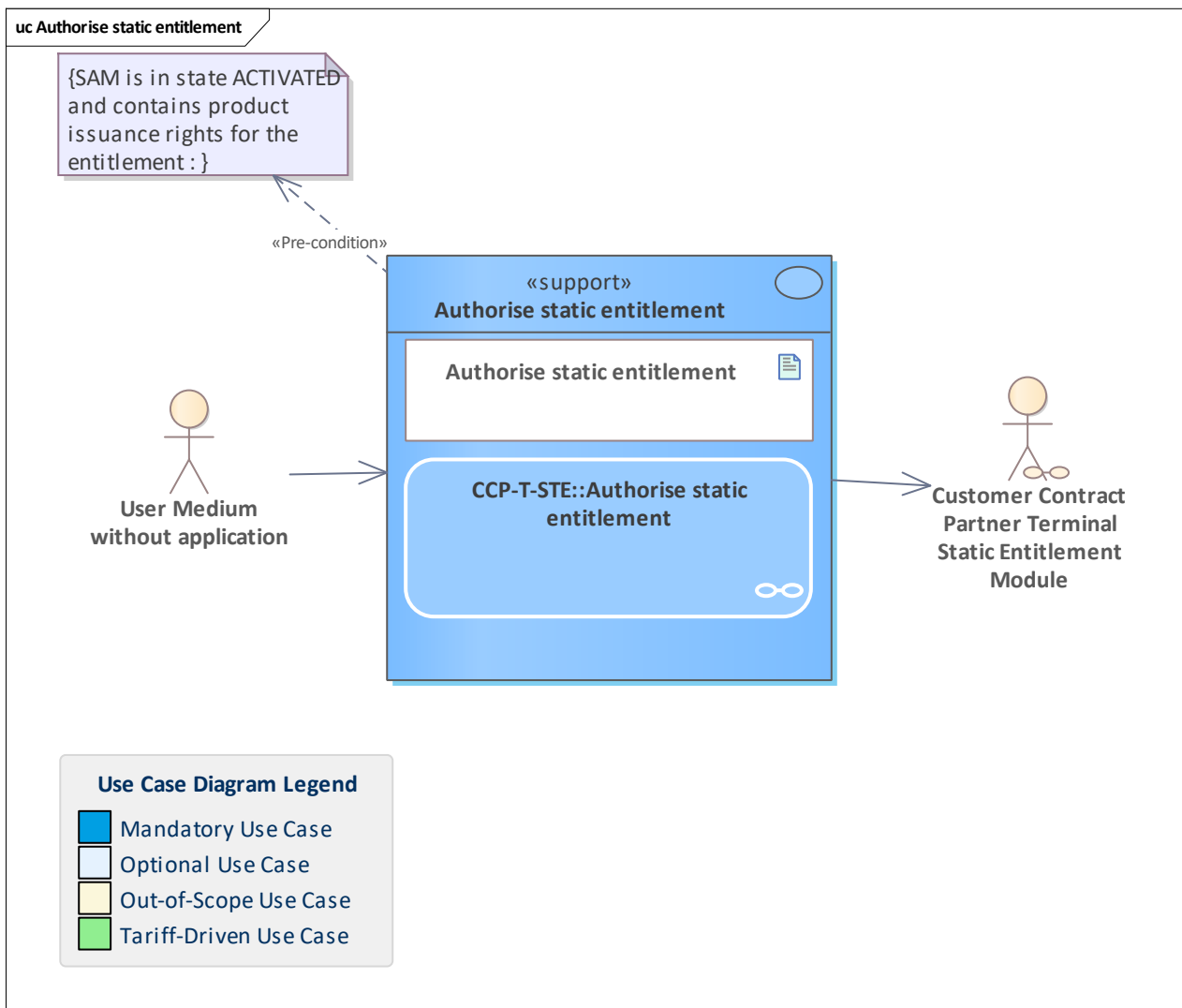


Figure 222: Authorise static entitlement

The customer contract partner terminal creates a static entitlement using a SAM. Supporting use case that performs the creation without the output to a user medium (mobile device, paper, etc.).

Note that this process only puts a single static entitlement data object into the static entitlements object. By running this process multiple times using the same SAM and afterwards merging the static entitlements objects, more than one entitlement (i.e. static entitlement data object) can be put into a single static entitlements object.

11.15 Autoload stored-value payment method

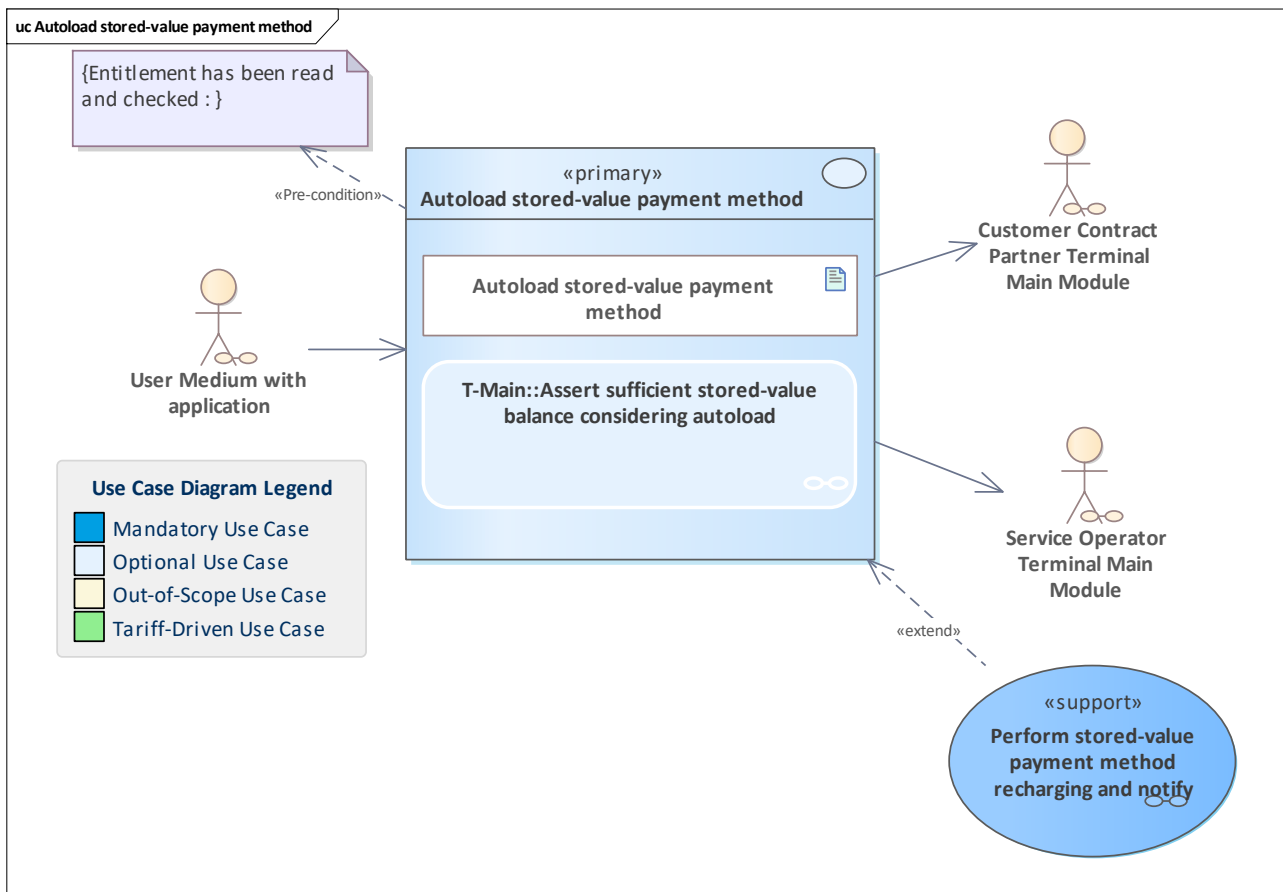


Figure 223: Autoload stored-value payment method

The terminal checks if sufficient stored-value funds are available and, if necessary, automatically triggers a recharge process if autoload was contractually agreed upon. In a CICO environment, the autoload option can also be available in SO terminals. In this case, the terminal must have a SAM with sufficient rights.

11.16 Block entitlement triggered by action order

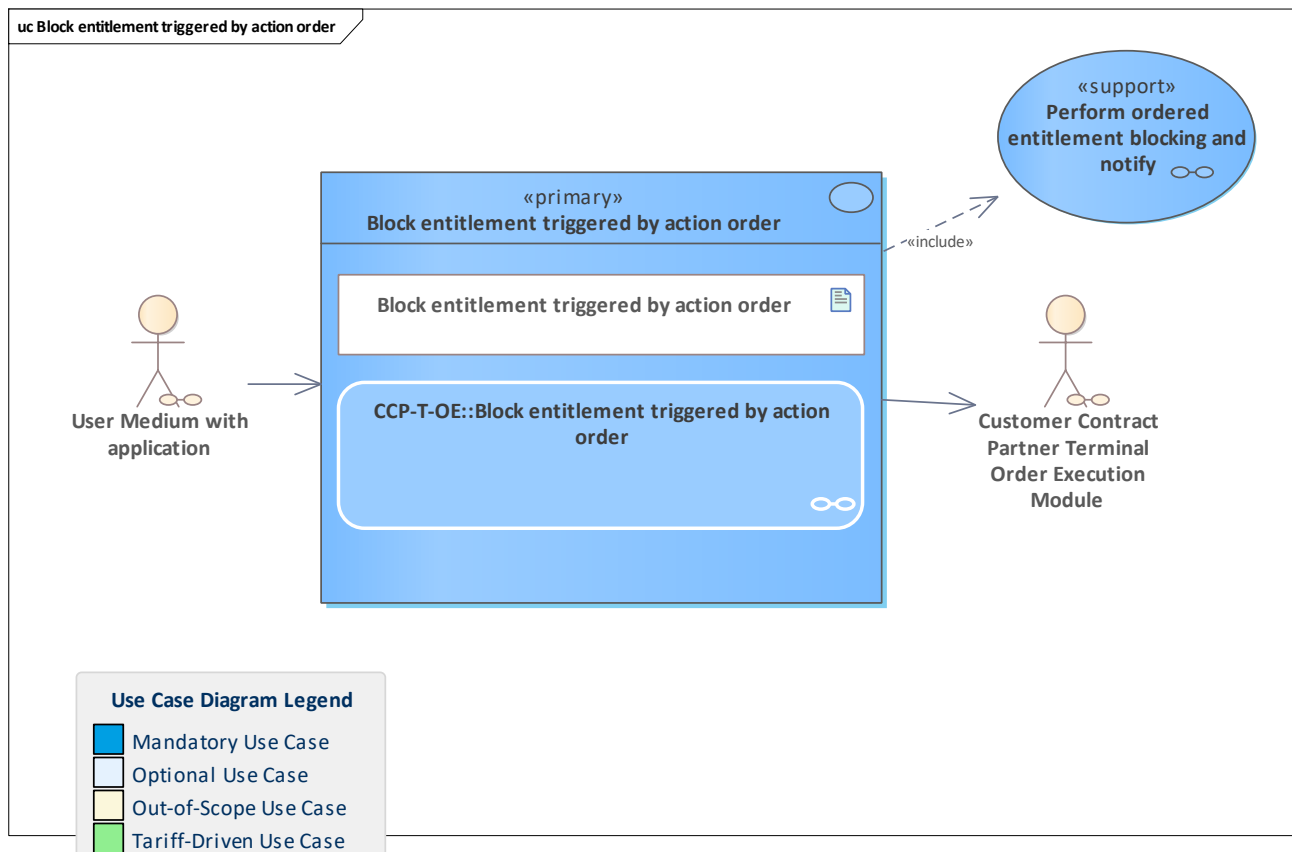


Figure 224: Block entitlement triggered by action order

An potentially relevant entitlement blocking order for a certain user medium is checked regarding the need to execute it and, if necessary, is executed.

To achieve this, the terminal does a lookup in the action list for the application instance ID. If a match is found, the entitlements are examined concerning their potential order ID. If the order ID matches with the ordered blocking action in the action list and the entitlement is not in the state [Entitlement blocked](#), the blocking is performed.

This use case aims at blocking an entitlement issued via action management which an unknown ID. Thus, a regular hotlist entry in the hotlist service is not possible.

Note: when an entitlement is blocked due to an action list entry, no extended logging regarding its status should be created during that same interaction. Extended logging regarding entitlement status should only be created for entitlements that were already invalid per status before the interaction with a terminal.

Note that this is not relevant for entitlements terminated via action list entries since these are also removed from the user medium application.

11.17 Cancel order

11.18 Cancel order

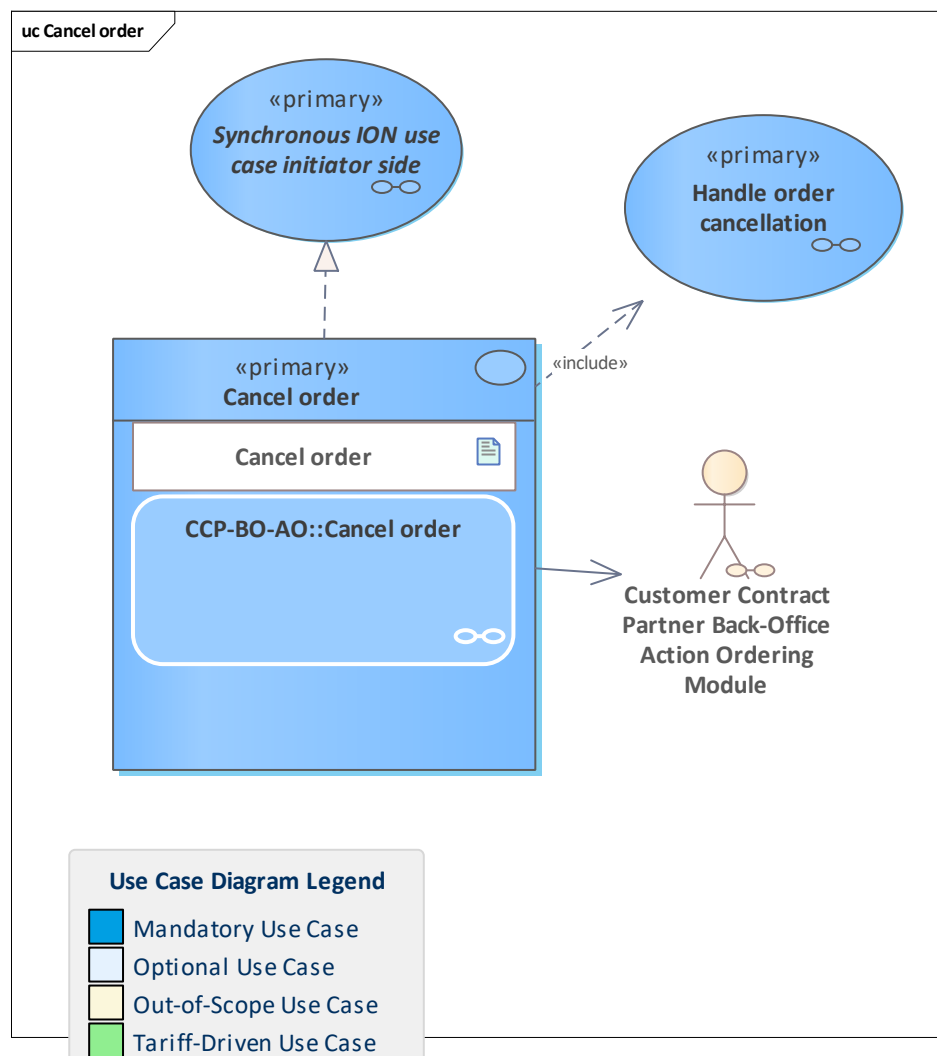


Figure 225: Cancel order

The ordering CCP cancels an order.

Cancelling an order will remove it from future action lists, but is not able to prevent the order from being executed before the next action list has reached all relevant terminals. In particular, the order may already have been executed and the execution notification may still be on its way to the relevant systems.

11.19 Change customer and discounts

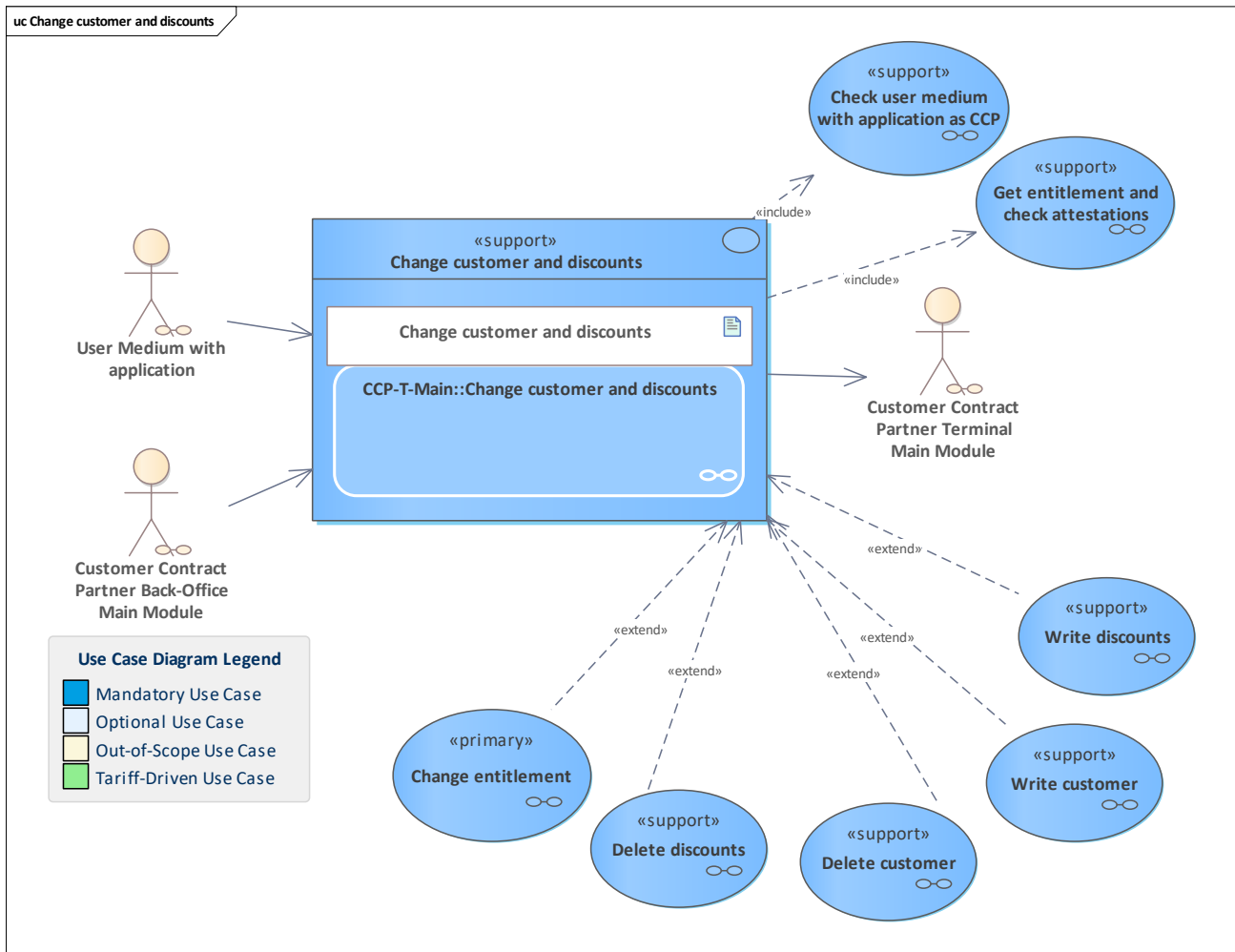


Figure 226: Change customer and discounts

Supporting use case of the CCP terminal triggered by the back-office system. Customer and discounts are changed on the user medium with application.

11.20 Change entitlement

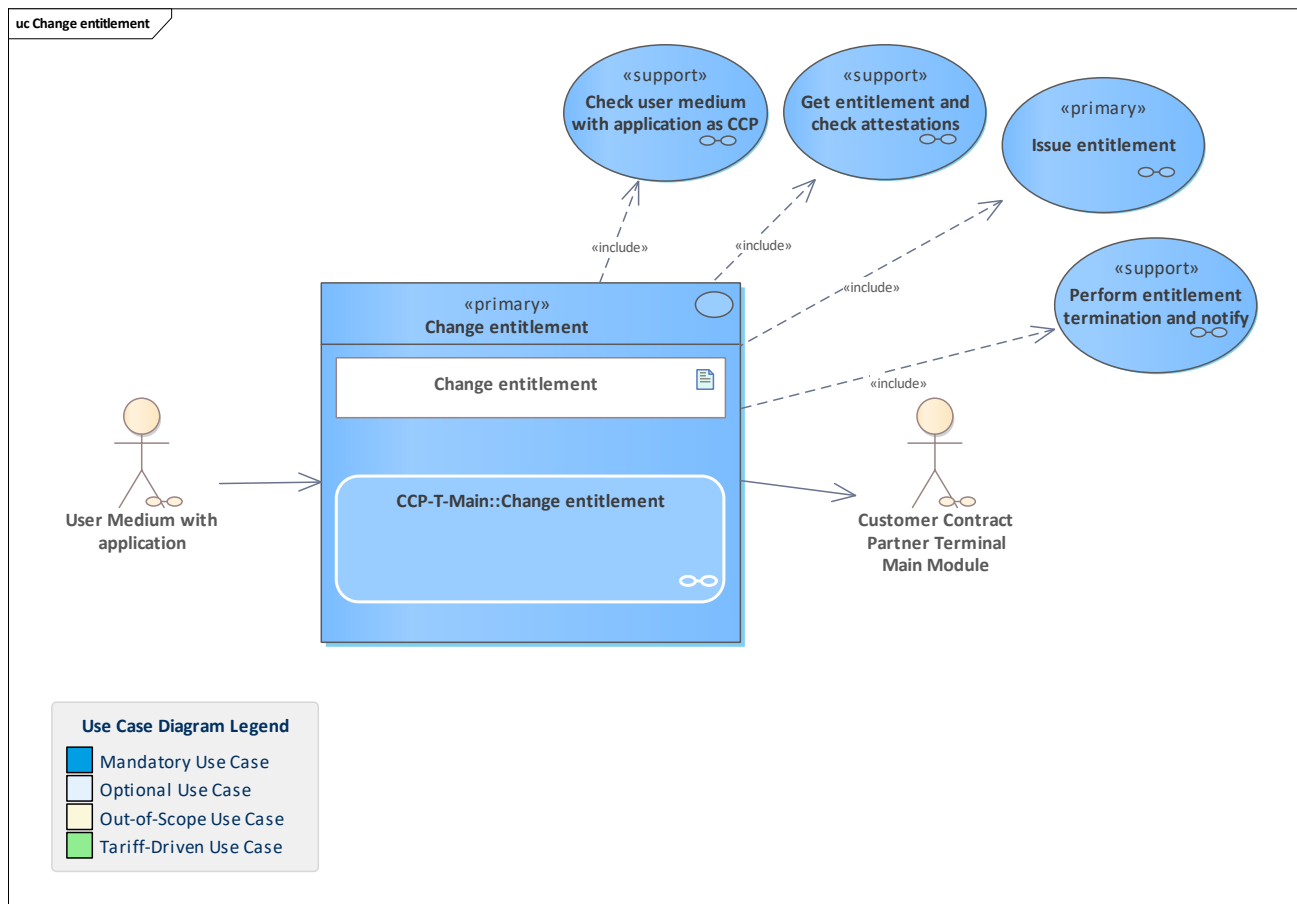


Figure 227: Change entitlement

An entitlement needs to be replaced with a new one, for example, due to changed product parameters.

Please note that it is assumed that there is no need for any payment transaction.

11.21 Change favourites

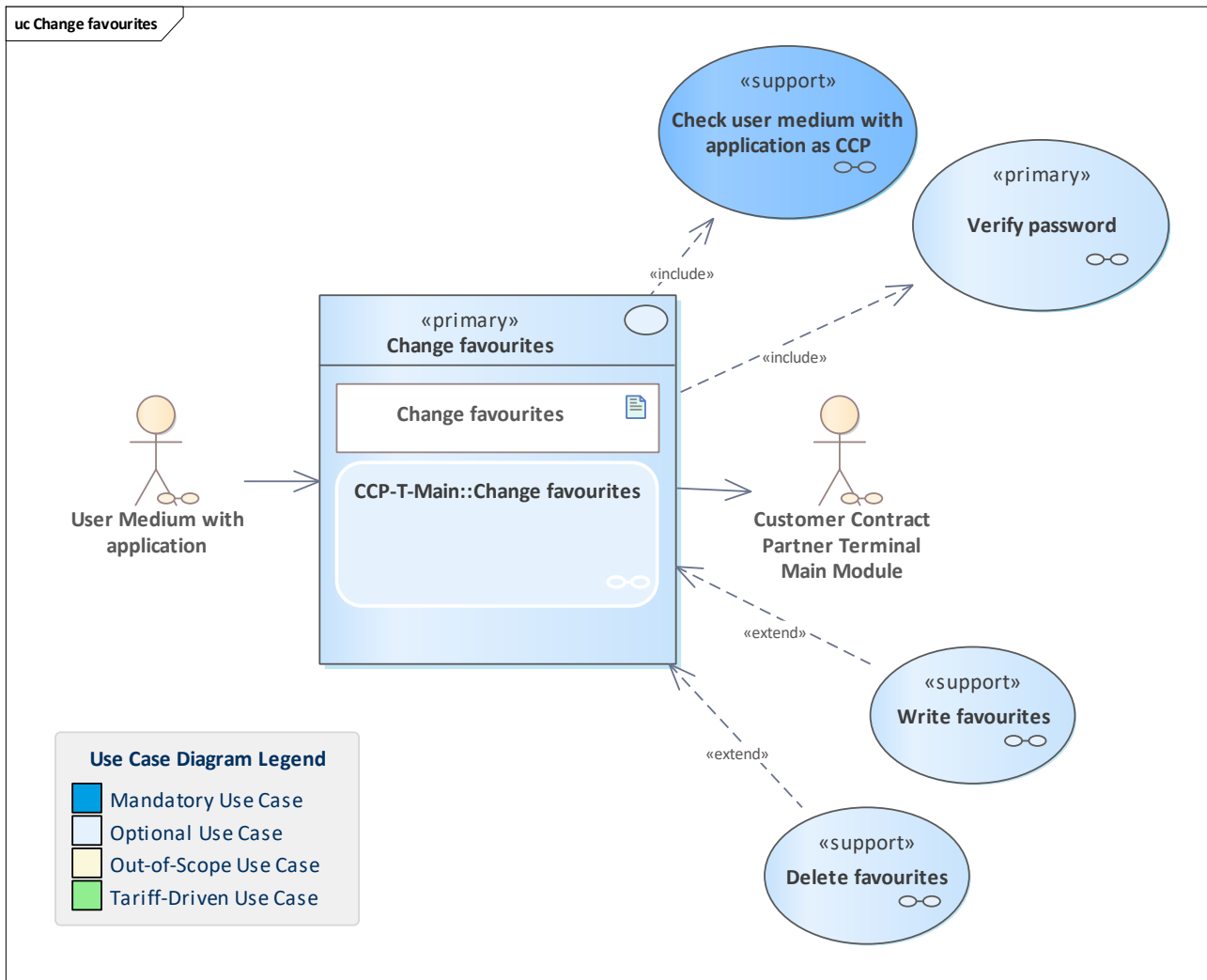


Figure 228: Change favourites

The customer wants to change his favourites aided by a CCP-T.
The customer must enter his password/PIN to change the favourites.

11.22 Change password

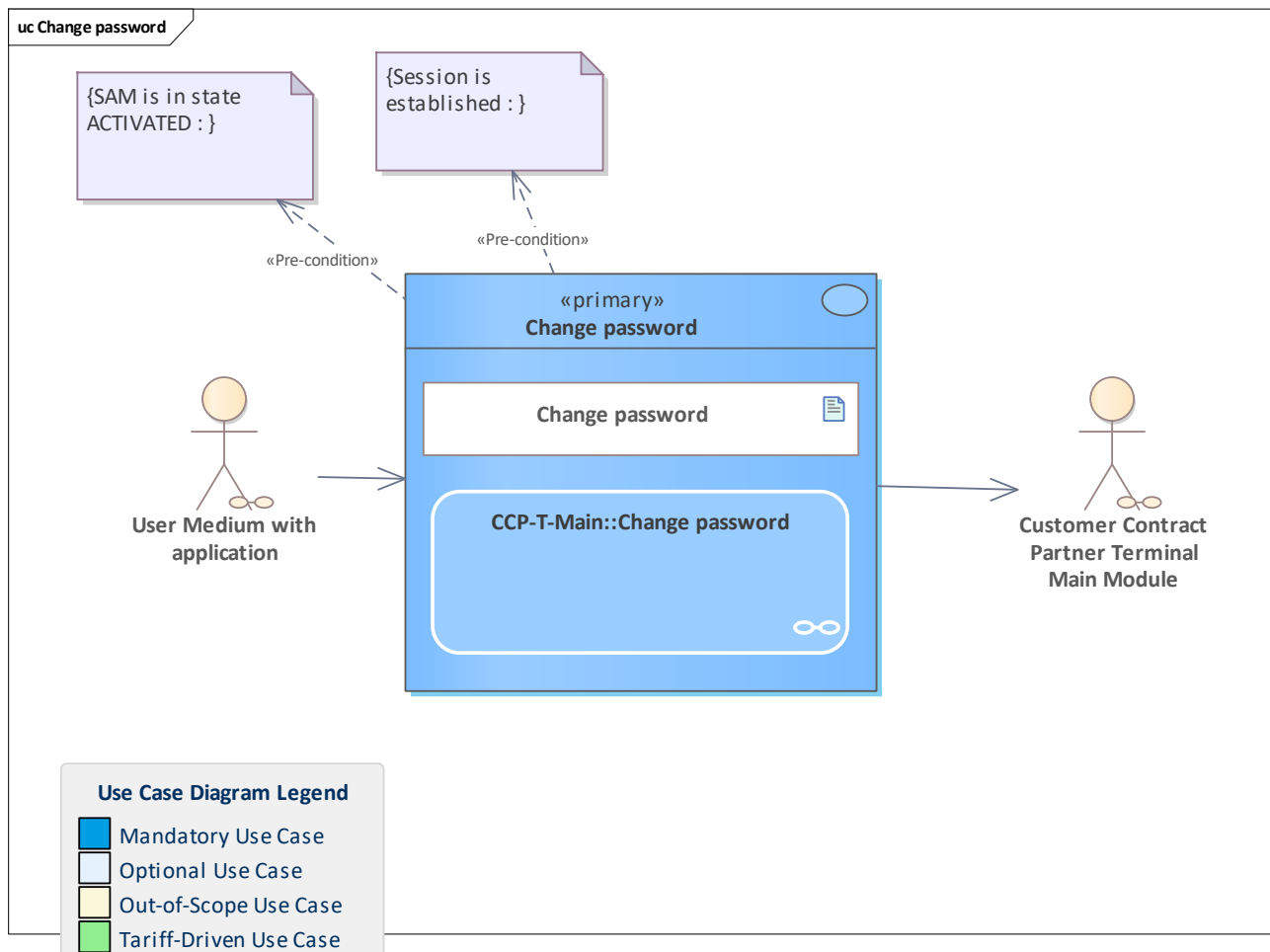


Figure 229: Change password

An end customer changes the password of a user medium with application.

11.23 Change static entitlement

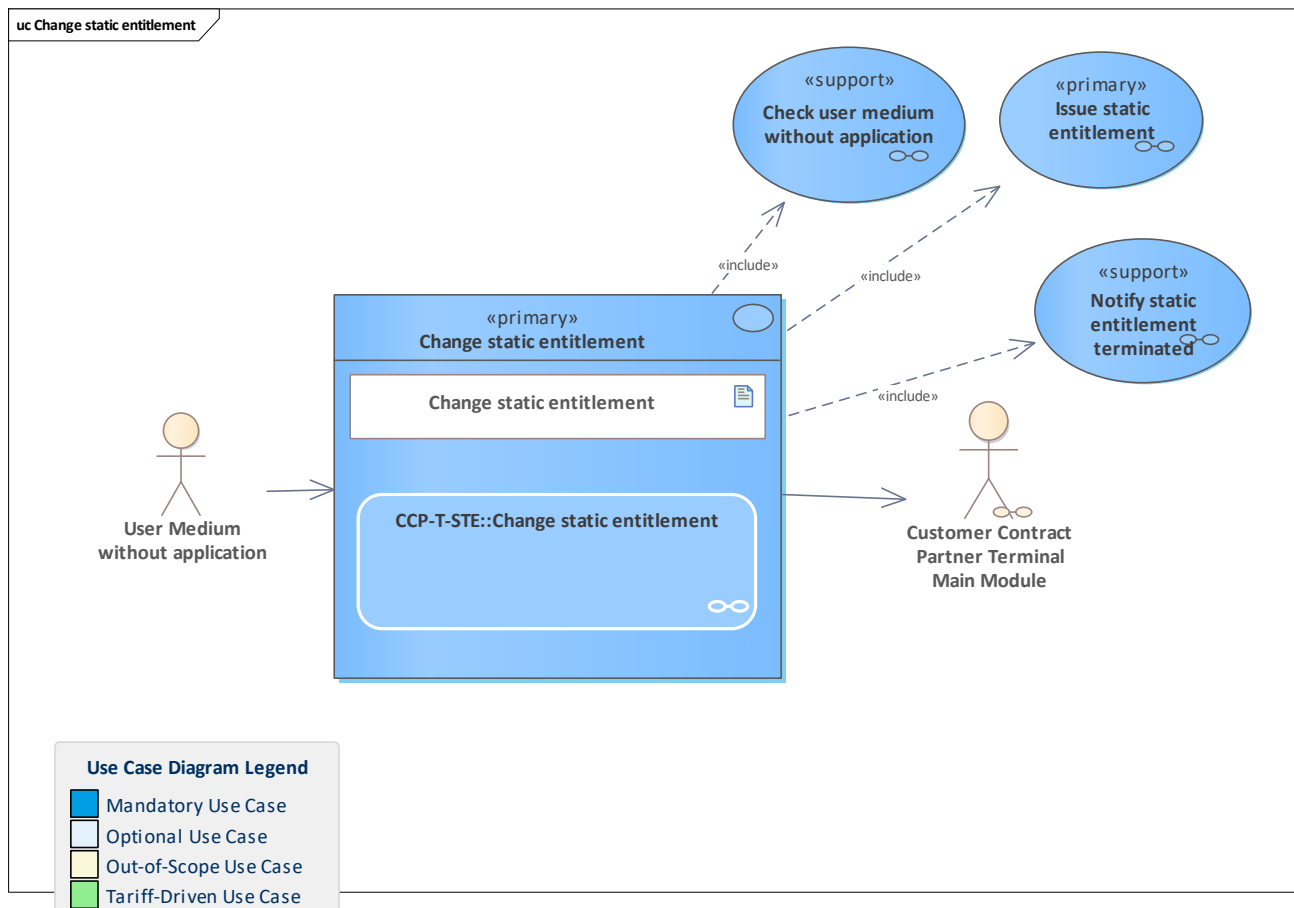


Figure 230: Change static entitlement

A static entitlement needs to be replaced with a new one, for example, due to changed product parameters.

Please note that it is assumed that there is no need for any payment transaction.

11.24 Change user tariff parameters

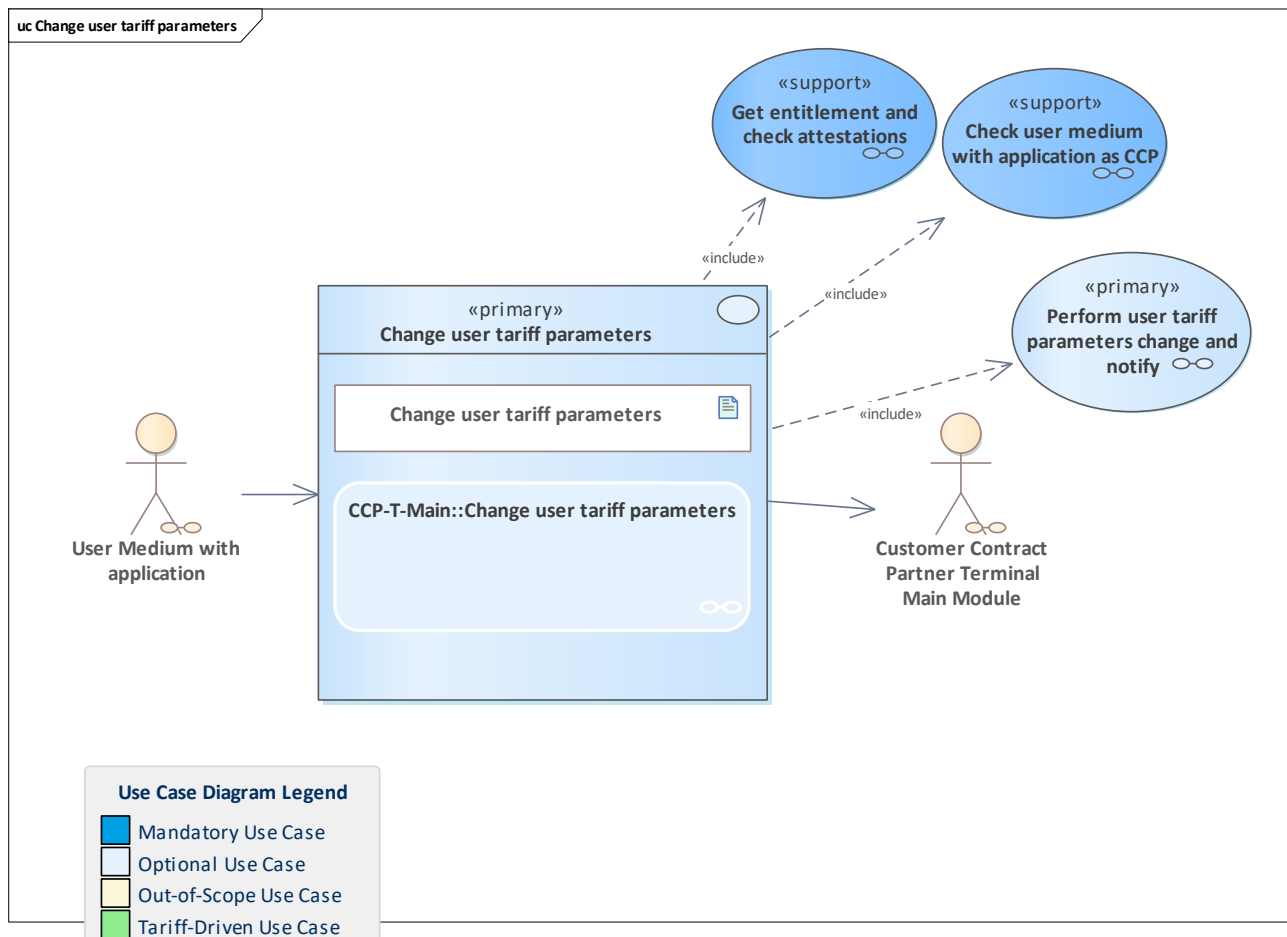


Figure 231: Change user tariff parameters

A customer can change user tariff parameters (UTP) such as transport rules or service class for a certain entitlement.

This use case is applicable only for the CICO process.

Please note that, UTP should be changed before checking in. If UTP are allowed to be changed during the journey from a tariff perspective, the terminal must check out, change UTP and check in again.

There is no validity period for changed user tariff parameters.

As long as the UTP remain in the changed version, they are passed through the terminals and copied to the next recording transaction.

A customer must roll back the changes at a CCP-T if requested.

11.25 Check and add SAM to hotlist

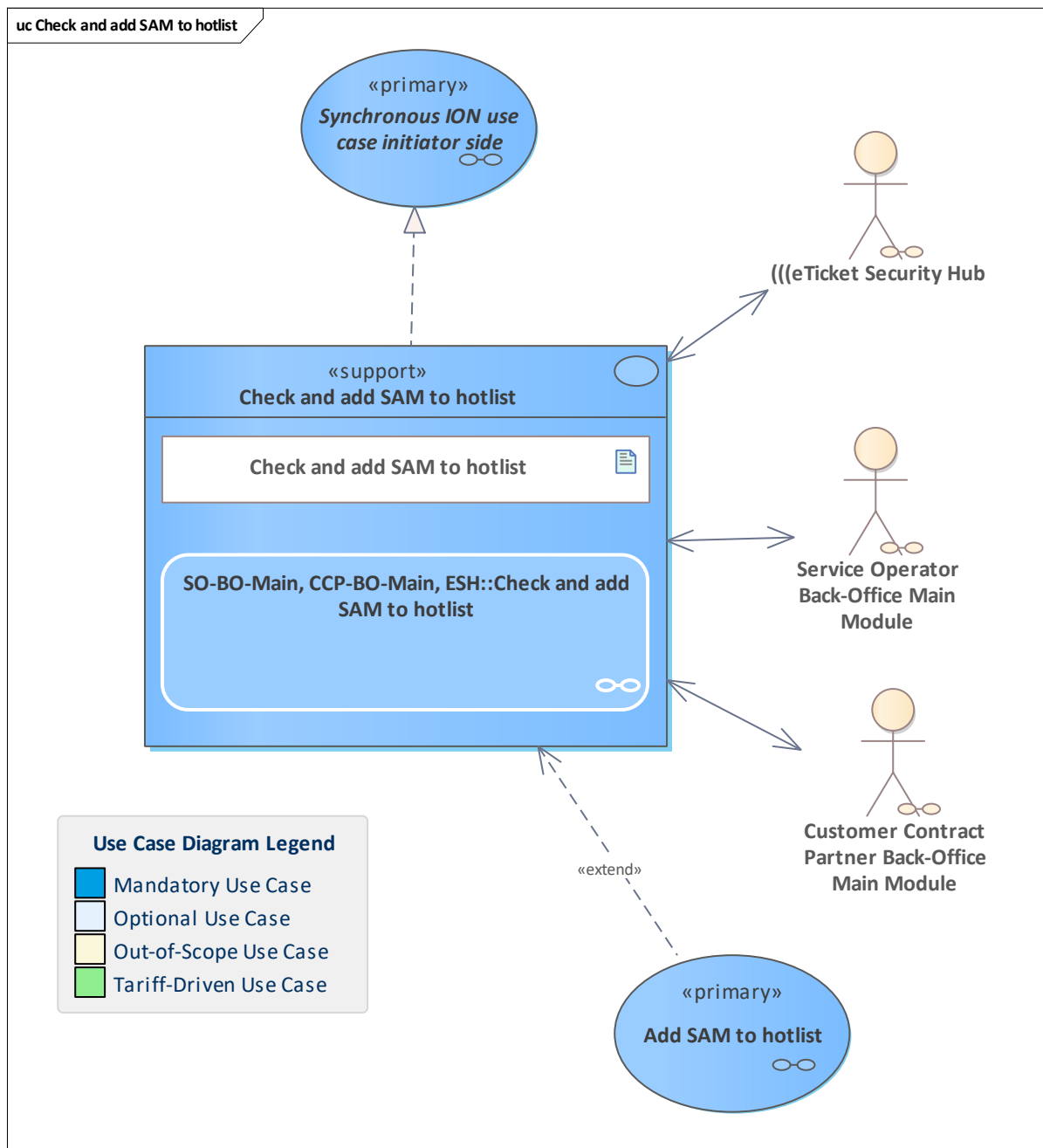


Figure 232: Check and add SAM to hotlist

Supporting use case for the Scheme Manager, SO or CCP.

The parameters required to add a SAM to the hotlist are checked, especially if another system has already requested the SAM to be added to the hotlist. In addition, the SAM owner may add specific counter information that is not available to a third party system.

The hotlist service system is requested to add the SAM to the SAM hotlist.

11.26 Check entitlement notifications against issuance notification from contractual perspective

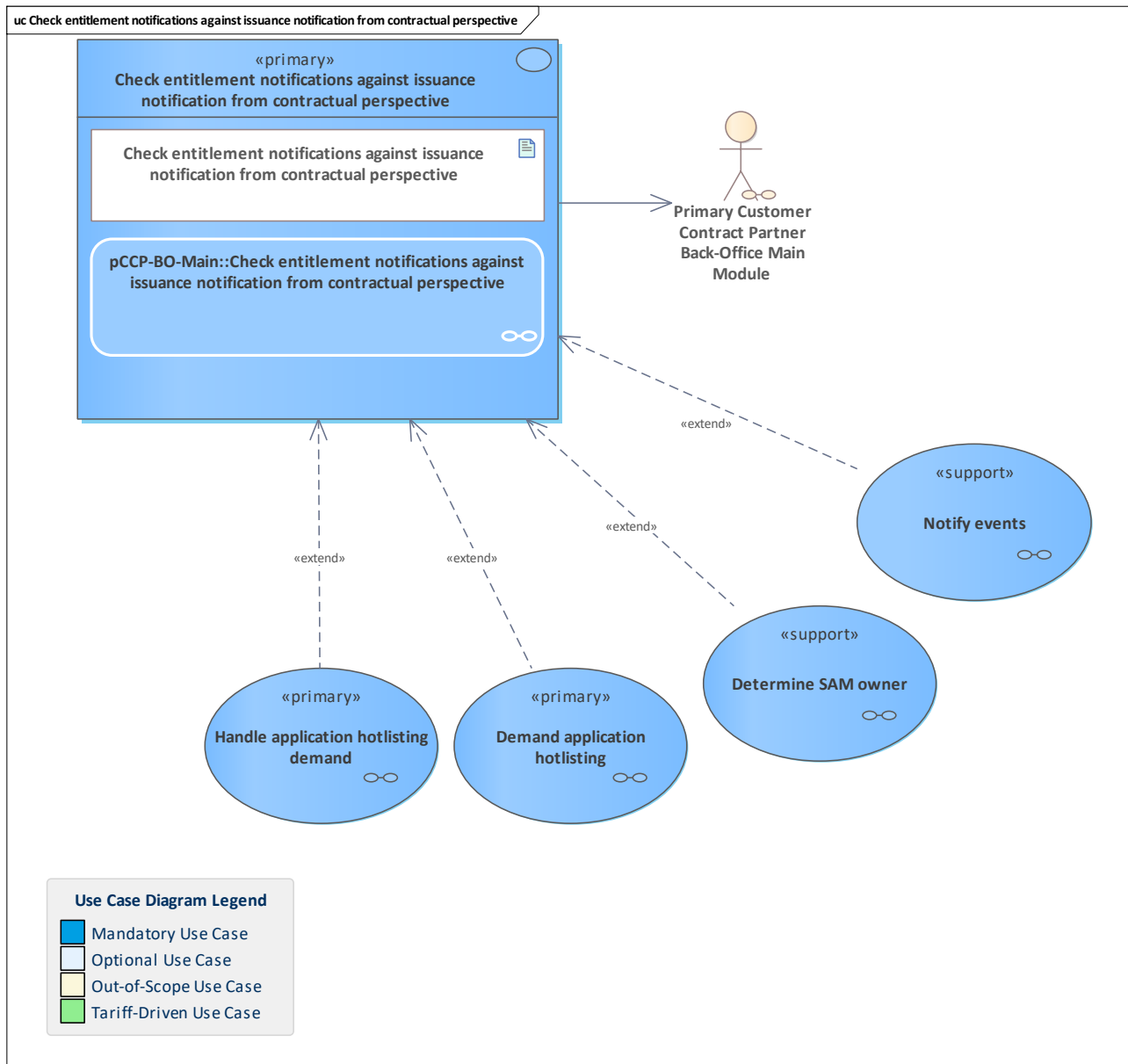


Figure 233: Check entitlement notifications against issuance notification from contractual perspective

Non-issuing entitlement notifications need to be checked against details that are only contained in the issuance notification, e.g., the entitlement validity period. This also makes sure that every entitlement seen live is known to the system / has been reported as issued.

11.27 Check entitlement notifications against issuance notification from product perspective

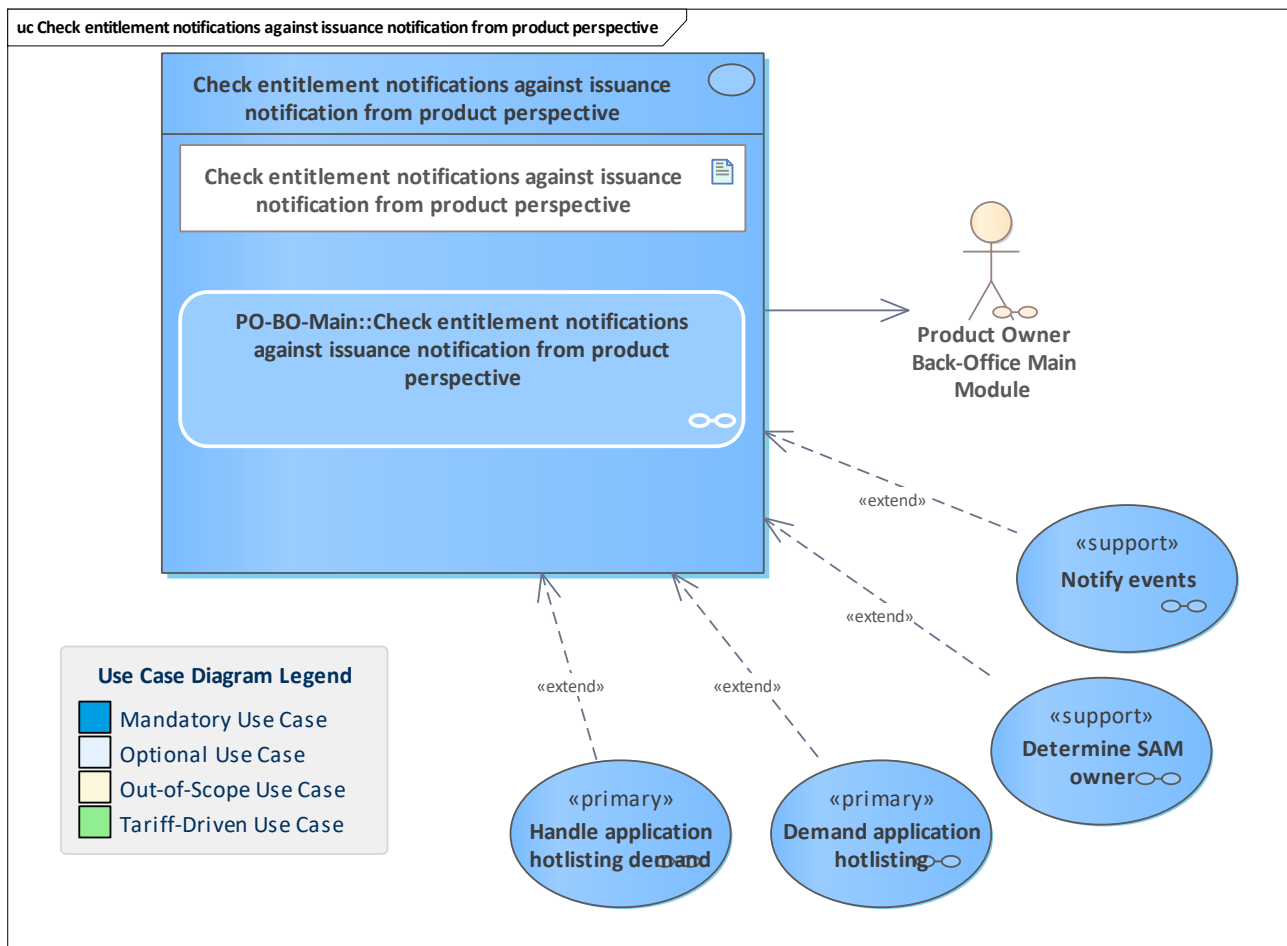


Figure 234: Check entitlement notifications against issuance notification from product perspective

Non-issuing entitlement notifications need to be checked against details that are only contained in the issuance notification, e.g., the entitlement validity period. This also makes sure that every entitlement seen live is known to the system / has been reported as issued.

11.28 Check for order obsolescence

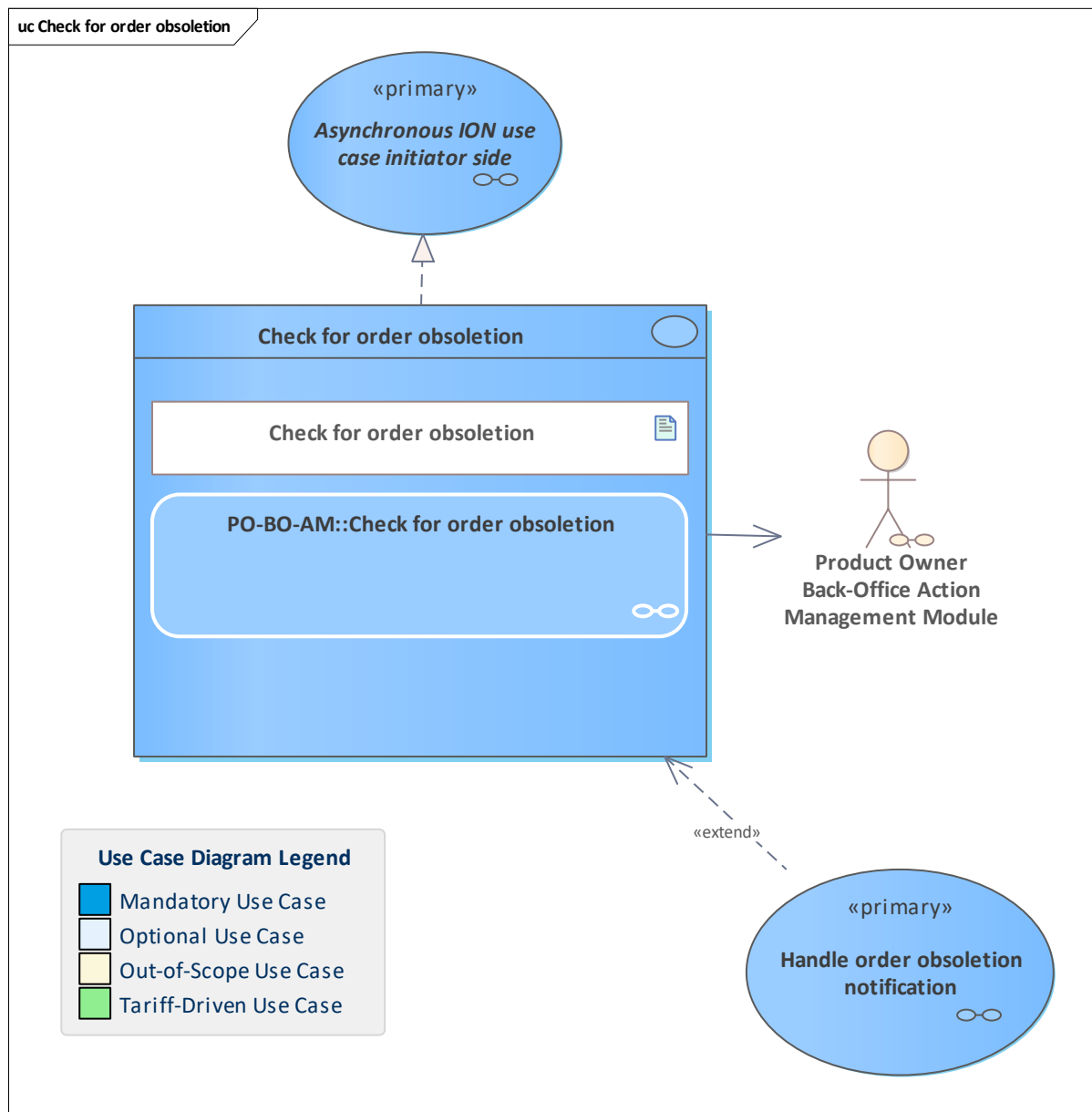


Figure 235: Check for order obsolescence

Check if an entitlement notification obsoletes an [Active](#) order for unblocking an entitlement. Such an order is marked as [Obsolete](#) and the ordering CCP is notified about it. Note that orders in state [Obsolete](#) are no longer part of the distributed action lists.

11.29 Check MOTICS requirements

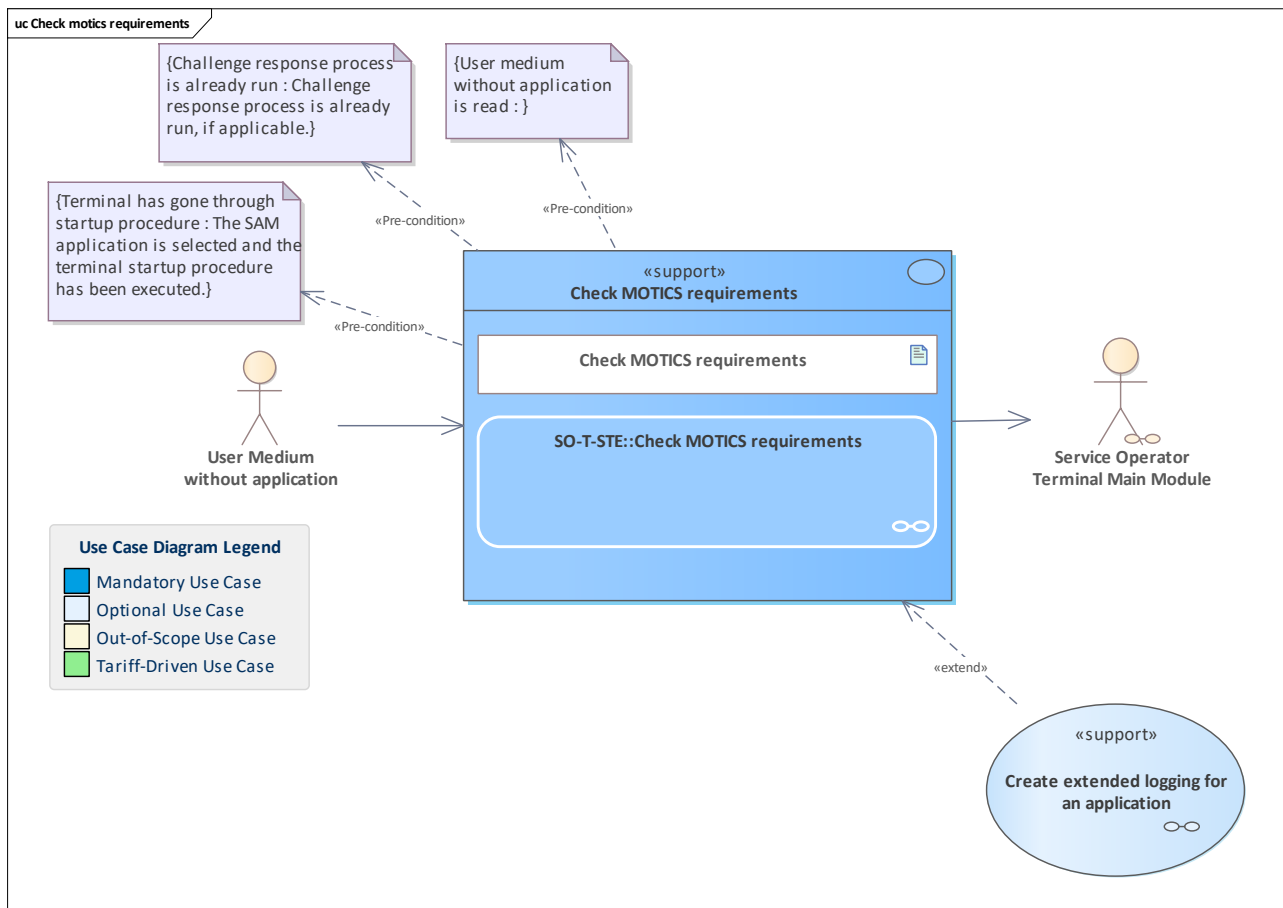


Figure 236: Check motics requirements

In this use case, the process of verifying the copy protection mechanism is defined within the framework of a ticket inspection or validation.

11.30 Check static entitlement notifications against issuance notification from contractual perspective

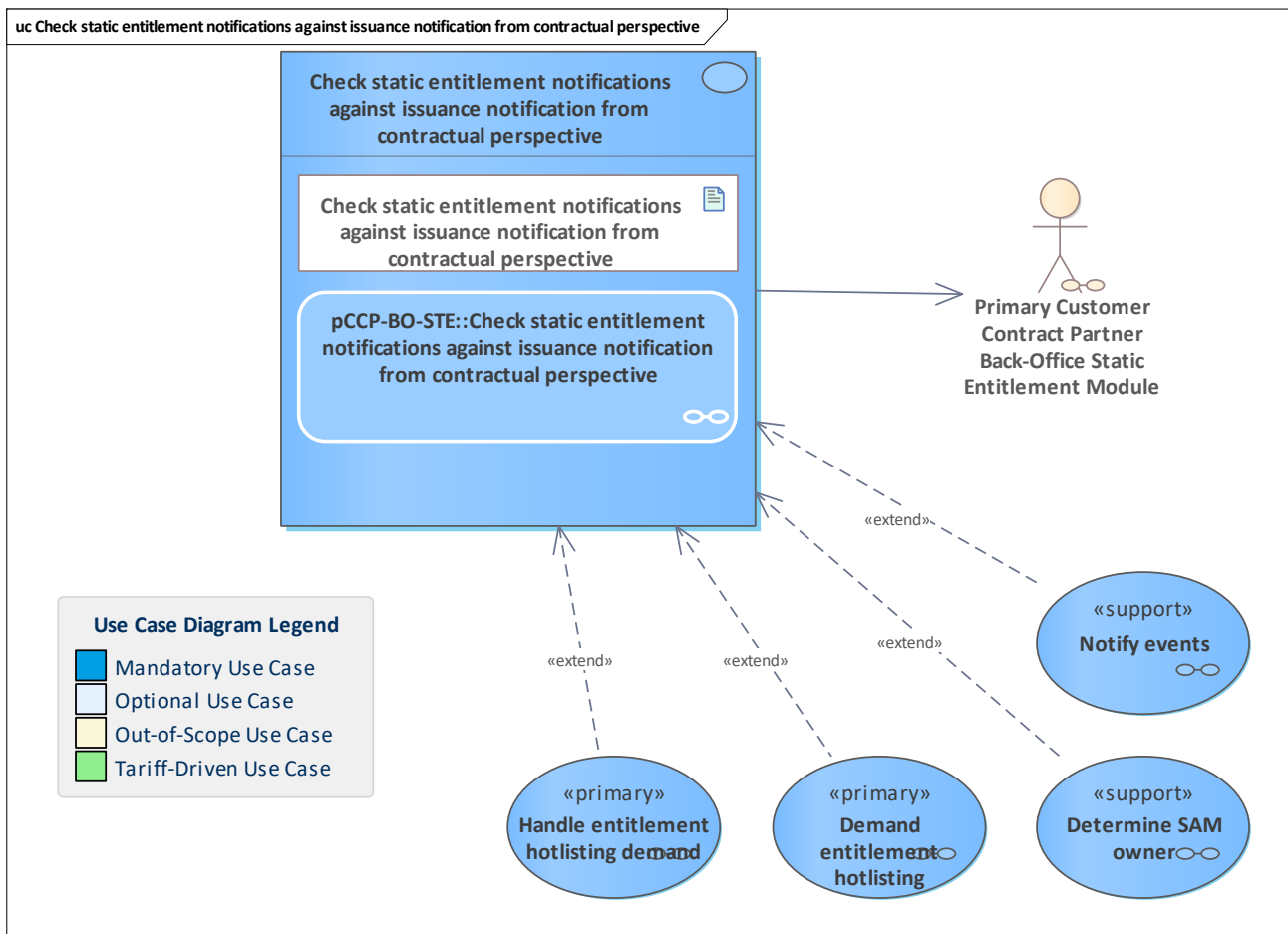


Figure 237: Check static entitlement notifications against issuance notification from contractual perspective

Non-issuing static entitlement notifications need to be checked for consistency with their issuance notification, e.g., the entitlement validity period. This also makes sure that every entitlement seen live is known to the system or has been reported as issued.

11.31 Check static entitlement notifications against issuance notification from product perspective

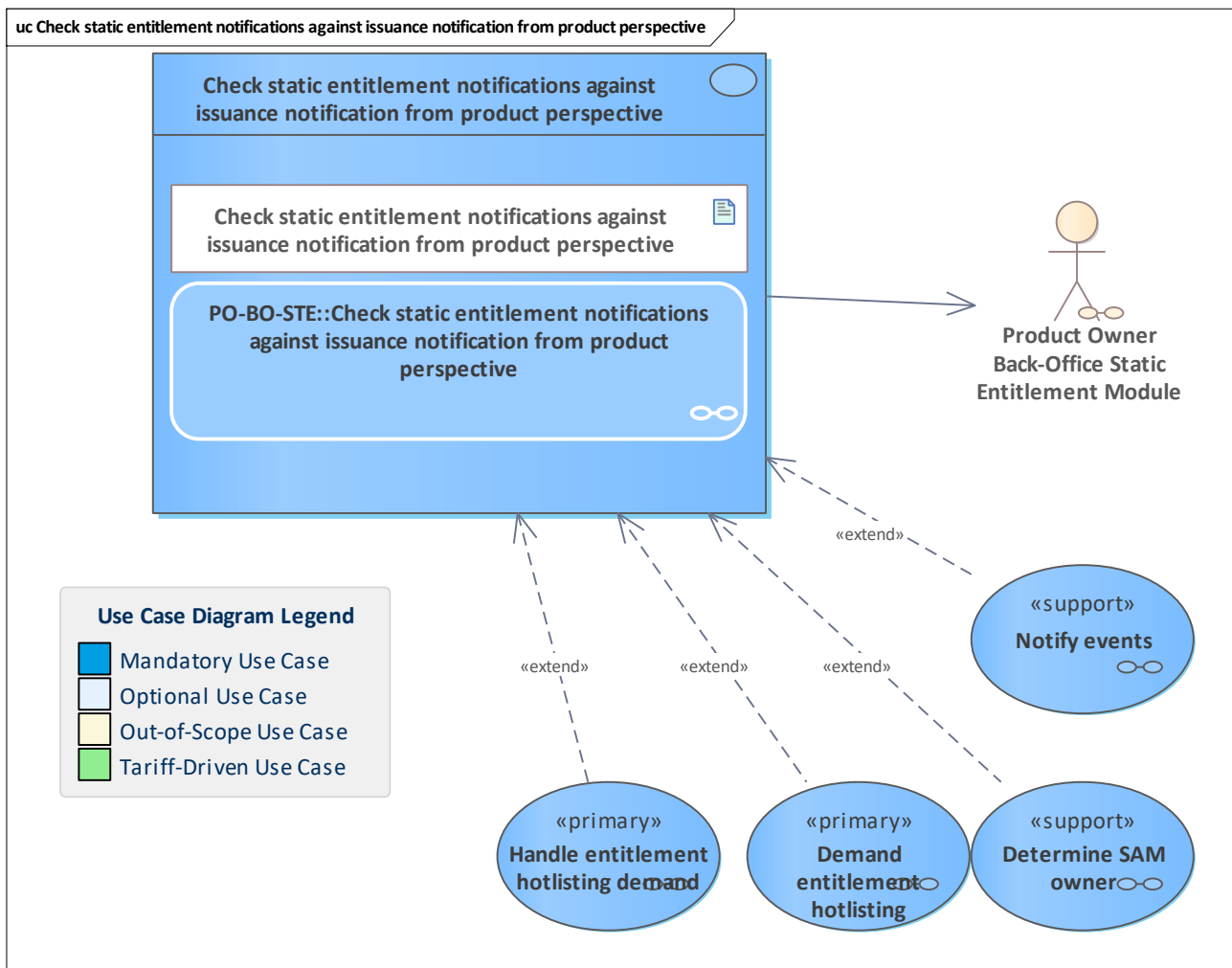


Figure 238: Check static entitlement notifications against issuance notification from product perspective

Non-issuing static entitlement notifications need to be checked for consistency with their issuance notification, e.g., the entitlement validity period. This also makes sure that every entitlement seen live is known to the system or has been reported as issued.

11.32 Check static entitlement notifications for plausibility

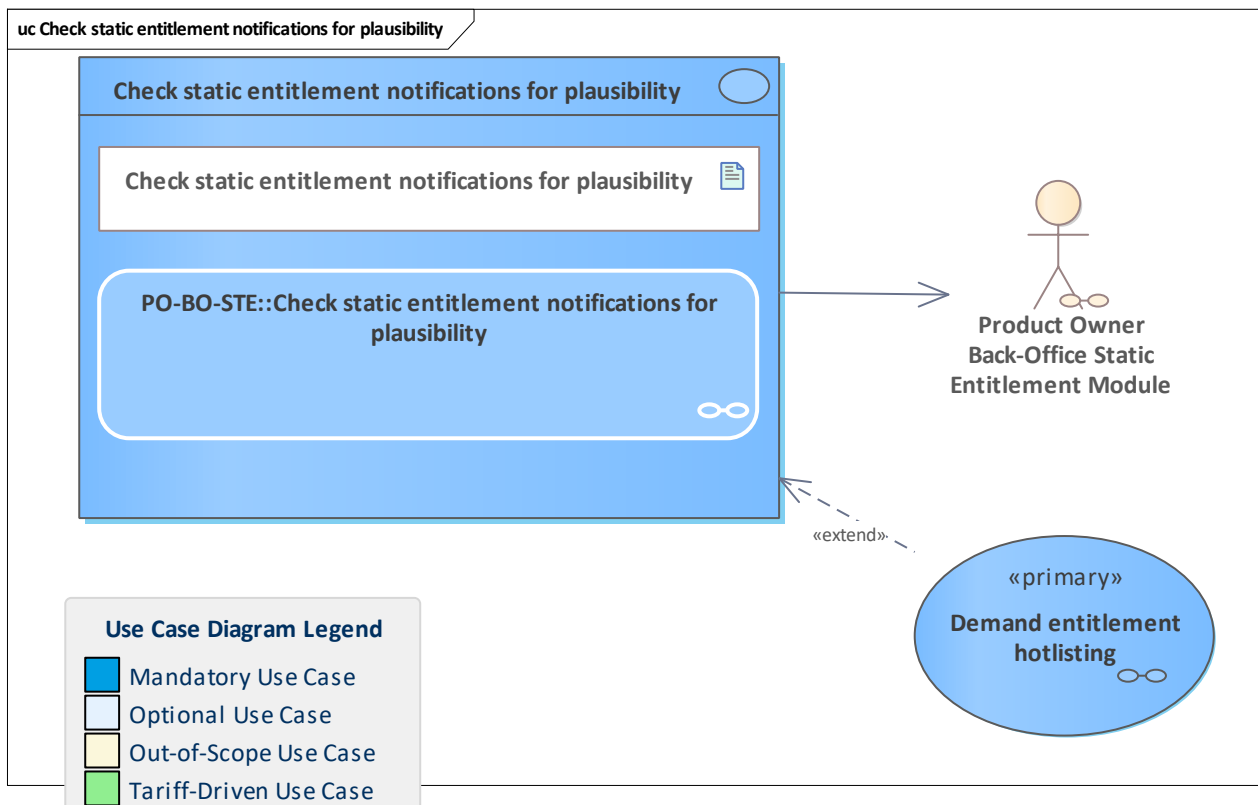


Figure 239: Check static entitlement notifications for plausibility

Check static entitlement notifications for plausibility to detect illegal copies and similar fraud scenarios.

11.33 Check user medium with application as CCP

11.34 Check user medium with application as CCP

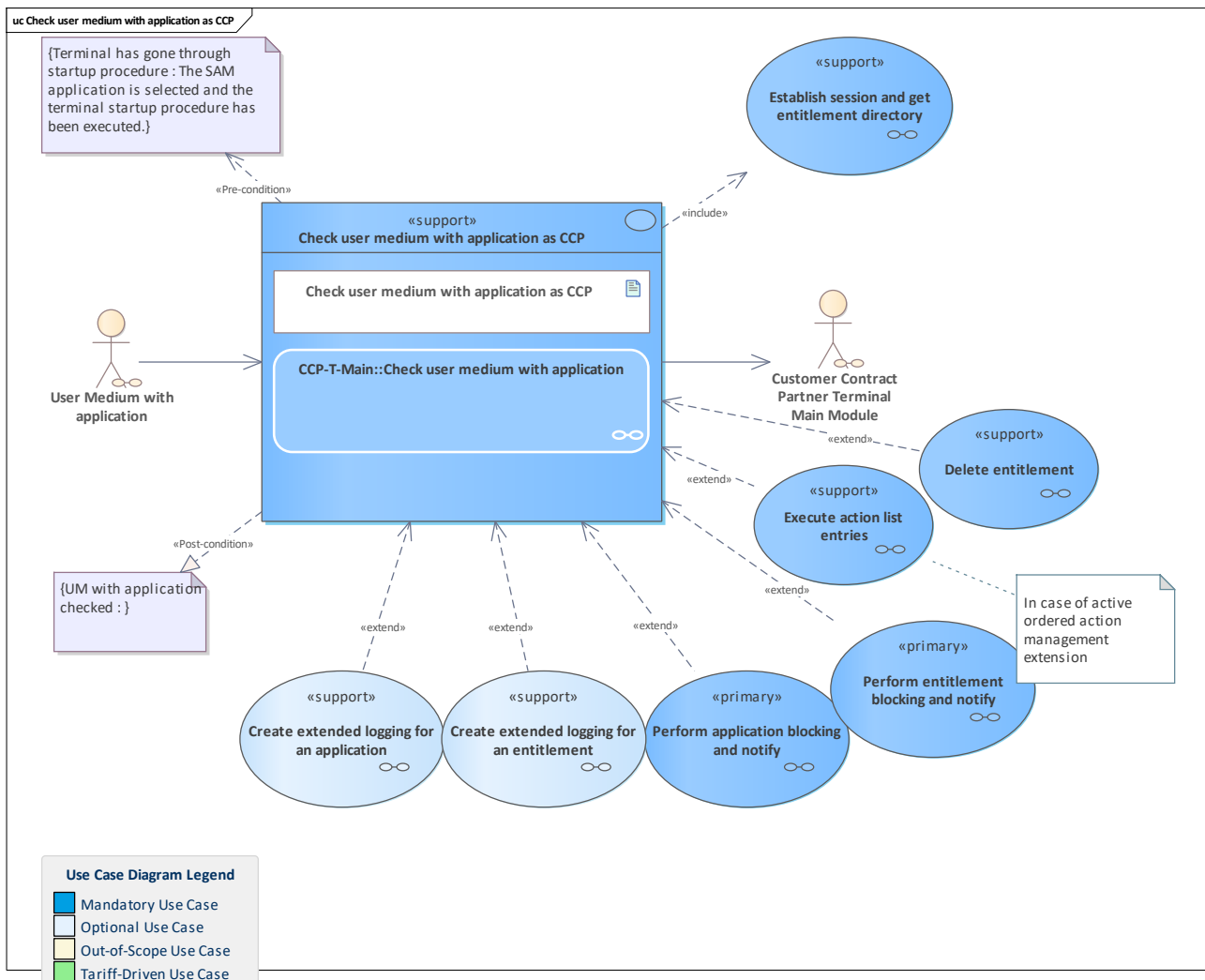


Figure 240: Check user medium with application as CCP

In the following, the processes are summarised for all use cases with the user medium with an application, which are generally to be realised only once at the beginning of a process.

This use case has the following steps:

- Establishing a secure messaging session
- Reading the application directory and entitlement directory
- Checking the application directory (hotlist, status, temporal validity)
- Checking the entitlement directory (hotlist, status, temporal validity)
- Potentially executing pending action orders

11.35 Check user medium with application as service operator

11.36 Check user medium with application as service operator

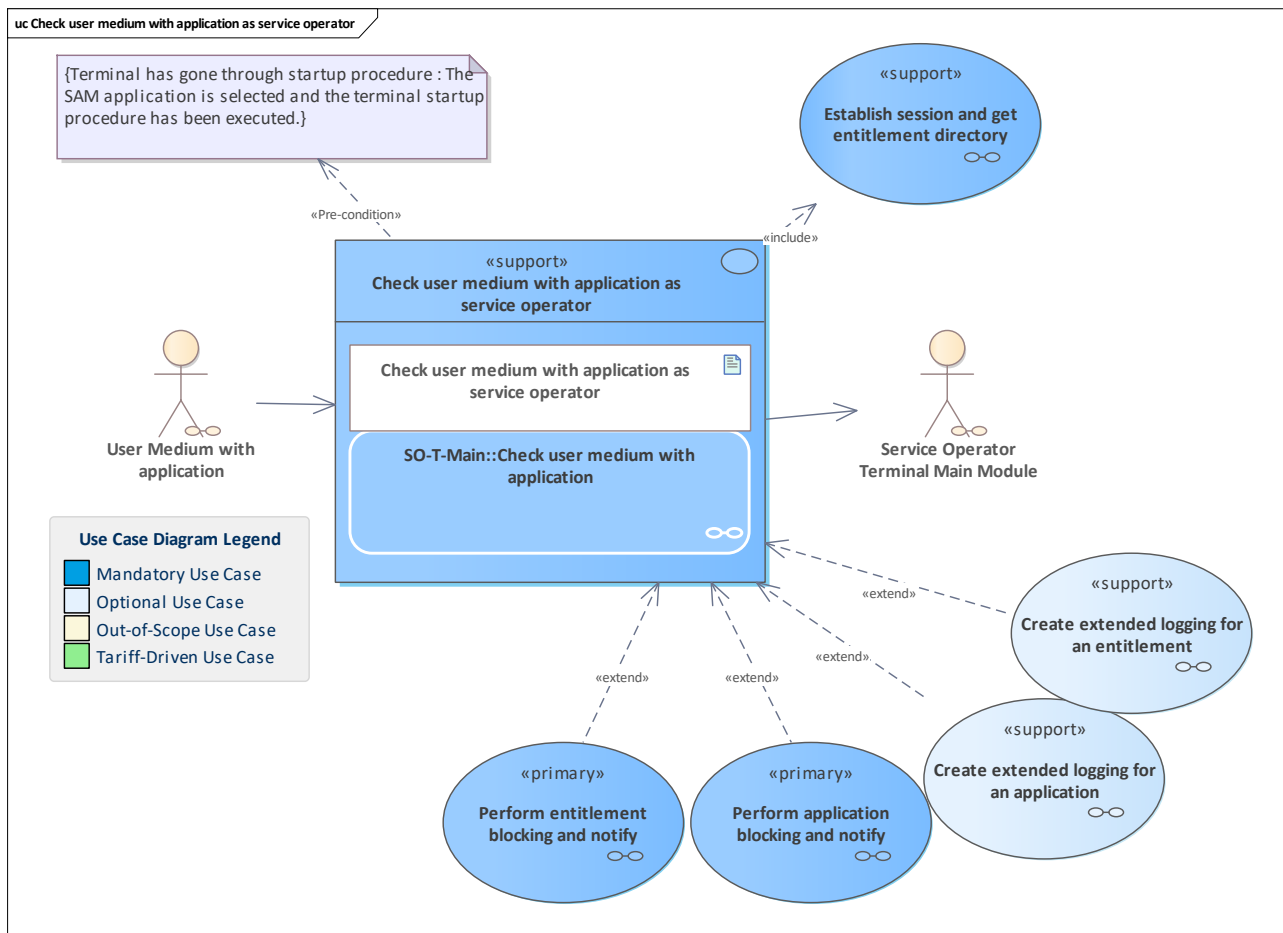


Figure 241: Check user medium with application as service operator

In the following, the processes are summarized for all use cases with the user medium with an application, which are generally to be realized only once at the beginning of a process.

This use case has the following steps:

- Establishing a secure messaging session
- Reading the application directory and entitlement directory
- Checking the application directory (hotlist, status, temporal validity)
- Checking the entitlement directory (hotlist, status, temporal validity)

11.37 Check user medium without application

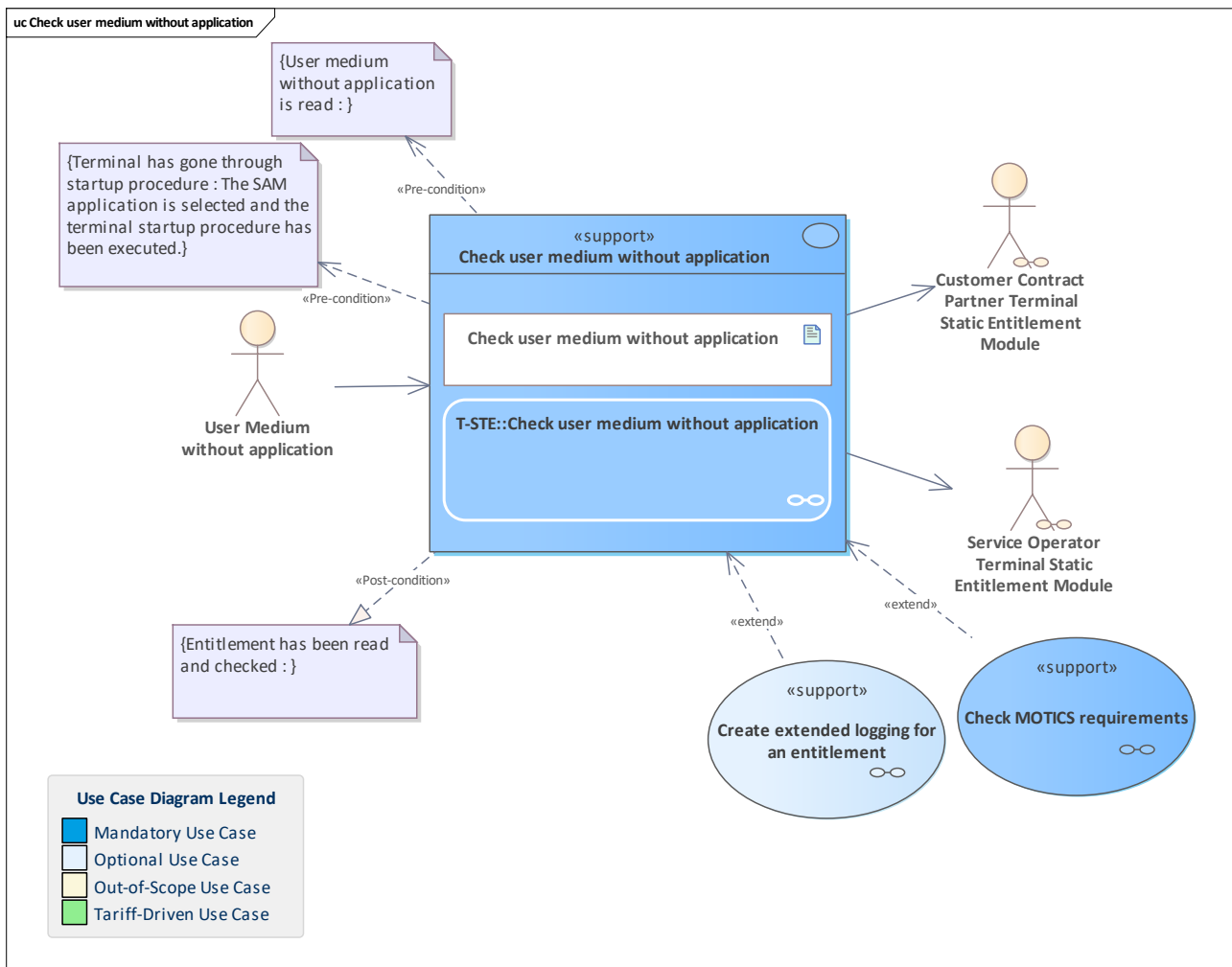


Figure 242: Check user medium without application

This use case has the following steps:

- Checking MOTICS requirements if applicable
- Checking static entitlements

11.38 Configure user medium application

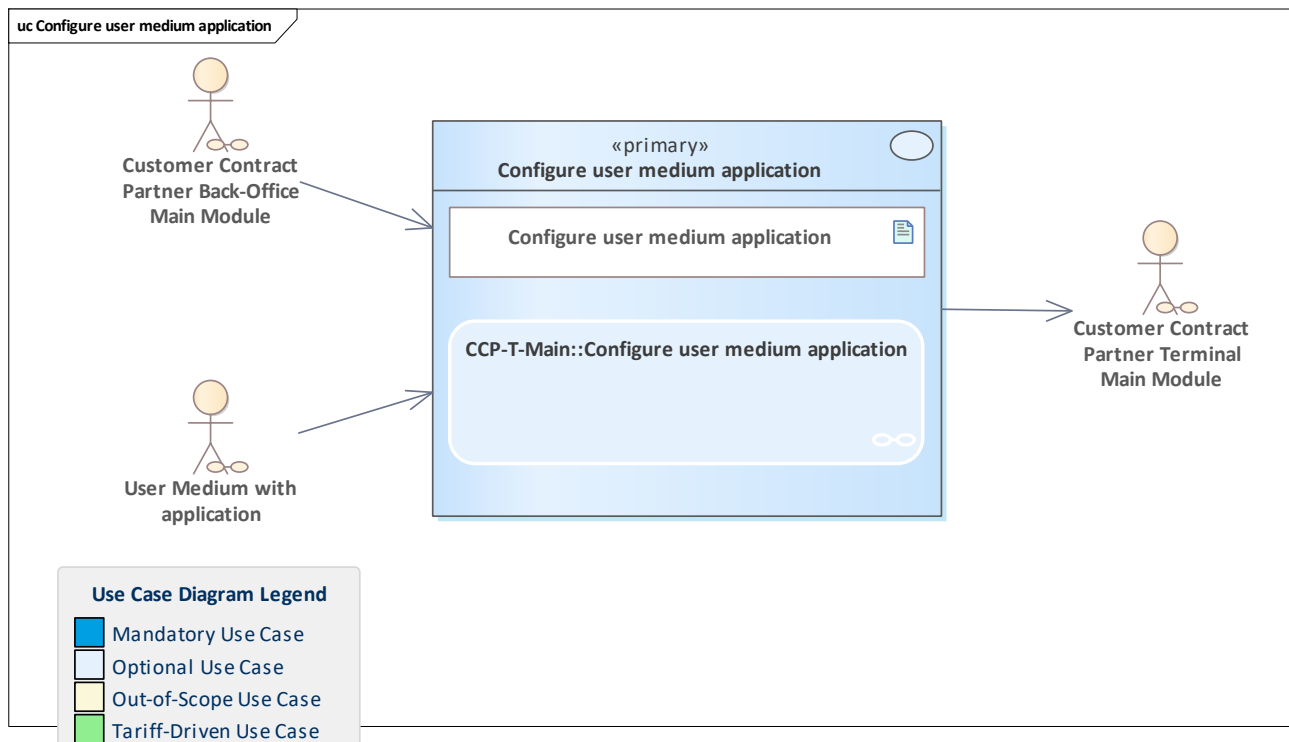


Figure 243: Configure user medium application

A user medium application is (re-)configured by a terminal.
For this use case, the terminal needs to interact with the MMS, which is only possible after organisational and contractual preparations, see [MMS Specification](#).

11.39 Create extended logging for an application

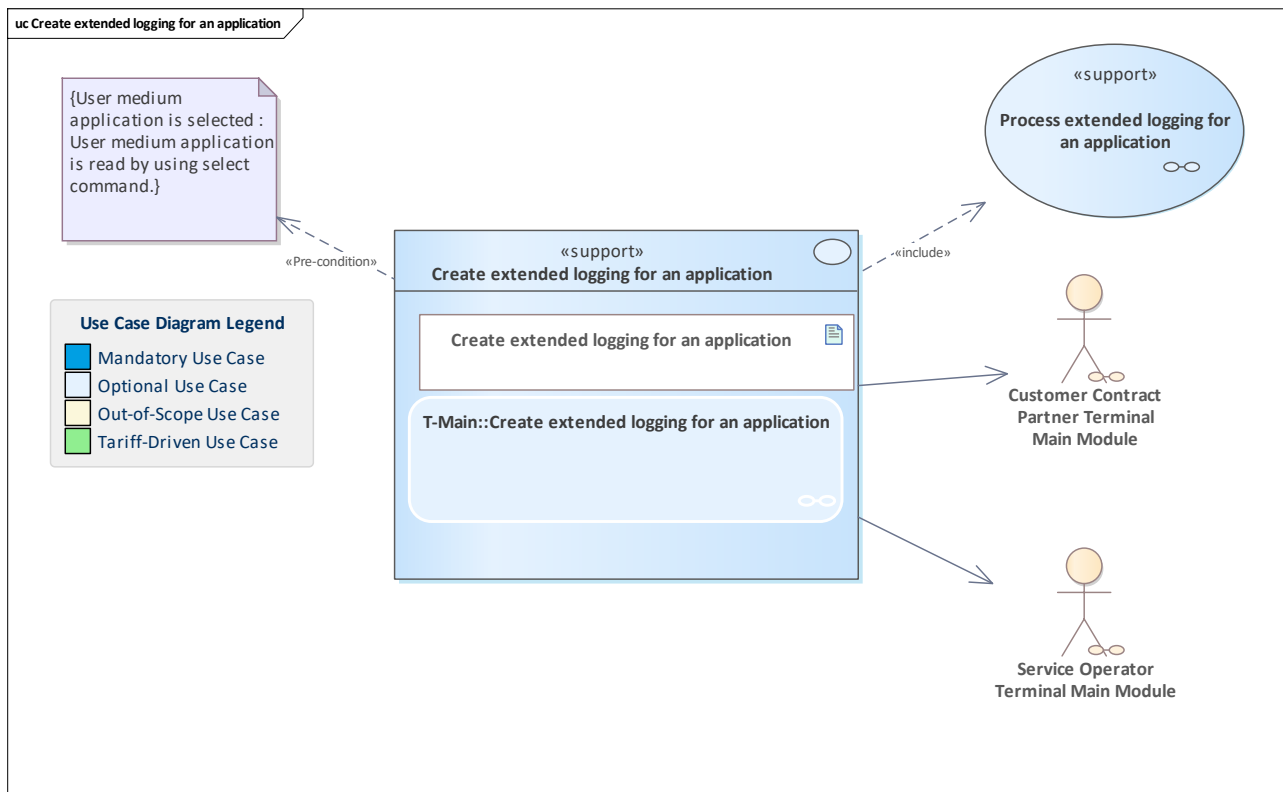


Figure 244: Create extended logging for an application

For monitoring purposes, logging related to an invalid application should be created by a terminal and analysed by the back-office system.

The extended logging of applications is triggered if a blocked or terminated application was presented or the validity period of the application has been expired.

The detailed reason can be found in [ValidationEnum](#).

11.40 Create extended logging for an entitlement

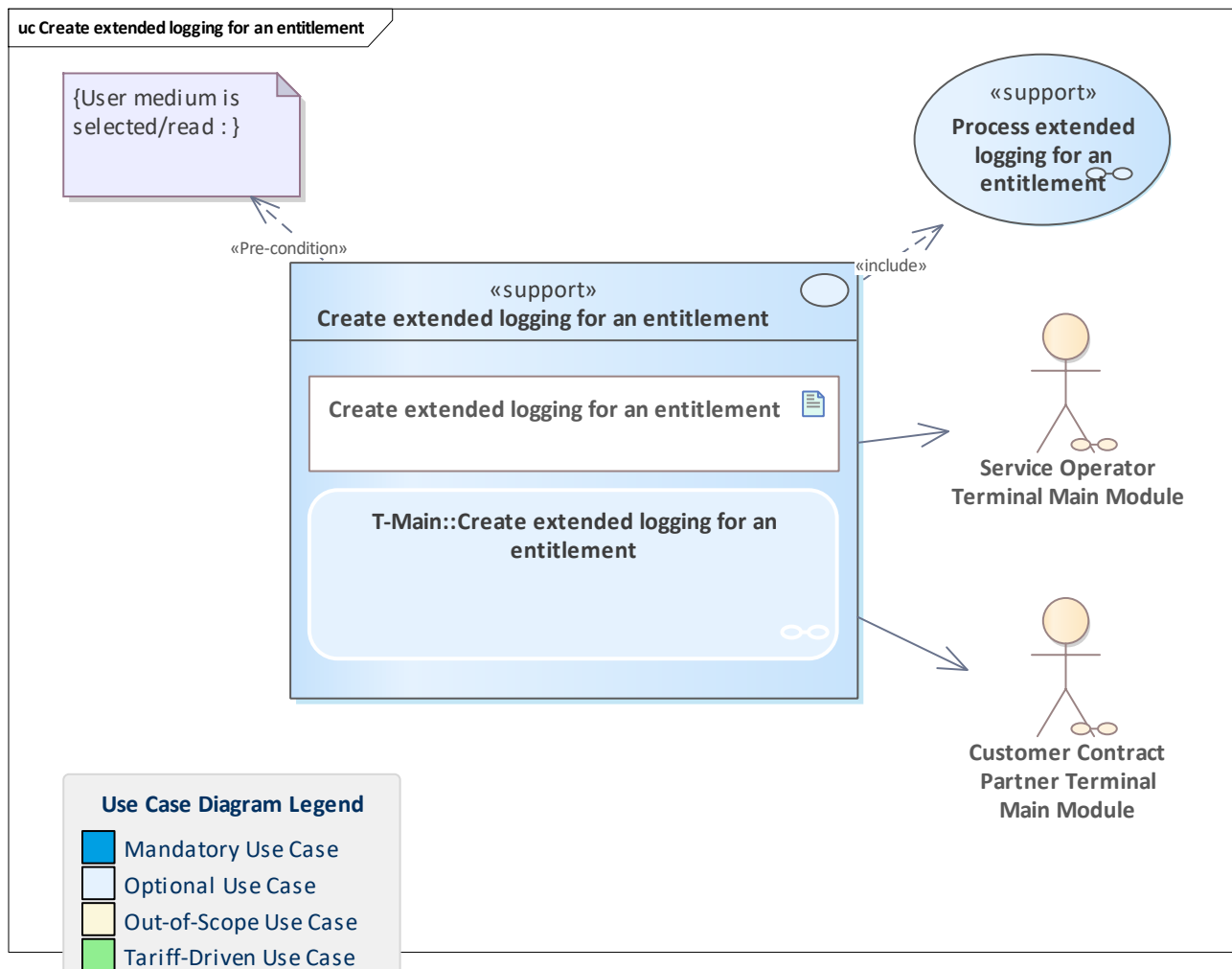


Figure 245: Create extended logging for an entitlement

For monitoring purposes, logging related to invalid entitlements should be created by a terminal and analysed by the back-office system.

The extended logging of entitlements is triggered if a blocked entitlement was presented (only chip-based), the validity period of the entitlement has been expired or the tariff conditions of the entitlement were invalid. The detailed reason can be found in [ValidationEnum](#).

If the [AppInstanceId](#) is available, this value must also be provided.

11.41 Credit account-based payment method

11.42 Credit account-based payment method

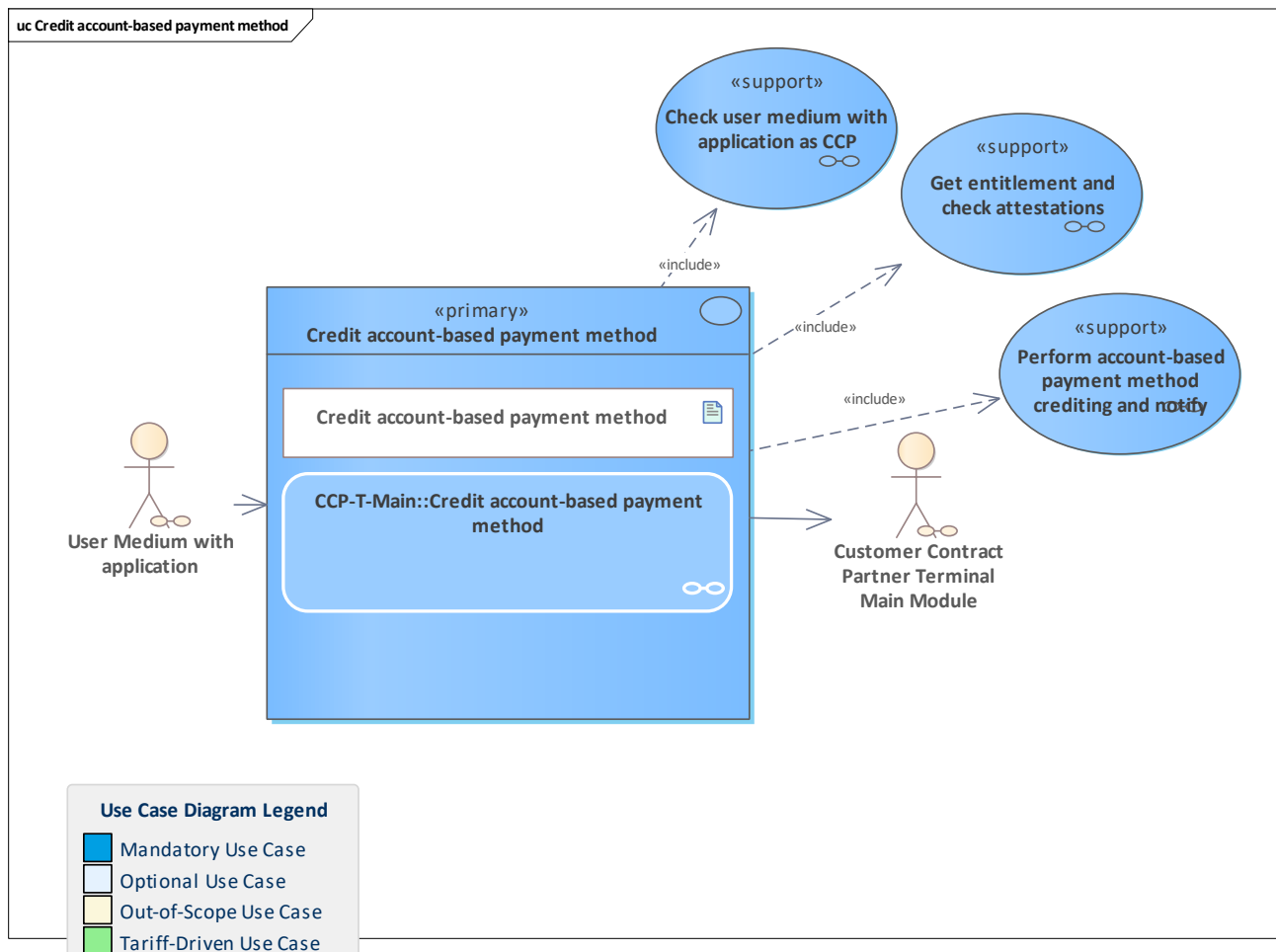


Figure 246: Credit account-based payment method

Credit an account-based payment method by a CCP terminal.
 The use case initially checks the account-based payment method.
 Finally, the credit is performed and notified to the CCP back-office system (same CCP as the terminal operator of the current terminal).
 Normally, this use case is combined with a reimbursement action of an electronic ticket.

11.43 Credit stored-value payment method

11.44 Credit stored-value payment method

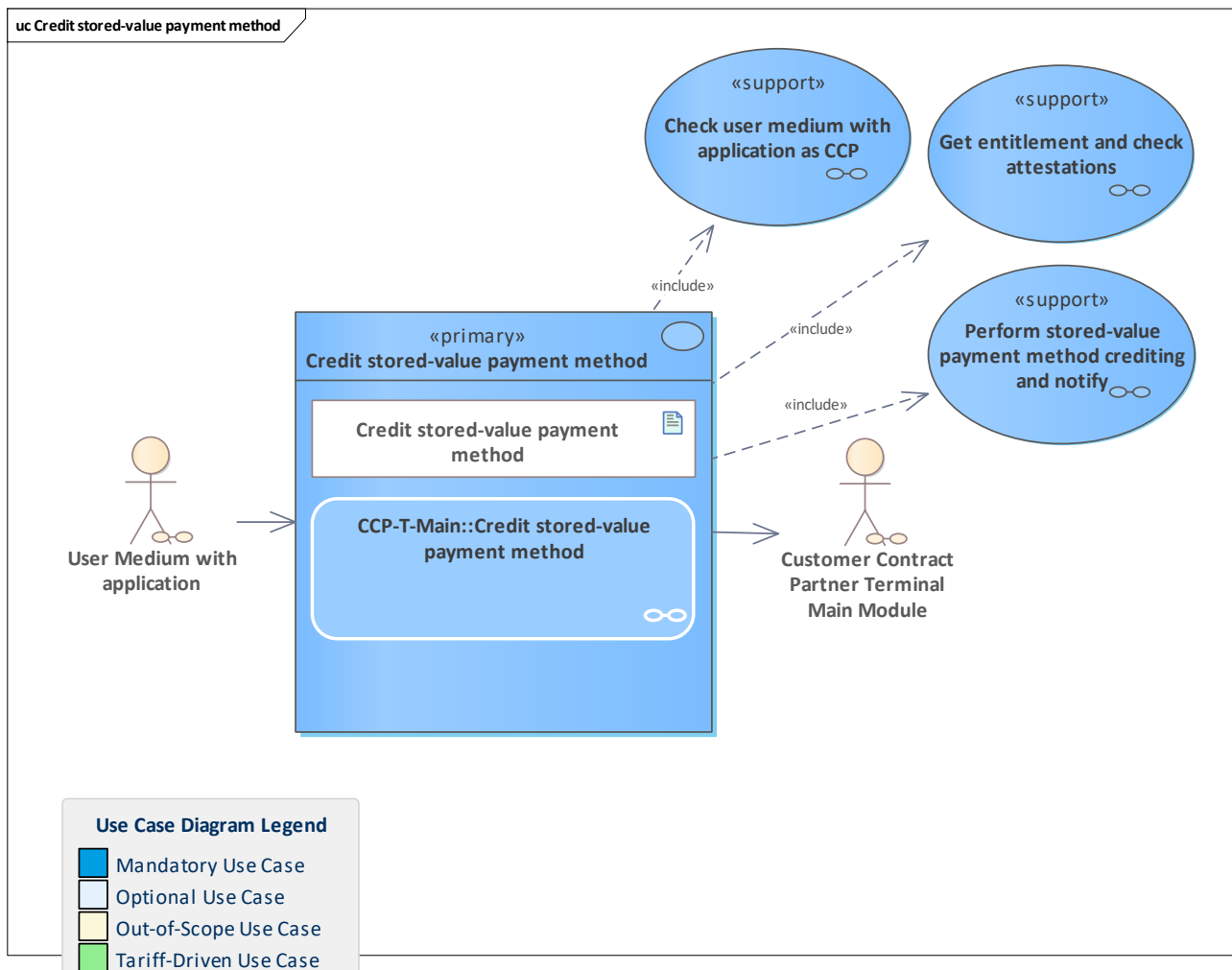


Figure 247: Credit stored-value payment method

Credit a stored-value payment method. Done by a CCP terminal e.g. as part of a reimbursement of a purchased electronic ticket.
Check the maximum allowed balance and trigger the action and notification.

11.45 Debit account-based payment method

11.46 Debit account-based payment method

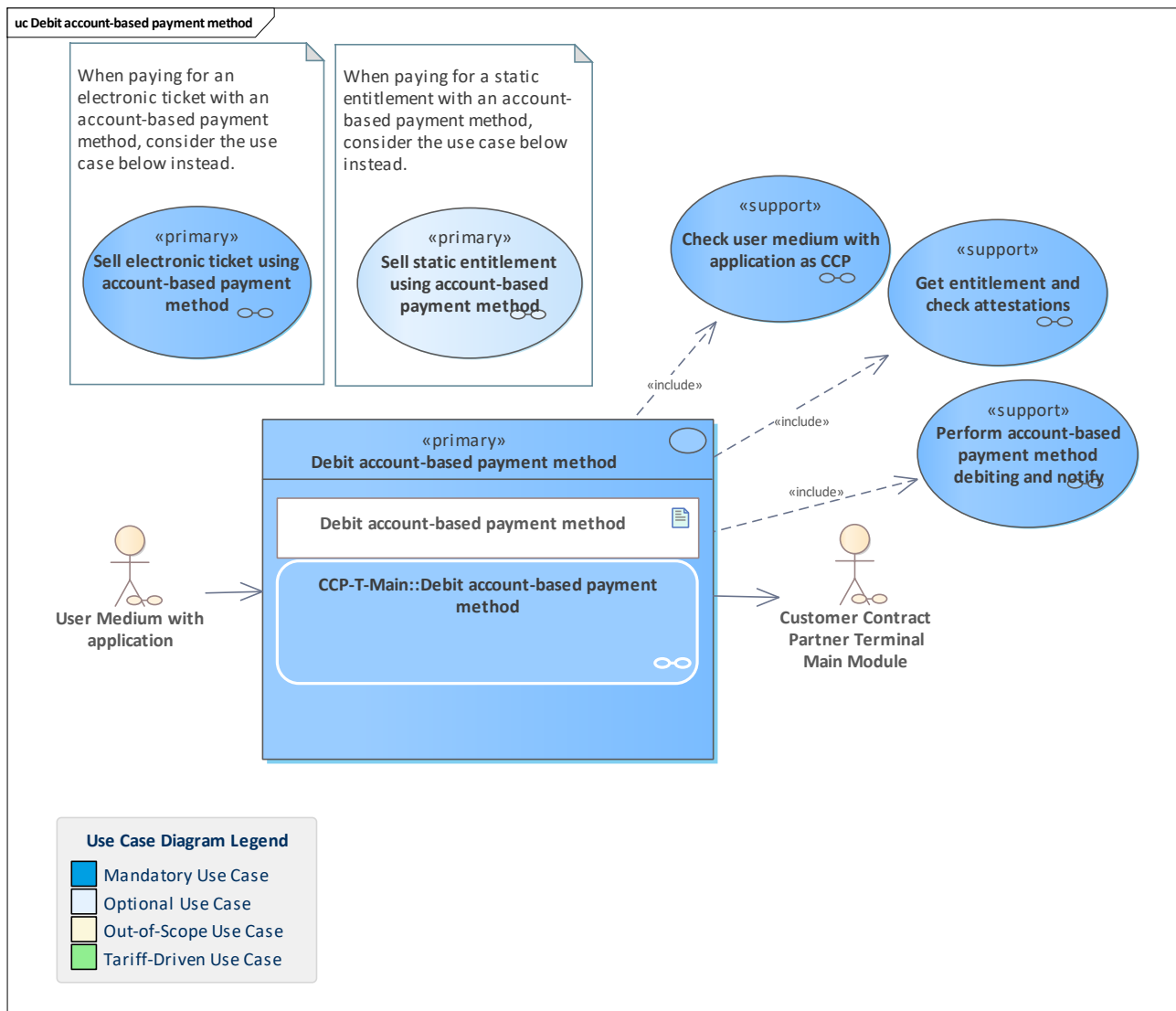


Figure 248: Debit account-based payment method

Debit an account-based payment method by a CCP terminal.
The use case initially checks the account-based payment method.
Finally, the debit is performed and notified to the CCP back-office system (same CCP as the terminal operator of the current terminal).
Normally, this use case is combined with a purchase transaction of an electronic ticket.

11.47 Debit stored-value payment method

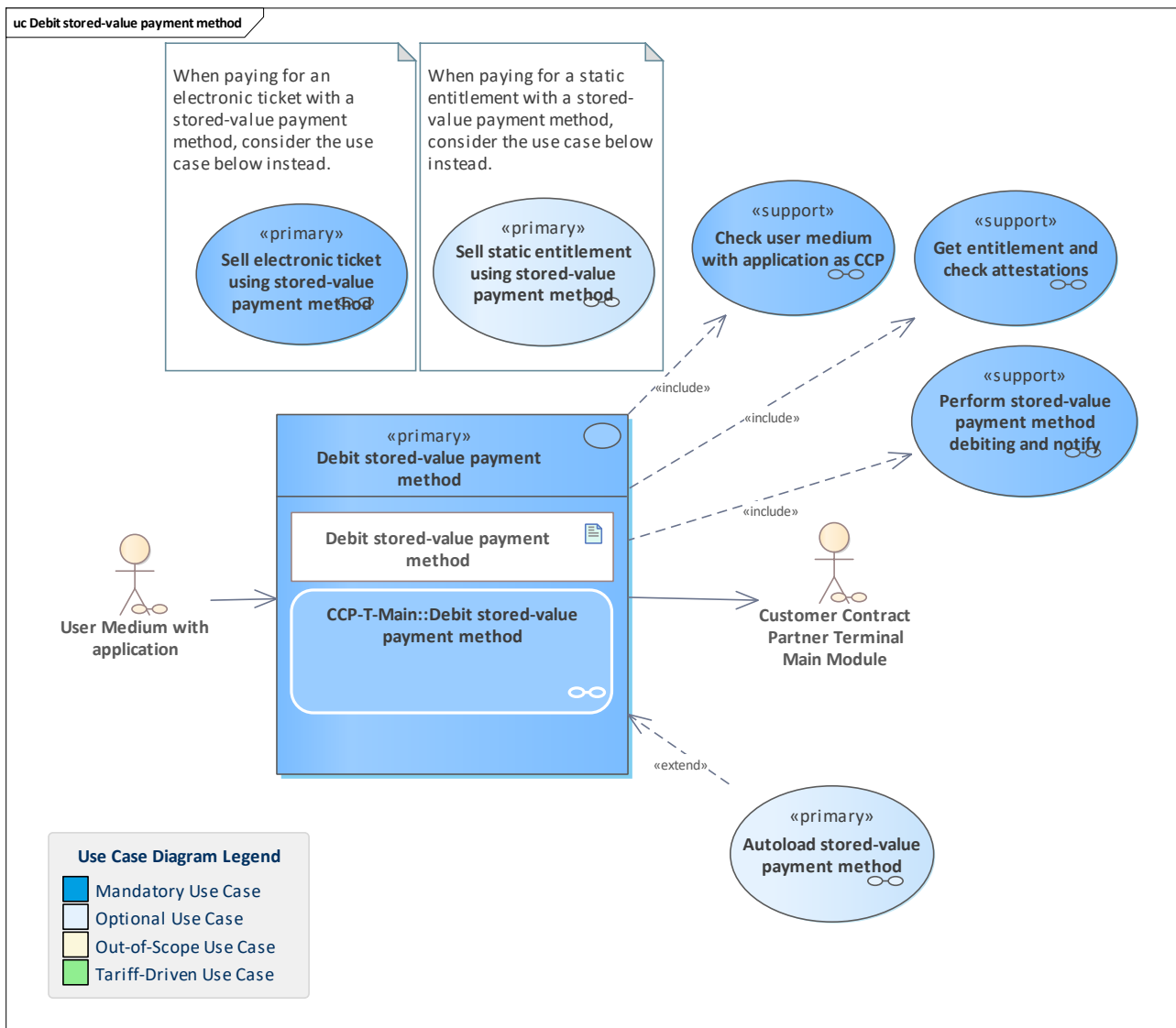


Figure 249: Debit stored-value payment method

Debit a stored-value payment method by a CCP terminal.

The use case checks the stored-value payment method and ensures, that the current balance in the stored-value account is sufficient for the intended debit action.

Finally, the debit is performed and notified to the CCP back-office system (same CCP as the terminal operator of the current terminal).

Normally, this use case is combined with a purchase transaction of an electronic ticket.

If autoload is configured for the customer, this could come into play if the balance is not sufficient.

11.48 Delete customer

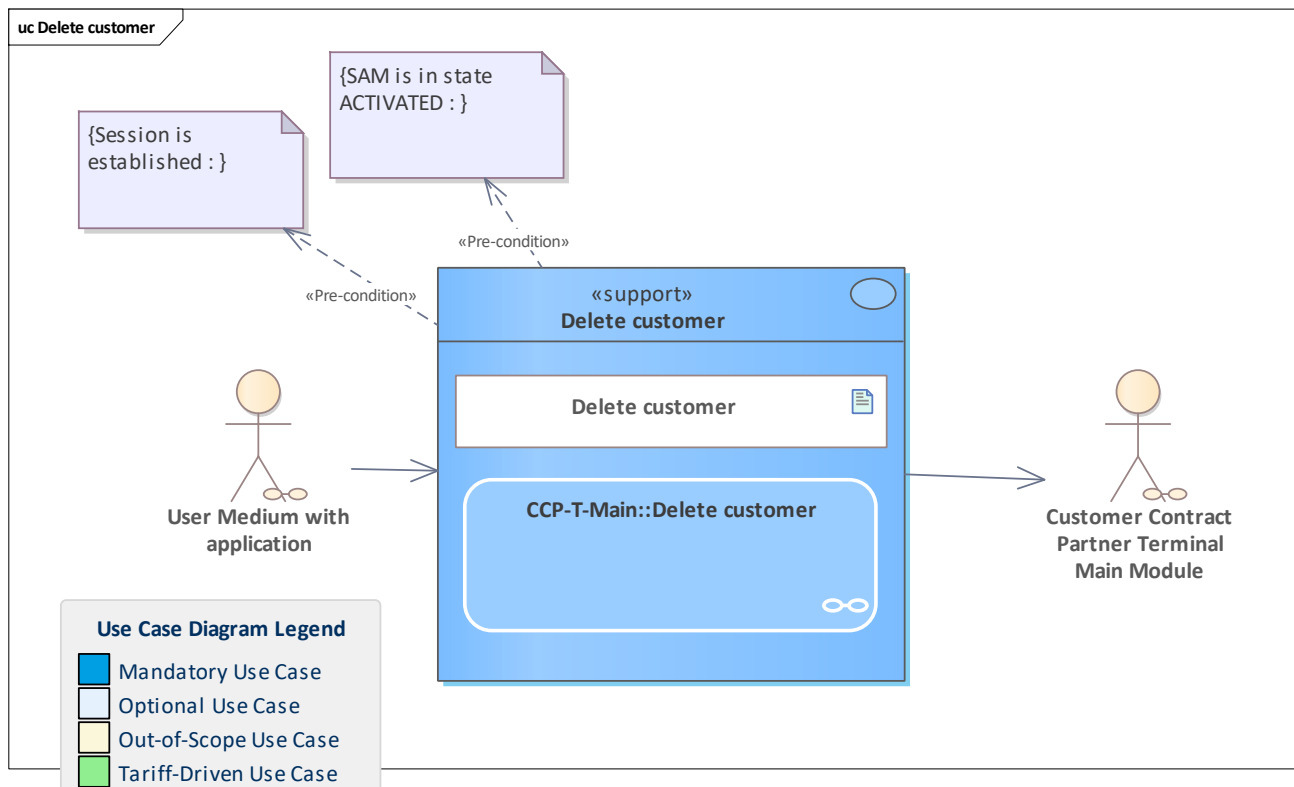


Figure 250: Delete customer

Delete the customer data object on the user medium with an application.

11.49 Delete discounts

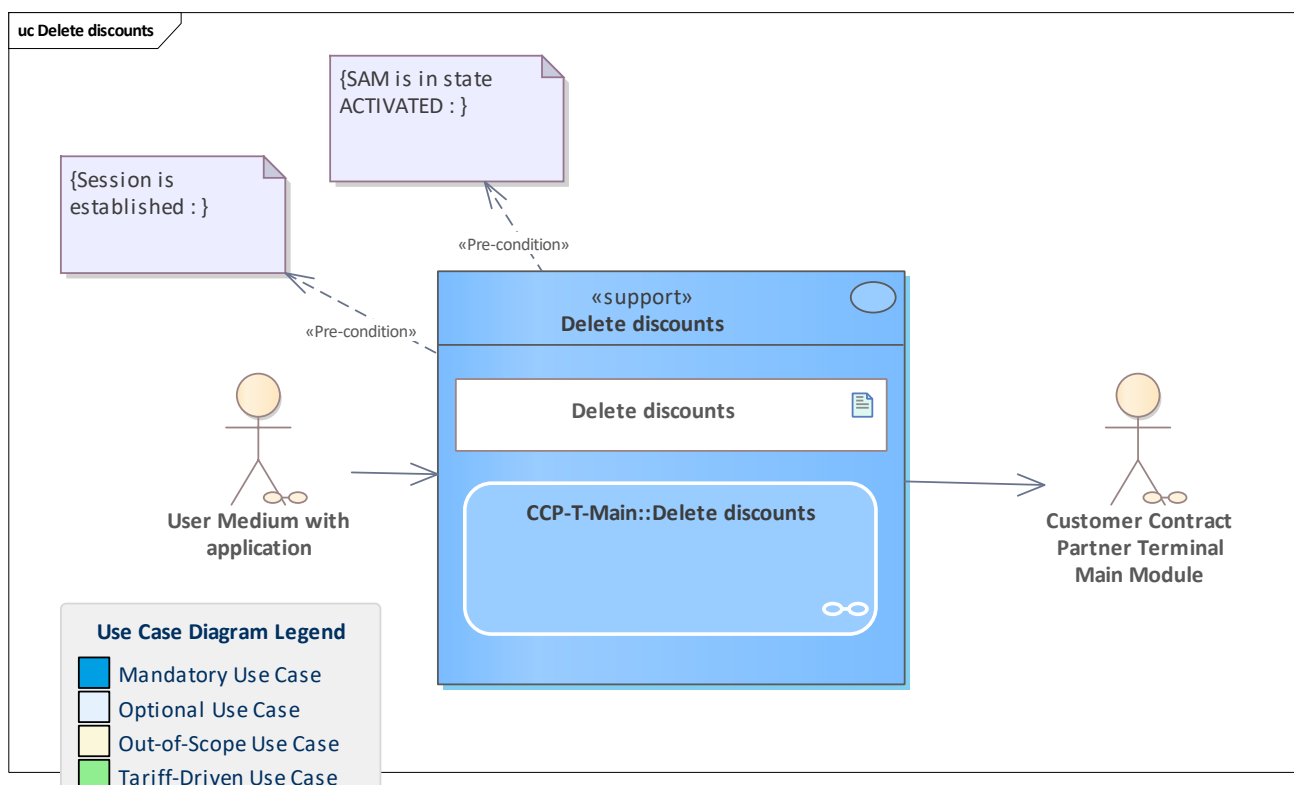


Figure 251: Delete discounts

Delete the discounts data object on the user medium with an application.

11.50 Delete entitlement

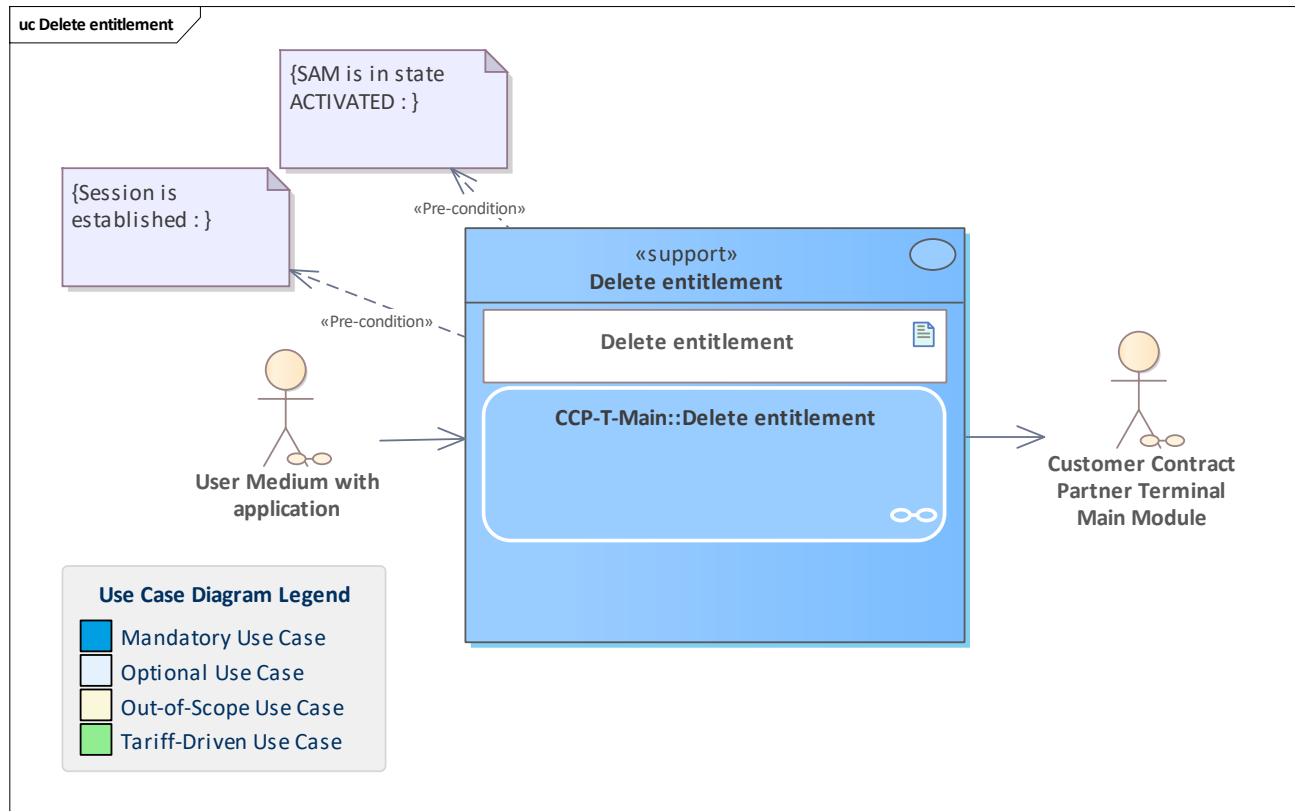


Figure 252: Delete entitlement

The terminal deletes an entitlement from the user medium application.

11.51 Delete favourites

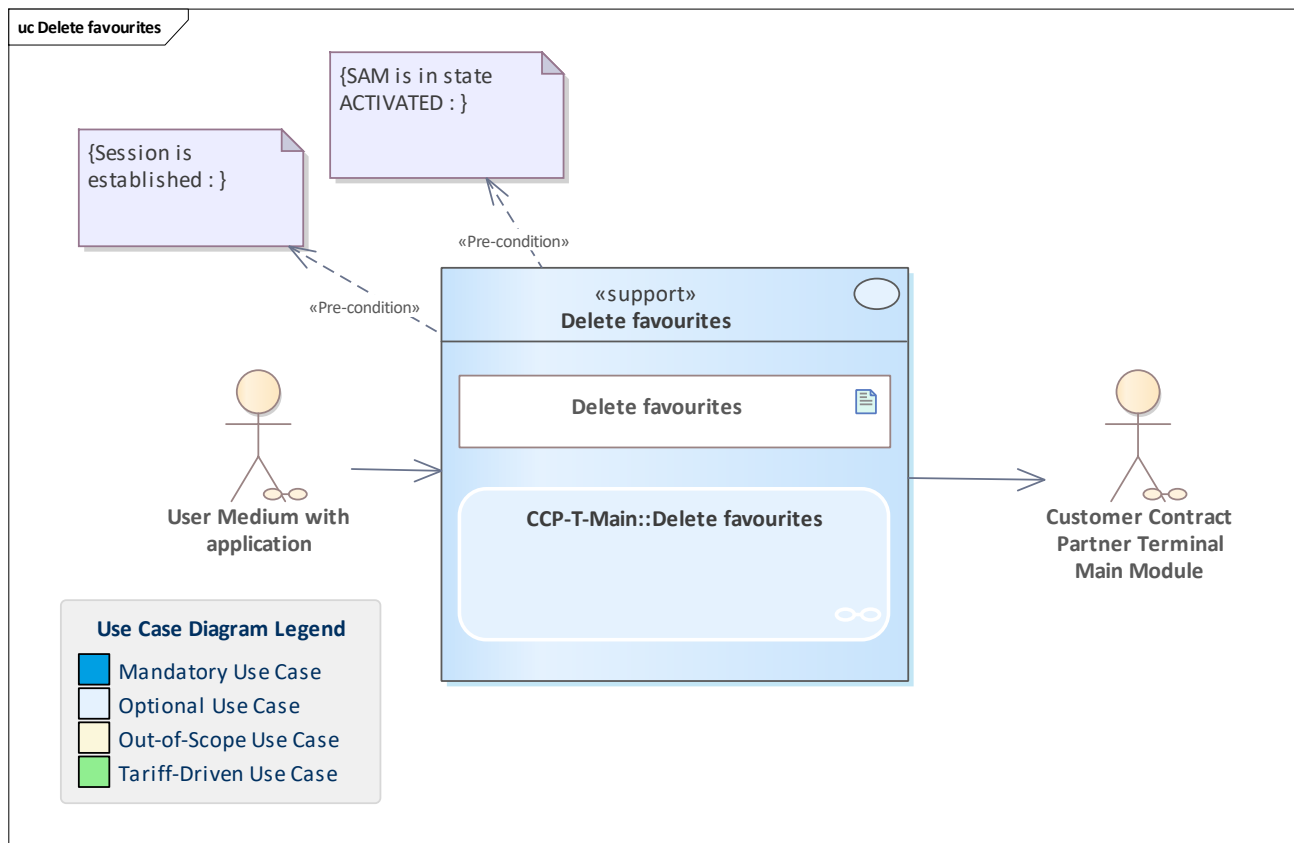


Figure 253: Delete favourites

Delete the favourites data object on the user medium with an application.

11.52 Demand account-based payment method charging

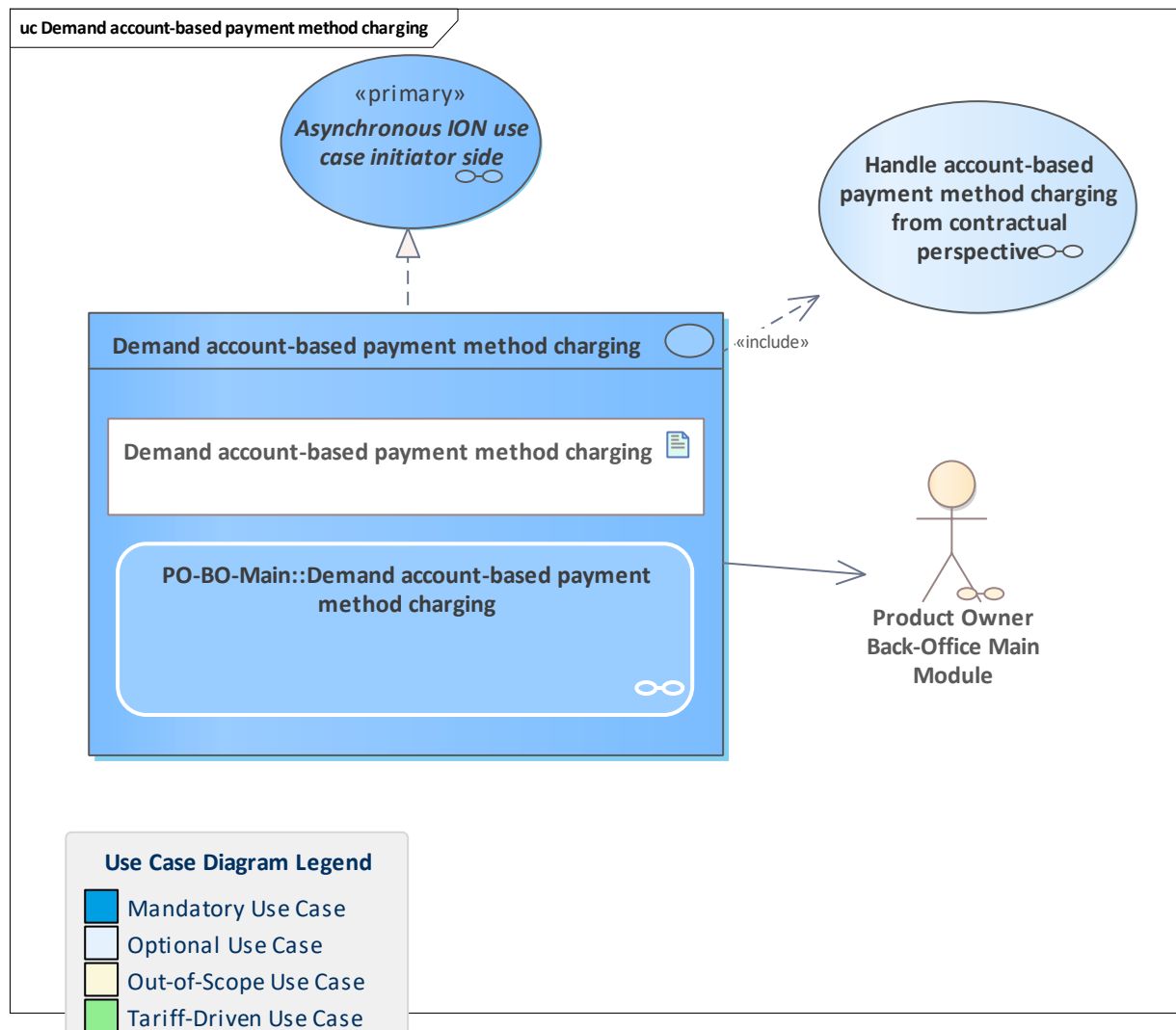


Figure 254: Demand account-based payment method charging

When using an account-based payment system in the CICO context:

The PO determines the price of all journeys based on received CICO messages and informs the pCCP about these journeys.

This is done in a scheduled process within a defined time interval. The process details are out of scope.

11.53 Demand application hotlisting

11.54 Demand application hotlisting

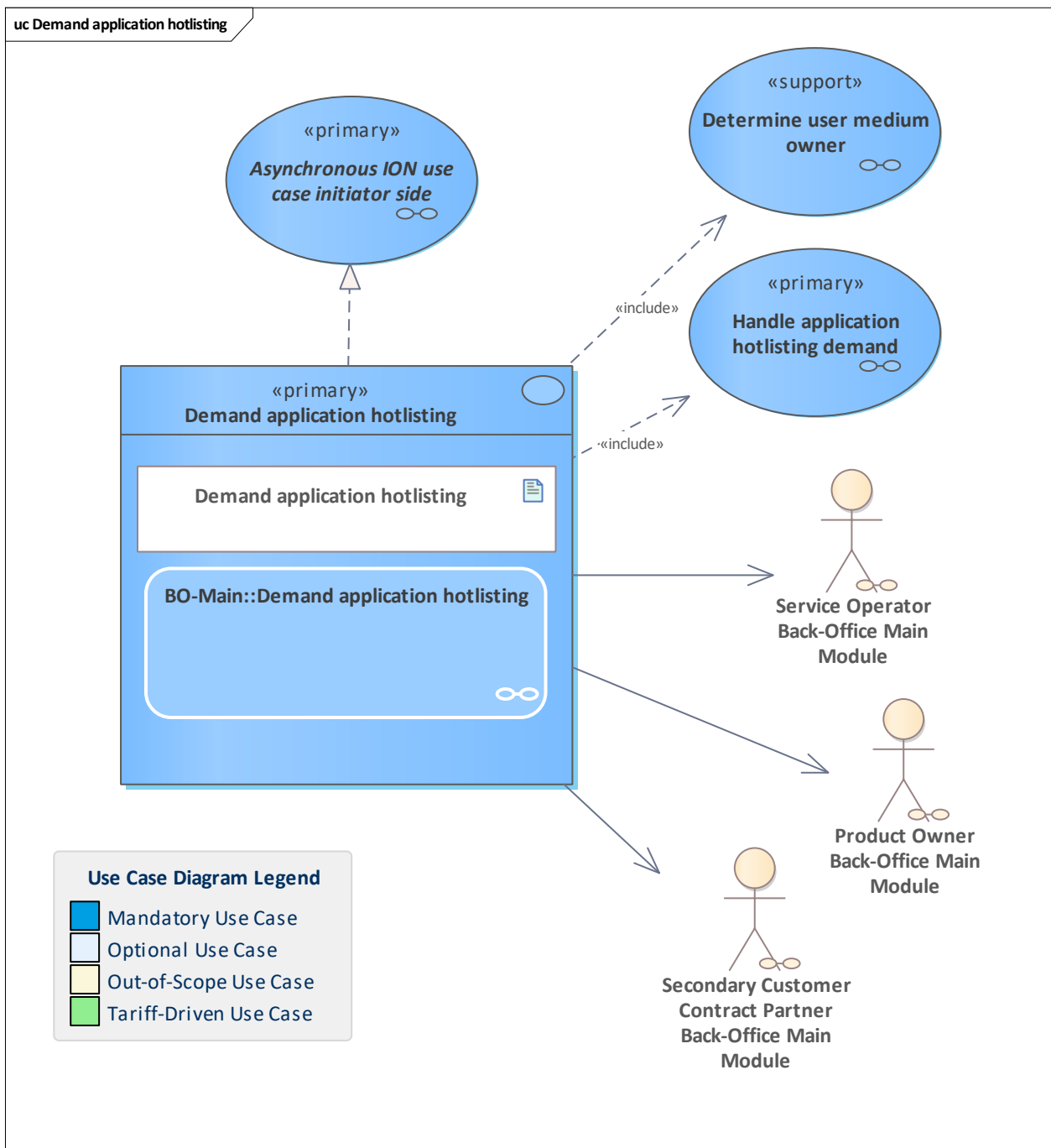


Figure 255: Demand application hotlisting

The PO, sCCP or SO as sender demands the pCCP that issued the application to the customer to hotlist the application in question.

The sender adds a reason (SO e.g. in case of a defective user medium, sCCP e.g. in case of a lost user medium) and further hotlisting parameters.

This application can be either a user medium application with an application instance ID or a MOTICS app with an SCE ID.

In most cases, the hotlist demand from a third party will be caused by monitoring.

11.55 Demand entitlement hotlisting

11.56 Demand entitlement hotlisting

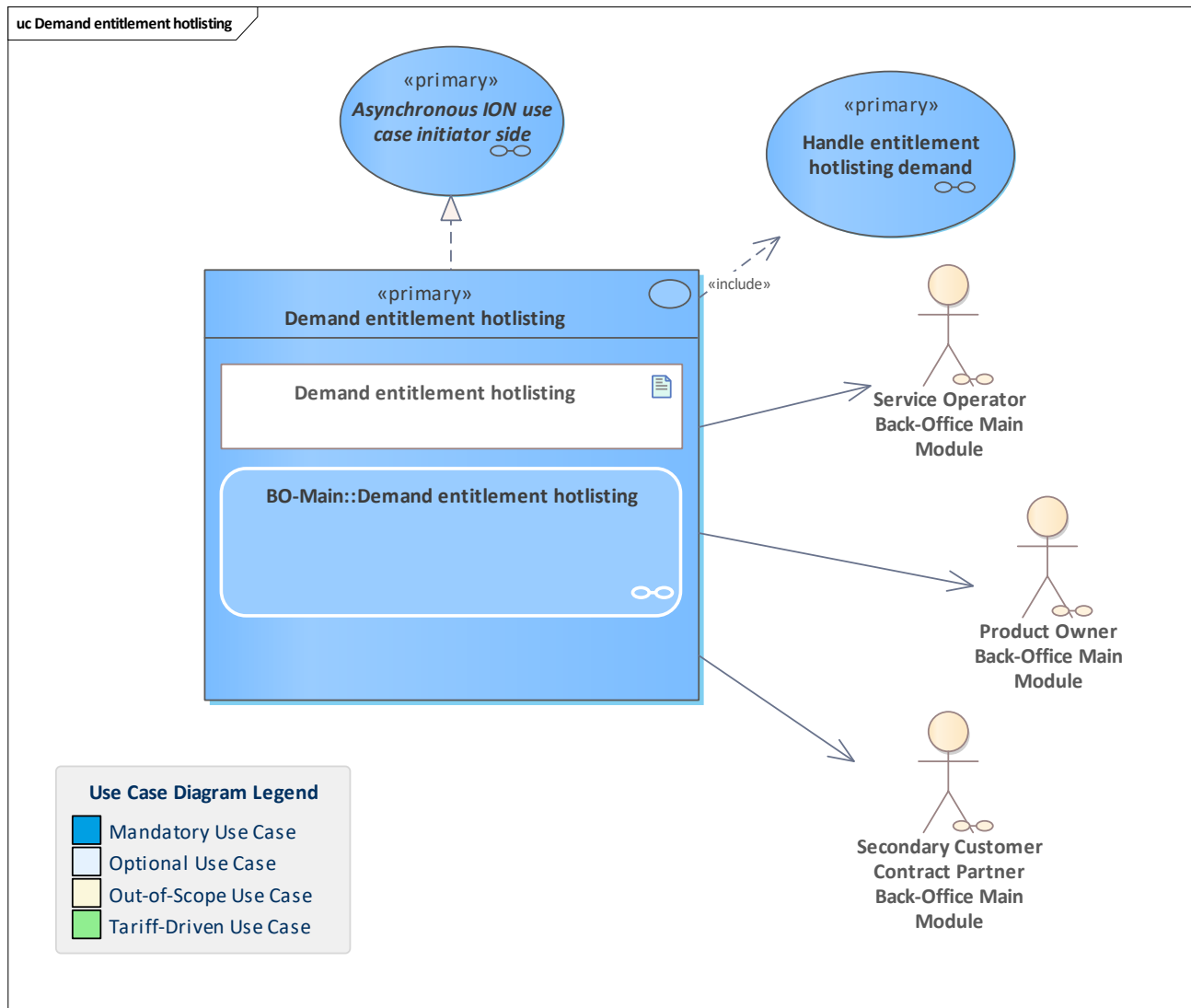


Figure 256: Demand entitlement hotlisting

In this use case, the PO, sCCP or SO as sender demands the pCCP that issued the entitlement to the customer to hotlist the entitlement in question.

The sender adds a reason and further hotlisting parameters.

This entitlement can be either located on a user medium with an application or a static entitlement coming from a barcode or a MOTICS app.

In most cases, the hotlisting demand from a third party will be caused by monitoring.

Possible reasons are:

- Inconsistencies have occurred during monitoring (the most likely reason)
- Offence against terms of carriage
- Payment delay
- Referenced product was deactivated
- Entitlement with the same ID already exists
- User medium/application which contains entitlements of a secondary customer contract partner (CCP) was replaced

11.57 Demand SAM hotlisting

11.58 Demand SAM hotlisting

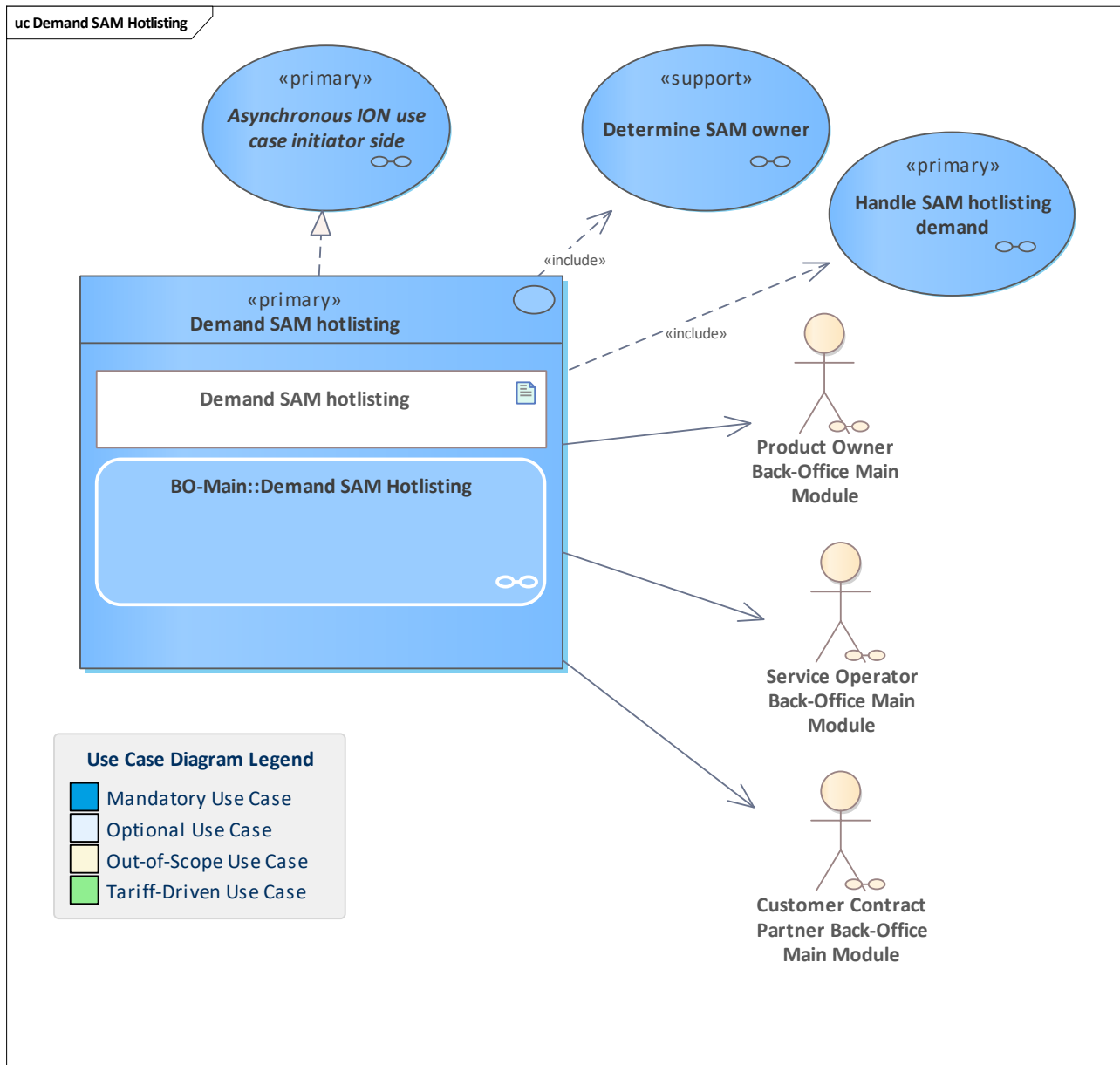


Figure 257: Demand SAM Hotlisting

A PO, SO or CCP sends a demand for hotlisting to the SAM Owner (SO or CCP). Reasons for this demand could either be loss or theft of a SAM that was used in one of the SO or CCP terminals. Another reason could be suspected fraud, which could be detected by monitoring. Before demanding the SAM hotlisting, the demander must find out the SAM owner to place the demand to the right receiver.

11.59 De-personalise application

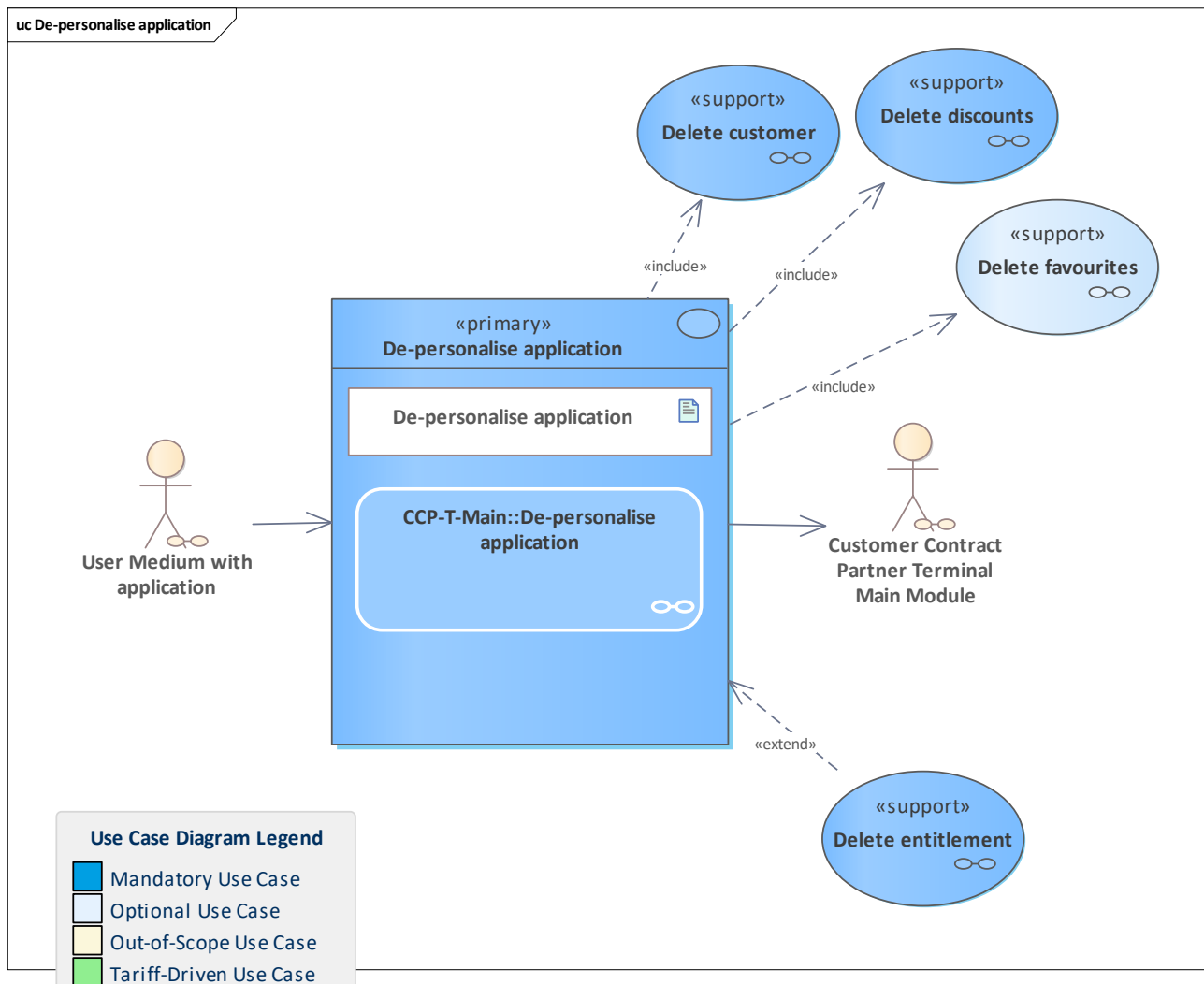


Figure 258: De-personalise application

Use case for the CCP terminal to remove all customer-related information from the user medium application.

This use case can be utilised to reuse the user medium for another customer or to remove personal data before terminating the user medium for data protection reasons.

11.60 Determine SAM owner

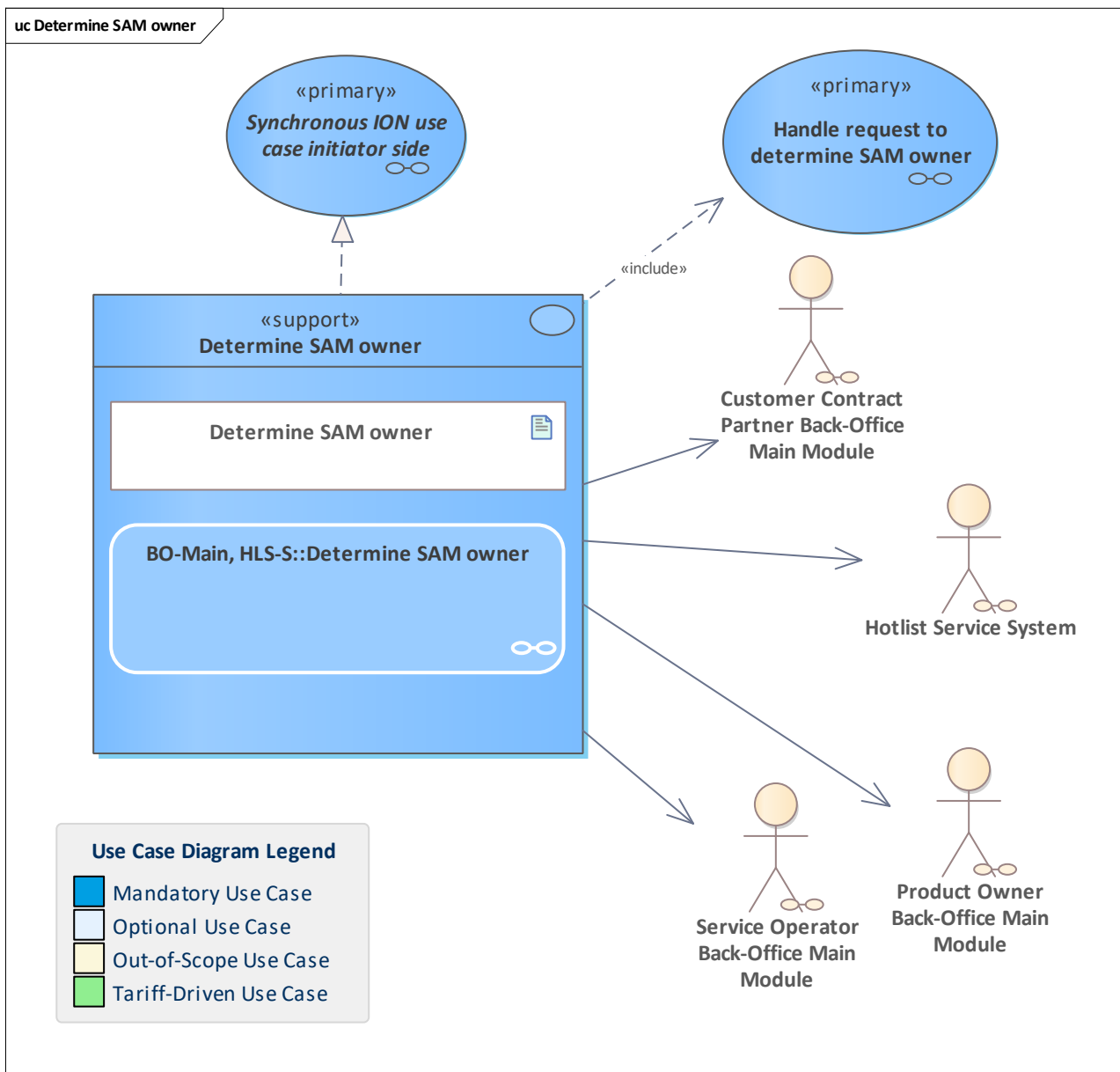


Figure 259: Determine SAM owner

Determine the SAM Owner (organisation ID and role) for a given SAM ID using the service provided by the ESH.

11.61 Determine UM app instance ID for Medium ID

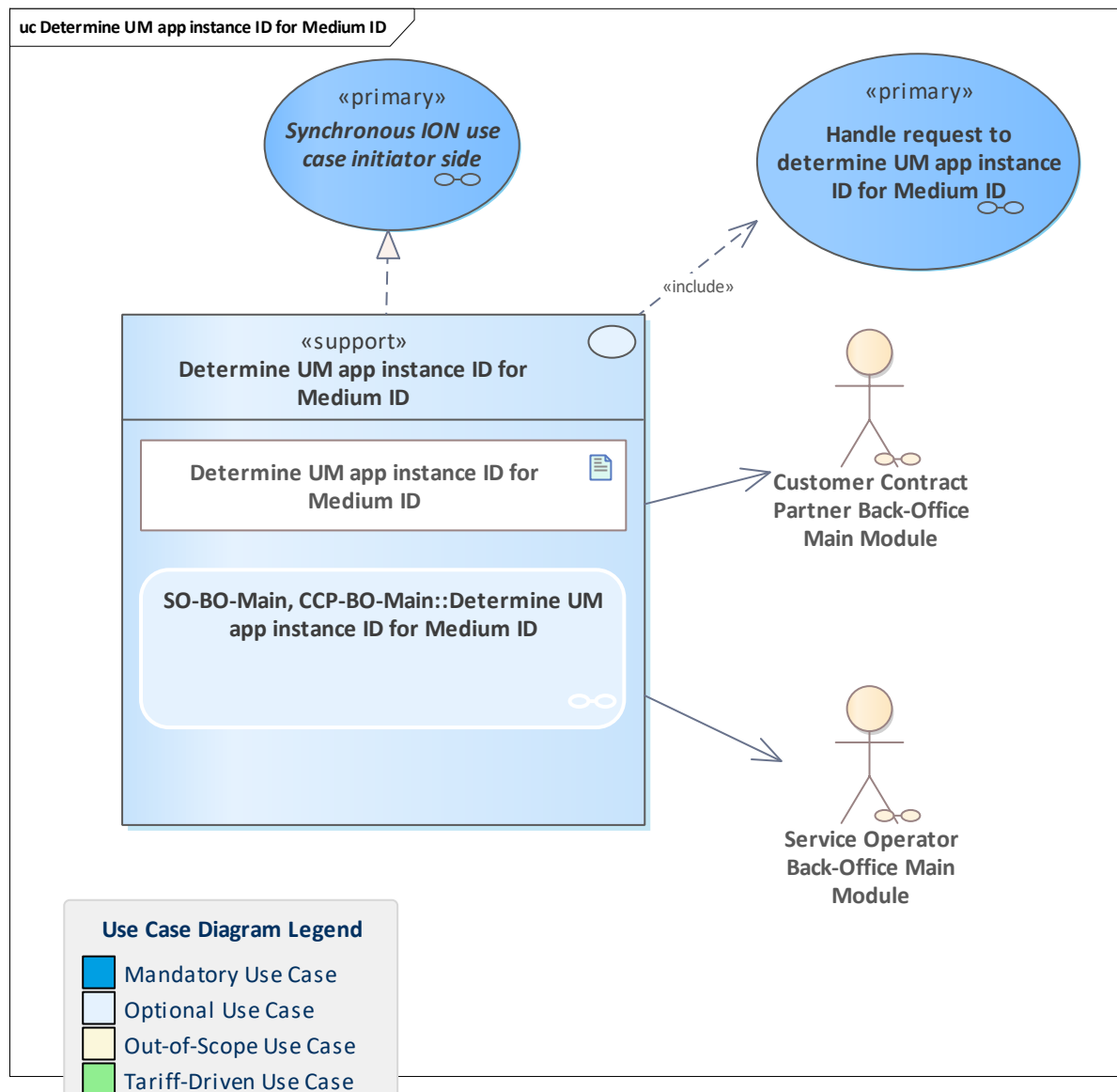


Figure 260: Determine UM app instance ID for Medium ID

Determine the user medium application instance ID for a given medium ID using the service provided by the ESH.

If the application instance ID is already printed on all user media of the transport company, then this use case is optional.

11.62 Determine user medium owner

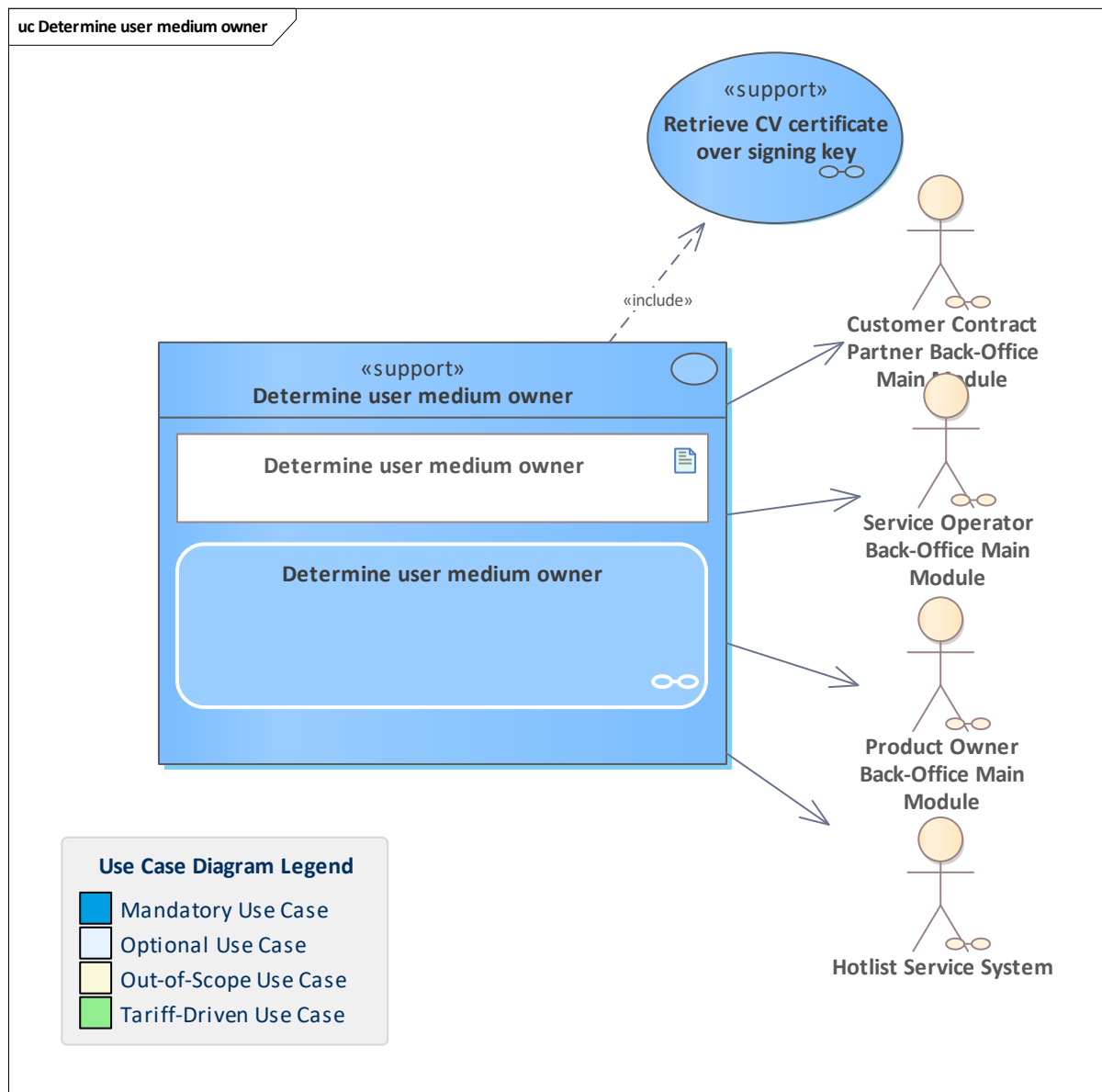


Figure 261: Determine user medium owner

Determine the owner of a user medium with an application or SCE (in case of MOTICS with static entitlement) in a back-office system using the ownership information contained in the corresponding certificate.

To determine the user medium owner, the matching CV certificate is to be fetched, so the caller retrieves the latest certificate over the signing key of an end entity.

Note that there might be several certificates for this end entity, that are not relevant here: superseded certificates and certificates for keys not used for signature purposes. Thus, the right certificate that delivers the owner organisation ID has to be filtered.

11.63 Determine valid entitlements for given app instance ID

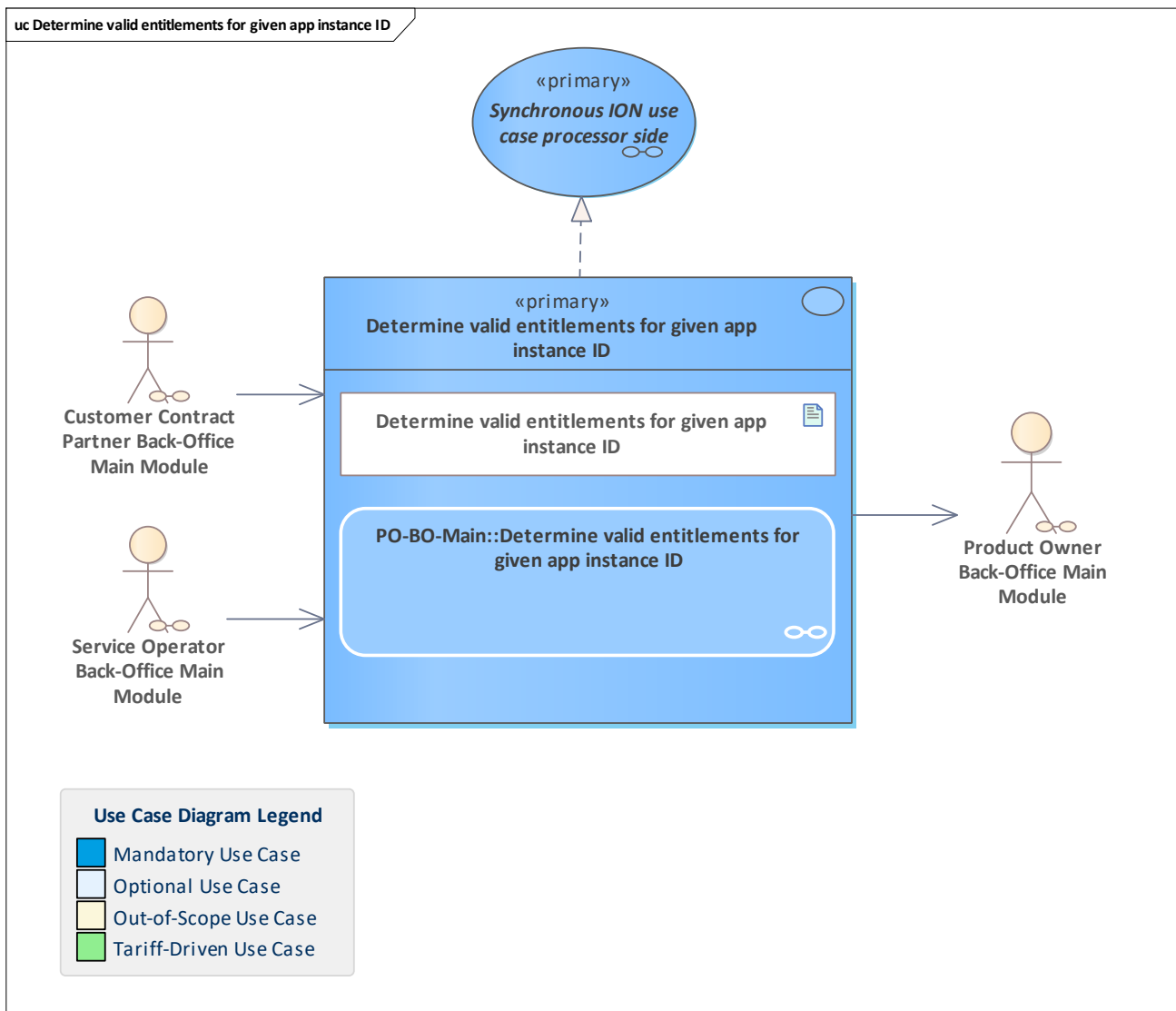


Figure 262: Determine valid entitlements for given app instance ID

Valid entitlements are determined based on the given application instance ID and given date "validOn".

Please note that the *validOn* has a data type of [ZonedDate](#) due to a possible grace period or delays.

Valid entitlements means that all entitlements for the requested application instance ID are returned that were valid at the time of validOn and were not on the entitlement hotlist.

The PO has no information about entries in the application hotlist or the blocking of the application with the requested application instance ID.

If valid entitlements are requested as part of an inspection process, the tariff checks must be executed on the CCP or SO side.

11.64 Display application data

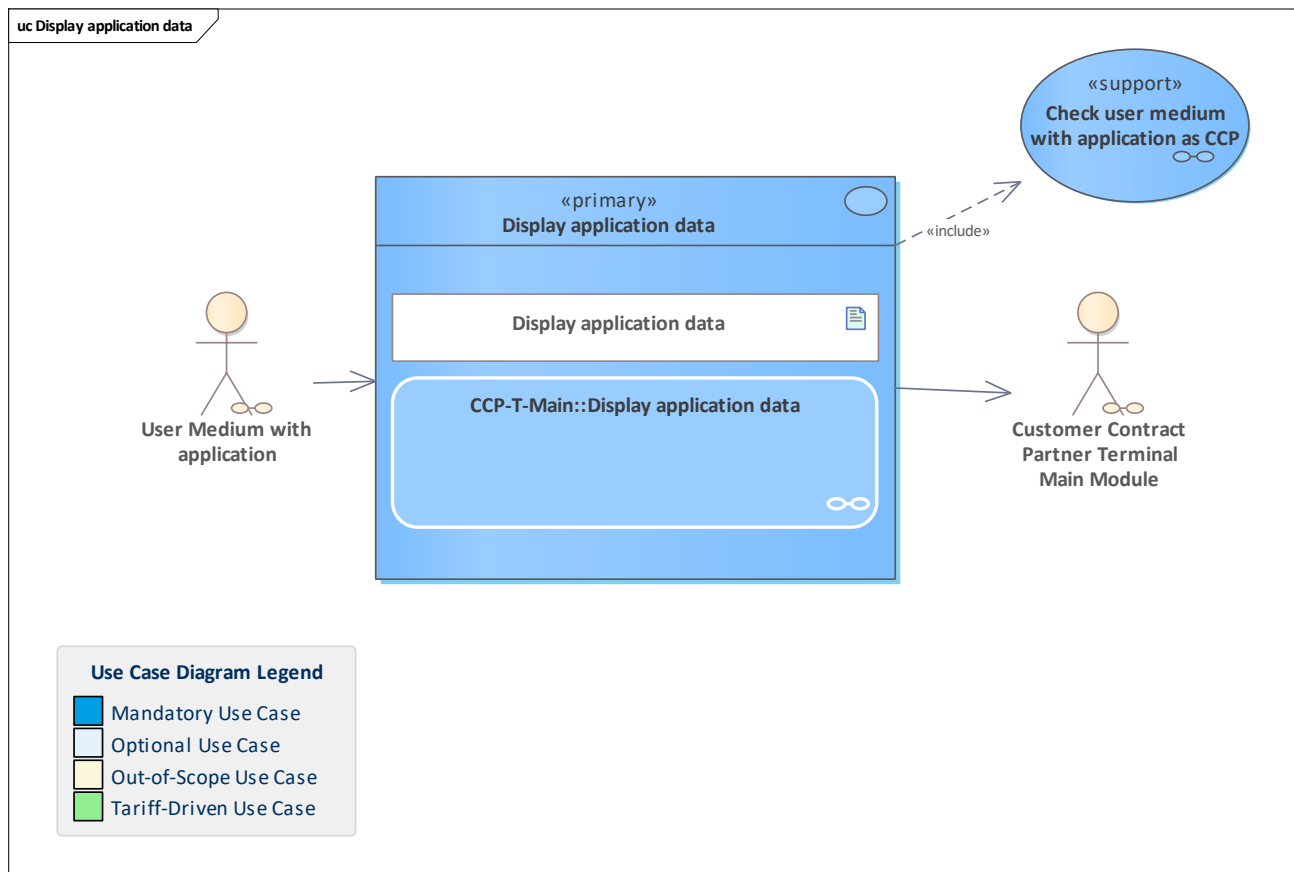


Figure 263: Display application data

The application data that is relevant for the customer is displayed using the application directory.

11.65 Display customer

11.66 Display customer

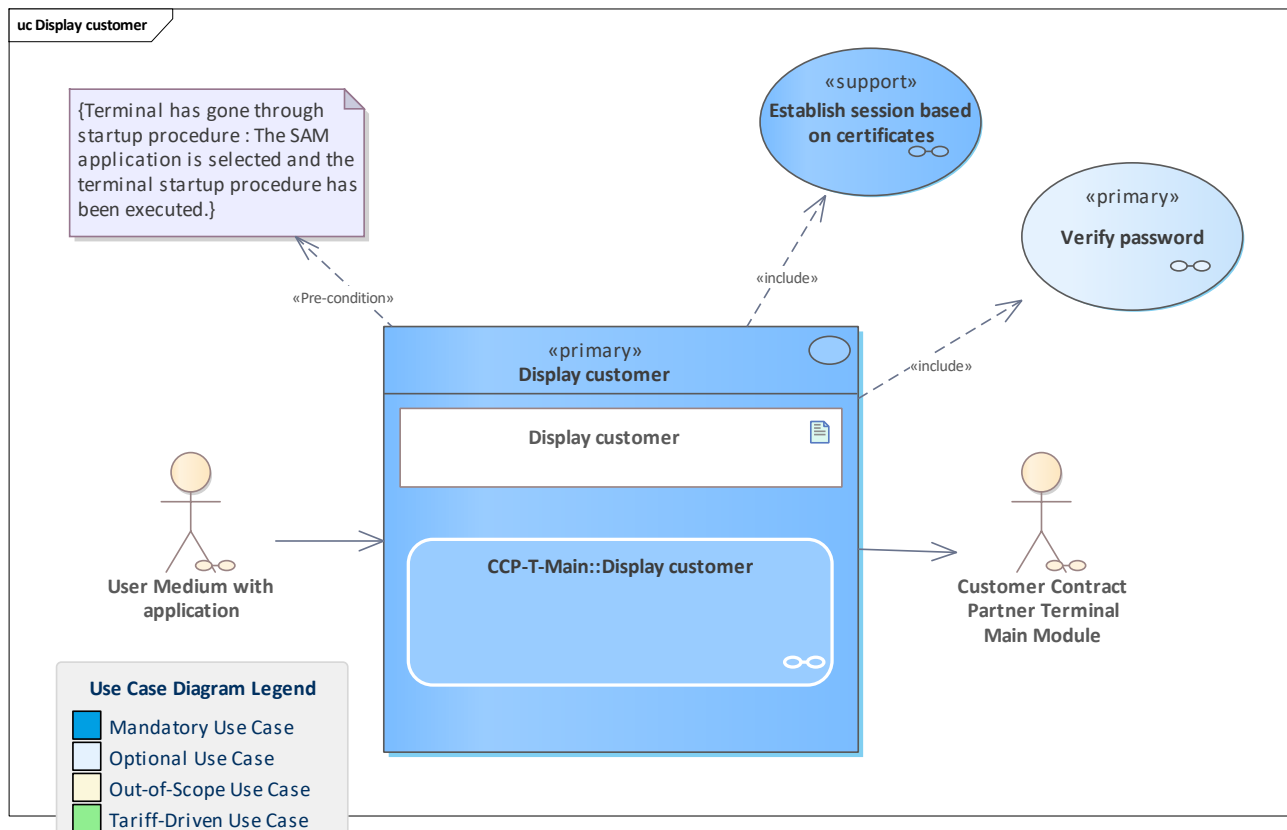


Figure 264: Display customer

The customer wants to display his user medium-located data on a CCP-T.
The customer must enter his password/PIN to show the customer data.

11.67 Display discounts

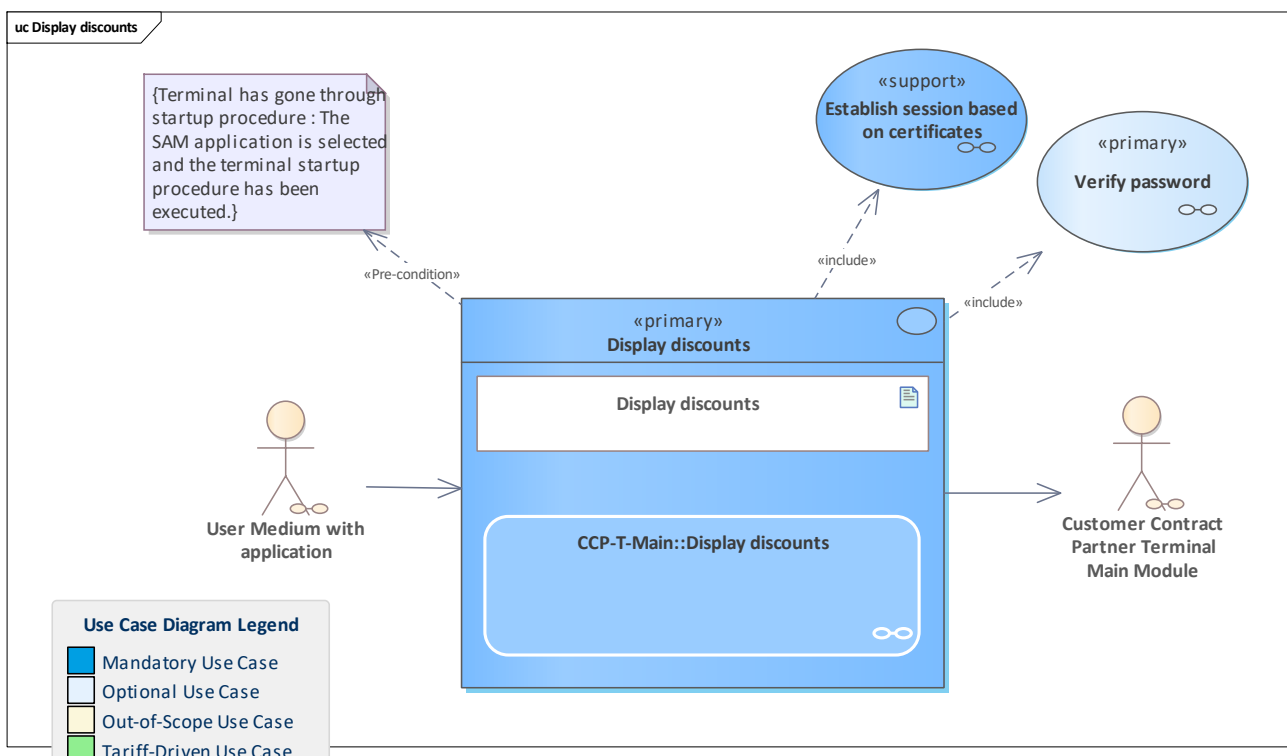


Figure 265: Display discounts

The customer wants to display his user medium-located discount information on a CCP-T. The customer must enter his password/PIN to show the discount information.

11.68 Display entitlement

11.69 Display entitlement

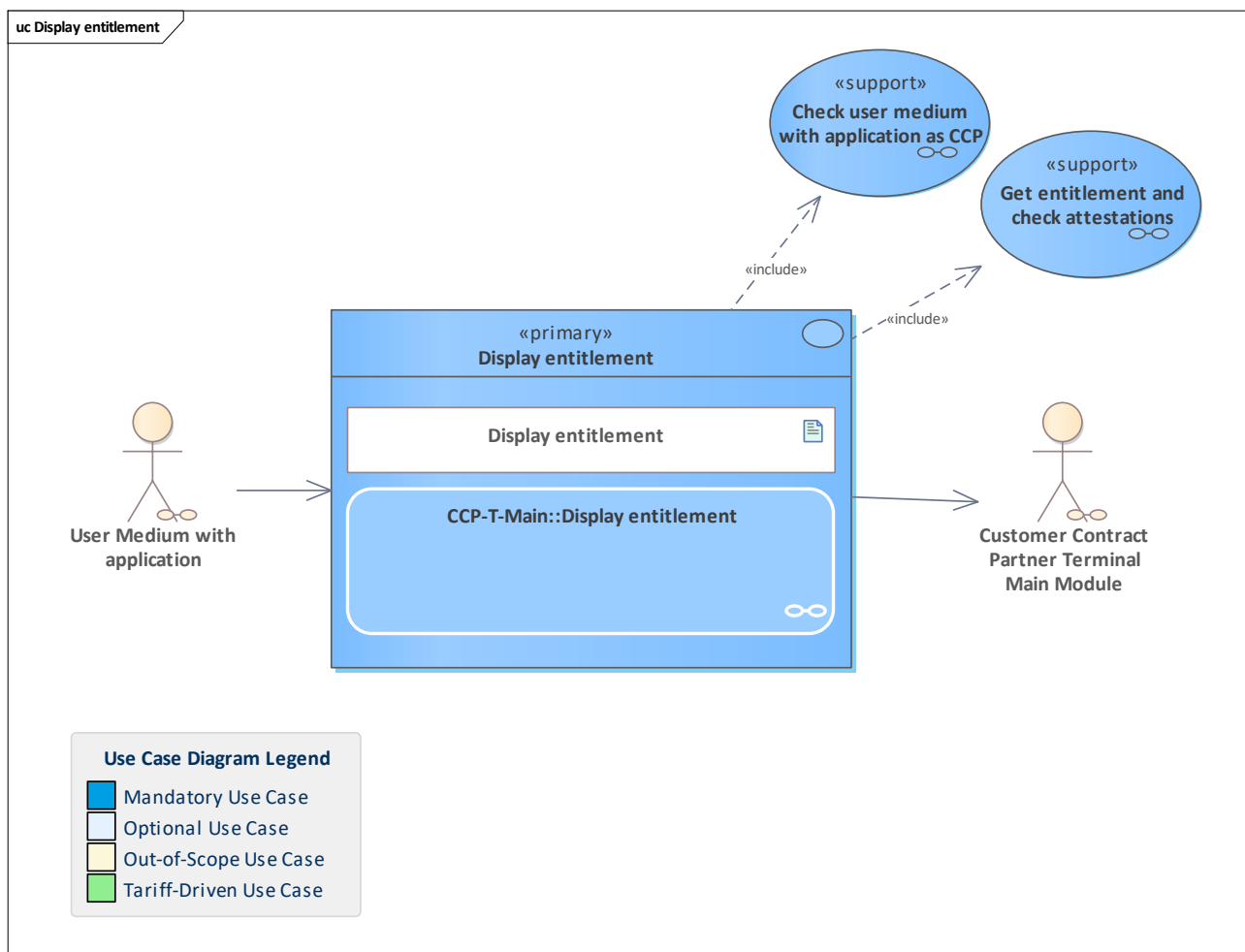


Figure 266: Display entitlement

Display an entitlement on a user medium with an application, independent of its status and validity period.

11.70 Display favourites

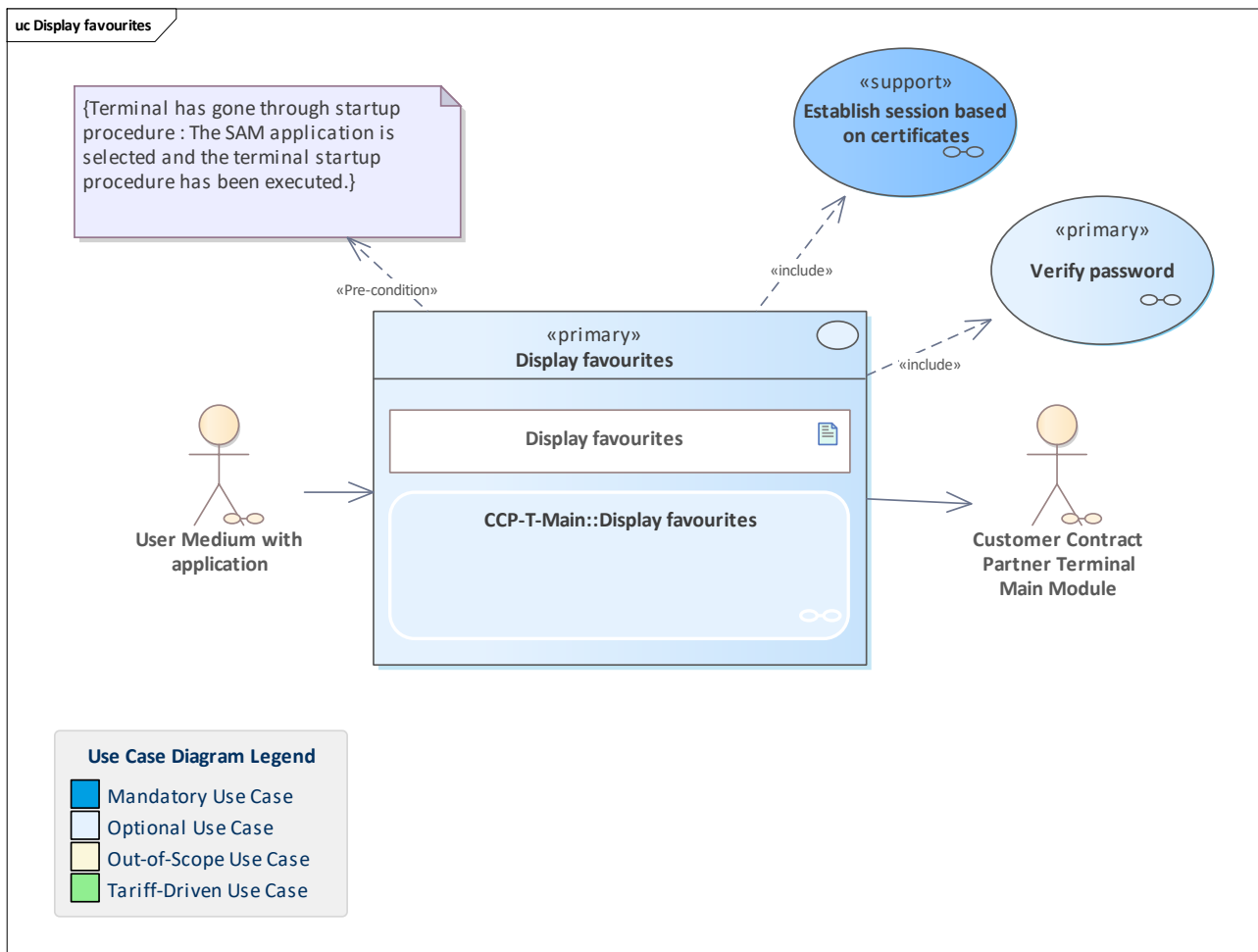


Figure 267: Display favourites

The customer wants to display his user medium-located favourites information on a CCP-T. The customer must enter his password/PIN to show the favourites information.

11.71 Display static entitlement

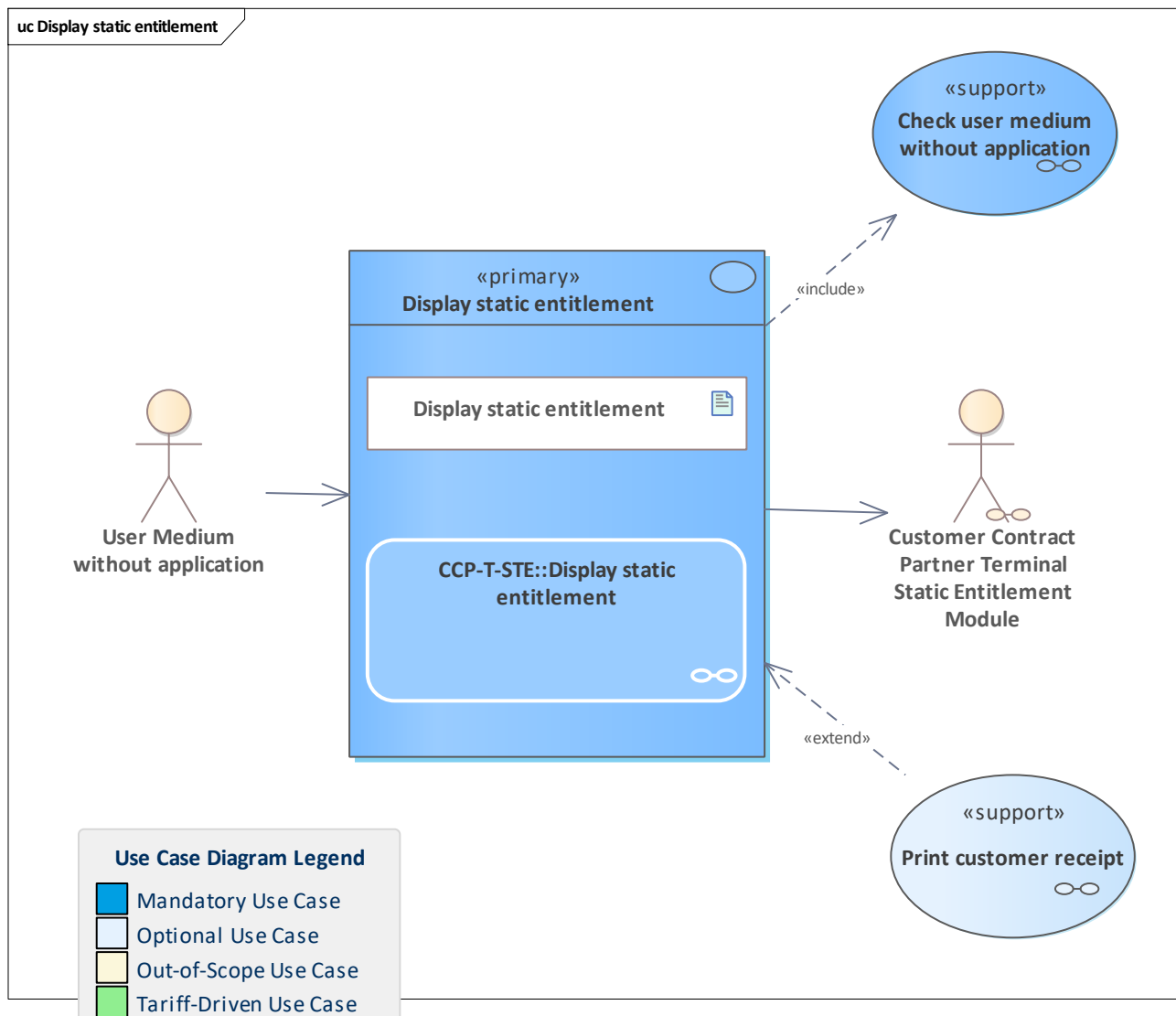


Figure 268: Display static entitlement

Details of a static entitlement are shown by customer service in a terminal.

11.72 Distribute SAM configuration

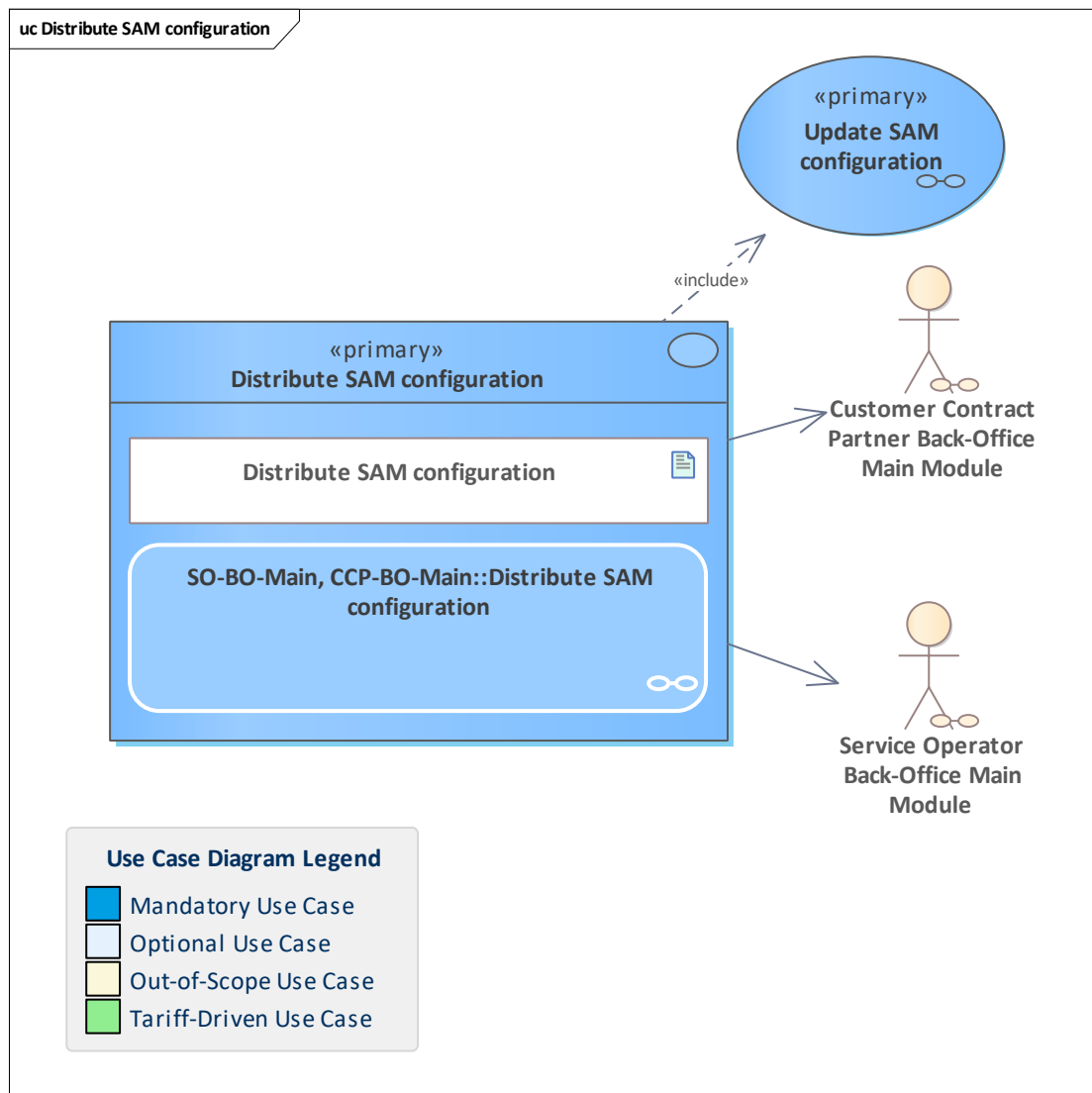


Figure 269: Distribute SAM configuration

Required SAM configuration data was received from the ESH. The back-office system distributes the script for each new SAM configuration.

In this use case, it is assumed that only one SAM configuration per terminal is required for the sake of simplicity.

11.73 Distribute tariff module

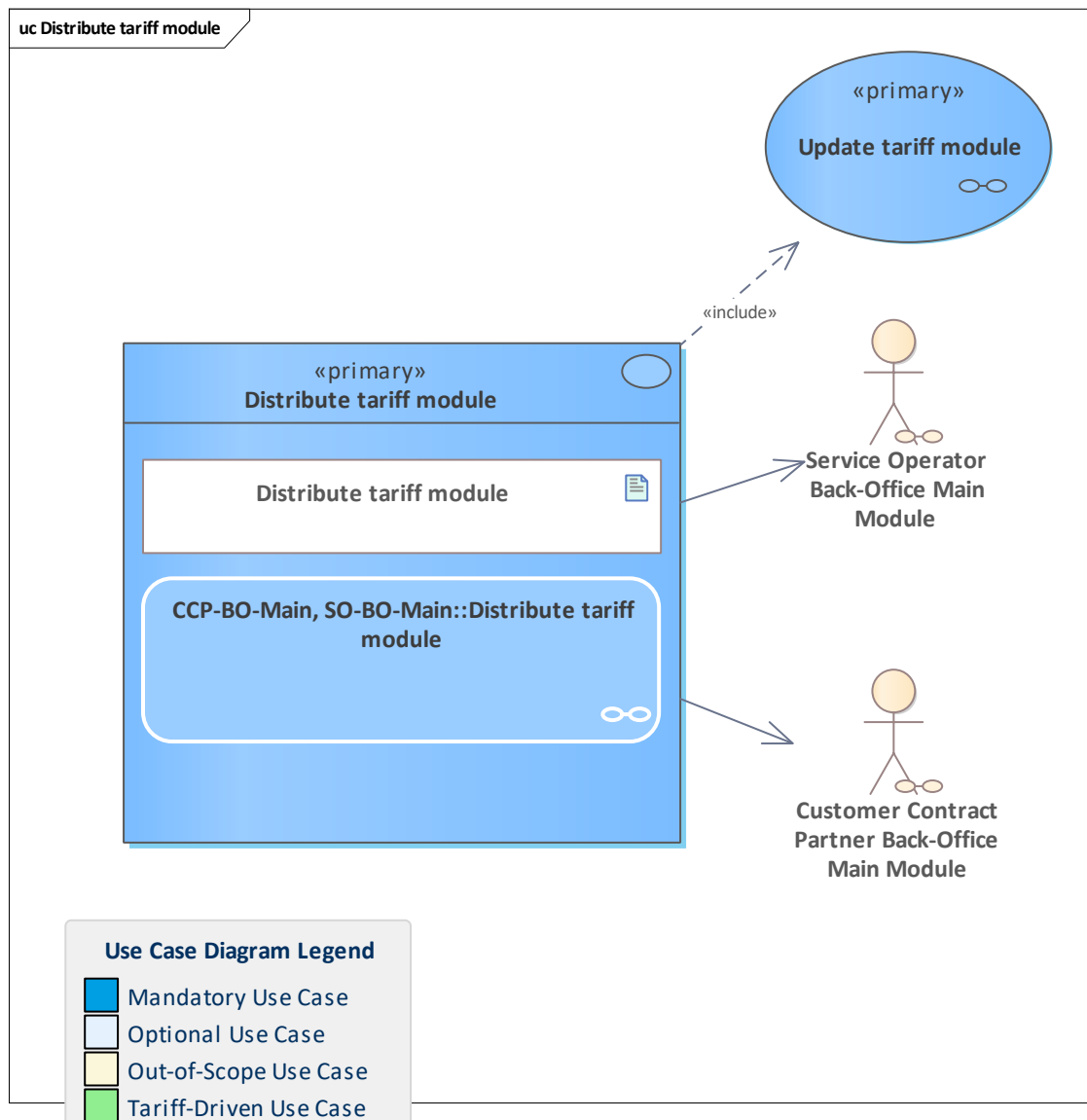


Figure 270: Distribute tariff module

The CPP and SO back-office system retrieves tariff modules from POs (out of scope) and stores them in its data store.

The SO and CCP create one tariff module for terminals based on the tariff modules received from the POs. The SO and CCP distributes this tariff module for its terminals. This process needs to run periodically to keep the tariff module up to date.

11.74 Distribute the SAM reset script

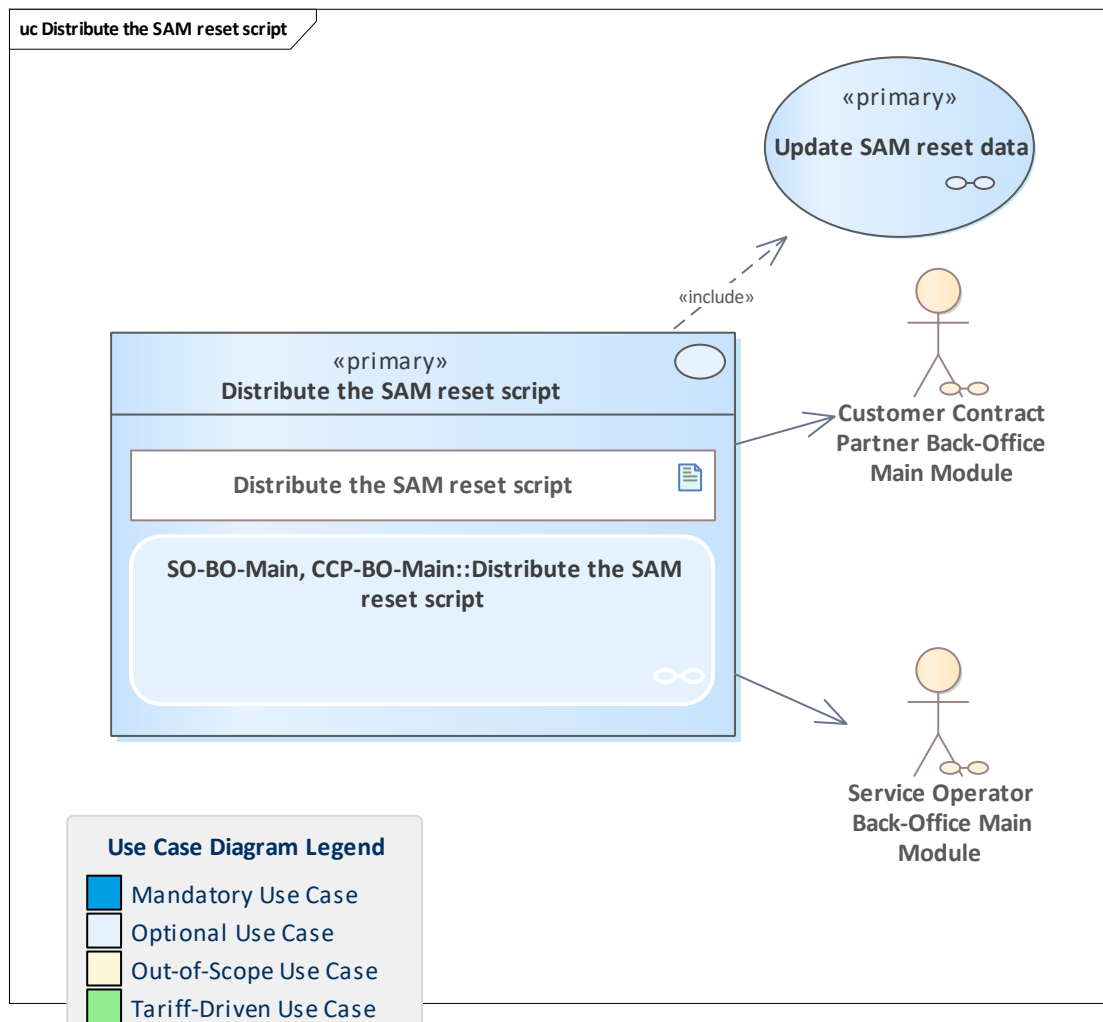


Figure 271: Distribute the SAM reset script

Required SAM reset data was received from the ESH. The back-office system distributes the script for each SAM that needs to be reset.

In this use case it is assumed that only one SAM reset per terminal is required for the sake of simplicity.

11.75 Establish session and get entitlement directory

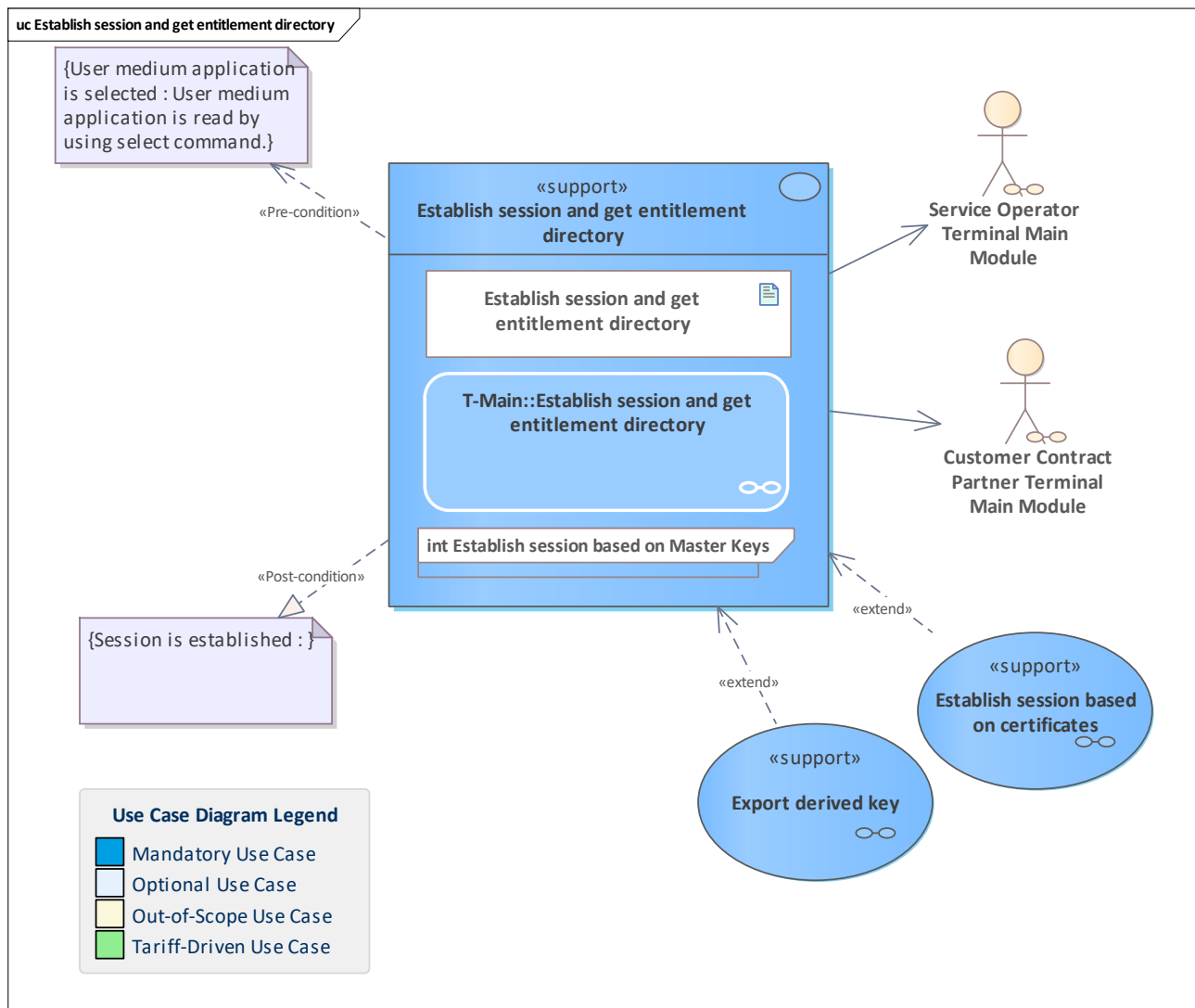


Figure 272: Establish session and get entitlement directory

The terminal establishes a session between the user medium and SAM and securely retrieves the entitlement directory.

If possible, the session is established based on symmetric master keys. Otherwise, the process falls back to certificate-based session establishment and tries to derive a key for the user medium such that future sessions can be established using that key.

The terminal checks that the core parts of the user medium application directory (initially retrieved via SELECT) are authentic.

11.76 Establish session based on certificates

11.77 Establish session based on certificates

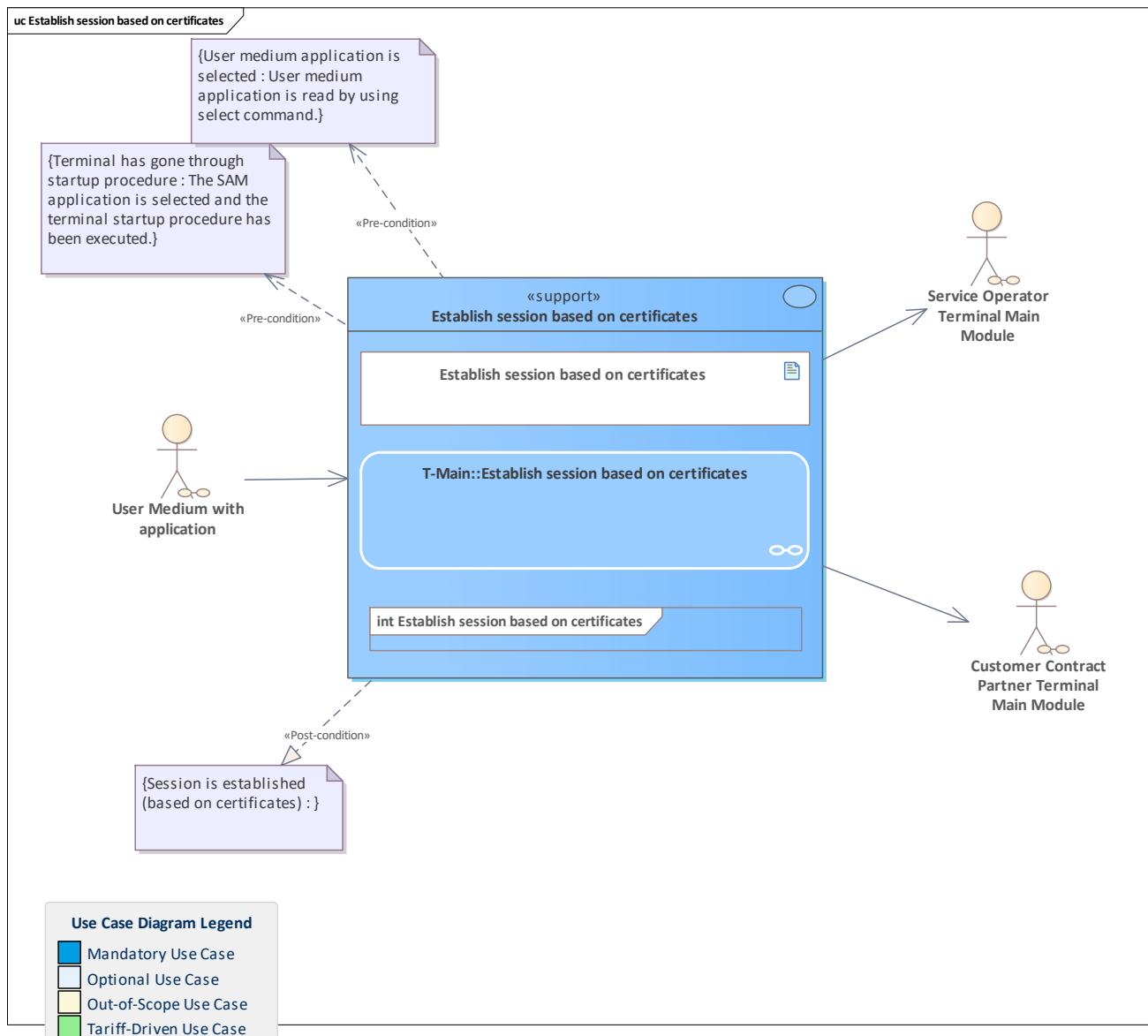


Figure 273: Establish session based on certificates

The terminal establishes a session between user medium and SAM based on certificates. This process can optionally be extended to use ephemeral keys to increase the security properties of the session. For the sake of simplicity, this is not shown. Whether to use ephemeral keys is at the discretion of the terminal.

11.78 Exchange user medium with application

11.79 Exchange user medium with application

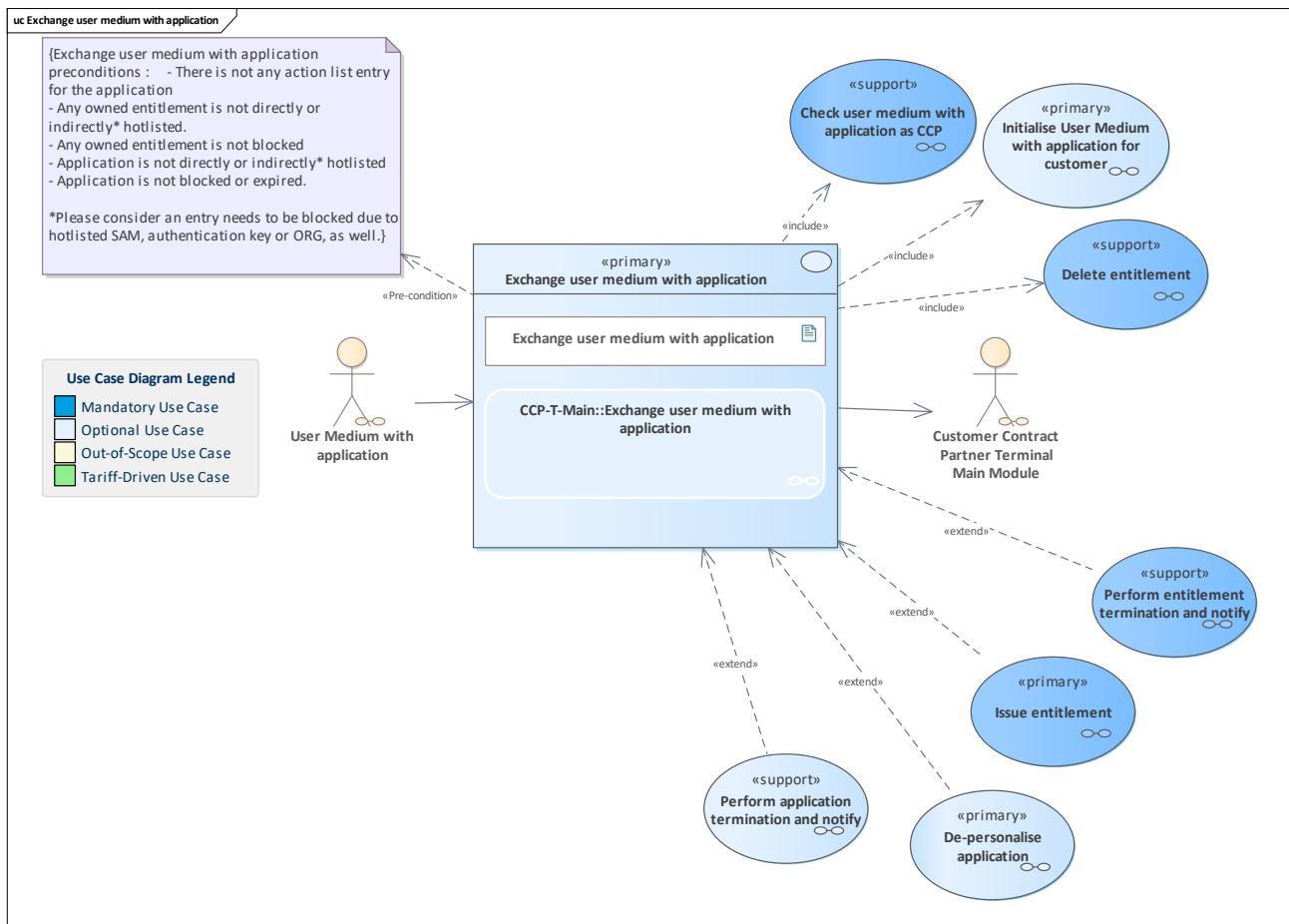


Figure 274: Exchange user medium with application

A user medium with an application is to be exchanged with the new one. This can be done only by the pCCP.

This process considers not only the migration of application data but also entitlements. Please note that only owned entitlements can be migrated.

Please note that it is assumed that there is no need for any payment transaction.

11.80 Execute action list entries

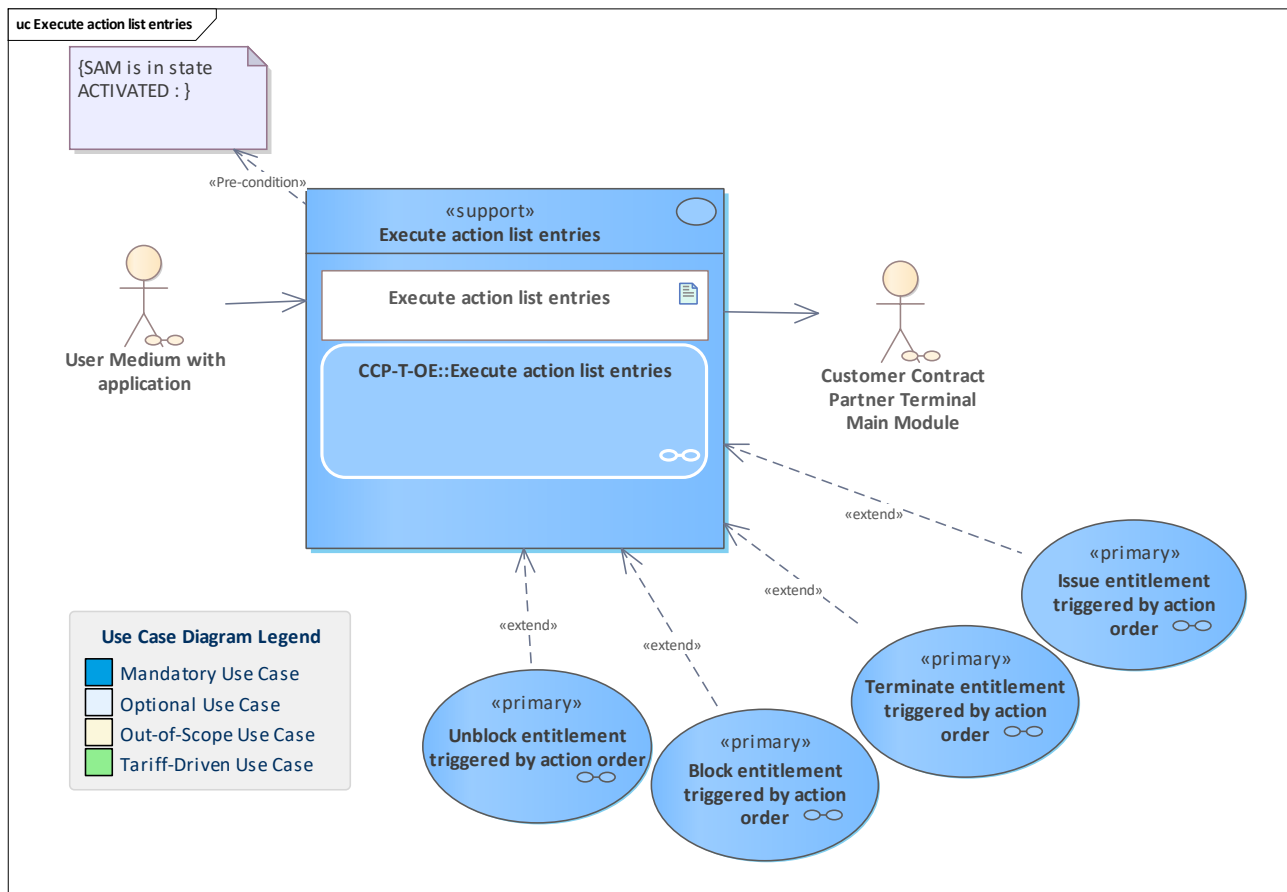


Figure 275: Execute action list entries

Check for action list entries and, if necessary, execute them.

Triggered by the general processes of [CCP-T-Main::Check user medium with application](#) if an action management extension exists. The action list is searched for possible actions for the user medium which is currently presented to the CCP terminal.

If one or more action entries exist for the application instance ID of the current user medium, it is executed.

Note that this process may only be run if the SAM is activated to guarantee that no termination is executed for which a linked (replacement) issuance cannot be executed.

11.81 Export derived key

11.82 Export derived key

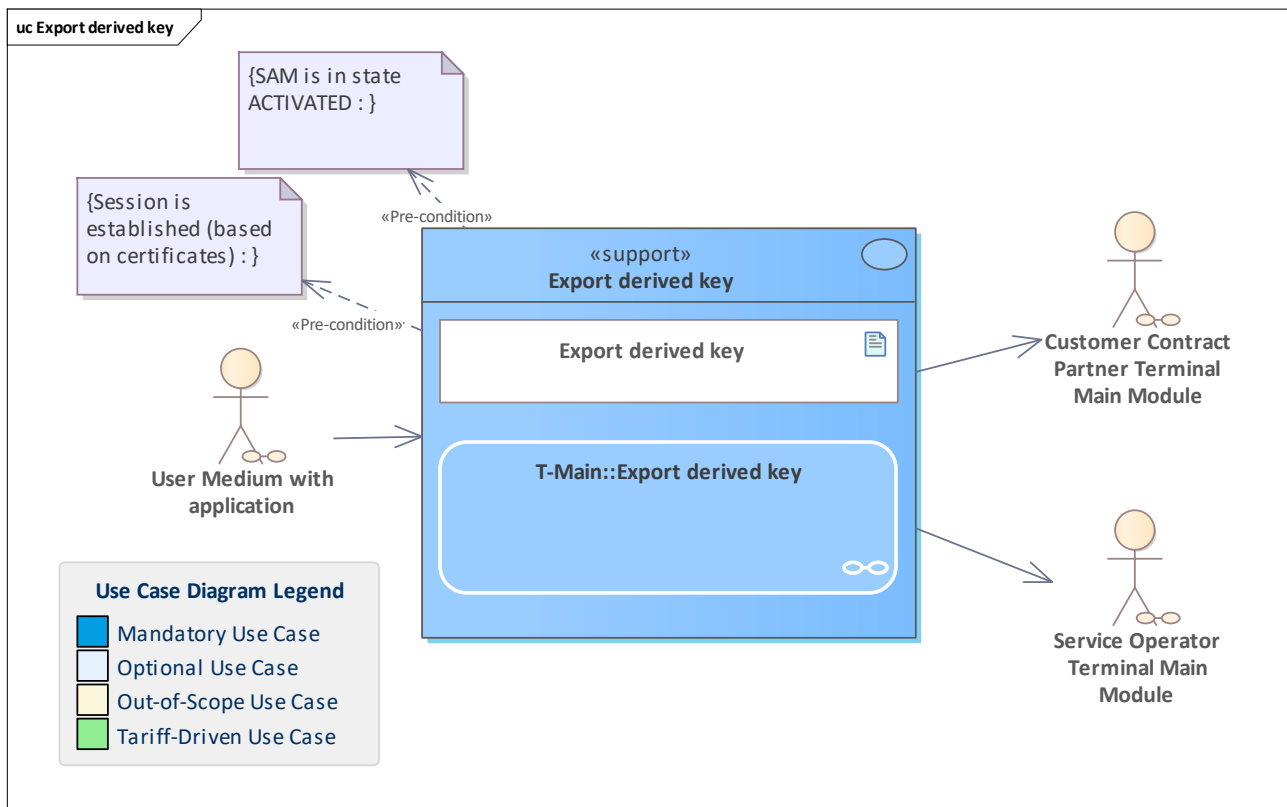


Figure 276: Export derived key

The terminal derives a key and exports it to the user medium application.

11.83 Generate current action list

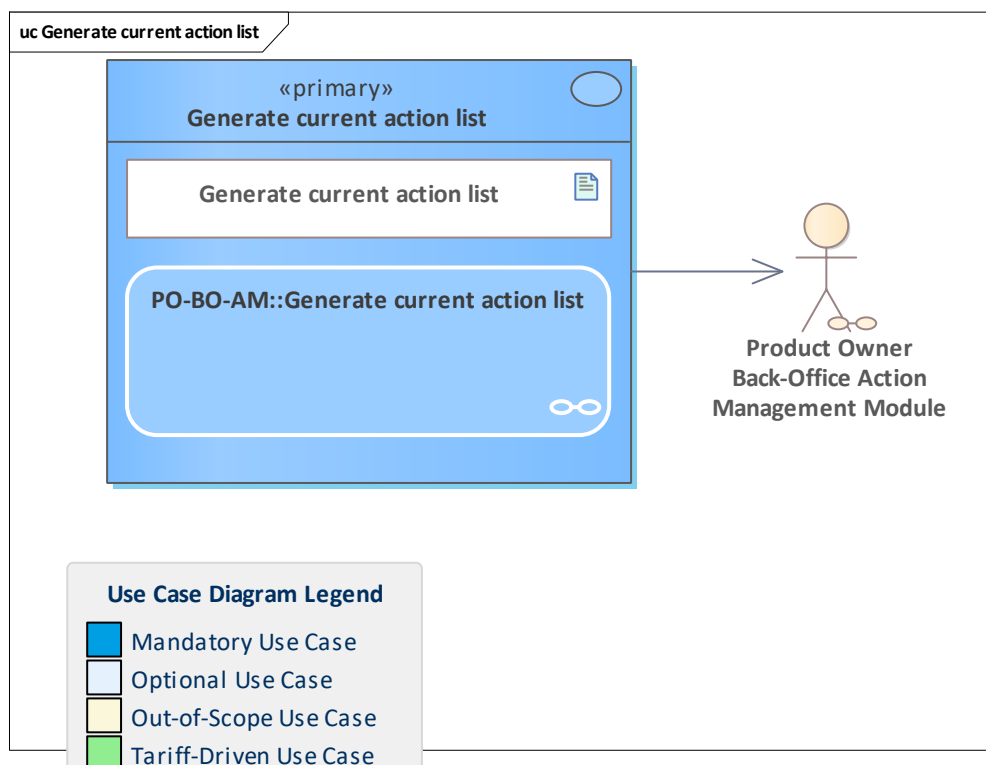


Figure 277: Generate current action list

The [Product Owner Back-Office Action Management Module](#) generates and stores the action list for a new cycle.

This process runs periodically on its own.

11.84 Generate current hotlists

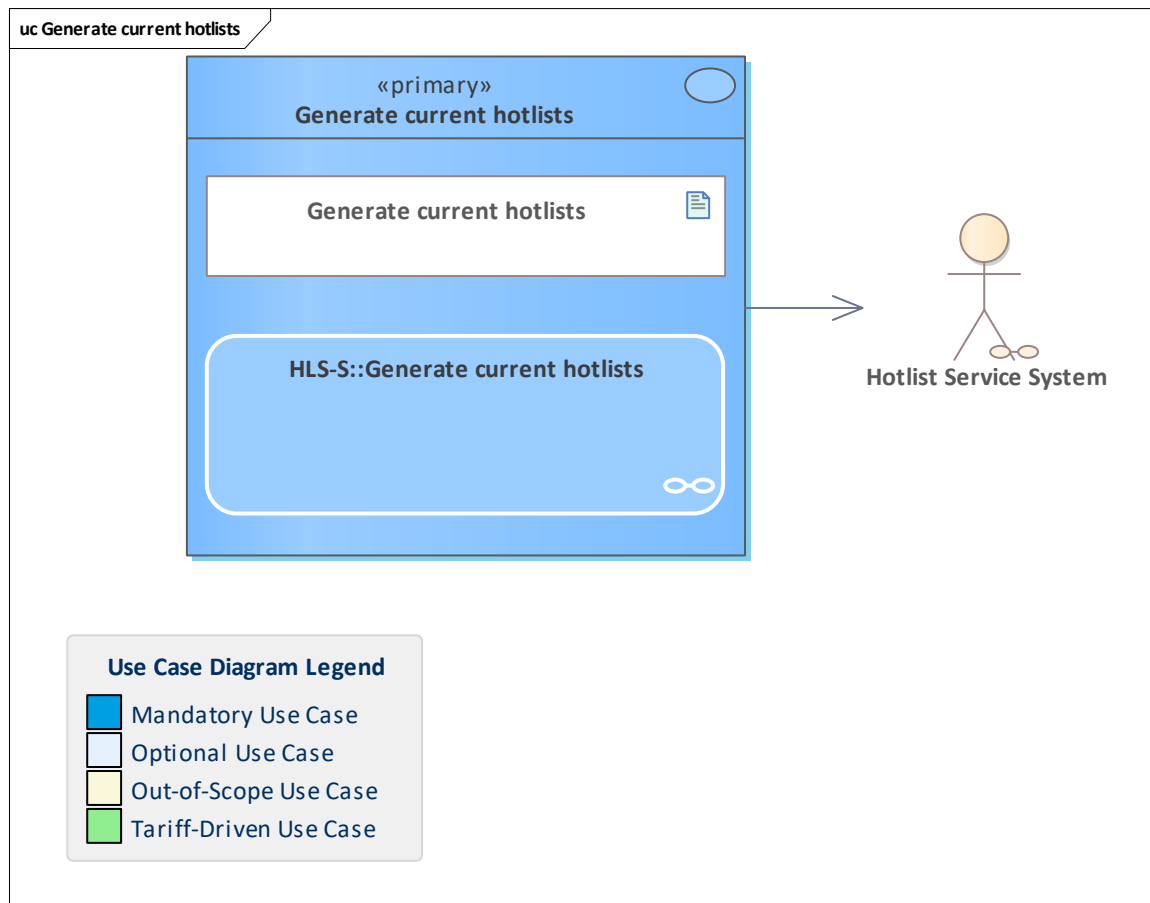


Figure 278: Generate current hotlists

Current hotlist inventories are generated and stored in the database for a later hotlist compilation when requested.

11.85 Get entitlement and check attestations

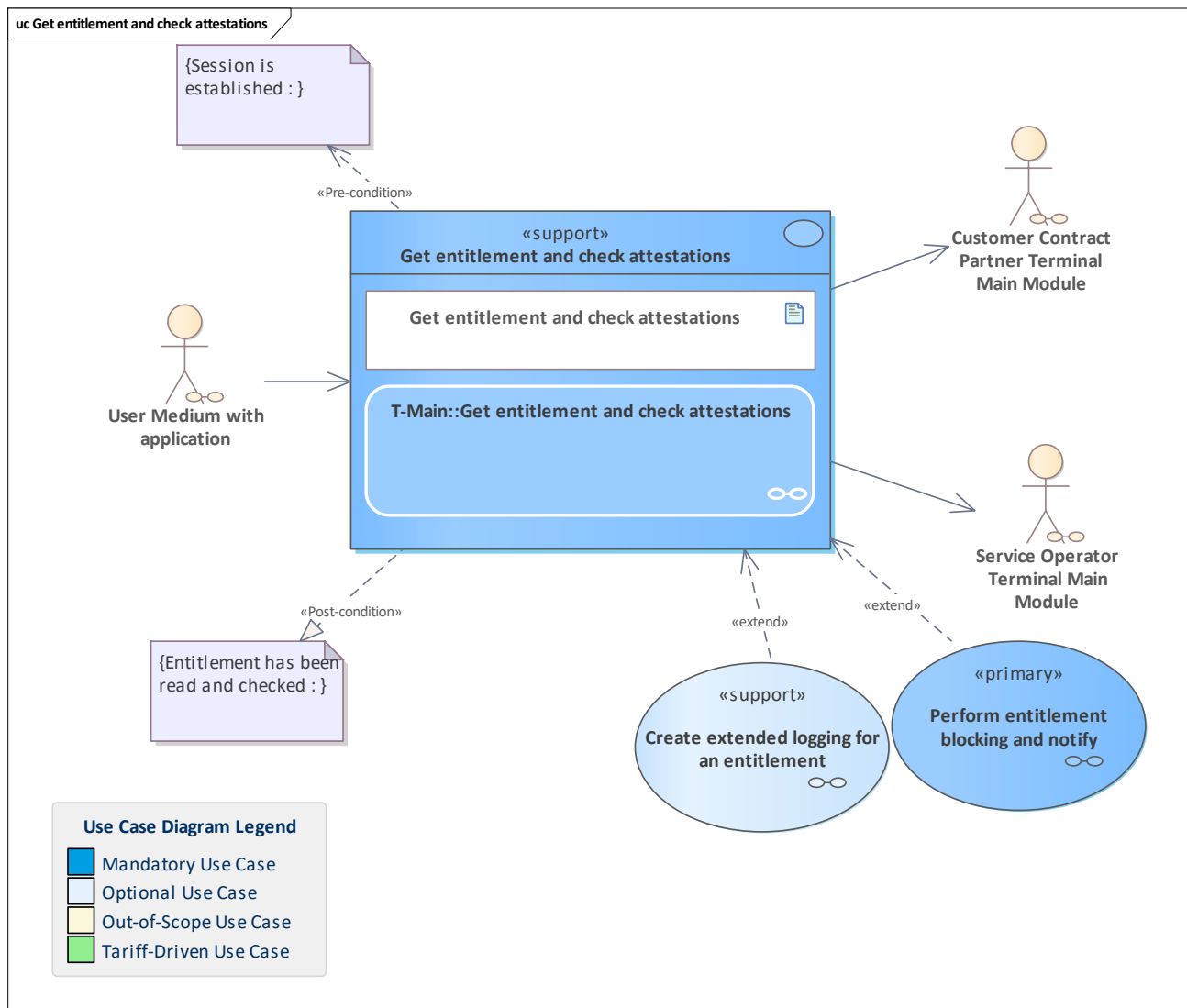


Figure 279: Get entitlement and check attestations

This supporting use case describes that

- the requested entitlement is read from the user medium and validated
- available attestations are checked against the hotlist
- in case of a hotlist match, the entitlement is blocked

11.86 Get product acceptance configuration list

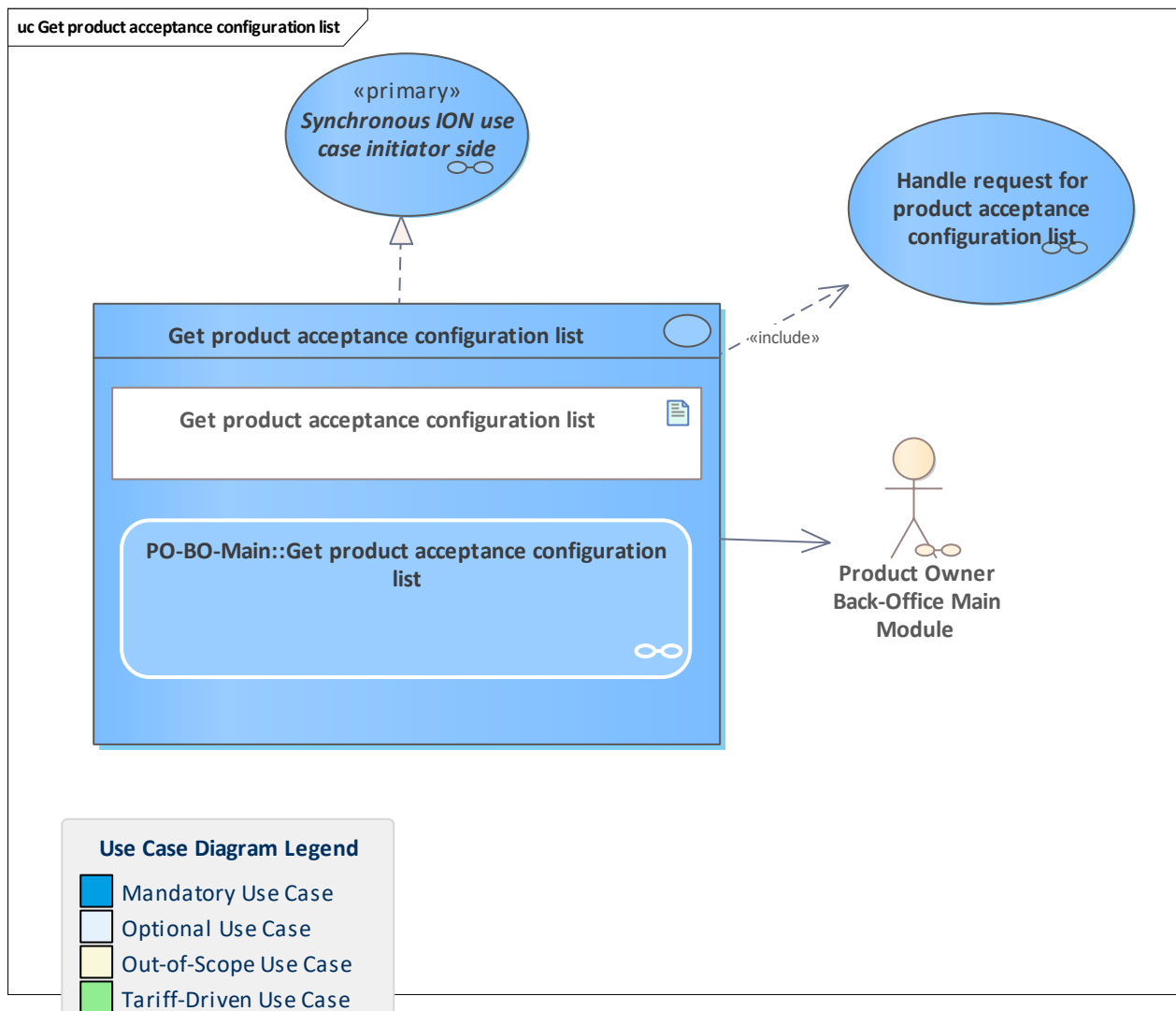


Figure 280: Get product acceptance configuration list

The PO retrieves its product acceptance configuration from the hotlist service system as a compressed list of product acceptance entries, where each entry contains the ID of the accepting organisation (SO or CCP), the product owner ID and, optionally, the product number. If the product number is omitted, the organisation accepts all products of the PO.

11.87 Get unclaimed list information

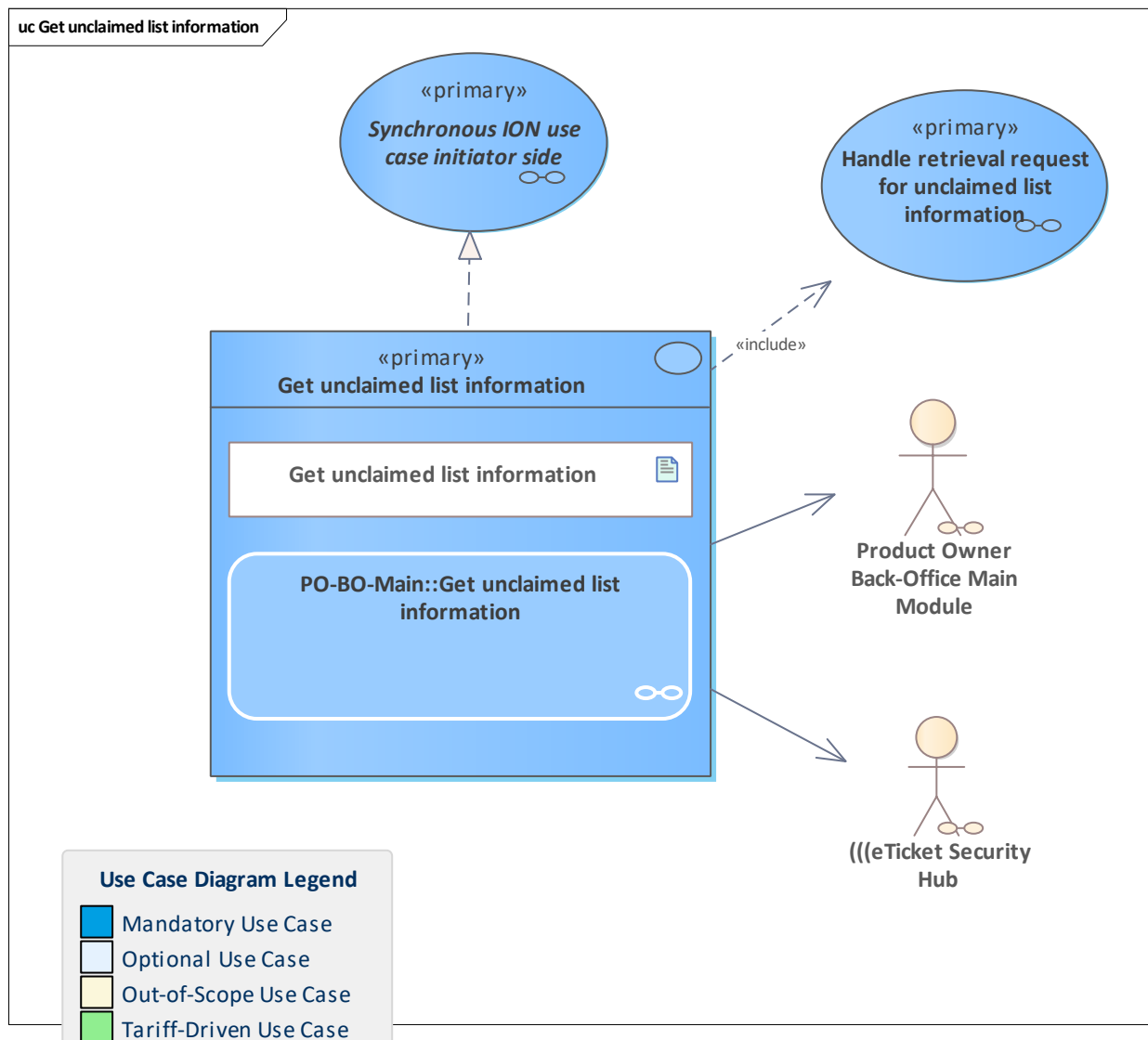


Figure 281: Get unclaimed list information

Use case that allows the PO to monitor the collection behaviour of its partner companies with the role SO and CCP.

The scheme manager monitors all SO, CCP and PO organisations.

The requestor gets the unclaimed list information from the hotlist service system for a period between the passed list cycle and the current list cycle. This information is registered and can be gathered for a regular report.

The following list types are expected for the organisation's roles:

- CCP and SO: application-, entitlement-, SAM-, organisation- and authentication-key-hotlist
- PO (included if the actor is the scheme manager): entitlement-, SAM- and organisation-hotlist

11.88 Handle account-based payment method charging from contractual perspective

11.89 Handle account-based payment method charging from contractual perspective

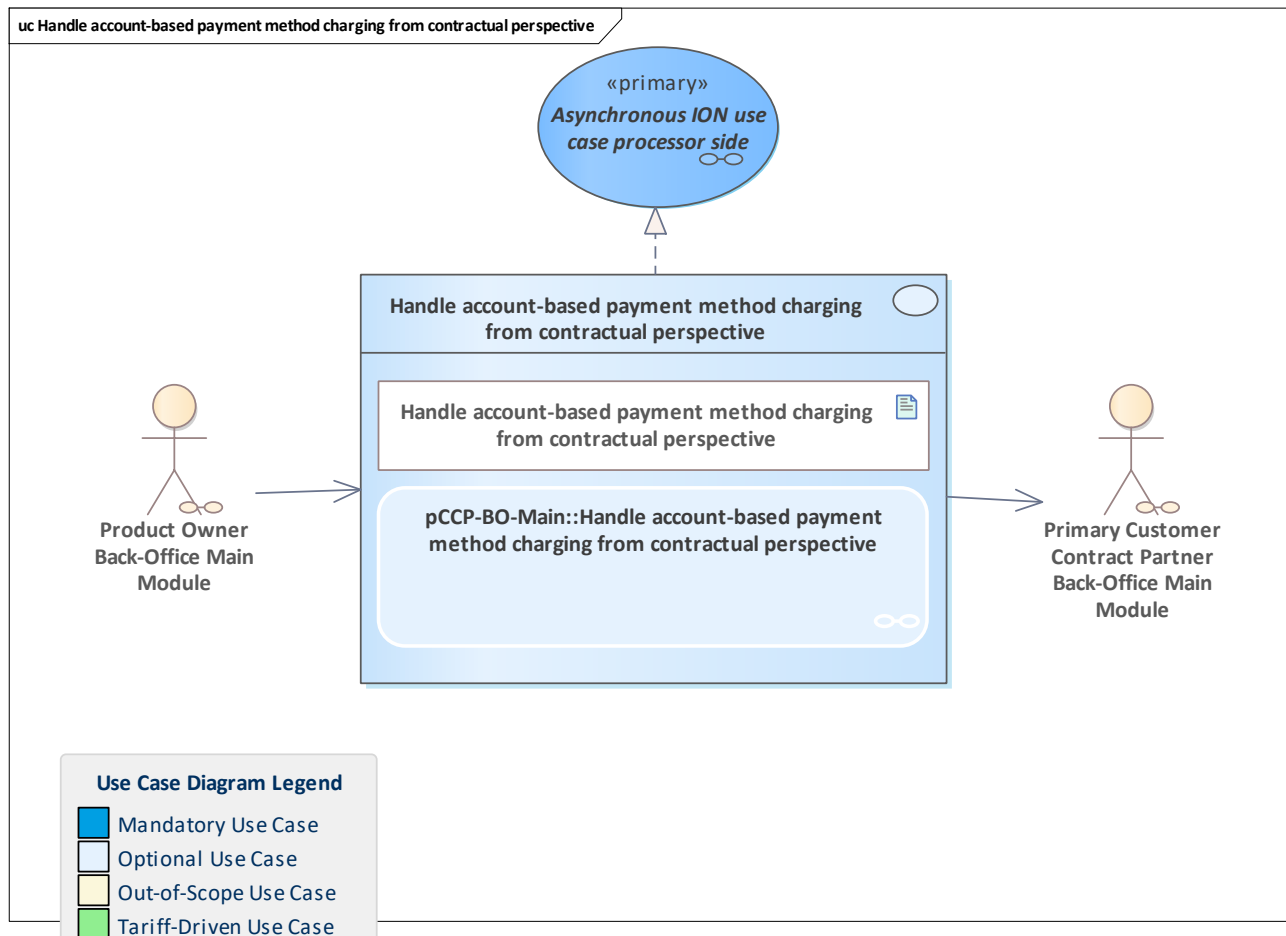


Figure 282: Handle account-based payment method charging from contractual perspective

Account-based payment method charging is received from the PO and handled.

The message contains a list of rated journeys with references to the messages about check-in and check-out procedures that had been sent before.

The list has to be verified and the price must be booked from the related account-based payment method.

Furthermore, the journey list can be used to provide an itemised bill e.g. via the web portal for the customer.

11.90 Handle account-based payment method credited notification from contractual perspective

11.91 Handle account-based payment method credited notification from contractual perspective

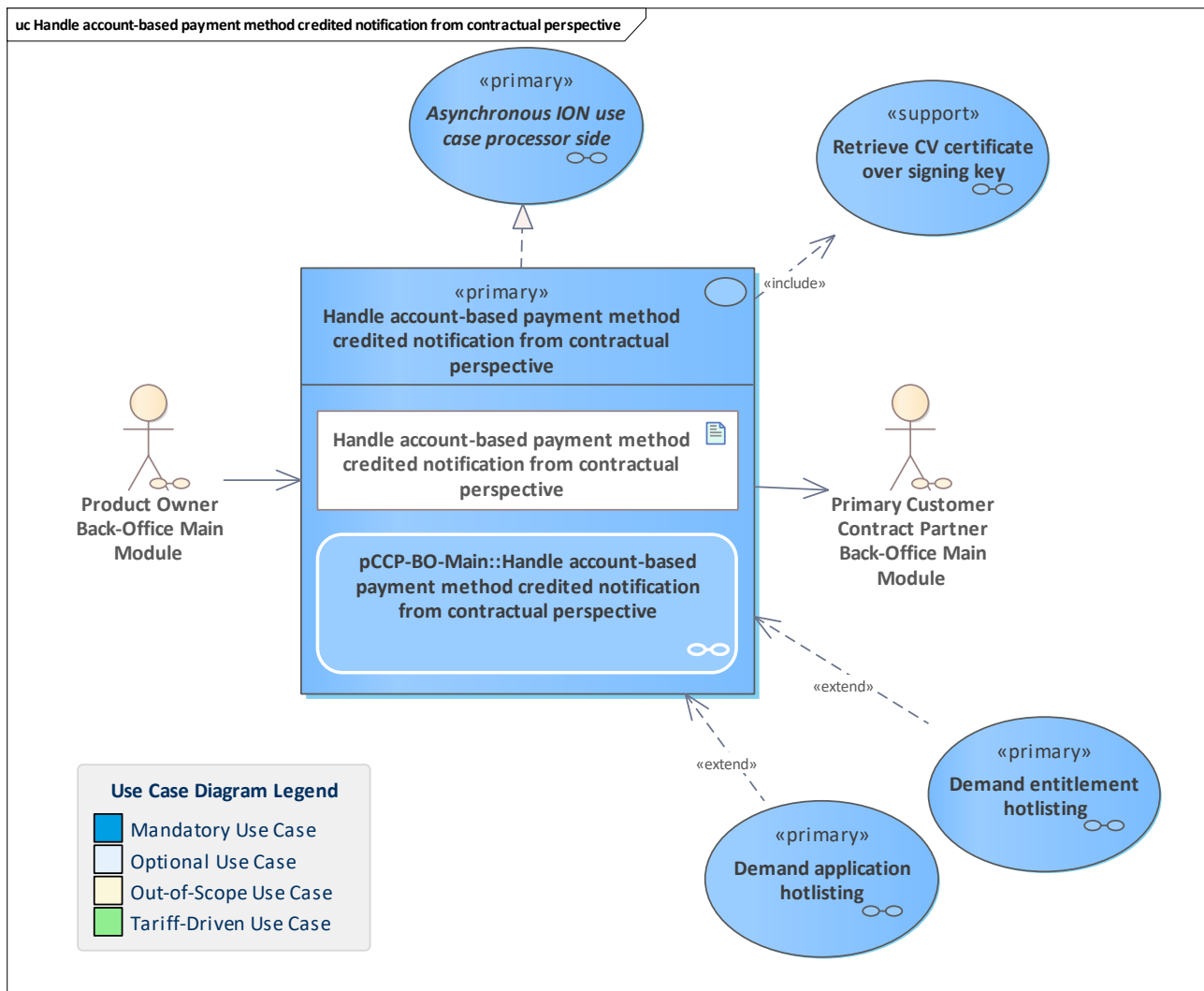


Figure 283: Handle account-based payment method credited notification from contractual perspective

Handle an account-based payment method credited notification from the contractual perspective.

The pCCP does its contractual checks and monitoring.

Note: if the pCCP itself performed the credit action, this use case takes inside the use case [Handle account-based payment method credited notification from operational perspective](#) and is not called by the PO.

In this case, this use case is not an [Asynchronous ION use case processor side](#).

11.92 Handle account-based payment method credited notification from operational perspective

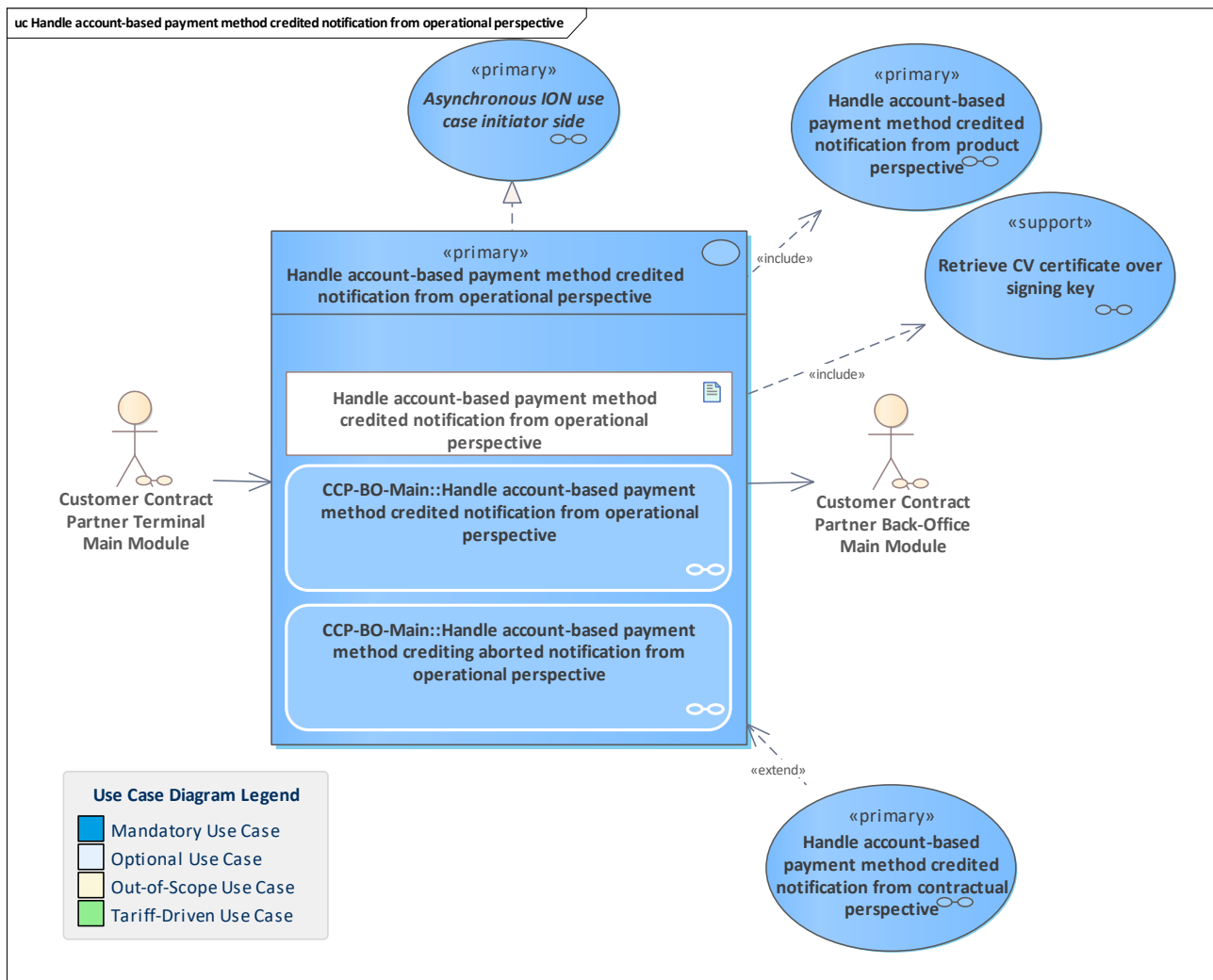


Figure 284: Handle account-based payment method credited notification from operational perspective

Handle an account-based payment method credited notification from the operational perspective.

The CCP back-office system receives the notification about the credit transaction of an account-based payment method and registers it.

Then it handles the notification from the operational perspective by performing checks and monitoring, e.g. verifying the signature of the credit action attestation.

Then, the notification is forwarded to the responsible PO system.

If the current CCP is the pCCP, it can also do the contractual checks and monitoring directly.

In the case of an abortion, the notification is registered for consistent monitoring. Since no counters are affected which are important for the PO, the notification is not forwarded.

11.93 Handle account-based payment method credited notification from product perspective

11.94 Handle account-based payment method credited notification from product perspective

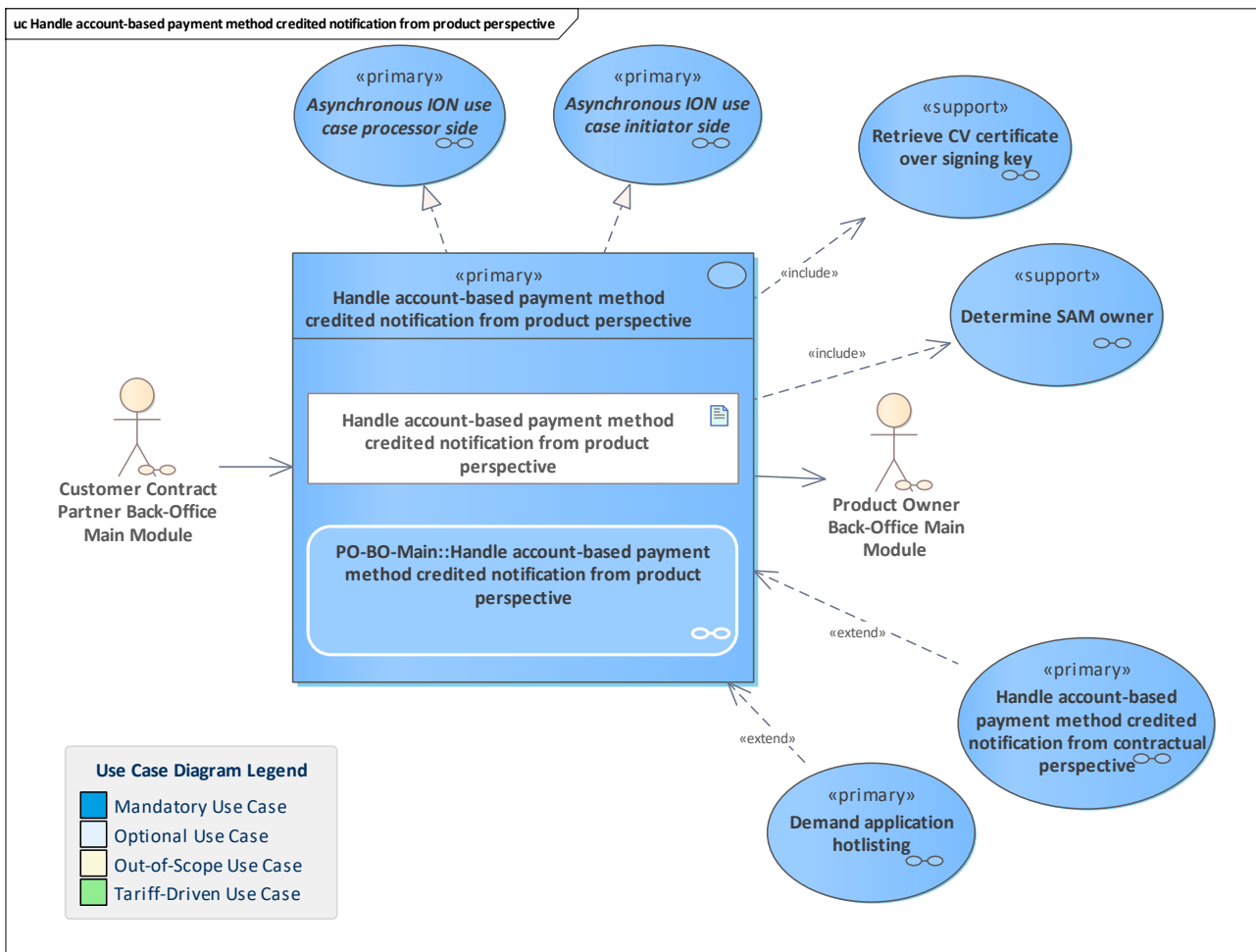


Figure 285: Handle account-based payment method credited notification from product perspective

Handle an account-based payment method credited notification from the product perspective. The PO back-office system receives and registers the notification about a performed credit with an account-based payment method from the CCP system. It does the checks and monitoring from the product owner perspective. If the sending CCP was not the pCCP, the notification is forwarded to the responsible pCCP system.

11.95 Handle account-based payment method debited notification from contractual perspective

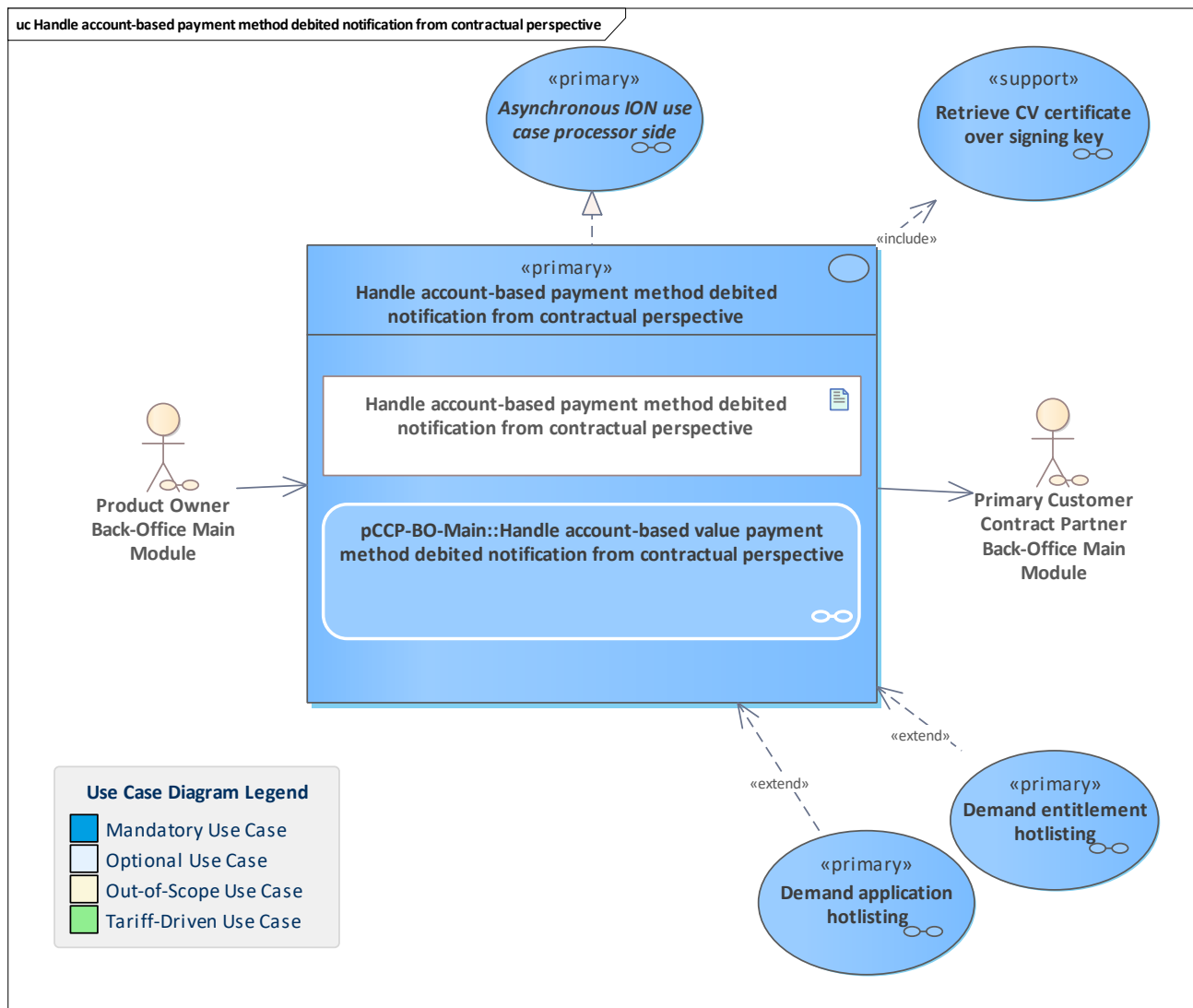


Figure 286: Handle account-based payment method debited notification from contractual perspective

Handle a notification about an account-based payment method debiting from the contractual perspective.

The pCCP does its contractual checks and monitoring.

Note: if the pCCP itself performed the debit action, this use case takes place inside the use case [Handle account-based payment method debited notification from operational perspective](#) and is not called by the PO.

In this case, this use case is not an [Asynchronous ION use case processor side](#).

11.96 Handle account-based payment method debited notification from operational perspective

11.97 Handle account-based payment method debited notification from operational perspective

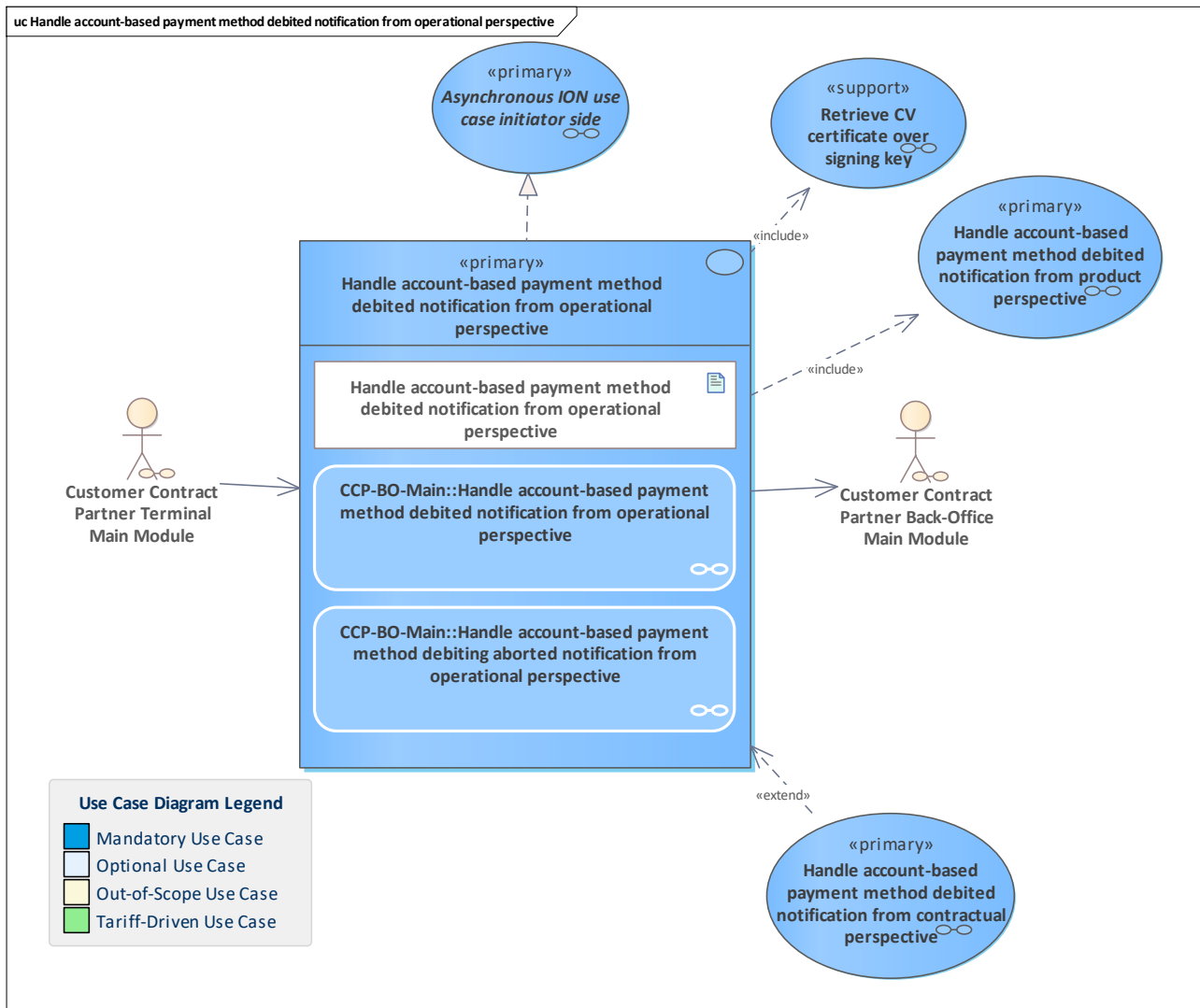


Figure 287: Handle account-based payment method debited notification from operational perspective

Handle a notification about an account-based payment method debiting from the operational perspective.

The CCP back-office system receives the notification about the debit action of an account-based payment method and registers it.

Then it handles the notification from the operational perspective by performing checks and monitoring, e.g. verifying the signature of the debit action attestation.

Then, the notification is forwarded to the responsible PO system.

If the current CCP is the pCCP, it can also do the contractual checks and monitoring directly.

In the case of an abortion, the notification is registered for consistent monitoring. Since no counters are affected which are important for the PO, the notification is not forwarded.

11.98 Handle account-based payment method debited notification from product perspective

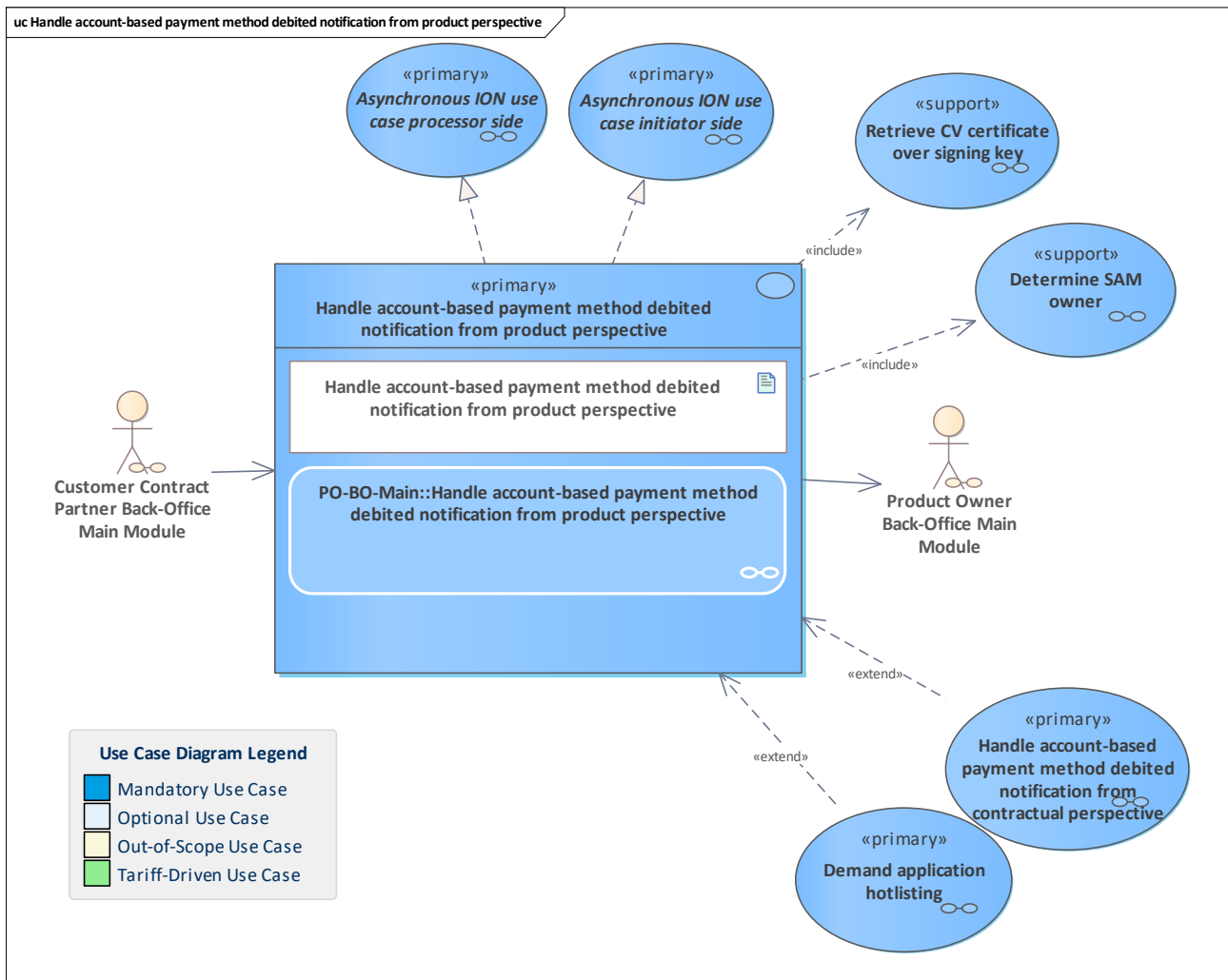


Figure 288: Handle account-based payment method debited notification from product perspective

Handle a notification about an account-based payment method debiting from the product perspective.

The PO back-office system receives and registers the notification about a performed debit with an account-based payment method from the CCP system.

It does the checks and monitoring from the product owner perspective.

If the sending CCP was not the pCCP, the notification is forwarded to the responsible pCCP system.

11.99 Handle application blocked notification from contractual perspective

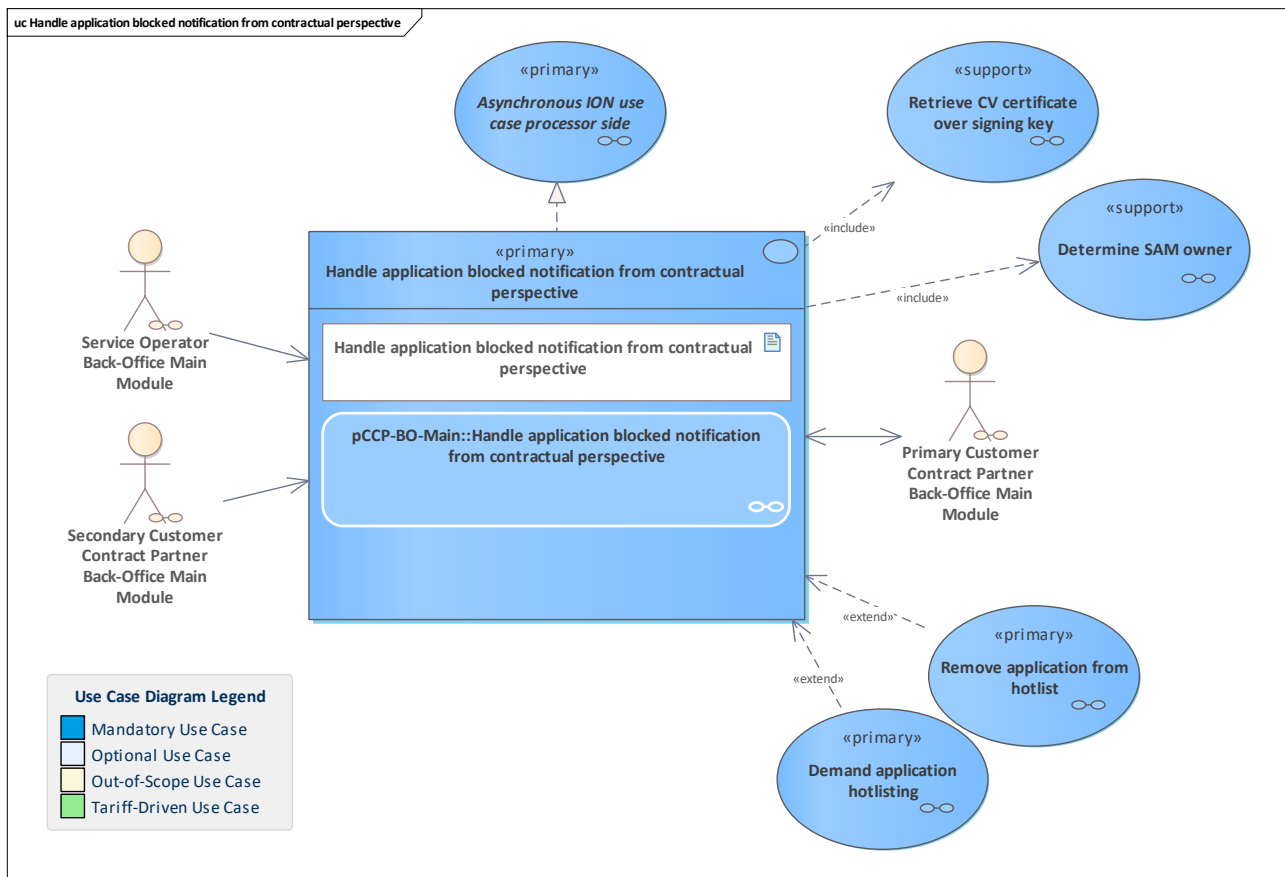


Figure 289: Handle application blocked notification from contractual perspective

The notification regarding user medium application blocked is received by the pCCP. The pCCP does its checks and monitoring from the contractual perspective regarding the correct execution of blocking. In this context, the SAM owner of the SAM that performed the blocking is determined. Furthermore, the signature of the blocking attestation is verified. If the blocking is correct, the pCCP initiates the removal of the application from the hotlist. **Note:** due to monitoring checks, the application may be demanded to be hotlisted. In this case, the pCCP will add the application directly to the hotlist again (or will not remove it).

11.100 Handle application blocked notification from operational perspective

11.101 Handle application blocked notification from operational perspective

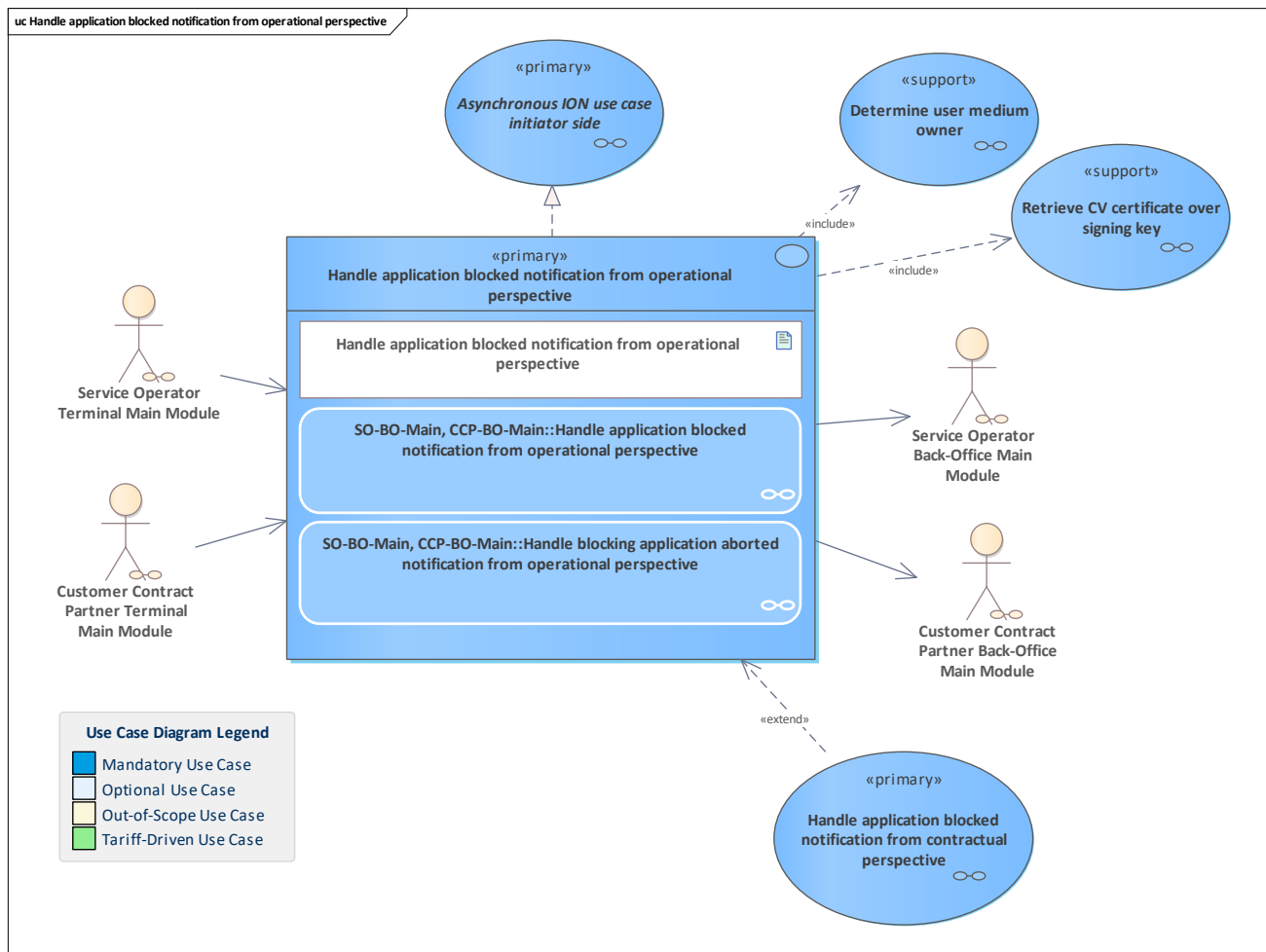


Figure 290: Handle application blocked notification from operational perspective

A terminal has blocked the application in a user medium with an application. In this use case, the SO and/or CCP back-office system receives the notification from the terminal and runs certain checks from the issuer perspective, such as the signature verification of the block application attestation. The pCCP of the application is informed if the current operator is a SO or sCCP. In the case of action abortion, terminal and SAM action data is sent by the terminal to the back-office system. The SO or the CCP back-office system registers the abortion for internal monitoring.

11.102 Handle application hotlisting demand

11.103 Handle application hotlisting demand

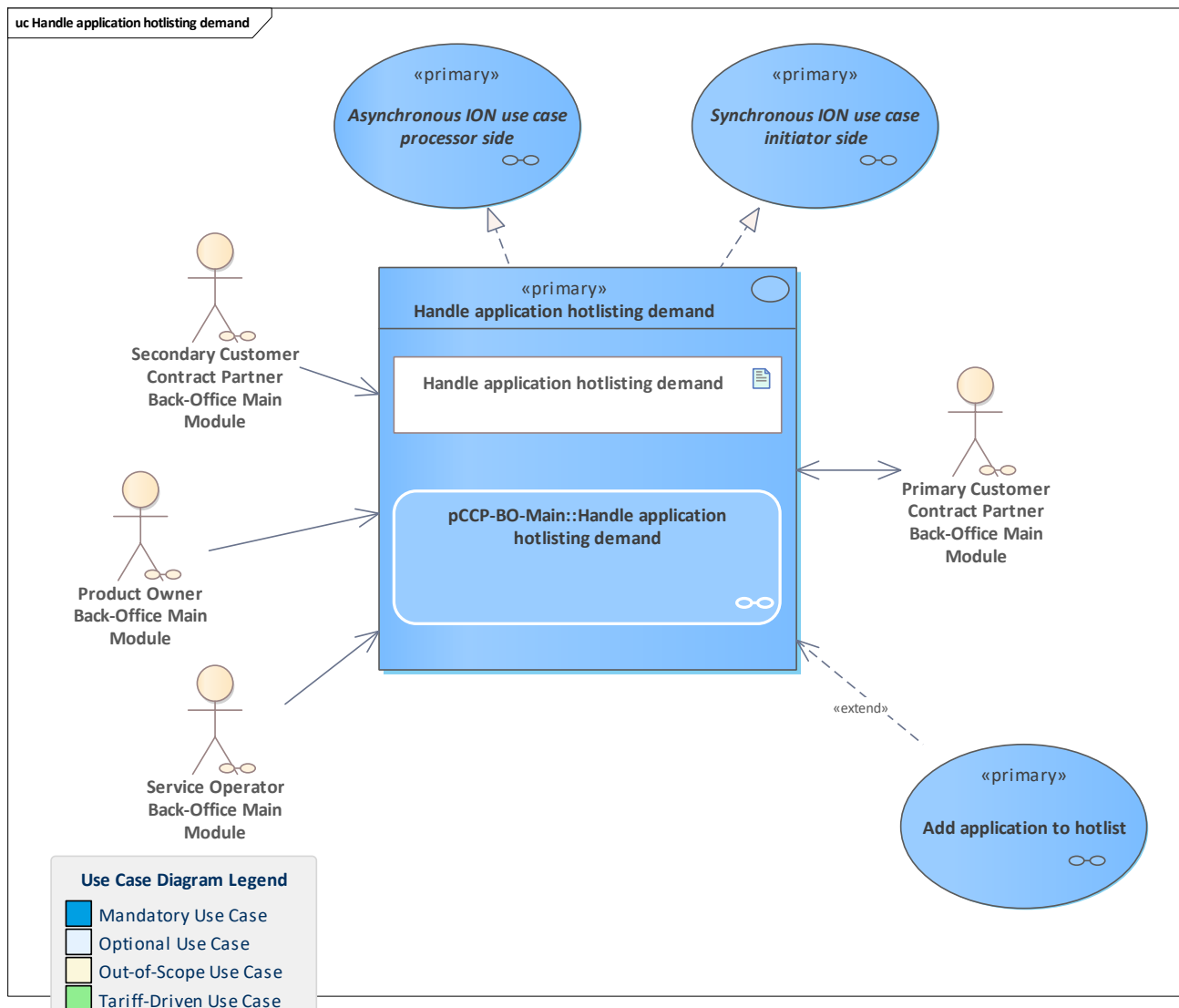


Figure 291: Handle application hotlisting demand

The pCCP that issued the application to a customer checks the hotlisting demand.

If the result of the check requires a hotlisting, the hotlist service system will be called to add the application to the hotlist.

Otherwise, the demand will be rejected and there is no need to communicate with the hotlist service system. The original caller of the demand is informed.

Please note that if there is more than one demand for hotlisting the same application, this has to be considered in the check for a required hotlisting but will not result in an exception.

Especially for the monitoring of a third-party system, it must be possible to demand hotlisting even if the same object was demanded for hotlisting in the past.

Several hotlisting requests for an application can be received. The final decision for a hotlisting must be decided in the pCCP, by considering all hotlisting demands, hotlisting demand revocations and hotlisting orders.

If the application is already hotlisted, the demand will not cause a further hotlist request towards the hotlist service system.

Note: in the ION context, the use case is asynchronous as processor (process the demand) and a synchronous use case as initiator due to the normally performed call to the hotlist service for adding the application to the hotlist.

11.104 Handle application terminated notification from contractual perspective

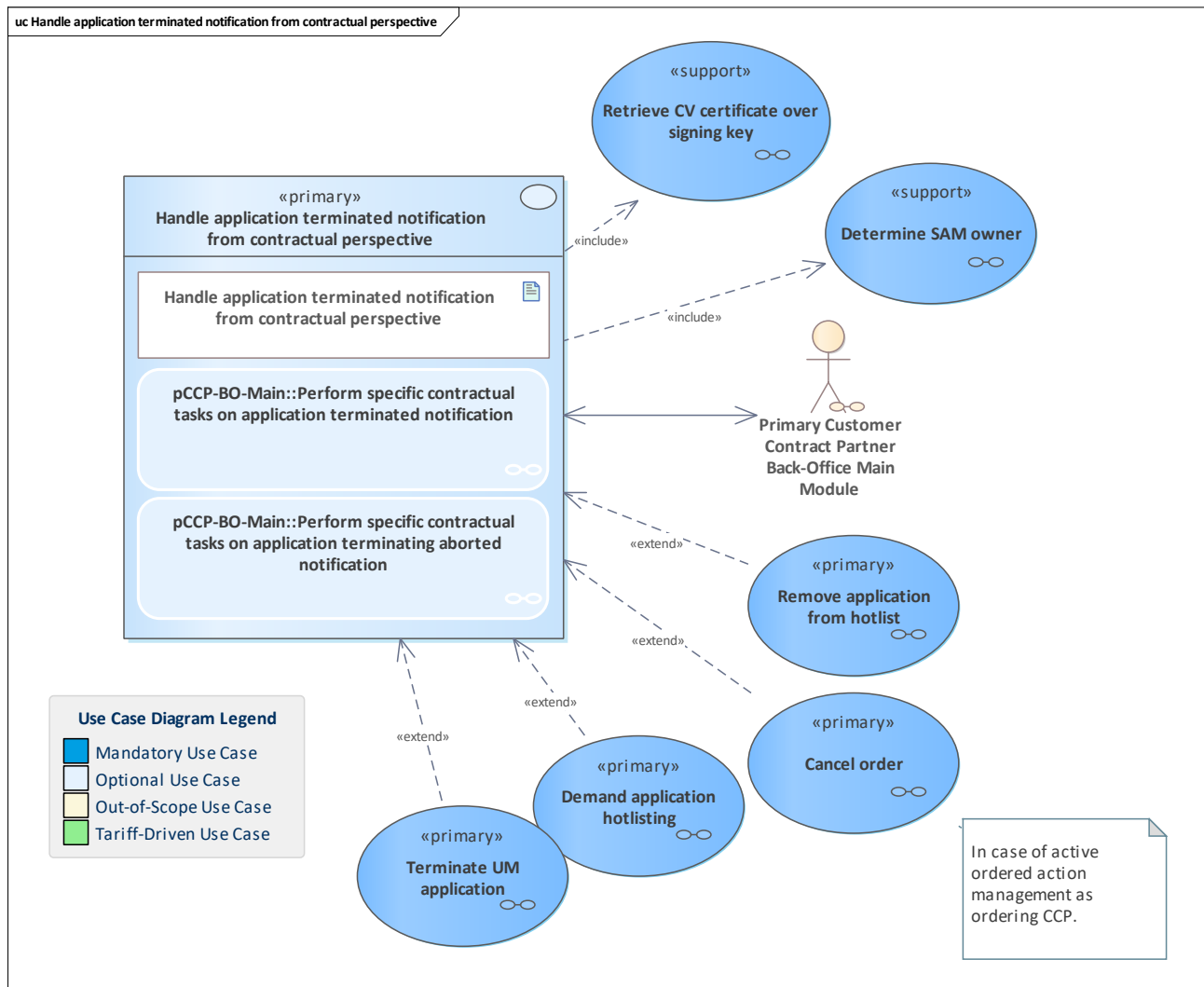


Figure 292: Handle application terminated notification from contractual perspective

Handle an application terminated notification from a contractual perspective.
The pCCP back-office system does its contractual checks and monitoring.
If the transaction of the termination was aborted, this might cause a clearing case.

11.105 Handle application terminated notification from operational perspective

11.106 Handle application terminated notification from operational perspective

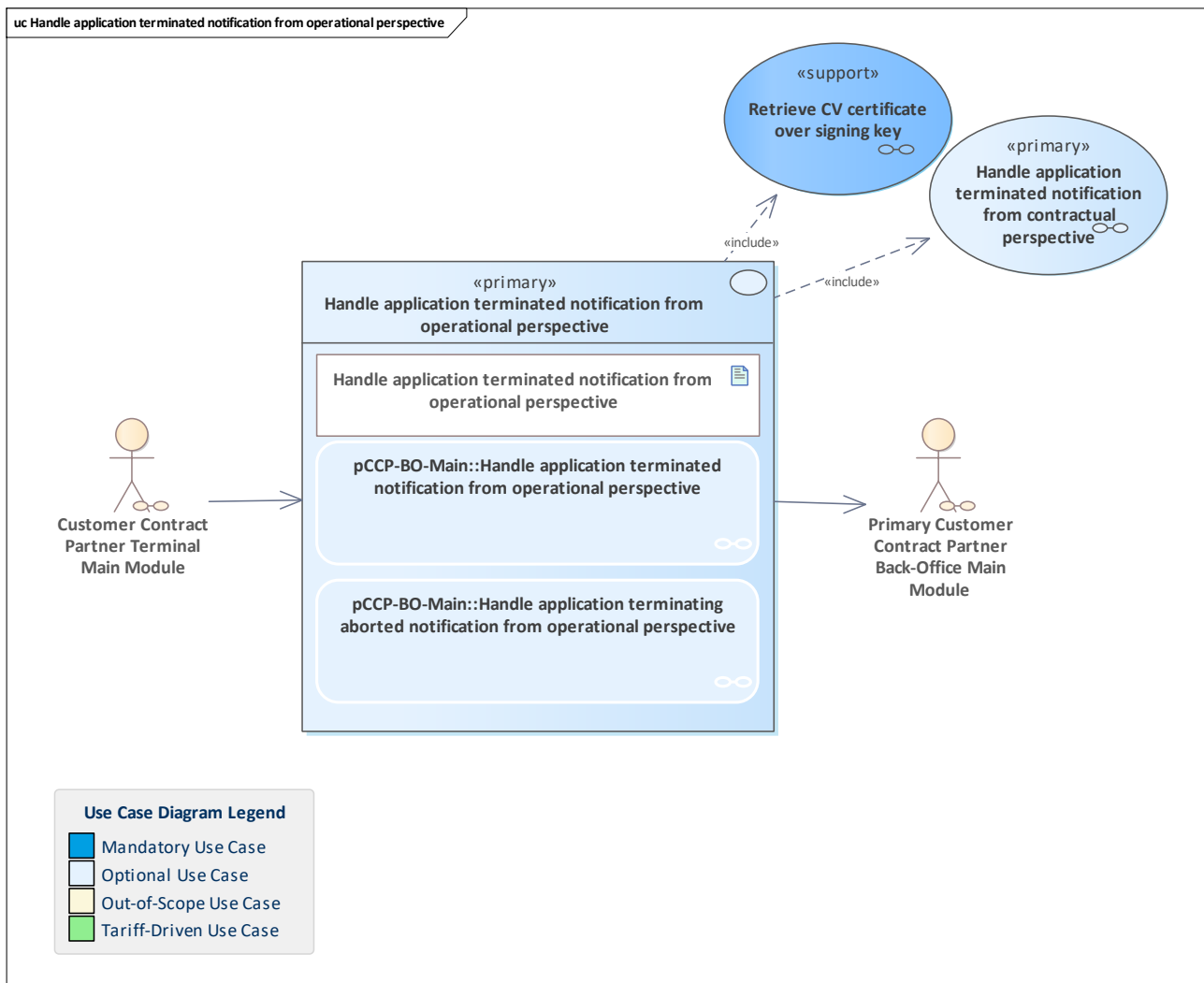


Figure 293: Handle application terminated notification from operational perspective

Handle an application termination from an operational perspective.

The notification is sent from the CCP terminal to the pCCP back-office system. The pCCP does its operational checks and monitoring and triggers the contractual handling.

If the termination transaction was aborted, the notification is registered and SAM counter values are stored for consistent monitoring.

11.107 Handle application unblocked notification from contractual perspective

11.108 Handle application unblocked notification from contractual perspective

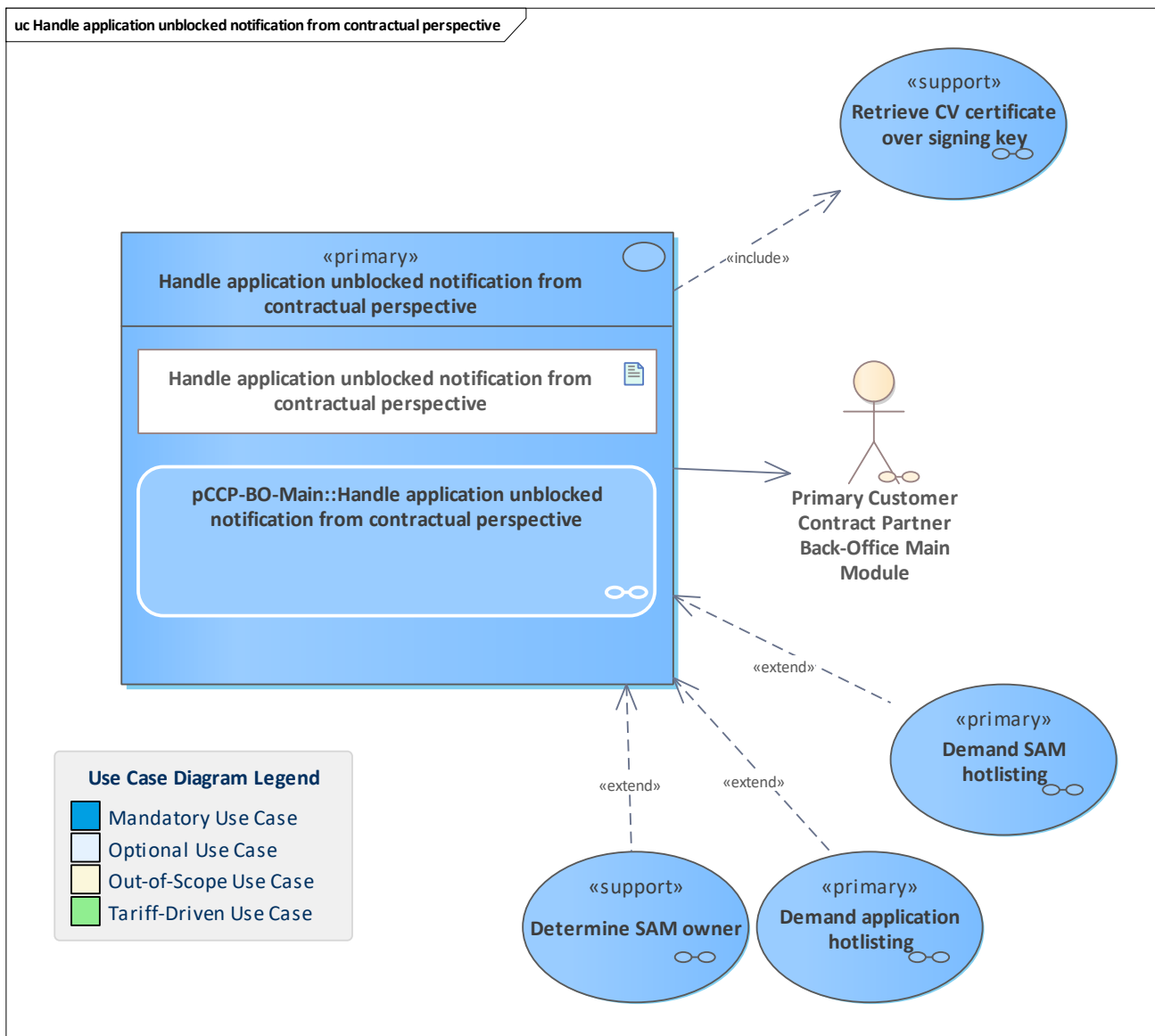


Figure 294: Handle application unblocked notification from contractual perspective

The pCCP checks the unblocked UM application notification from the contractual perspective.

11.109 Handle application unblocked notification from operational perspective

11.110 Handle application unblocked notification from operational perspective

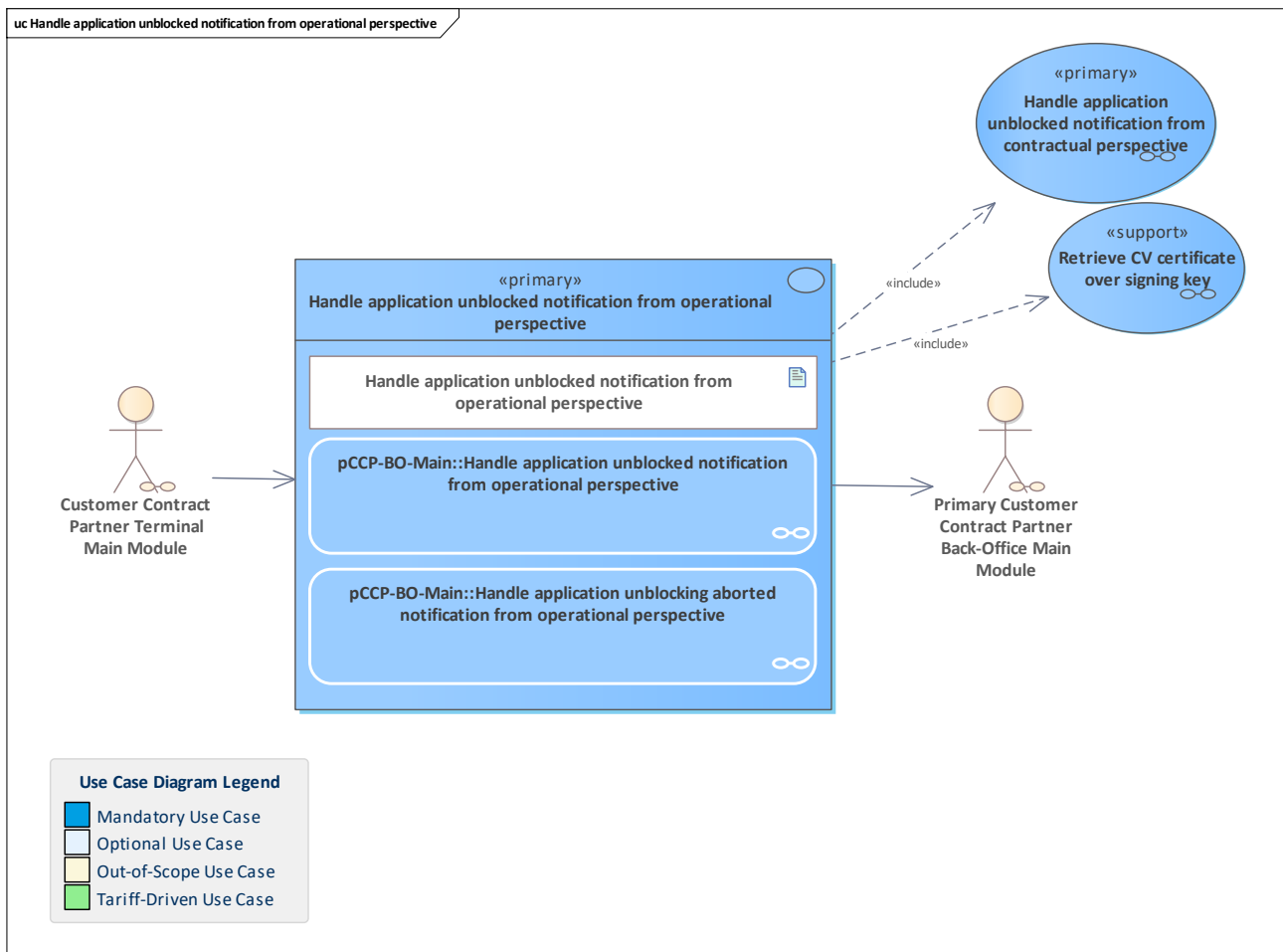


Figure 295: Handle application unblocked notification from operational perspective

A terminal has unblocked the application in an user medium with an application. In this use case, the pCCP receives the notification from its terminal and runs certain checks from operational perspective, such as the signature verification of the unblocking attestation. Then the contractual checks are done. As an alternative flow, a potential transaction abortion is handled.

11.111 Handle application XY notification from contractual perspective

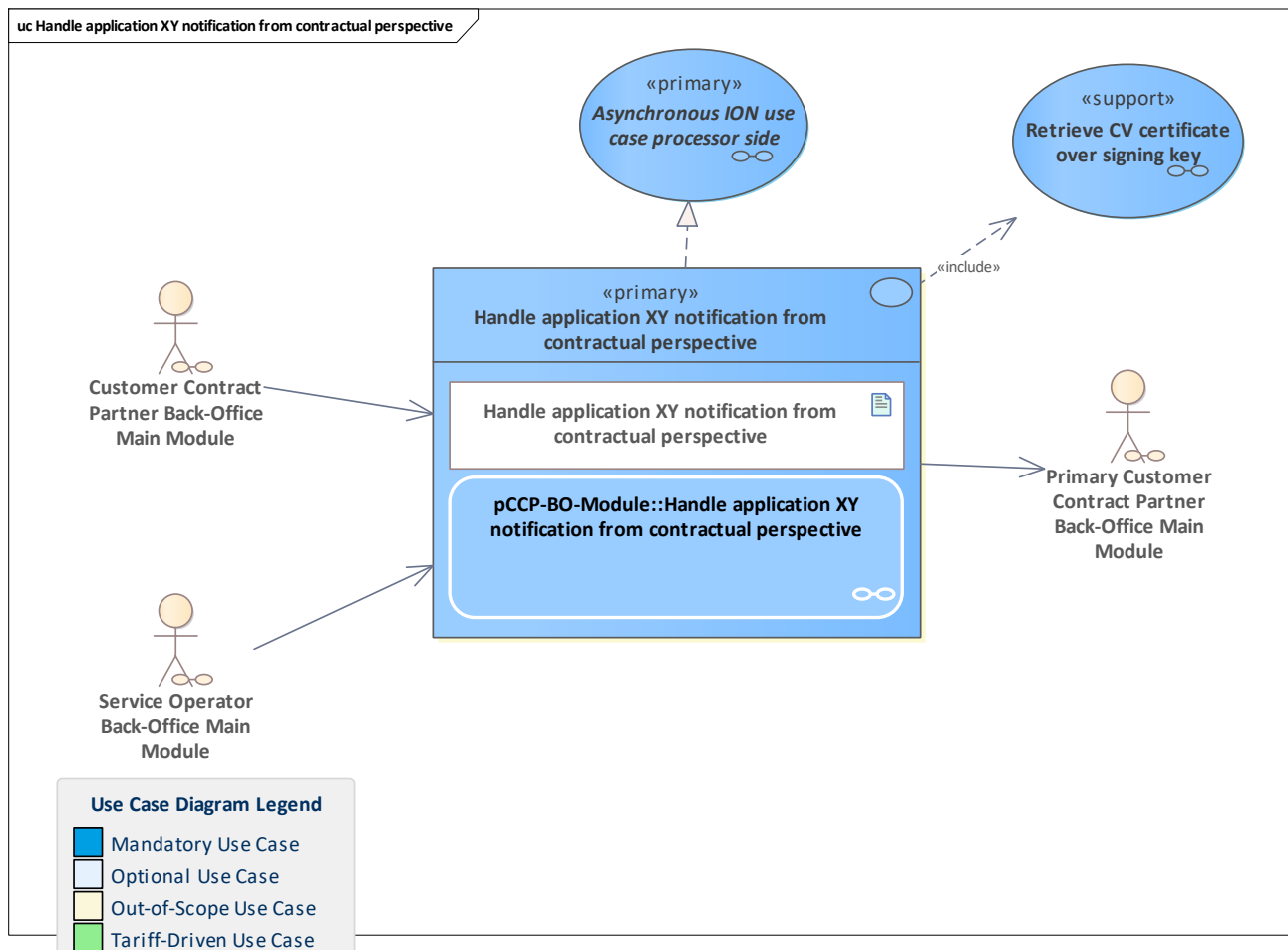


Figure 296: Handle application XY notification from contractual perspective

A CCP system processes a notification about an action executed on an owned application. The processing is done from the contractual perspective focusing on the application lifecycle aspects.

This use case has two entry points:

- [pCCP-BO-Module::Handle application XY notification from contractual perspective](#)
This entry point is used in the non-owned scenarios
- [pCCP-BO-Module::Perform specific contractual tasks on application XY notification](#)
This entry point is used in the owned scenarios

The first includes the latter, in which most of the processing is done.

11.112 Handle application XY notification from operational perspective

11.113 Handle application XY notification from operational perspective

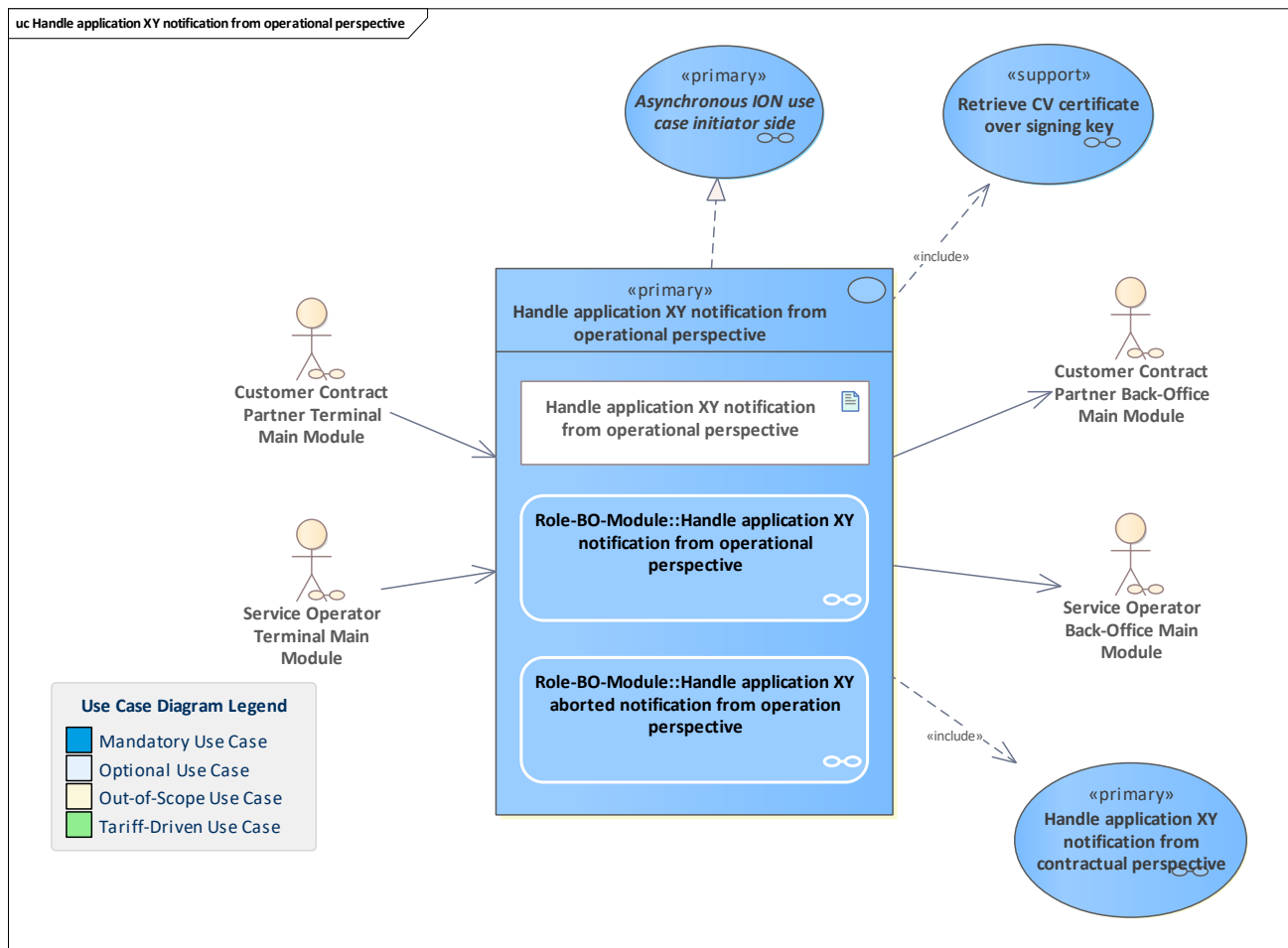


Figure 297: Handle application XY notification from operational perspective

A back-office system belonging to a terminal that executed a UM action with an application processes the notification about that action execution. The processing is done from the operational perspective, focusing on the terminal-side of the action execution, i.e. logging the used SAM counter values. Depending on the use case and the ownership of the application the action was executed on, other back-office systems are informed about the action execution.

11.114 Handle authentication key hotlisting demand

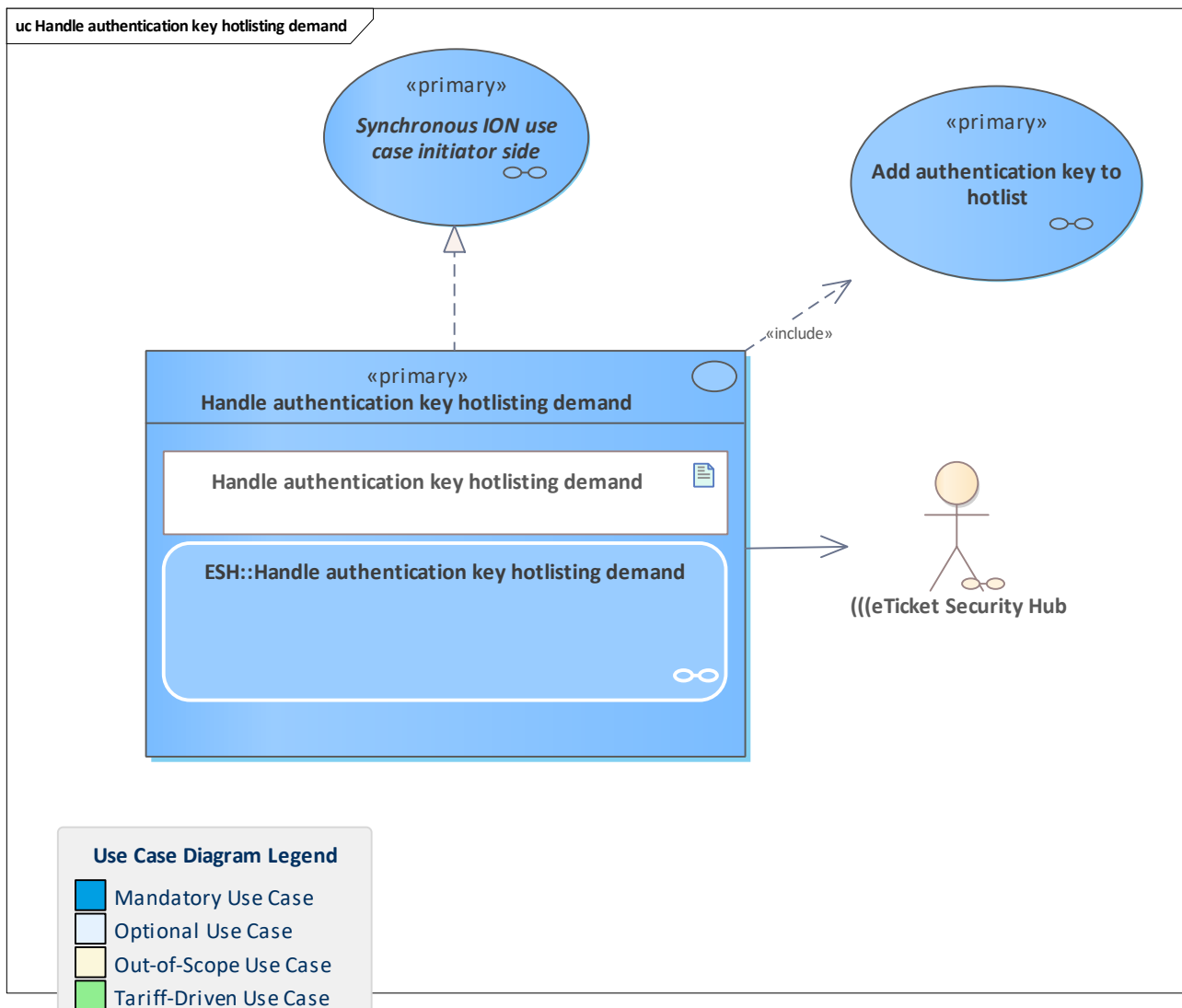


Figure 298: Handle authentication key hotlisting demand

Use case to add the current symmetric authentication key (for masterkey-based sessions between SAM and user medium).

Either the demand for hotlisting authentication key has been received via the service management or the key usage period has expired. If the authentication key must be hotlisted, then the use case can be started.

The scheme manager's ESH creates the request for adding the authentication key to the hotlist and sends it to the hotlist service system. The result will be updated in the ESH.

Please note that authentication key hotlisting may have a huge impact on the (((etiCORE environment. Each terminal has to consider the entry in the hotlist and must get the SAM and user medium to use the next generation of the authentication key to establish the session.

11.115 Handle check-in notification from contractual perspective

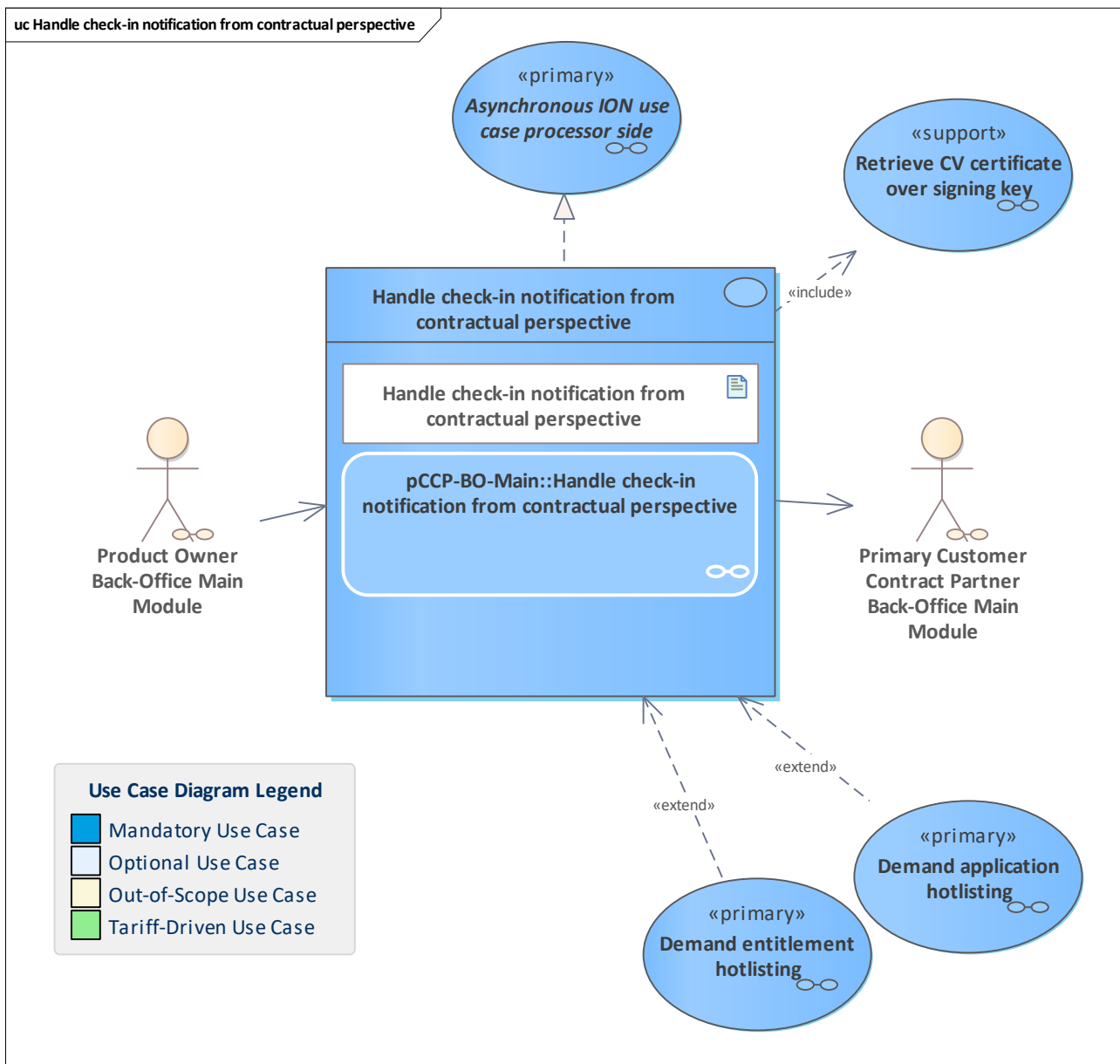


Figure 299: Handle check-in notification from contractual perspective

Handle a check-in operation from the contractual perspective.
The pCCP receives the notification from the PO system, registers it and does its contractual monitoring checks.
For a potential later billing, this notification will be referenced by the PO system when sending the collected and rated recording events.

11.116 Handle check-in notification from operational perspective

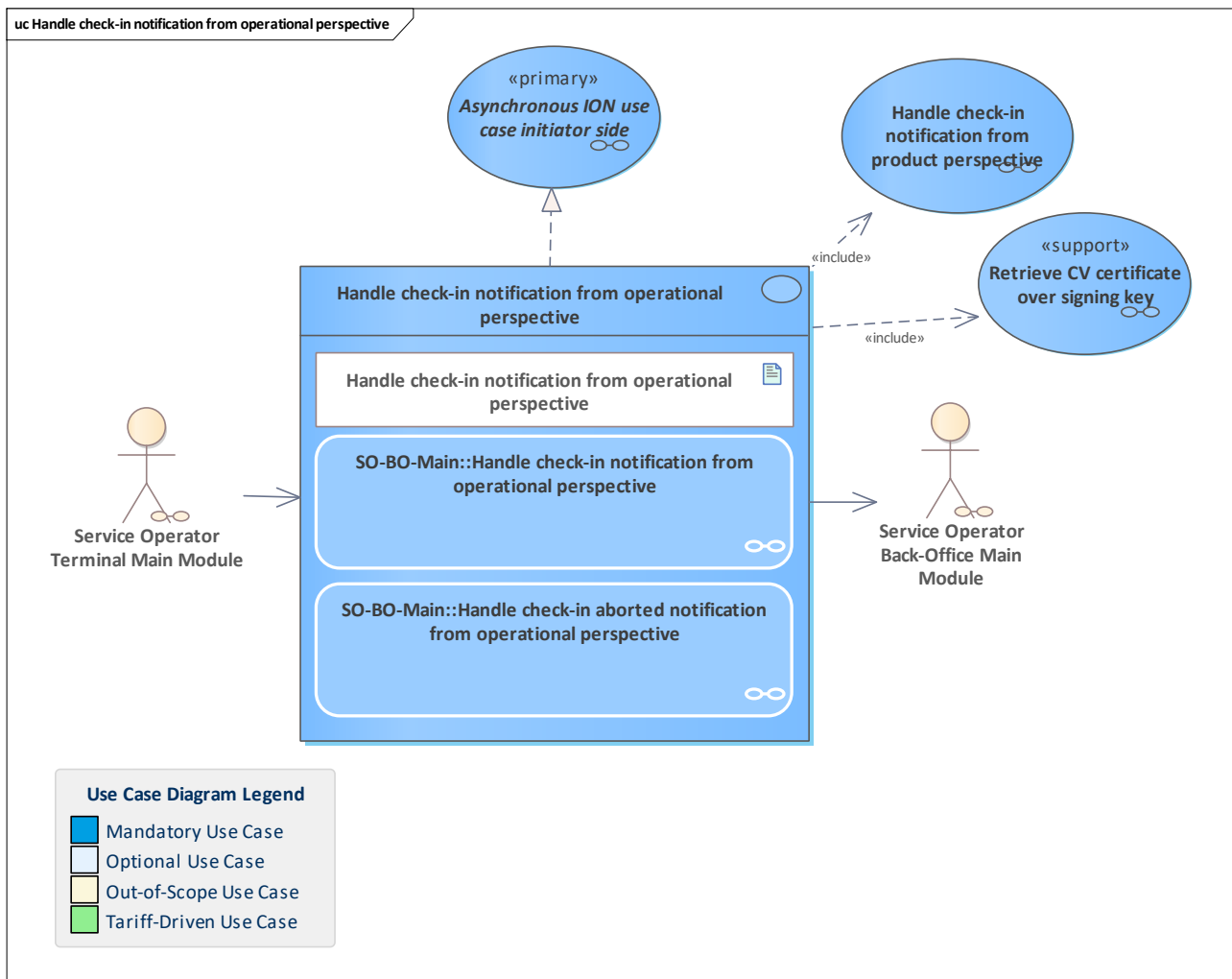


Figure 300: Handle check-in notification from operational perspective

Handle an check-in operation from the operational perspective.

The check-in notification is sent by the CCP terminal to the CCP back-office system. The notification will be checked and monitored, e.g. by verifying the signature of the embedded attestation.

Finally, the notification is forwarded to the PO system. This can be done either directly with a single message or in a scheduled process that sends a list of notifications.

In the case of action abortion, terminal and SAM action data is sent by the terminal to the back-office system. The CCP back-office system registers the abortion for internal monitoring and data consistency. Since no counters are affected which are important for the PO, in case of an abortion, the notification is not forwarded.

11.117 Handle check-in notification from product perspective

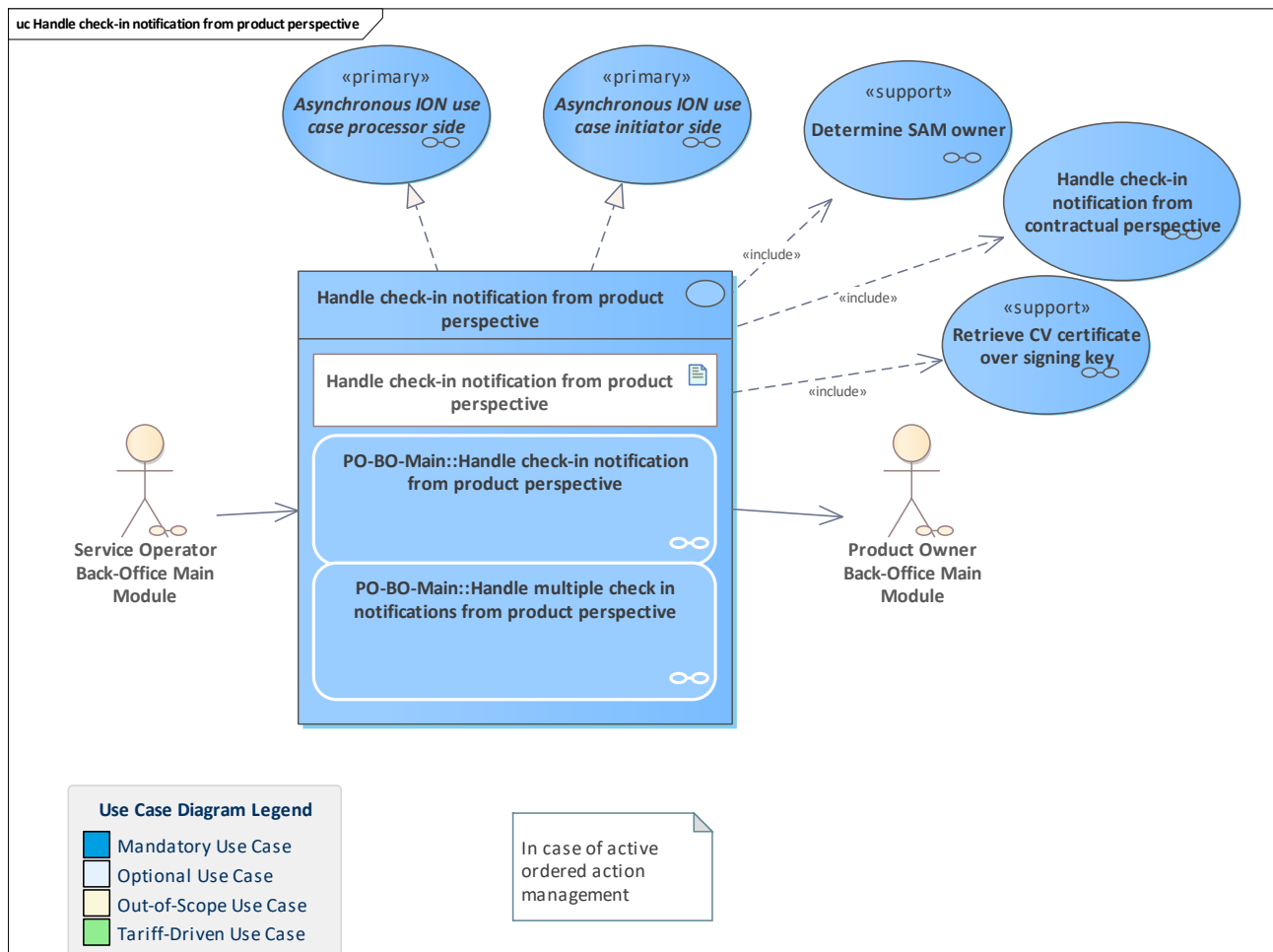


Figure 301: Handle check-in notification from product perspective

Handle a check-in operation from the product perspective.

This use case describes the processing of the check-in notification in the PO back-office system. The SO sends the notification, the PO registers it and does its monitoring checks. After these checks, the notification is forwarded to the CCP back-office system.

Furthermore, check-in notifications can be collected in the SO system and then sent as a list in a scheduled process to the PO.

11.118 Handle check-out notification from contractual perspective

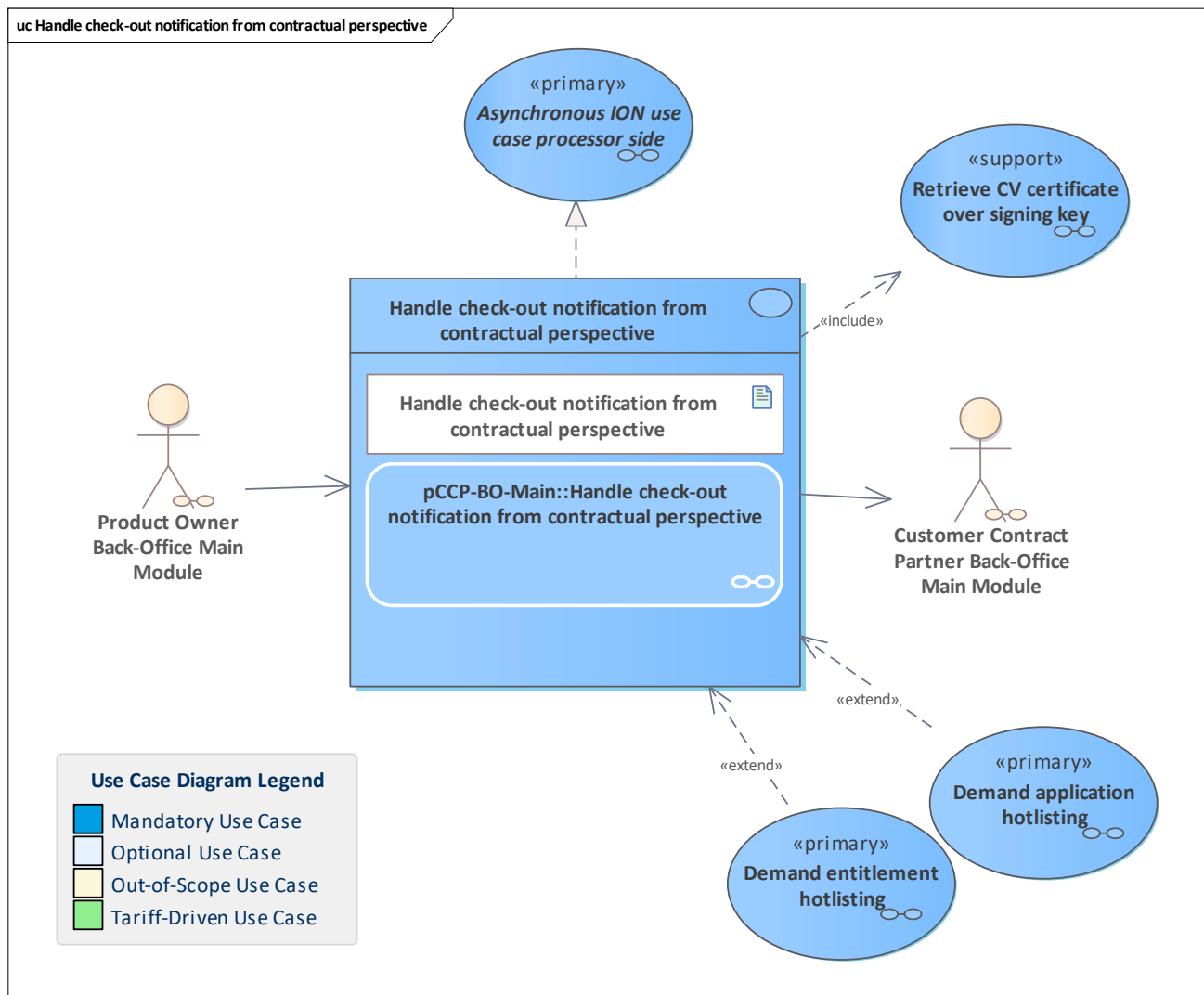


Figure 302: Handle check-out notification from contractual perspective

Handle check-out operation from the contractual perspective.

The pCCP receives the notification from the PO system, registers it and does its contractual monitoring checks.

For a potential later billing, this notification will be referenced by the PO system when sending the collected and rated recording events.

11.119 Handle check-out notification from operational perspective

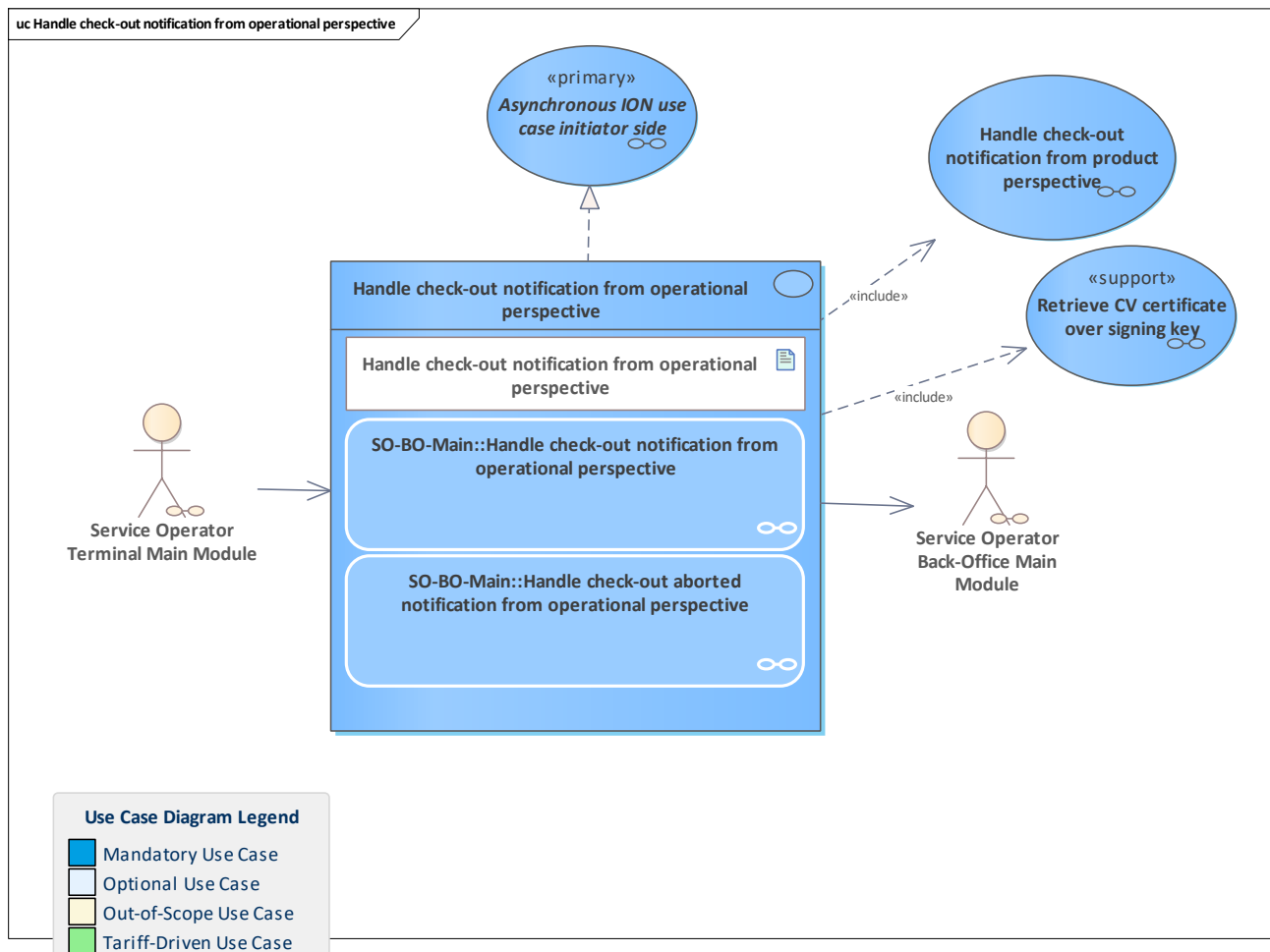


Figure 303: Handle check-out notification from operational perspective

Handle an check-out operation from the operational perspective.

The check-out notification is sent by the CCP terminal to the CCP back-office system. The notification will be checked and monitored, e.g. by verifying the signature of the embedded attestation.

Finally, the notification is forwarded to the PO system. This can be done either directly with a single message or in a scheduled process that sends a list of notifications.

In the case of action abortion, terminal and SAM action data is sent by the terminal to the back-office system. The CCP back-office system registers the abortion for internal monitoring and data consistency. Since no counters are affected which are important for the PO, in case of an abortion, the notification is not forwarded.

11.120 Handle check-out notification from product perspective

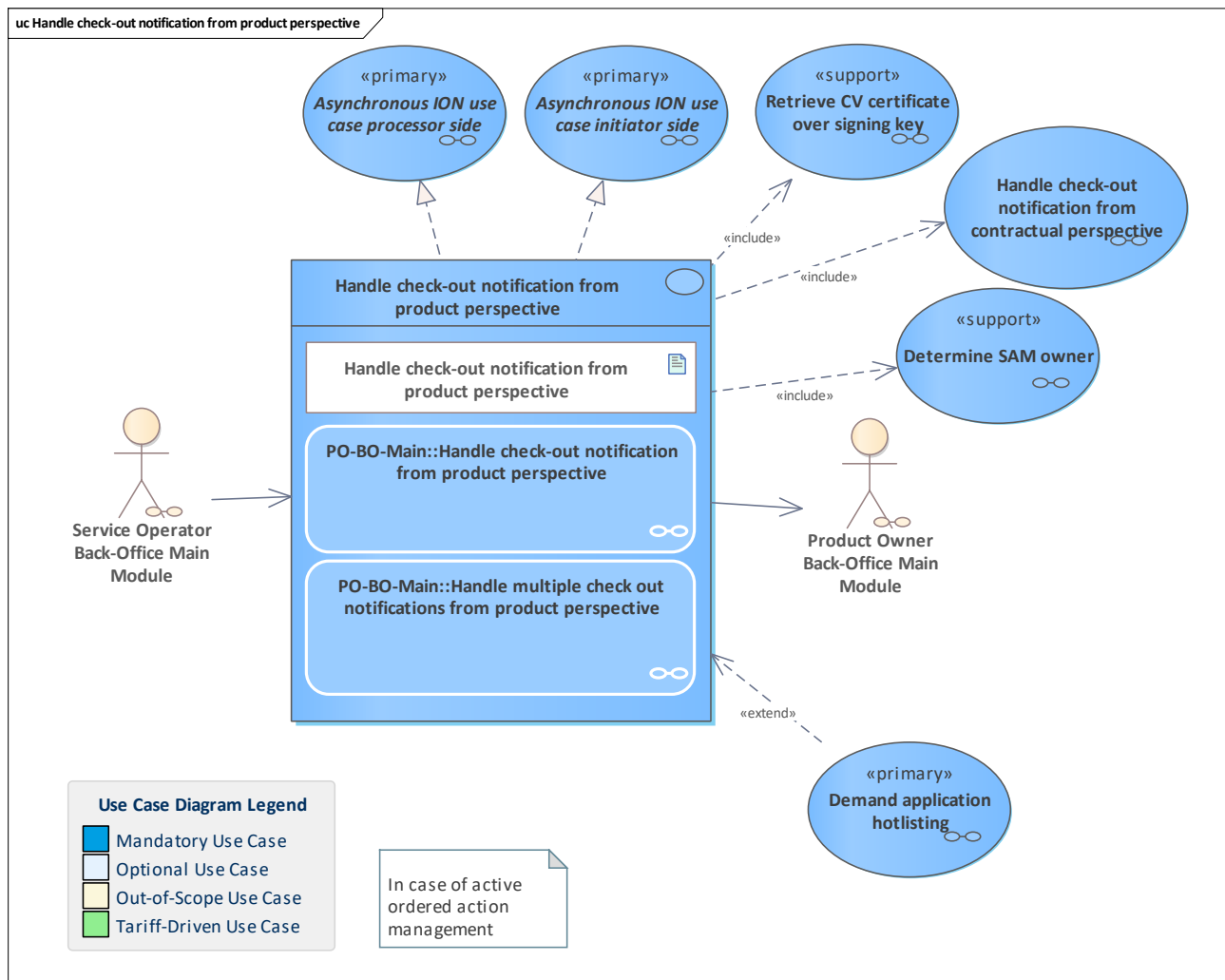


Figure 304: Handle check-out notification from product perspective

Handle a check-out operation from the product perspective.

This use case describes the processing of the check-out notification in the PO back-office system.

The SO sends the notification, the PO registers it and does its monitoring checks. After these checks, the notification is forwarded to the CCP back-office system.

Furthermore, check-out notifications can be collected in the SO system and then sent as a list in a scheduled process to the PO.

11.121 Handle defective user medium with application

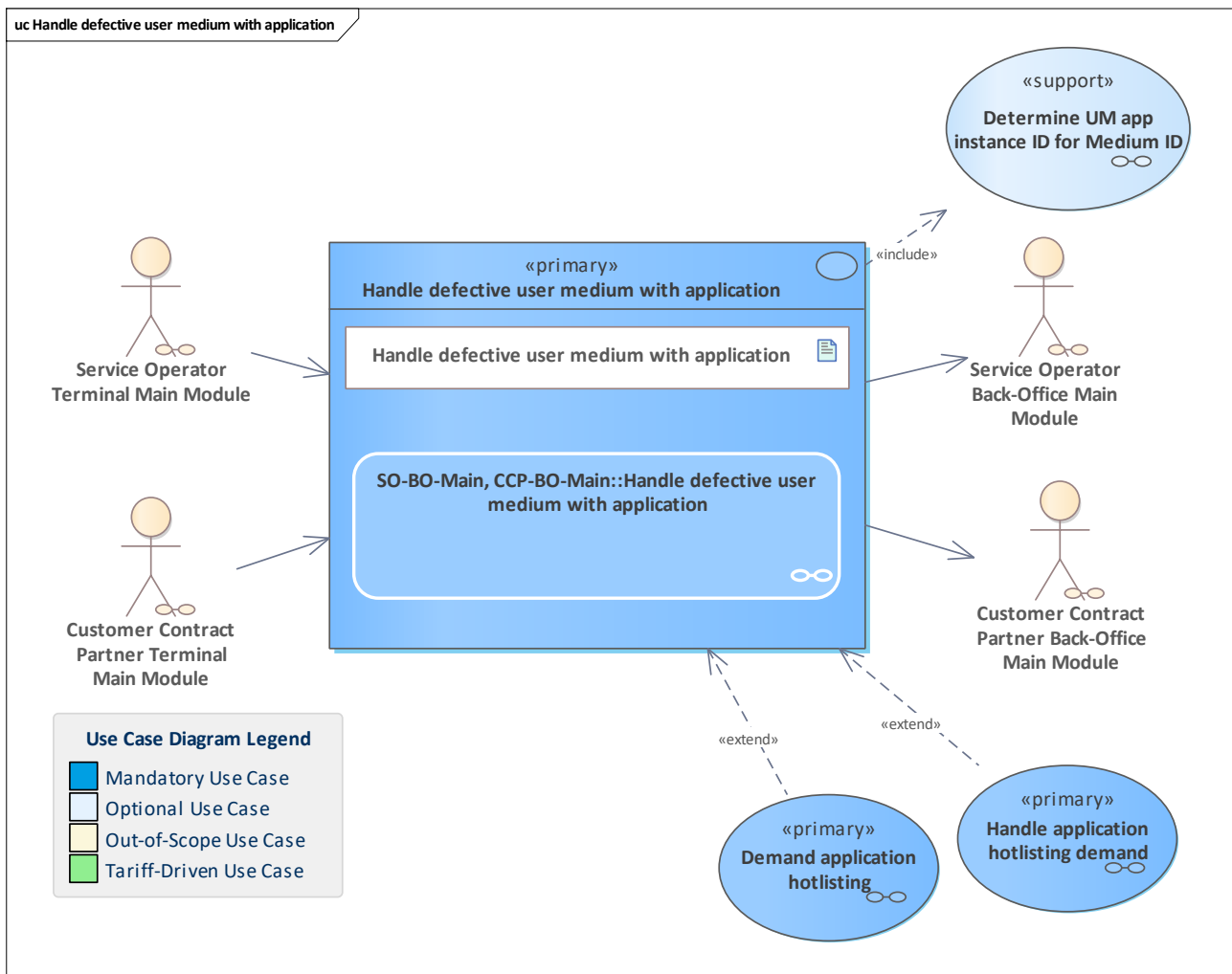


Figure 305: Handle defective user medium with application

The terminal operator (CCP or SO) back-office system receives the log data of the terminal which detected the defective user medium.

The message also contains the medium ID which can be the application instance ID. If the medium ID is proprietary, the application instance ID must be determined for the received medium ID.

The application of the application instance ID must be hotlisted. Depending on the role in the current process (pCCP versus sCCP/SO) either a hotlist demand is sent to the application instance owner, or the pCCP communicates directly with the hotlist service.

11.122 Handle discarded messages information

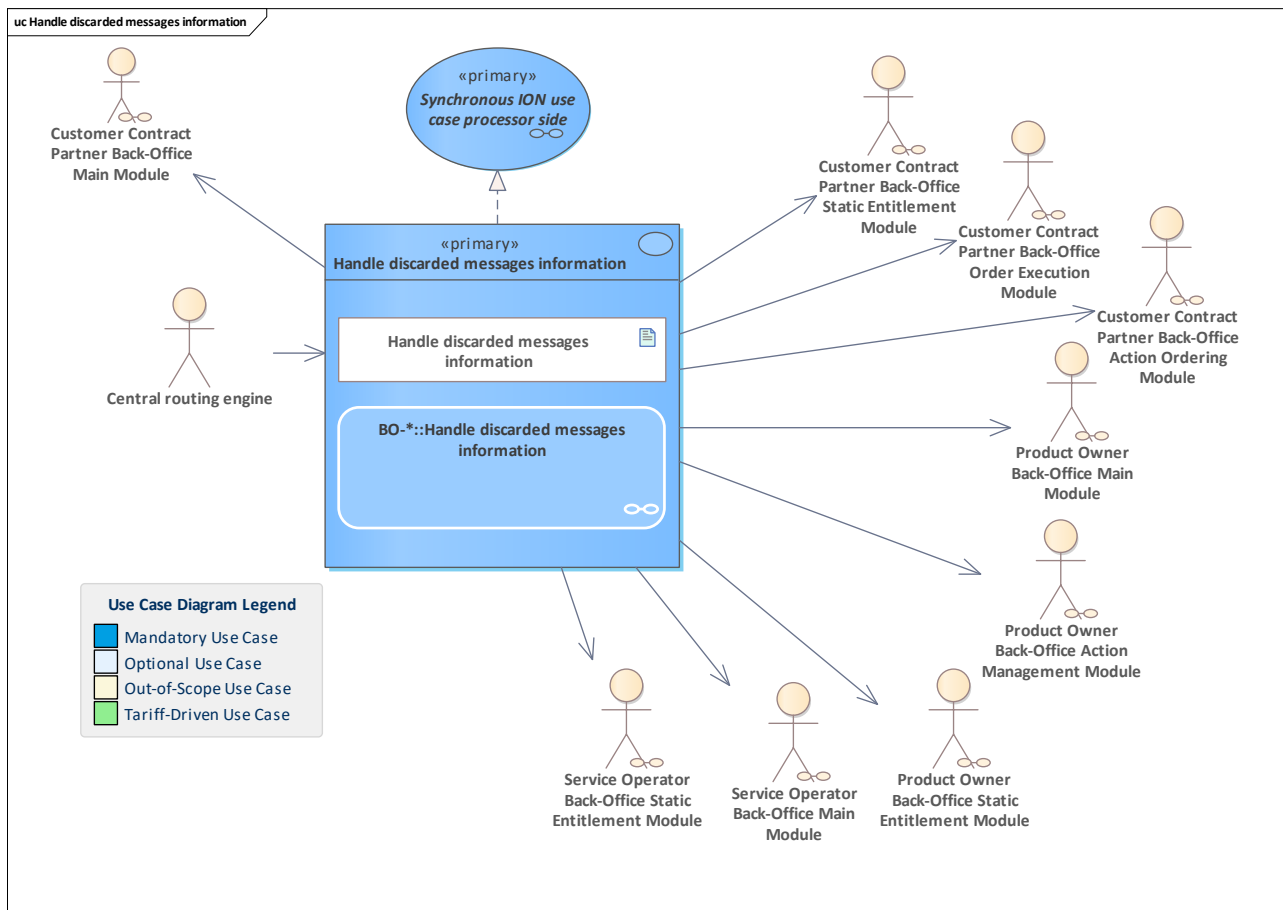


Figure 306: Handle discarded messages information

Process information about discarded asynchronous messages which were provided for store & forward but could not be delivered to the specified recipient.

The CRE will send message information to a

- [Customer Contract Partner System](#)
- [Ordering Customer Contract Partner System](#)
- [Executing Customer Contract Partner System](#)
- [Customer Contract Partner STE System](#)
- [Service Operator System](#)
- [Service Operator STE System](#)
- [Product Owner System](#)
- [Product Owner Action Management System](#)
- [Product Owner STE System](#)

These systems and modules work with asynchronous messages which can possibly be discarded.

11.123 Handle entitlement blocked notification from contractual perspective

11.124 Handle entitlement blocked notification from contractual perspective

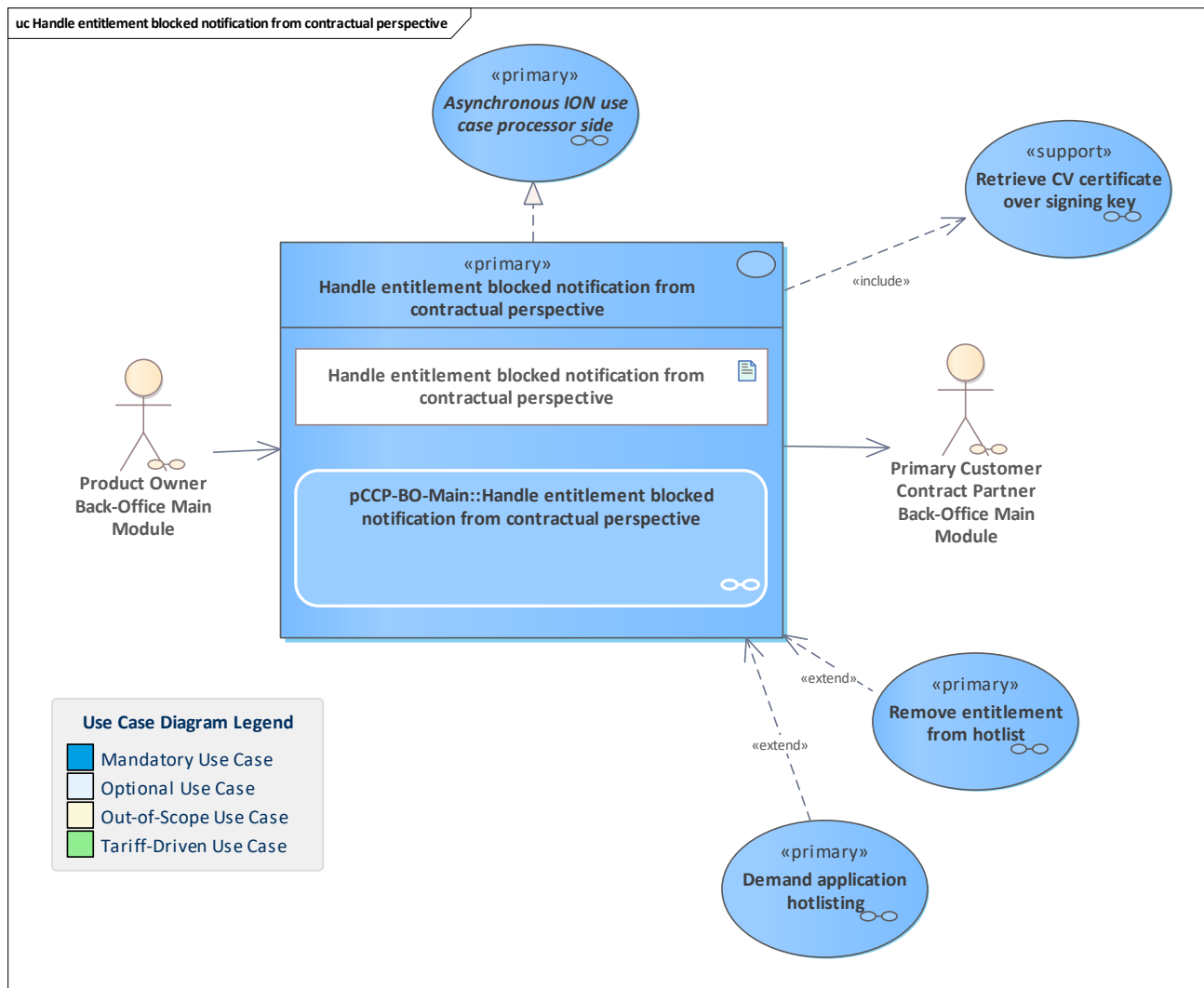


Figure 307: Handle entitlement blocked notification from contractual perspective

The entitlement blocked notification is received by the pCCP. The pCCP does its checks and monitoring from the contractual perspective regarding the correct execution of blocking. In this context, the signature of the blocking attestation is verified. If the blocking was correct, the pCCP demands that the entitlement is removed from the hotlist.

Note: if the pCCP itself performed the blocking, this use case takes inside the use case [Handle entitlement blocked notification from operational perspective](#) and is not called by the PO. In this case, this use case is not an [Asynchronous ION use case processor side](#).

Note: due to monitoring checks, the application may be demanded to be hotlisted. In this case, the pCCP will add the application directly to the hotlist.

11.125 Handle entitlement blocked notification from operational perspective

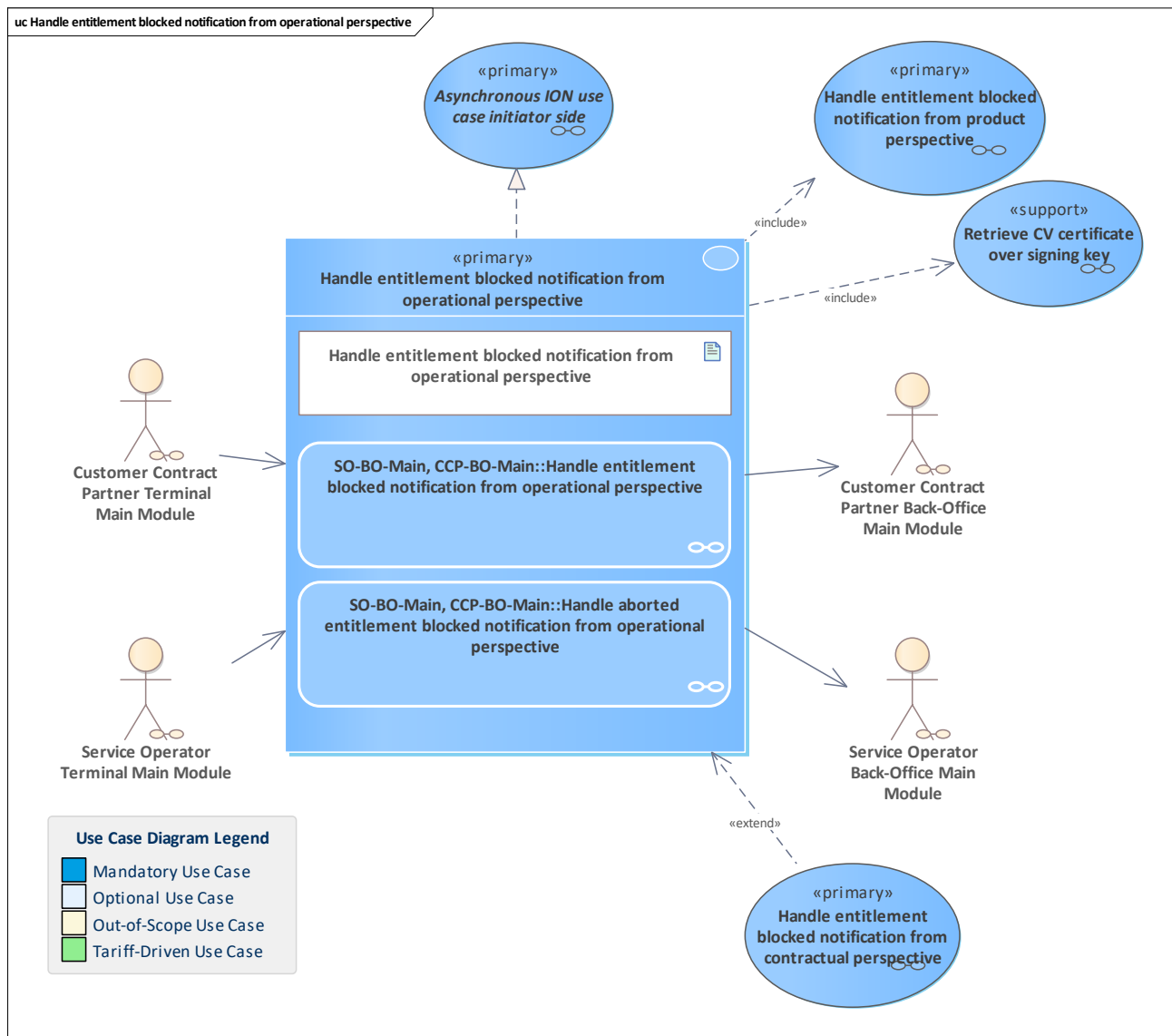


Figure 308: Handle entitlement blocked notification from operational perspective

The entitlement blocked notification is sent by the terminal to the SO or CCP back-office system. The notification will be checked.

If the pCCP blocked its own entitlement:

- the notification will be sent to the PO
- request the hotlist service system to remove the entitlement from the entitlement hotlist.

If the sCCP or SO blocked an entitlement:

- the notification will be sent to the PO (and the PO will forward it later to the pCCP)
- In case of action abortion, terminal and SAM action data is sent by the terminal to the back-office system. The SO or CCP back-office system registers the abortion for internal monitoring and data consistency.

11.126 Handle entitlement blocked notification from product perspective

11.127 Handle entitlement blocked notification from product perspective

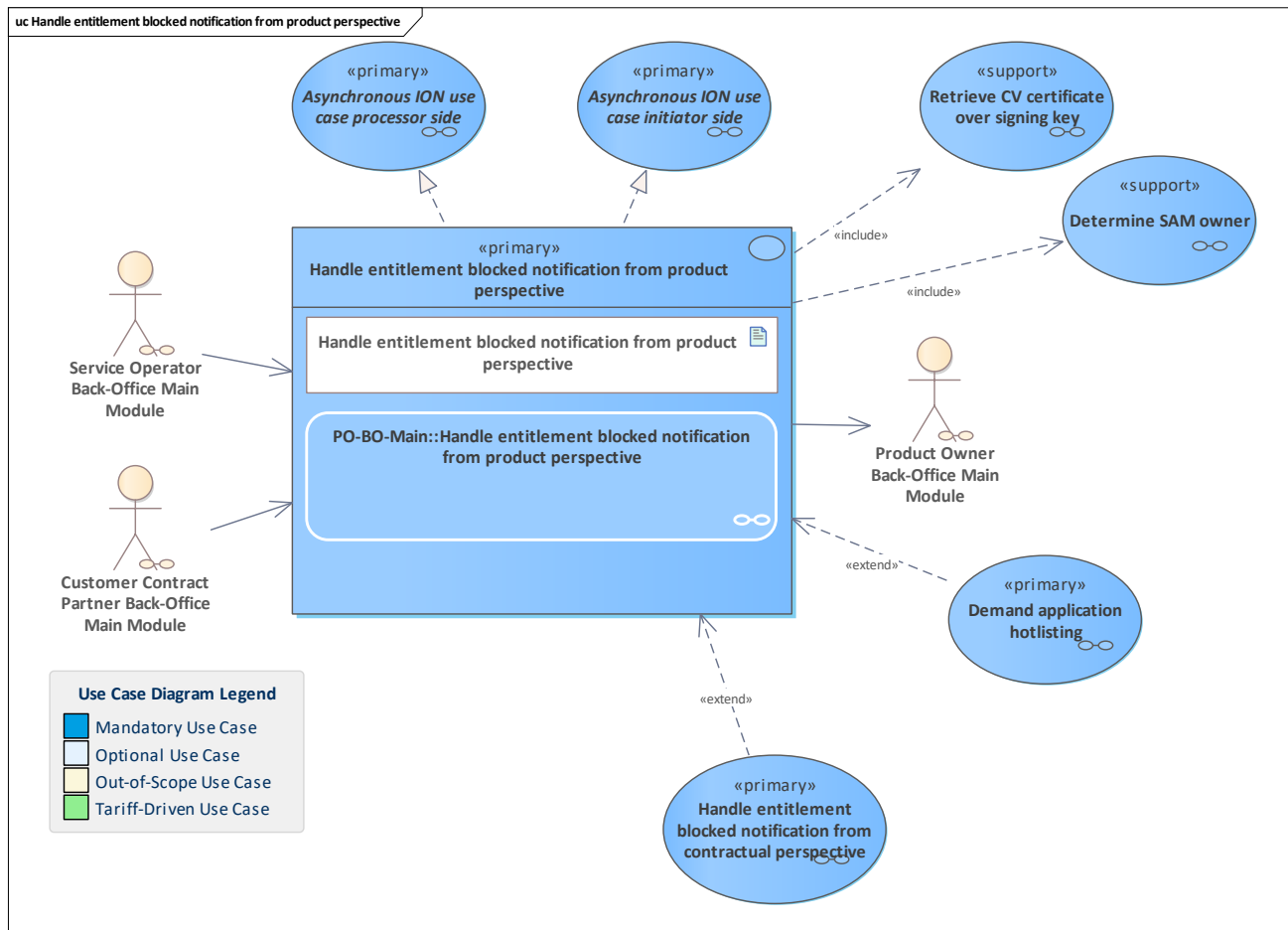


Figure 309: Handle entitlement blocked notification from product perspective

The entitlement blocked notification is received by the PO.

The PO registers the entitlement blocked notification and performs its checks and monitoring from the product perspective regarding the correct execution of blocking. In this context, the signature of the blocking attestation is verified and the SAM owner of the SAM that performed the blocking is determined.

- if the sender is a sCCP or SO, forward the notification to pCCP
- if the sender is the pCCP, do not forward the notification. In this case, the current use case is not an [Asynchronous ION use case initiator side](#).

11.128 Handle entitlement blocking order

11.129 Handle entitlement blocking order

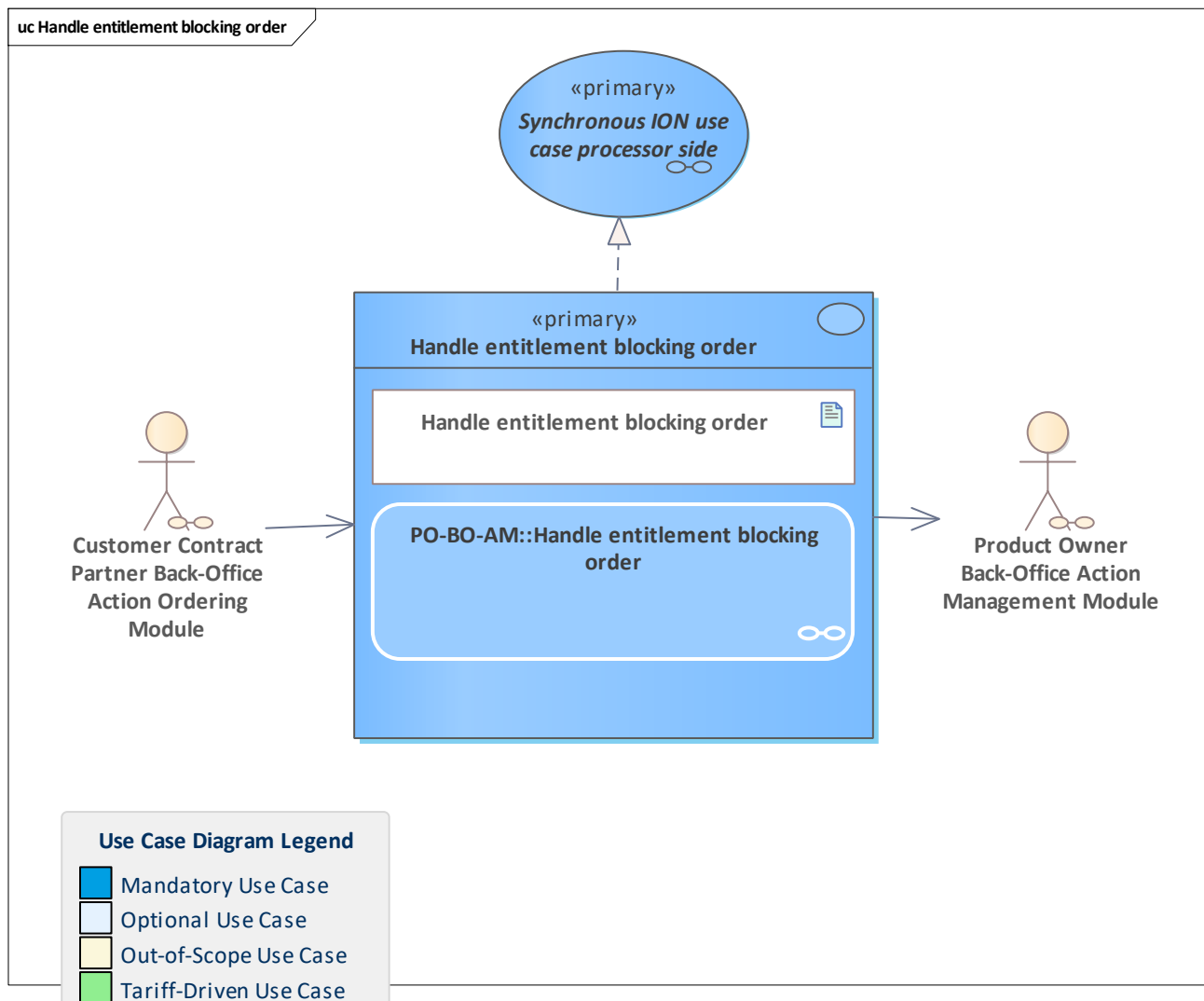


Figure 310: Handle entitlement blocking order

The PO back-office system with an action management extension handles an entitlement blocking order.

If the order passes all checks, it is added to the order inventory and may be considered for the next action list.

11.130 Handle entitlement hotlisting demand

11.131 Handle entitlement hotlisting demand

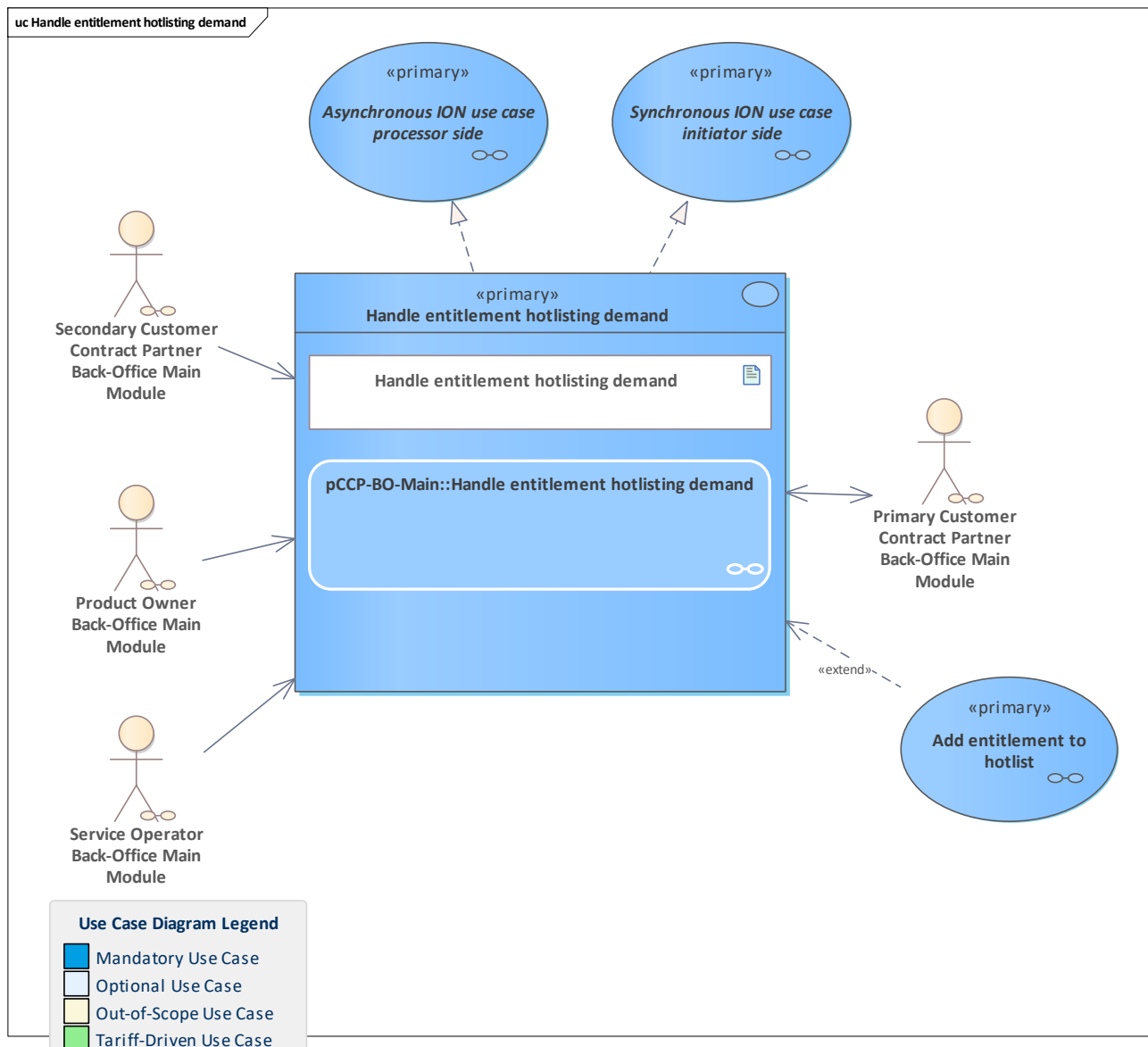


Figure 311: Handle entitlement hotlisting demand

The pCCP that issued the entitlement to a customer checks the hotlisting demand. If the result of the check requires a hotlisting, the hotlist service system will be called to add the entitlement to the hotlist. Otherwise, the demand will be rejected and there is no need to communicate with the hotlist service system. The original caller of the demand is informed. Please note that if there is more than one demand for hotlisting the same entitlement, this has to be considered in the check for a required hotlisting but will not result in an exception. Especially for the monitoring of a third-party system, it must be possible to demand hotlisting even if the same object had previously been demanded for hotlisting. Several hotlisting requests for an entitlement can be received. The final decision for a hotlisting must be decided in the pCCP, by considering all hotlisting demands, hotlisting demand revocations and hotlisting orders. If the entitlement is already hotlisted, the demand will not cause a further hotlist request towards the hotlist service system.

Note: in the ION context, the use case is asynchronous as processor (process the demand, [Asynchronous ION use case processor side](#)) and a synchronous use case as initiator

([Synchronous ION use case initiator side](#)) due to the request to the hotlist service for adding the entitlement to the hotlist.

11.132 Handle entitlement inspected notification from contractual perspective

11.133 Handle entitlement inspected notification from contractual perspective

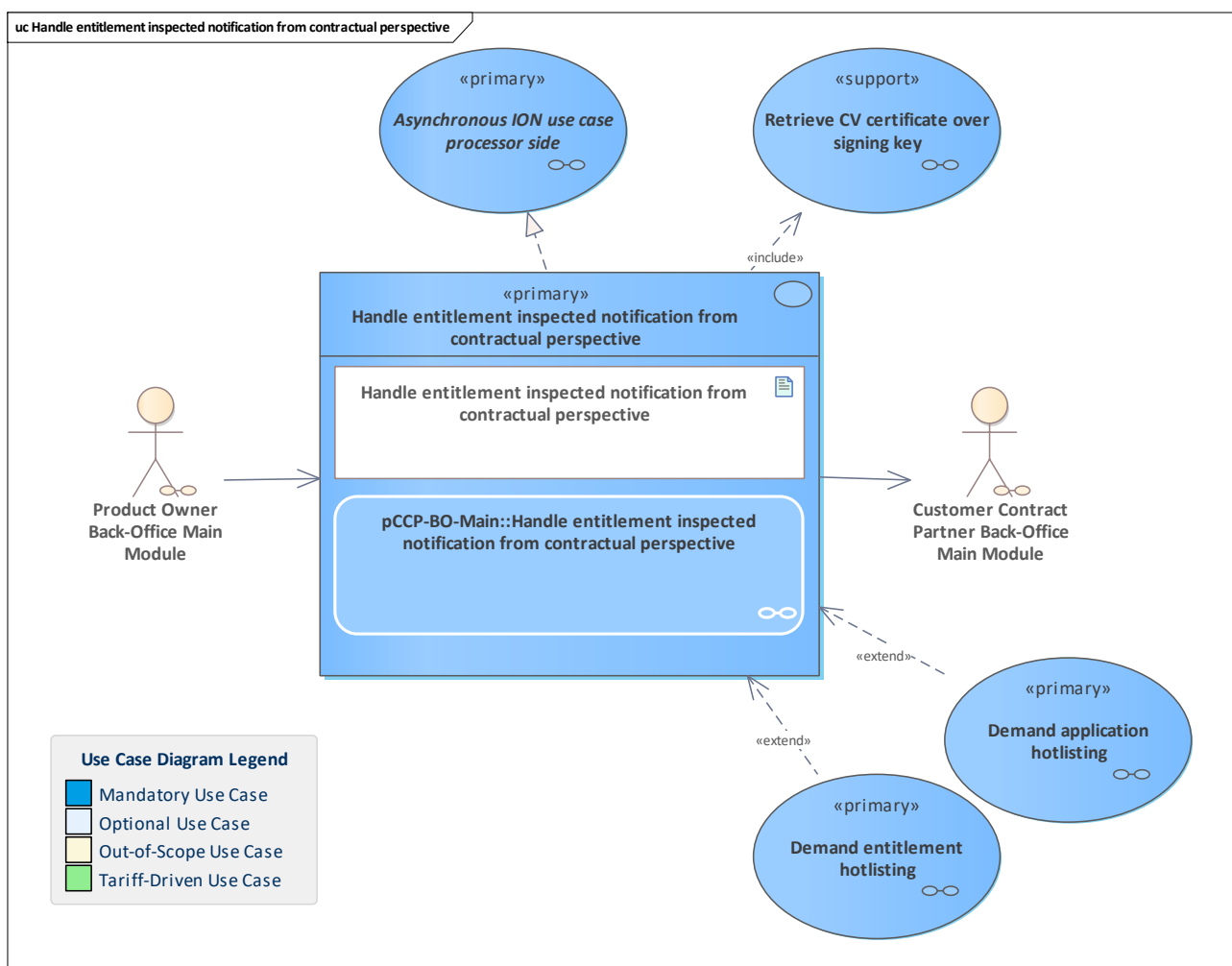


Figure 312: Handle entitlement inspected notification from contractual perspective

This use case describes the processing of the notification of an inspected entitlement in the back-office system of the pCCP.

The pCCP receives the notification from the PO system, registers it and does its contractual monitoring checks.

11.134 Handle entitlement inspected notification from operational perspective

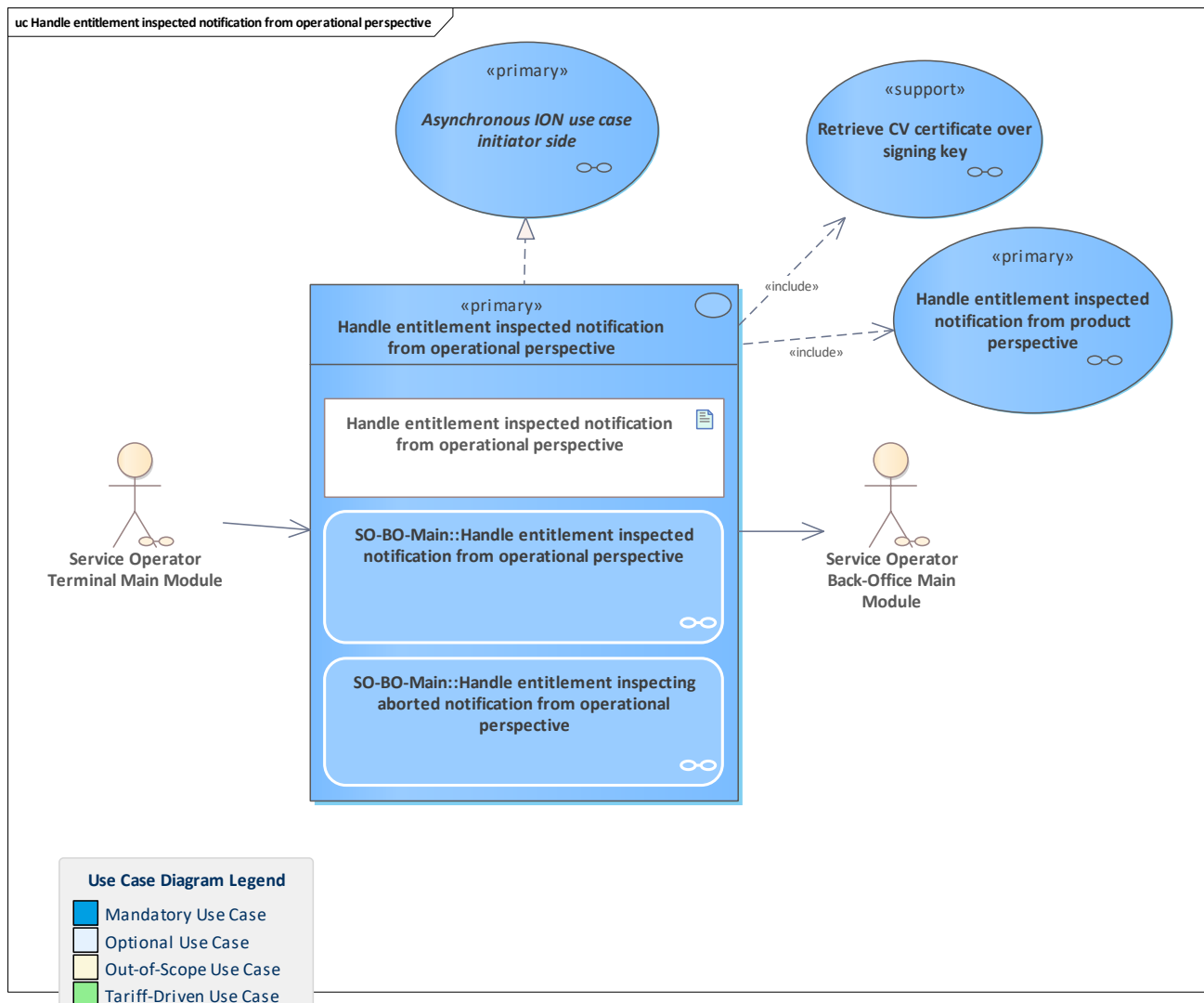


Figure 313: Handle entitlement inspected notification from operational perspective

This use case describes the processing of the inspection notification received from the SO terminal in the back-office system of the SO.

The notification will be checked and monitored, e.g. by verifying the signature of the embedded attestation.

Finally, the notification is forwarded to the PO system. This can be done either directly with a single message or in a scheduled process that sends a list of notifications.

11.135 Handle entitlement inspected notification from product perspective

11.136 Handle entitlement inspected notification from product perspective

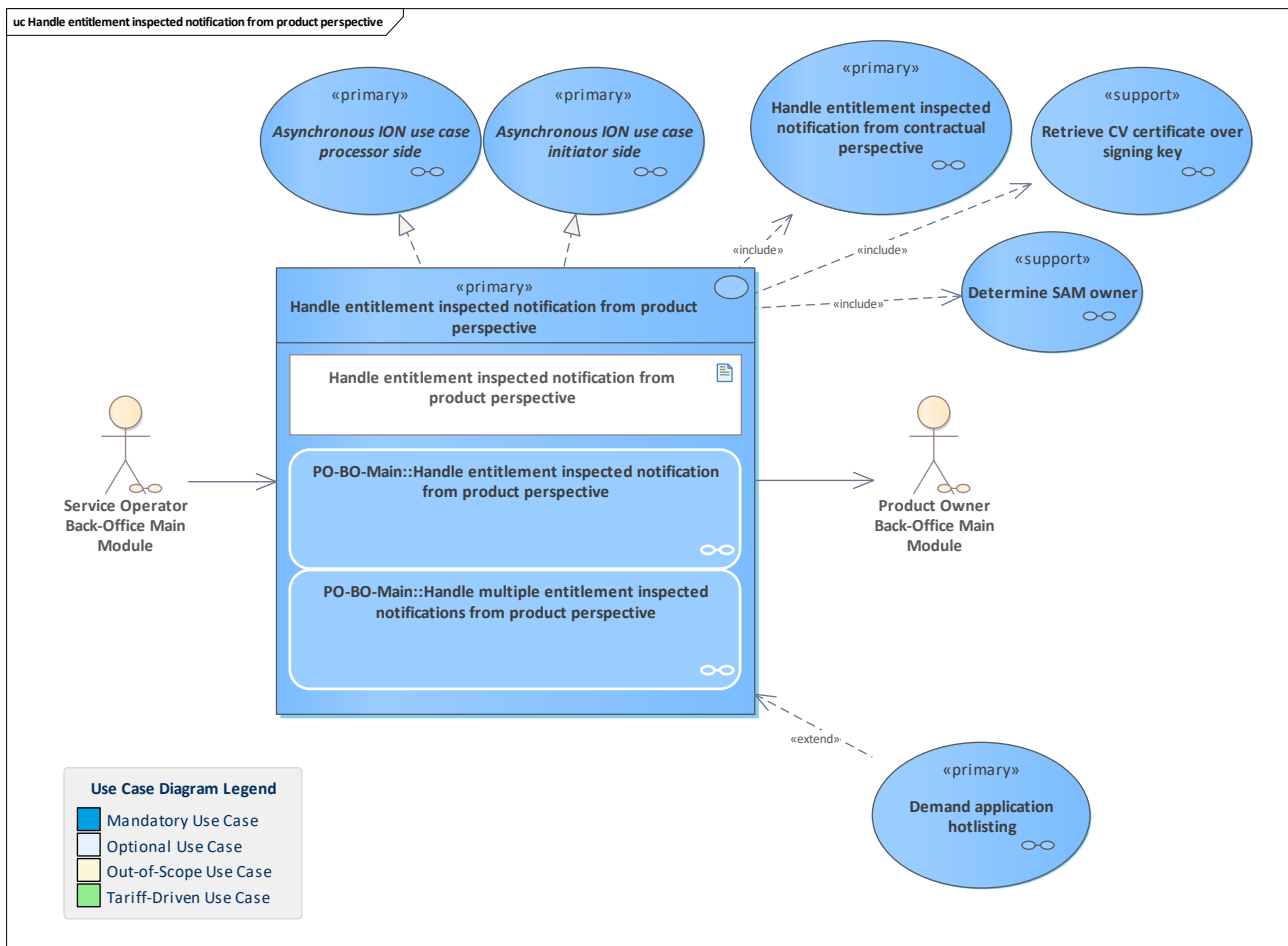


Figure 314: Handle entitlement inspected notification from product perspective

This use case describes the processing of the notification of an inspected entitlement in the PO back-office system.

The SO sends the notification, the PO registers it and does its monitoring checks. After these checks, the notification is forwarded to the CCP back-office system.

Furthermore, entitlement inspected notifications can be collected in the SO system and then sent as a list in a scheduled process to the PO.

11.137 Handle entitlement issuance order

11.138 Handle entitlement issuance order

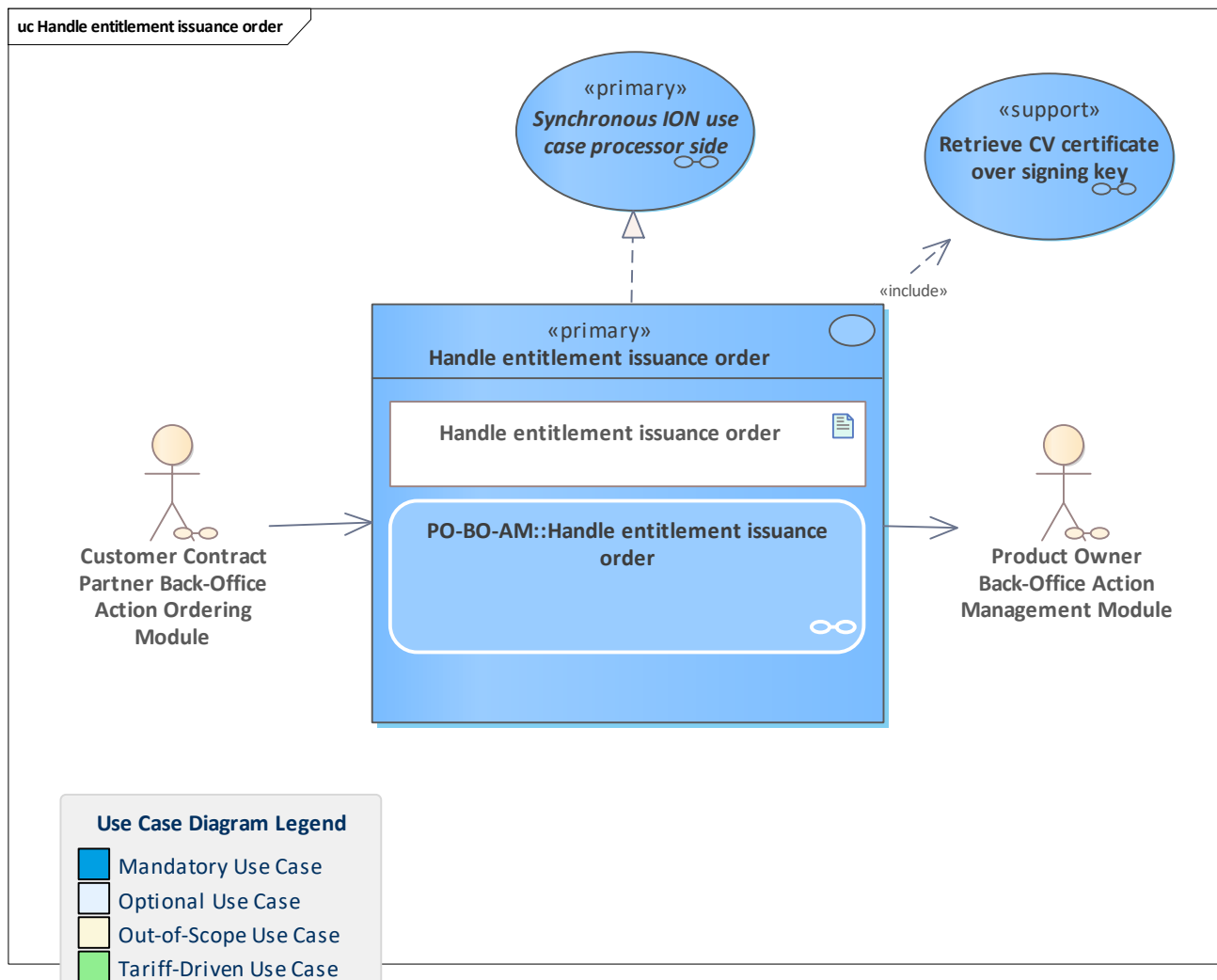


Figure 315: Handle entitlement issuance order

The PO back-office system with an action management extension handles an entitlement issuance order.

If the order passes all checks, it is added to the order inventory and may be considered for the next action list.

11.139 Handle entitlement issued notification from contractual perspective

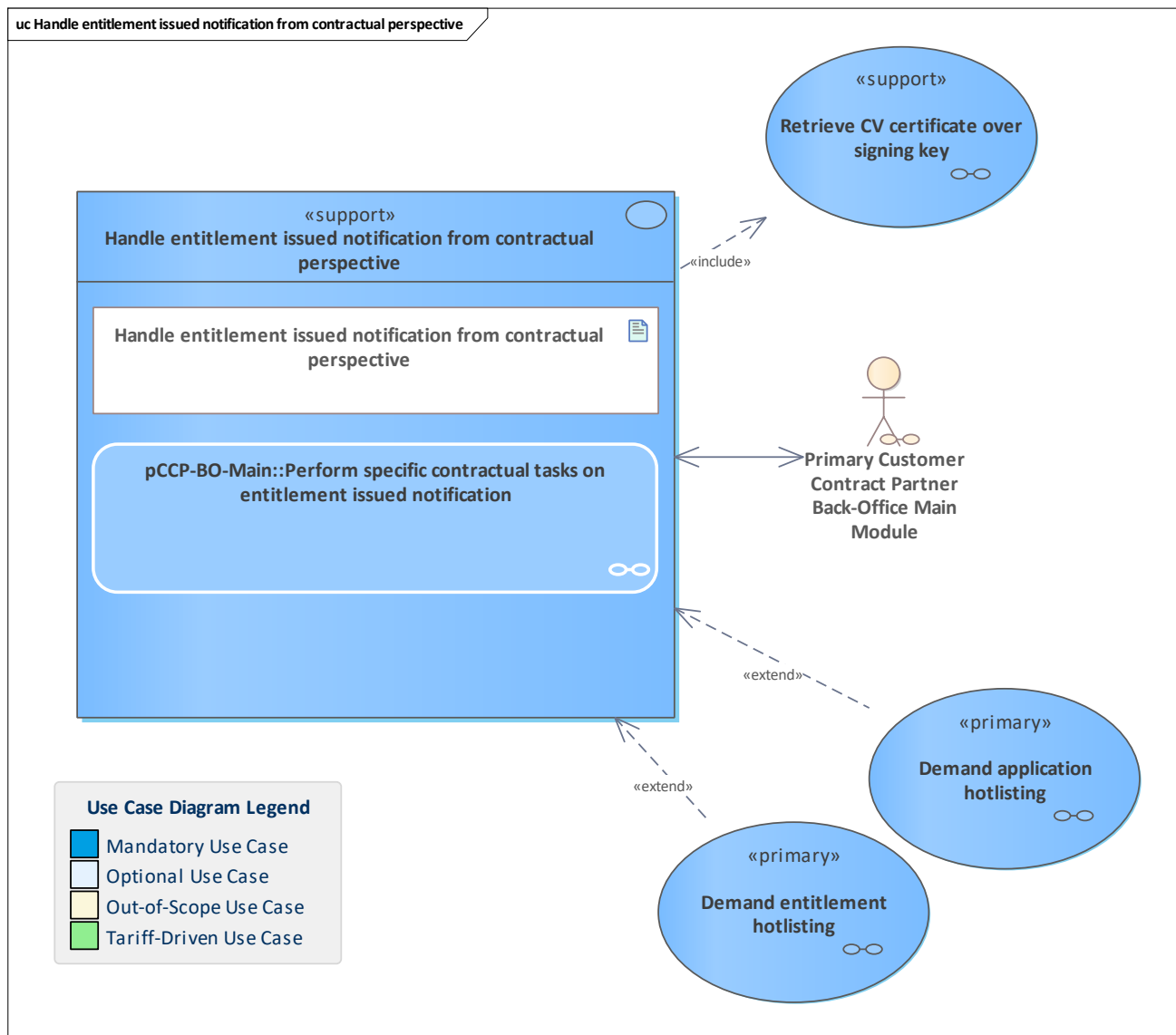


Figure 316: Handle entitlement issued notification from contractual perspective

Handle a notification about the issuance of an owned entitlement from the contractual perspective by performing the related checks and monitoring. Note that the application instance ID of the user medium the entitlement was issued to can be uniquely identified via *umAppInstanceId*, which is part of the [SignedEntitlementIssuedAttestation](#) that is contained in the [EntitlementIssuedNotification](#).

11.140 Handle entitlement issued notification from operational perspective

11.141 Handle entitlement issued notification from operational perspective

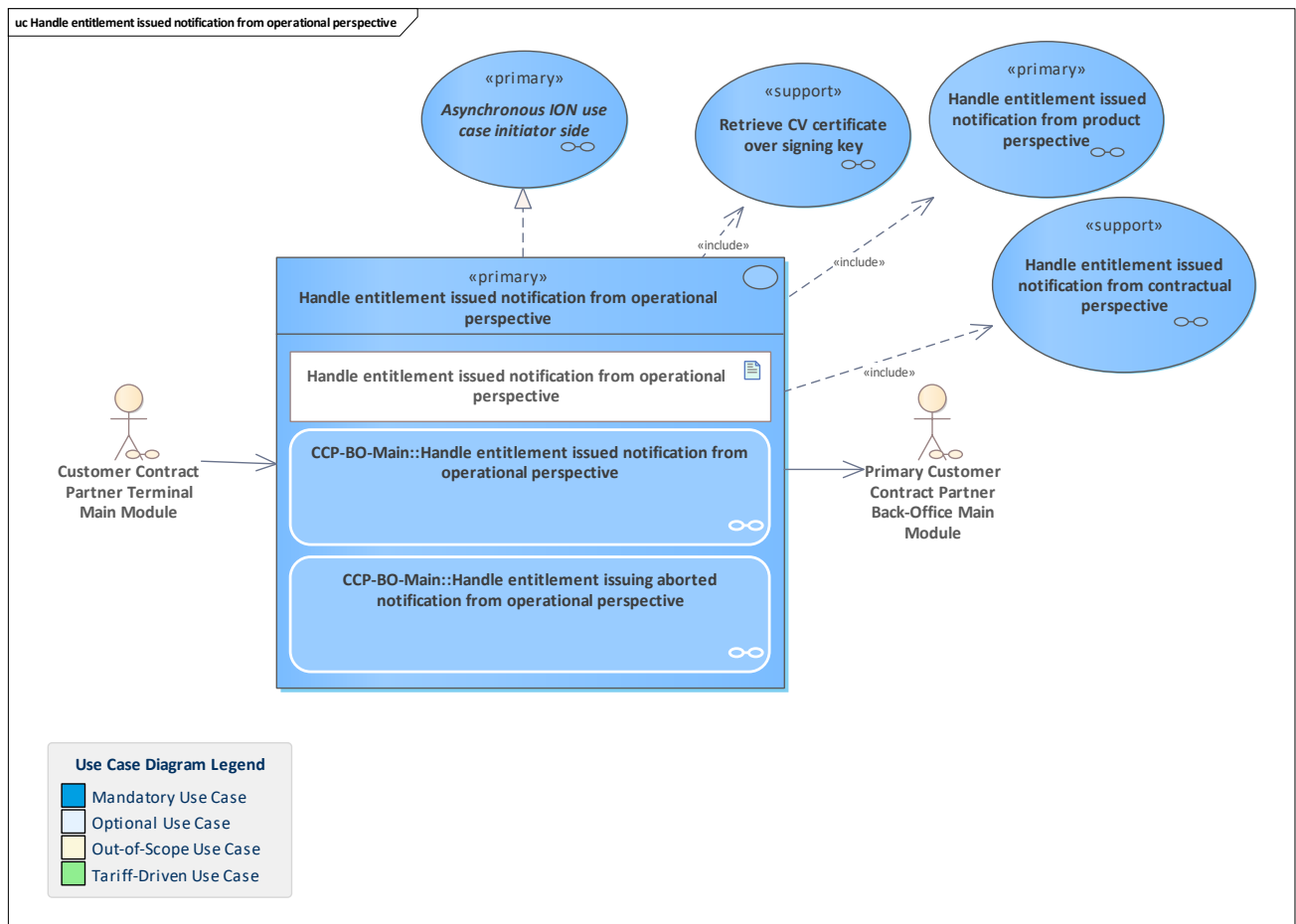


Figure 317: Handle entitlement issued notification from operational perspective

The pCCP of the entitlement receives the notification about an entitlement issuance and registers it.

Then it handles the notification from the operational perspective by performing checks and monitoring, e.g. verifying the signature of the issuance attestation.

Since it is the pCCP, it also does the contractual checks and monitoring.

Finally, the notification is forwarded to the responsible PO system. This can be done either with a single message or in a scheduled process with a list of messages.

In case of an abortion, the notification is also sent to the PO system to announce the used SAM- and product issuance counters.

Note: this use case is without the context of ordered action execution.

11.142 Handle entitlement issued notification from product perspective

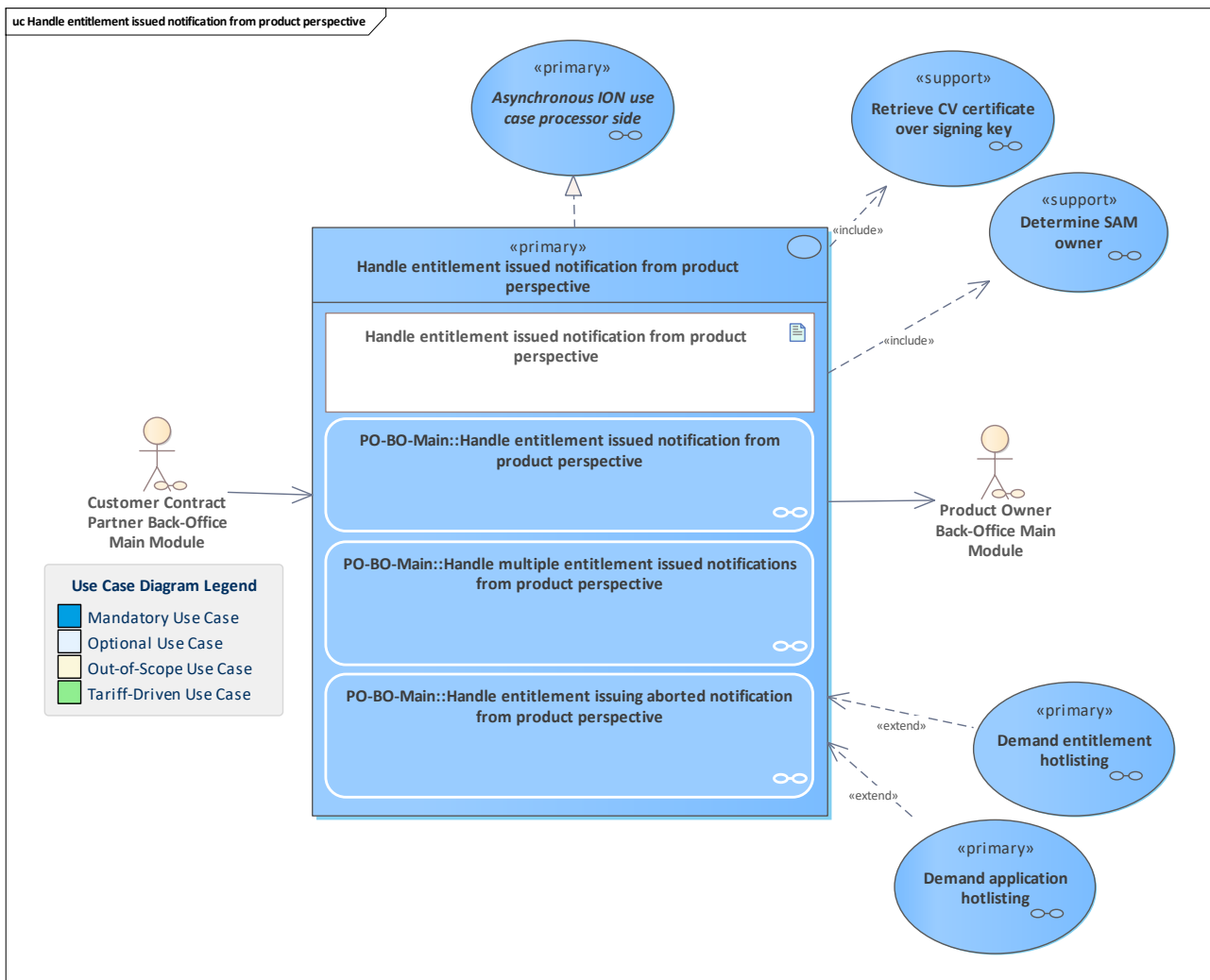


Figure 318: Handle entitlement issued notification from product perspective

The PO back-office system receives the notification about an entitlement issuance and handles it from the product perspective. It registers the notification and does checks and monitoring. In the case of a list of notifications, this list is processed instead of a single notification. In the case of an abortion notification, the PO system has to register the SAM and product issuance counter for consistent monitoring.

Note: the application instance ID of the user medium the entitlement was issued to can be uniquely identified via the request field *umAppInstanceId*, which is part of the [SignedEntitlementIssuedAttestation](#) that is contained in the [EntitlementIssuedNotification](#).

Note: for this use case, the issuance was not triggered by an action order.

11.143 Handle entitlement terminated notification from contractual perspective

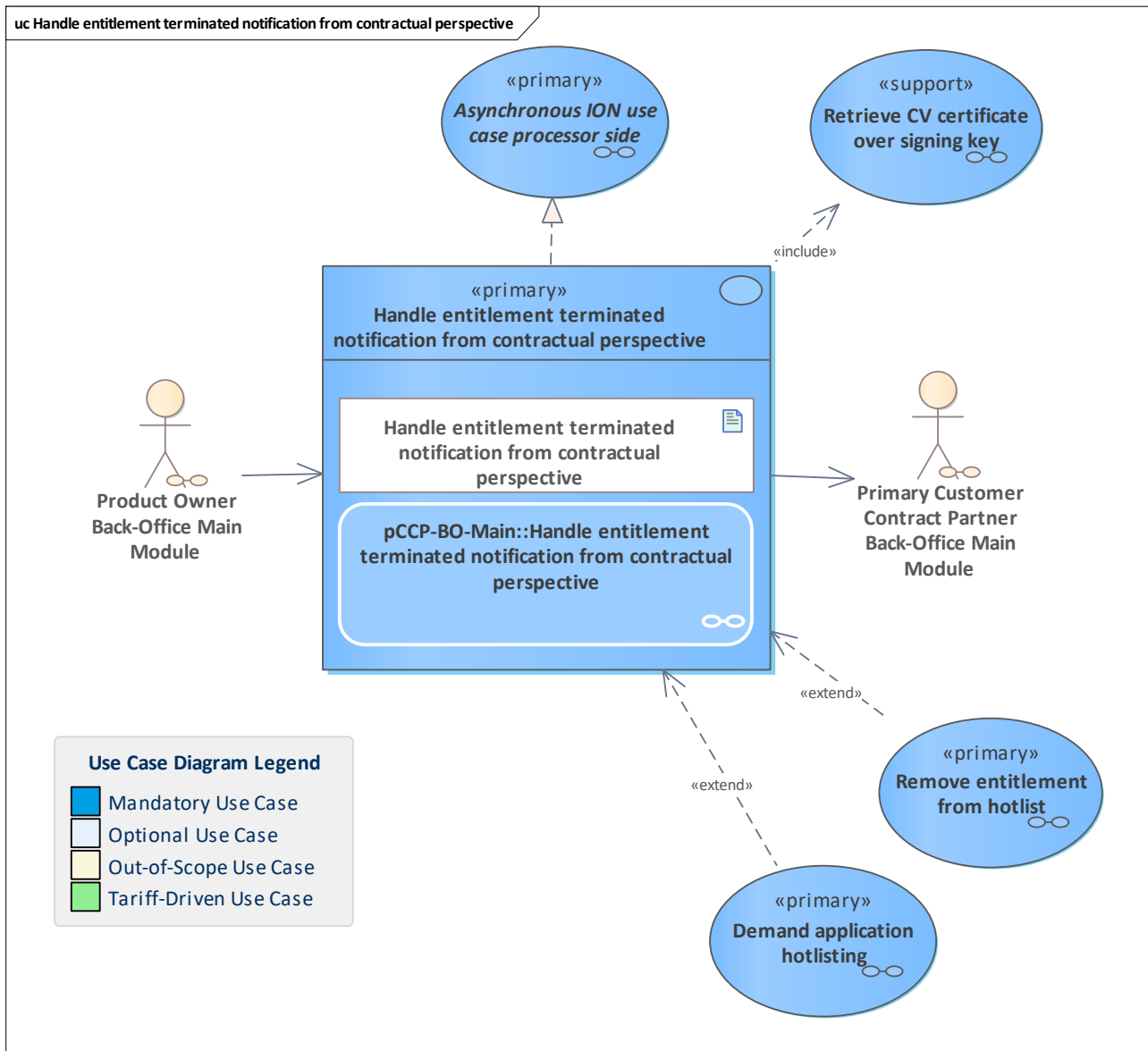


Figure 319: Handle entitlement terminated notification from contractual perspective

Handle an entitlement terminated notification from the contractual perspective.

The entitlement terminated notification is received by the pCCP.

The pCCP does its checks and monitoring from the contractual perspective regarding the correct execution of termination. In this context, the signature of the termination attestation is verified.

Note: this closes a potentially related user account concerning the entitlement.

If the termination was correct, the pCCP checks if the entitlement has to be removed from the hotlist.

Note: if the pCCP itself performed the termination, this use case takes place inside the use case [Handle entitlement terminated notification from operational perspective](#) and is not called by the PO. In this case, this use case is not an [Asynchronous ION use case processor side](#).

11.144 Handle entitlement terminated notification from operational perspective

11.145 Handle entitlement terminated notification from operational perspective

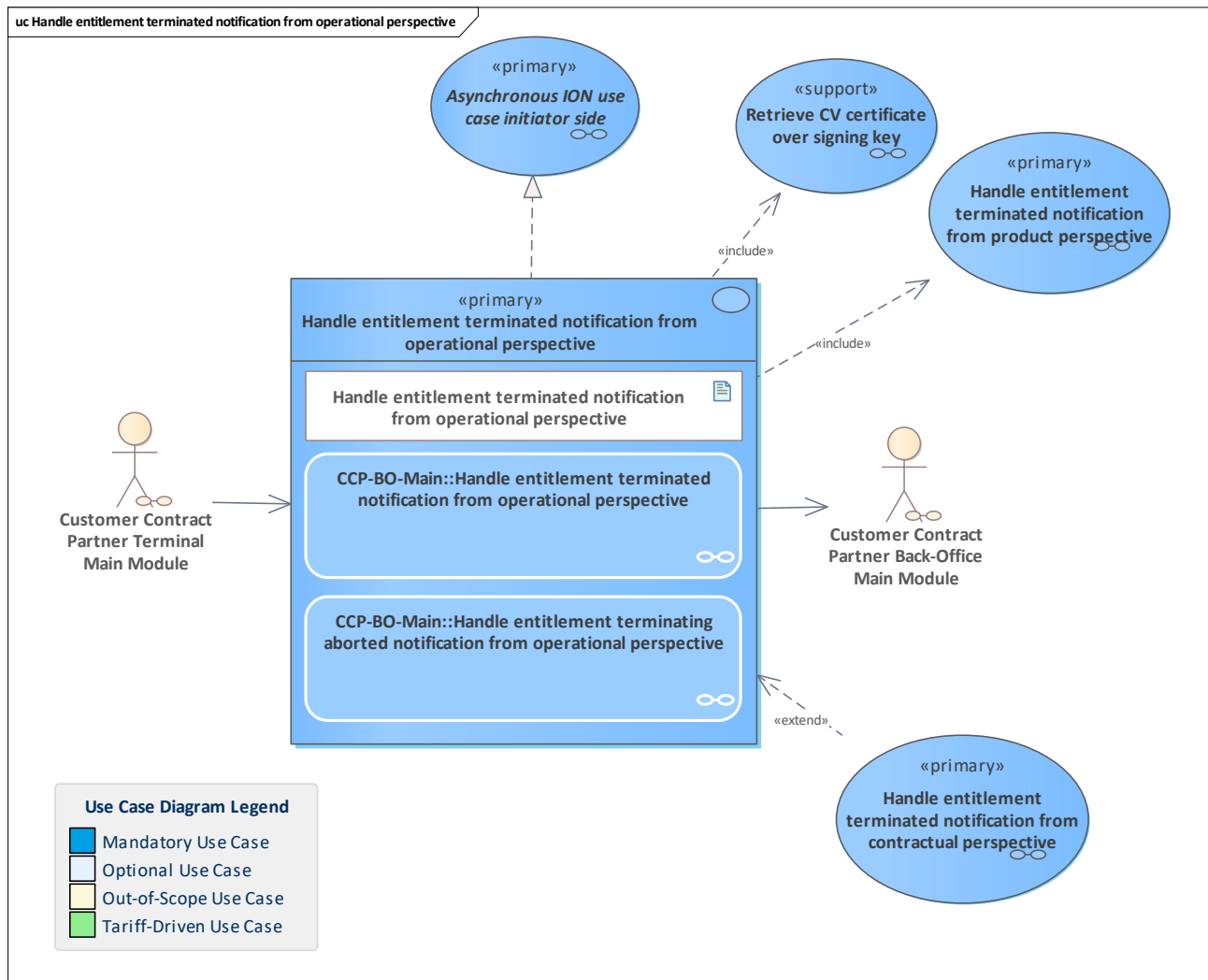


Figure 320: Handle entitlement terminated notification from operational perspective

Handle an entitlement terminated notification from the operational perspective. The entitlement terminated notification is sent by the CCP terminal to the CCP back-office system. The notification will be checked and monitored, e.g. by verifying the signature of the embedded attestation.

If the pCCP has terminated its own entitlement:

- the notification will be sent to the PO
- the pCCP does its contractual checks and monitoring

If a sCCP has terminated the entitlement:

- the notification will be sent to the PO (and the PO will forward it later to the pCCP)

In the case of action abortion, terminal and SAM action data is sent by the terminal to the back-office system. The CCP back-office system registers the abortion for internal monitoring and data consistency.

11.146 Handle entitlement terminated notification from product perspective

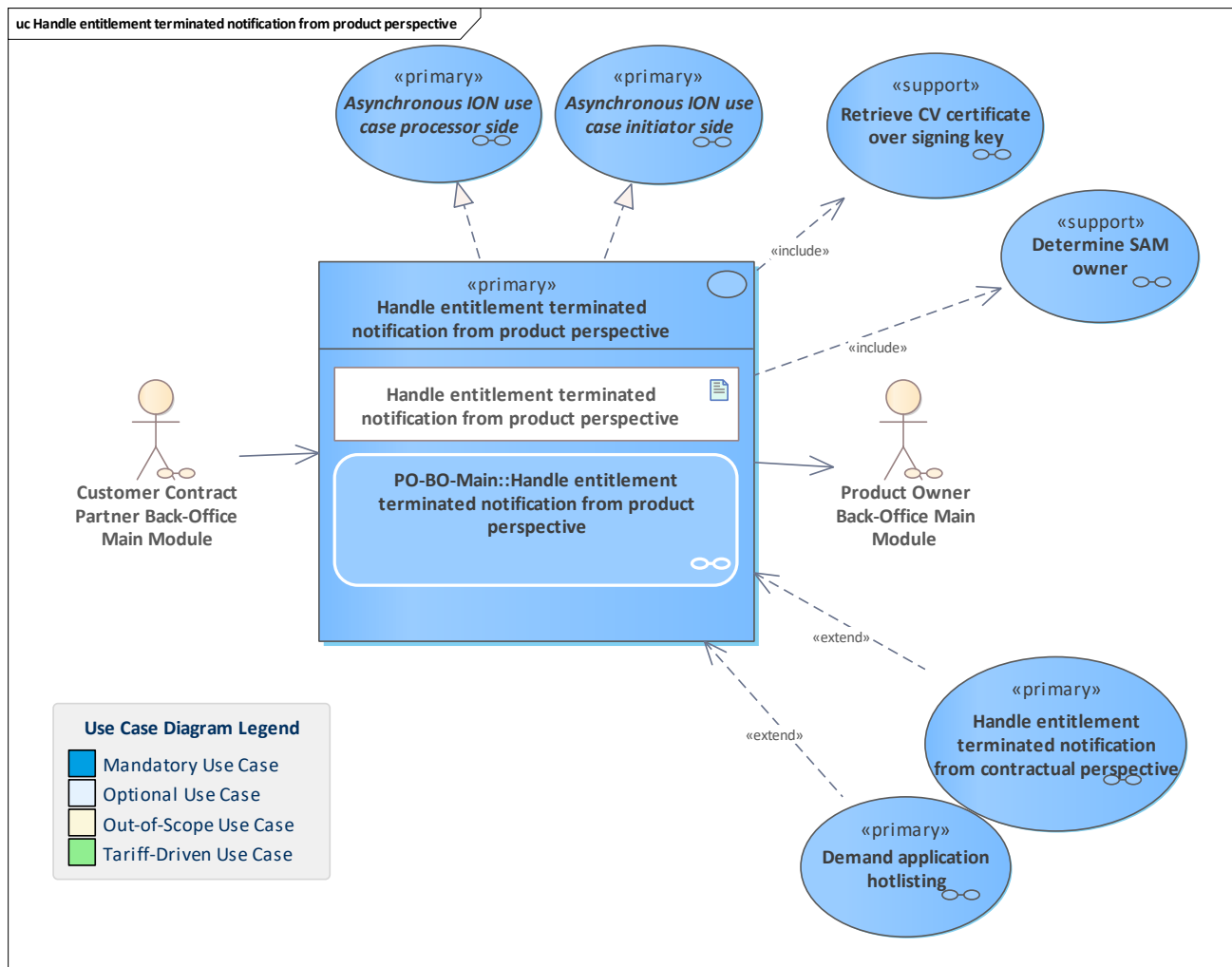


Figure 321: Handle entitlement terminated notification from product perspective

Handle an entitlement terminated notification from the product perspective.

The entitlement terminated notification is received by the PO.

The PO registers the entitlement terminated notification and performs its checks and monitoring from the product perspective regarding the correct execution of the termination. In this context, the signature of the termination attestation is verified and the SAM owner of the SAM that performed the termination is determined.

- if the sender is a sCCP: forward the notification to the pCCP
- if the sender is the pCCP, do not forward the notification. In this case, the current use case is not an asynchronous ION use case as initiator ([Asynchronous ION use case initiator side](#)).

Note: in the ION context, the use case is asynchronous as processor (process the notification) and an asynchronous use case as initiator due to the asynchronous call to the pCCP.

11.147 Handle entitlement termination order

11.148 Handle entitlement termination order

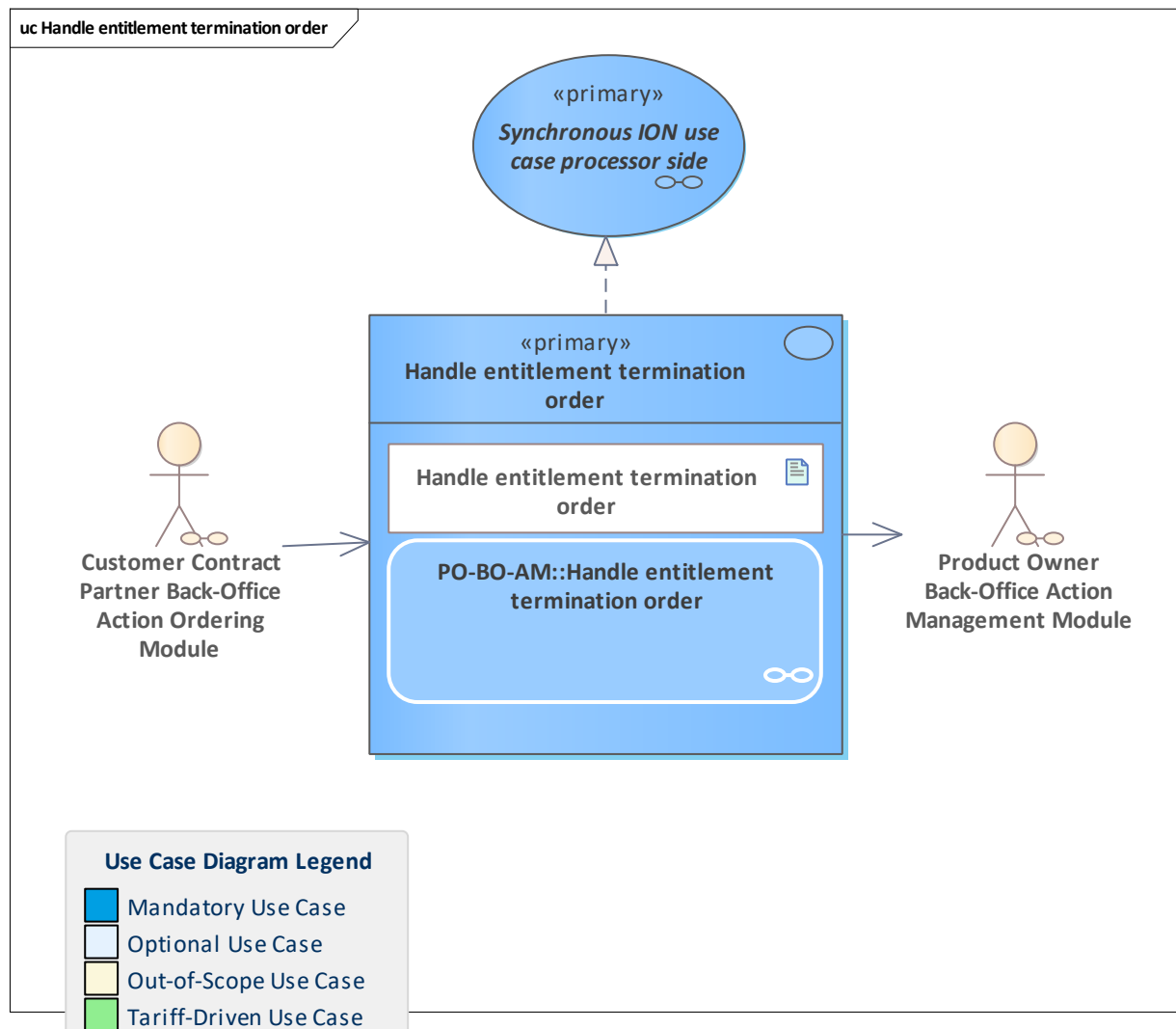


Figure 322: Handle entitlement termination order

The PO back-office system with an action management extension handles an entitlement termination order.

If the order passes all checks, it is added to the order inventory and may be considered for the next action list.

11.149 Handle entitlement unblocked notification from contractual perspective

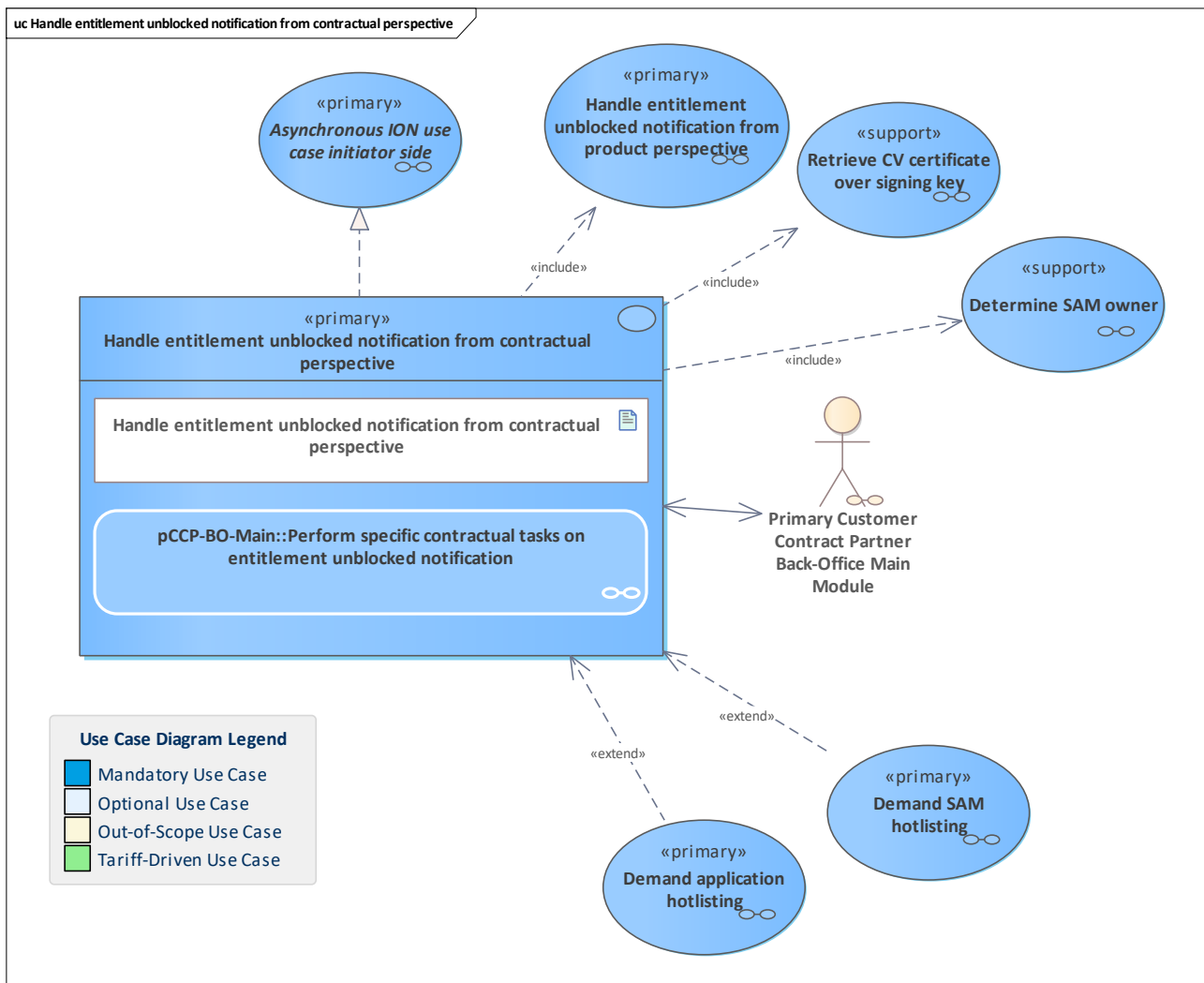


Figure 323: Handle entitlement unblocked notification from contractual perspective

Handle an entitlement unblocked notification from the contractual perspective. The pCCP does its checks and monitoring from the contractual perspective regarding the correct execution of the unblock action. In this context, the signature of the embedded attestation is verified.

11.150 Handle entitlement unblocked notification from operational perspective

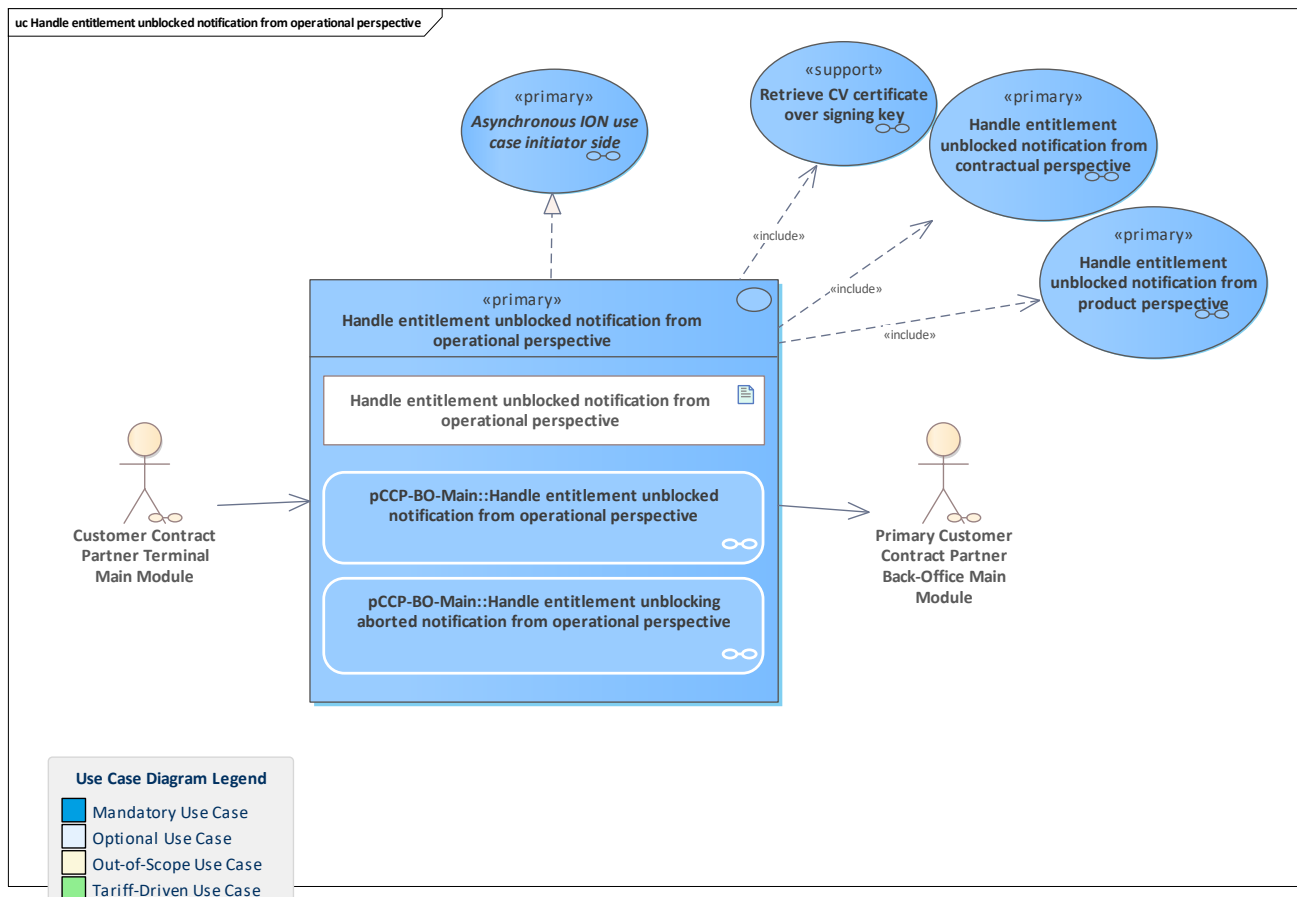


Figure 324: Handle entitlement unblocked notification from operational perspective

A terminal has unblocked the entitlement in a user medium with an application. In this use case, the pCCP back-office system receives the notification from the terminal and runs certain checks from the operational perspective such as the signature verification of the attestation of the unblocking. Then the contractual checks are done. Finally, the notification is forwarded to the PO. As an alternative flow, a potential transaction abortion is handled.

11.151 Handle entitlement unblocked notification from product perspective

11.152 Handle entitlement unblocked notification from product perspective

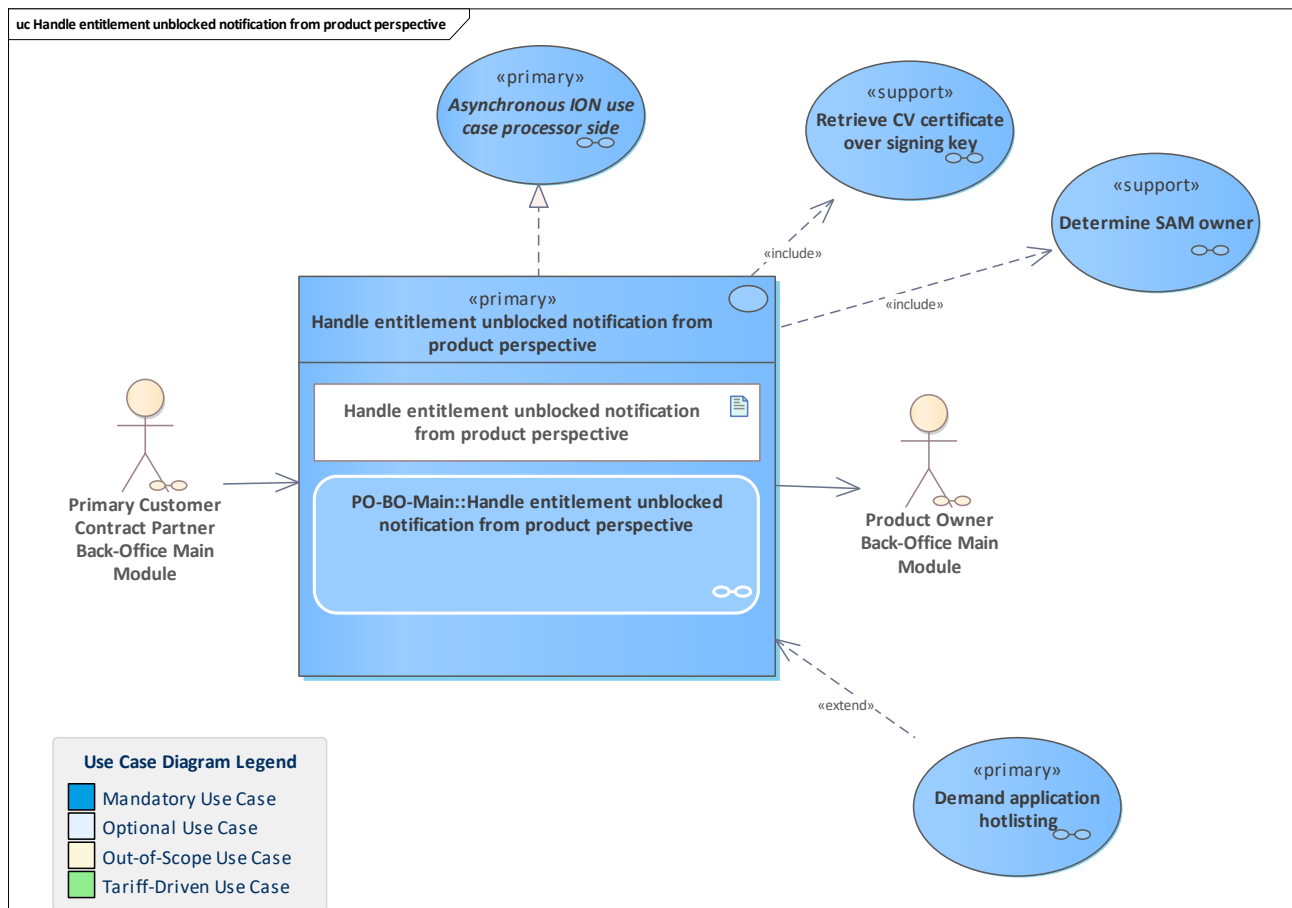


Figure 325: Handle entitlement unblocked notification from product perspective

Handle an entitlement unblocked notification from the product perspective.

The entitlement unblocked notification is received by the PO.

The PO registers entitlement unblocked notification and does its checks and monitoring from the product perspective regarding the correct execution of unblocking. In this context, the signature of the embedded attestation is verified and the SAM owner of the SAM that performed the action is determined.

Note: in the ION context, the use case is asynchronous as processor (process the notification) and an asynchronous use case as initiator due to the asynchronous call to the pCCP.

11.153 Handle entitlement unblocking order

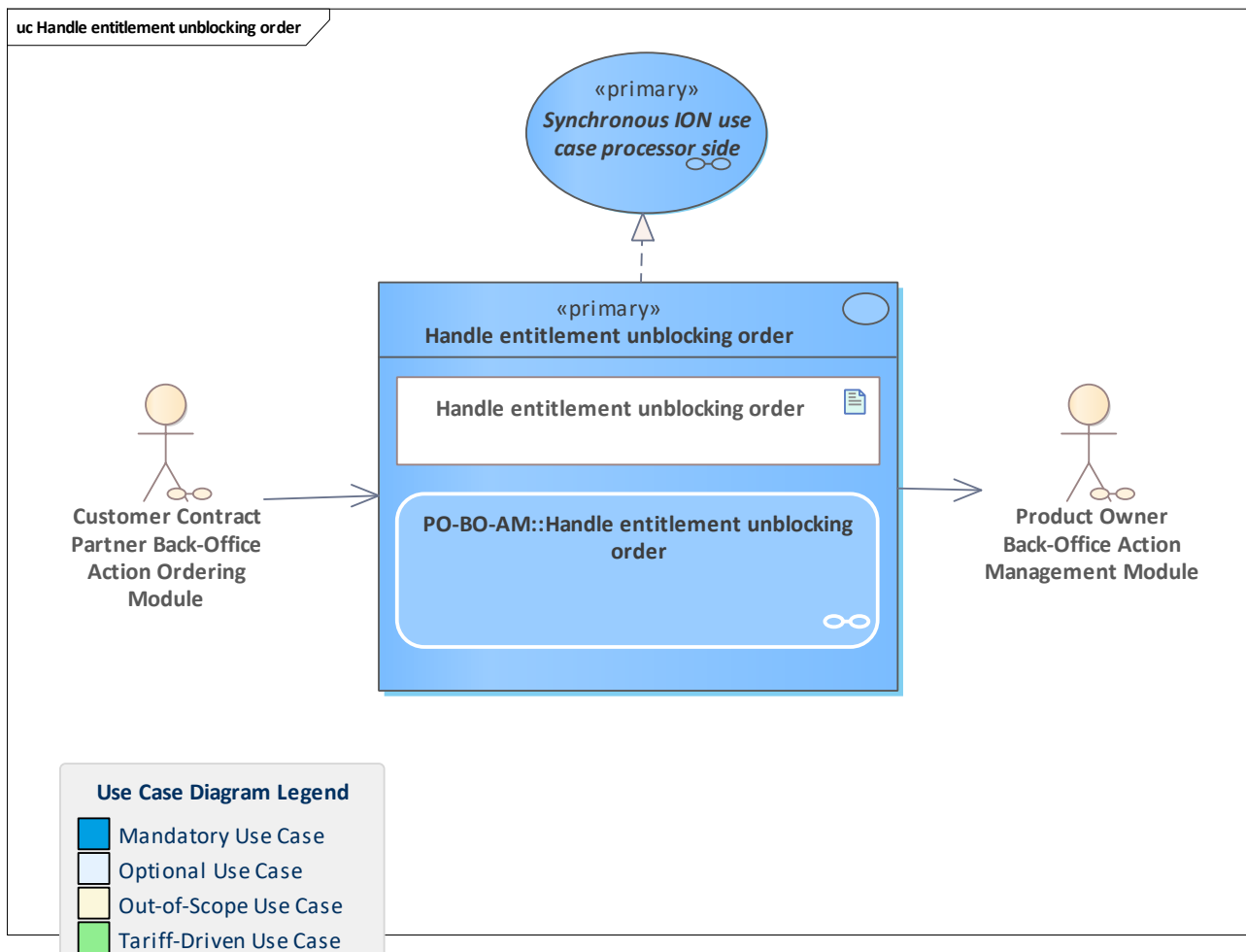


Figure 326: Handle entitlement unblocking order

The [Product Owner Back-Office Action Management Module](#) handles an entitlement unblocking order.

If the order passes all checks, it is added to the order inventory and may be considered for the next action list.

11.154 Handle entitlement validated notification from contractual perspective

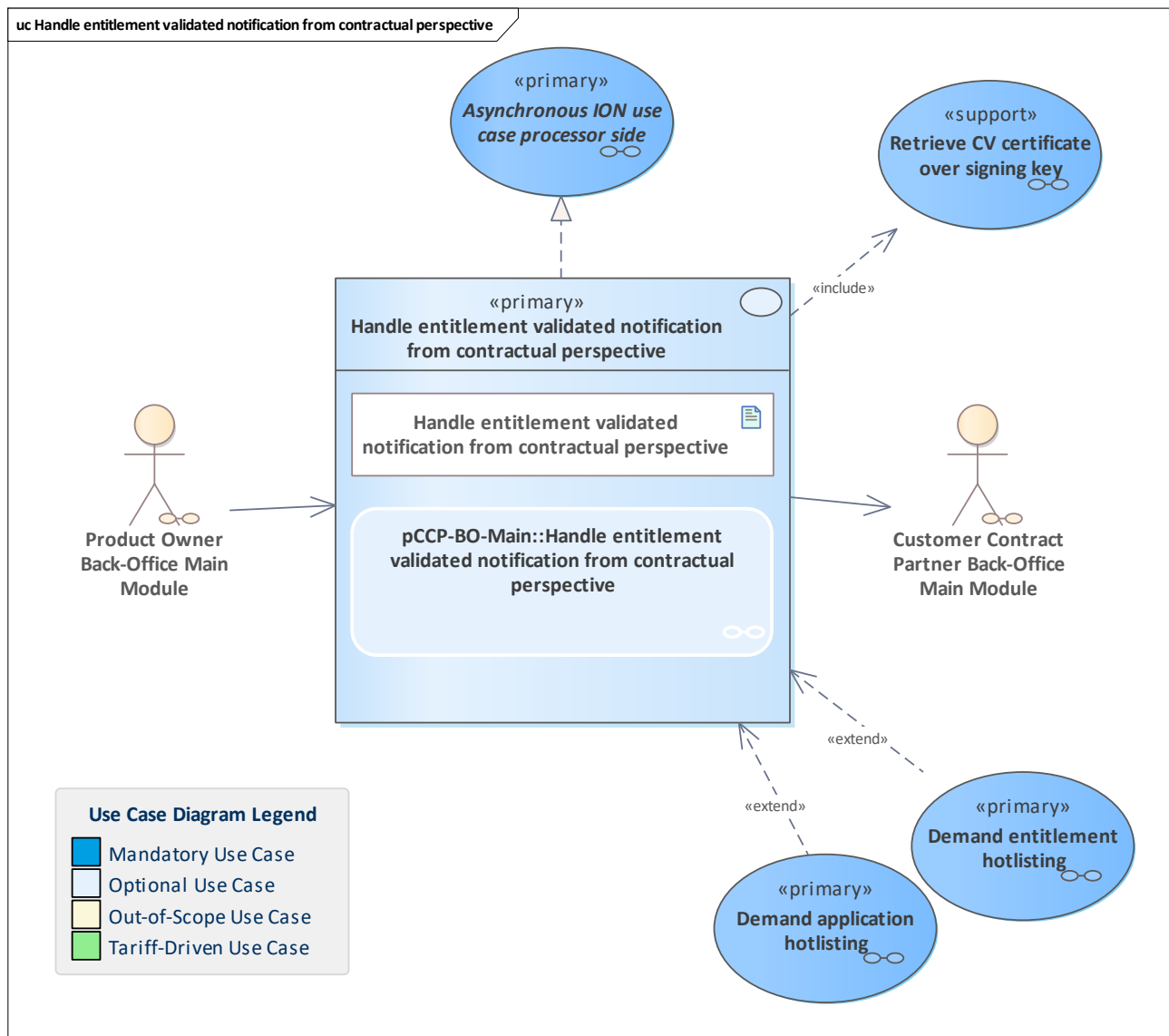


Figure 327: Handle entitlement validated notification from contractual perspective

Handle an entitlement validated notification from the contractual perspective. The entitlement validated notification is received by the pCCP. The pCCP does its checks and monitoring from the contractual perspective regarding the correct execution of the validation. In this context, the signature of the embedded attestation is verified.

11.155 Handle entitlement validated notification from operational perspective

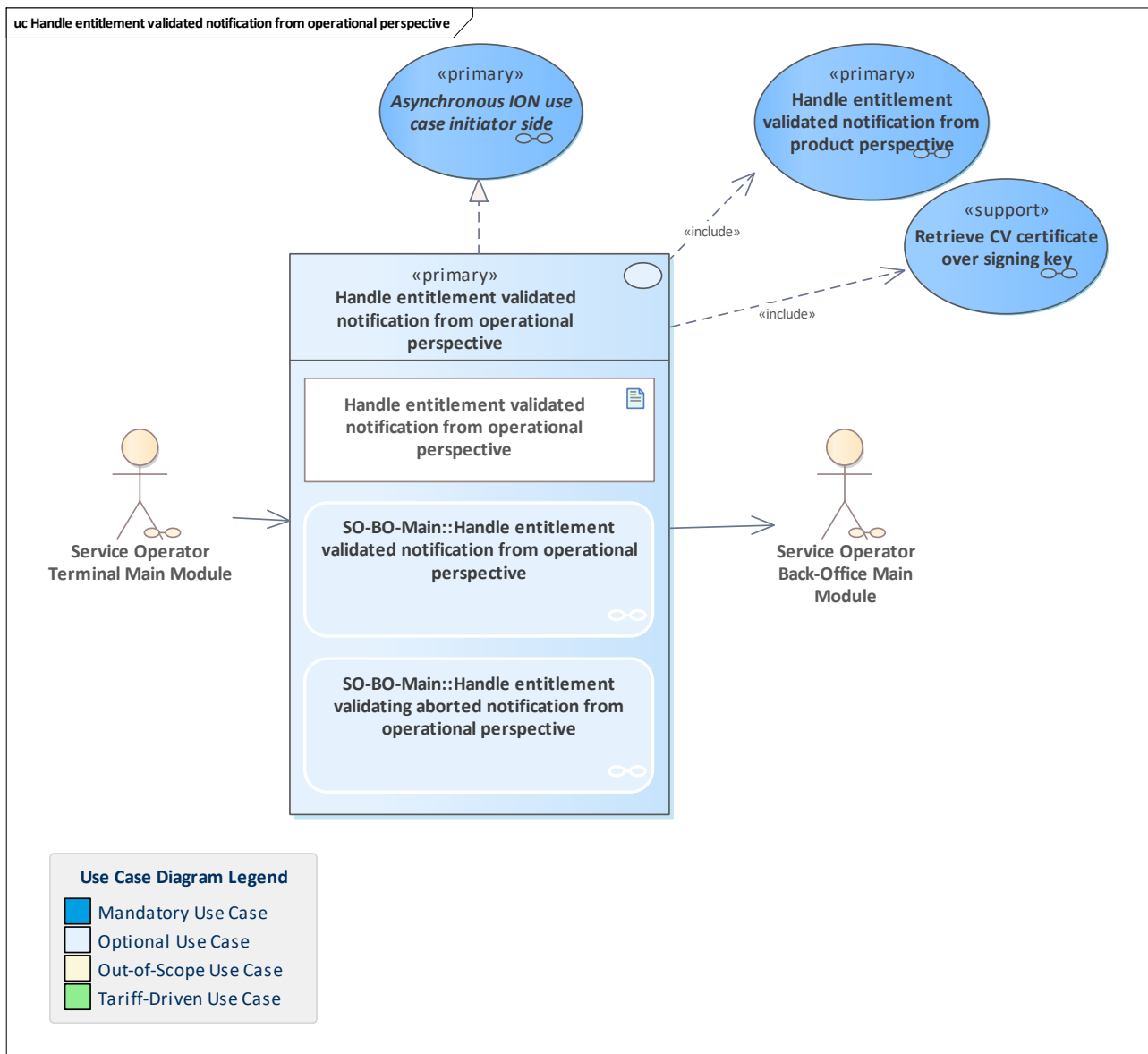


Figure 328: Handle entitlement validated notification from operational perspective

Handle an entitlement validated notification from the operational perspective. The entitlement validation notification is sent by the SO terminal to the SO back-office system. The notification will be checked and monitored, e.g. by verifying the signature of the embedded attestation. Finally, the notification will be sent to the PO (and the PO will forward it later to the pCCP) In the case of action abortion, terminal and SAM action data is sent by the terminal to the back-office system. The CCP back-office system registers the abortion for internal monitoring and data consistency.

11.156 Handle entitlement validated notification from product perspective

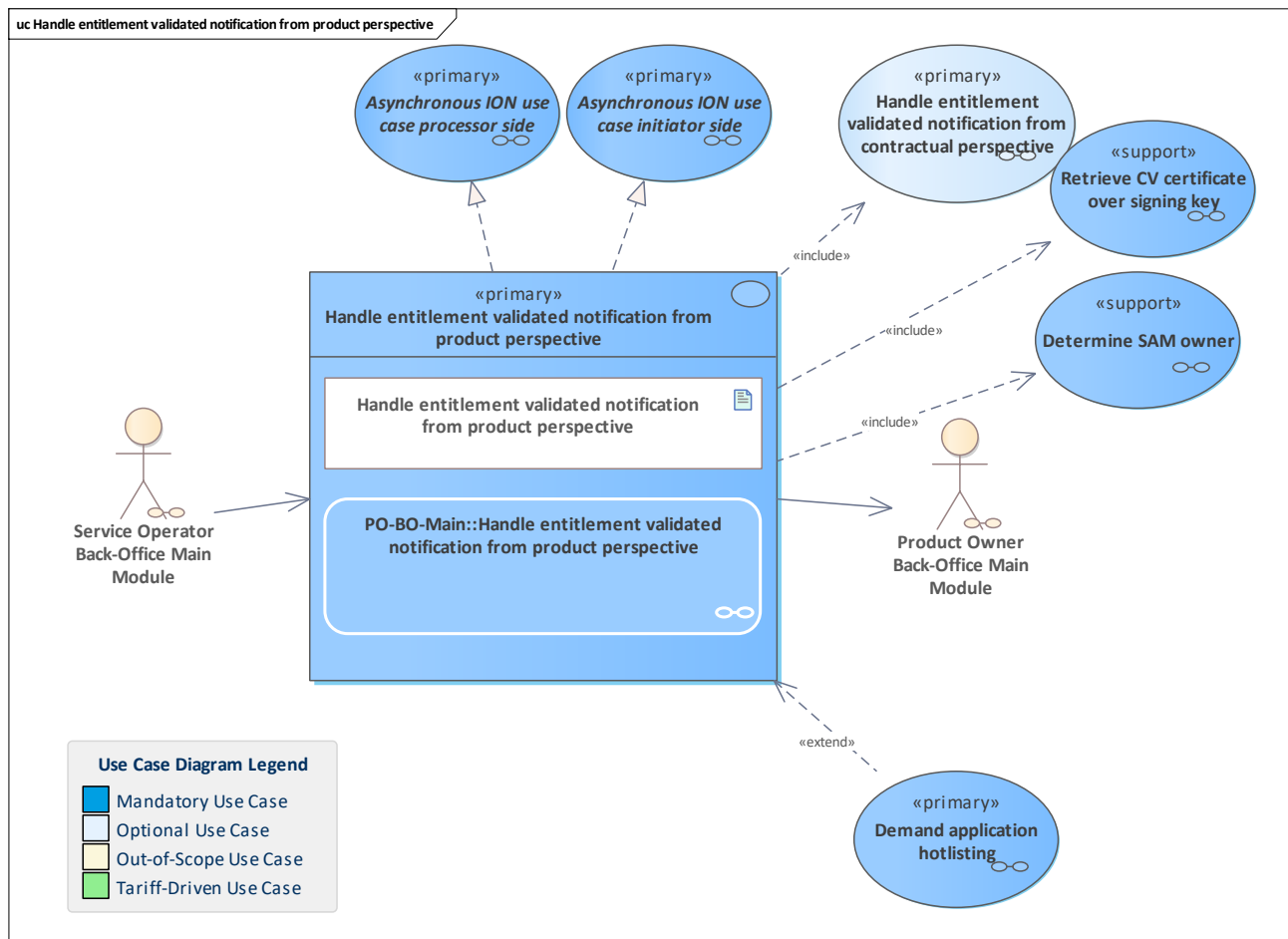


Figure 329: Handle entitlement validated notification from product perspective

Handle an entitlement validated notification from the product perspective.

The entitlement validated notification is received by the PO.

The PO registers the entitlement validated notification and does its checks and monitoring from the product perspective regarding the correct execution of validation. In this context, the signature of the validation attestation is verified and the SAM owner of the SAM that performed the validation is determined.

Finally, the notification is forwarded to the pCCP

Note: in the ION context, the use case is asynchronous as processor (process the notification) and an asynchronous use case as initiator due to the asynchronous call to the pCCP.

11.157 Handle entitlement XY notification from contractual perspective

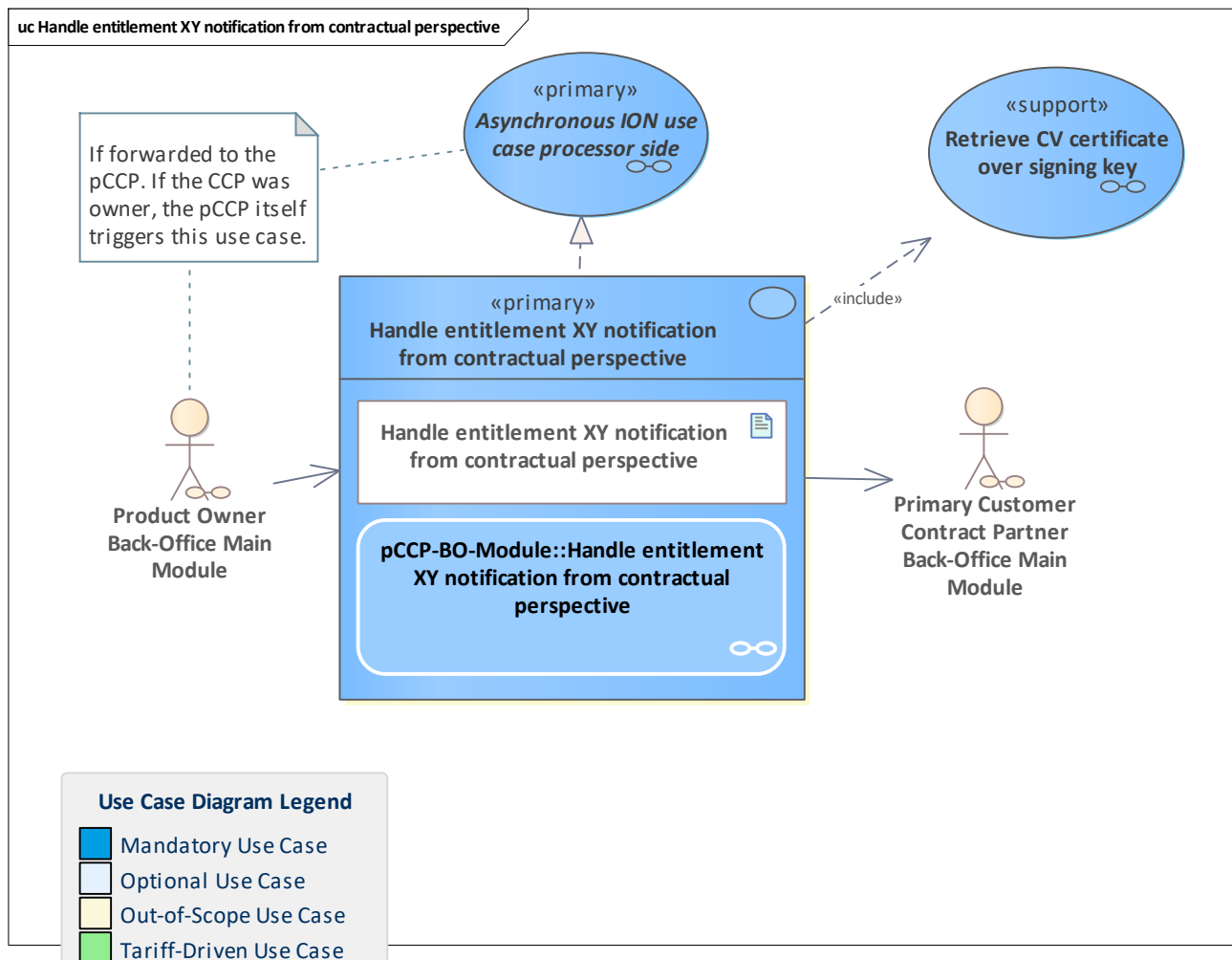


Figure 330: Handle entitlement XY notification from contractual perspective

A CCP system processes a notification about an action executed on an owned entitlement. The processing is done from the contractual perspective focusing on the entitlement lifecycle and payment aspects.

This use case has two entry points:

- [pCCP-BO-Module::Handle entitlement XY notification from contractual perspective](#)
This entry point is used in the non-owned scenarios
- [pCCP-BO-Module::Perform specific contractual tasks on entitlement XY notification](#)
This entry point is used in the owned scenarios

The first includes the latter, in which most of the processing is done.

11.158 Handle entitlement XY notification from operational perspective

11.159 Handle entitlement XY notification from operational perspective

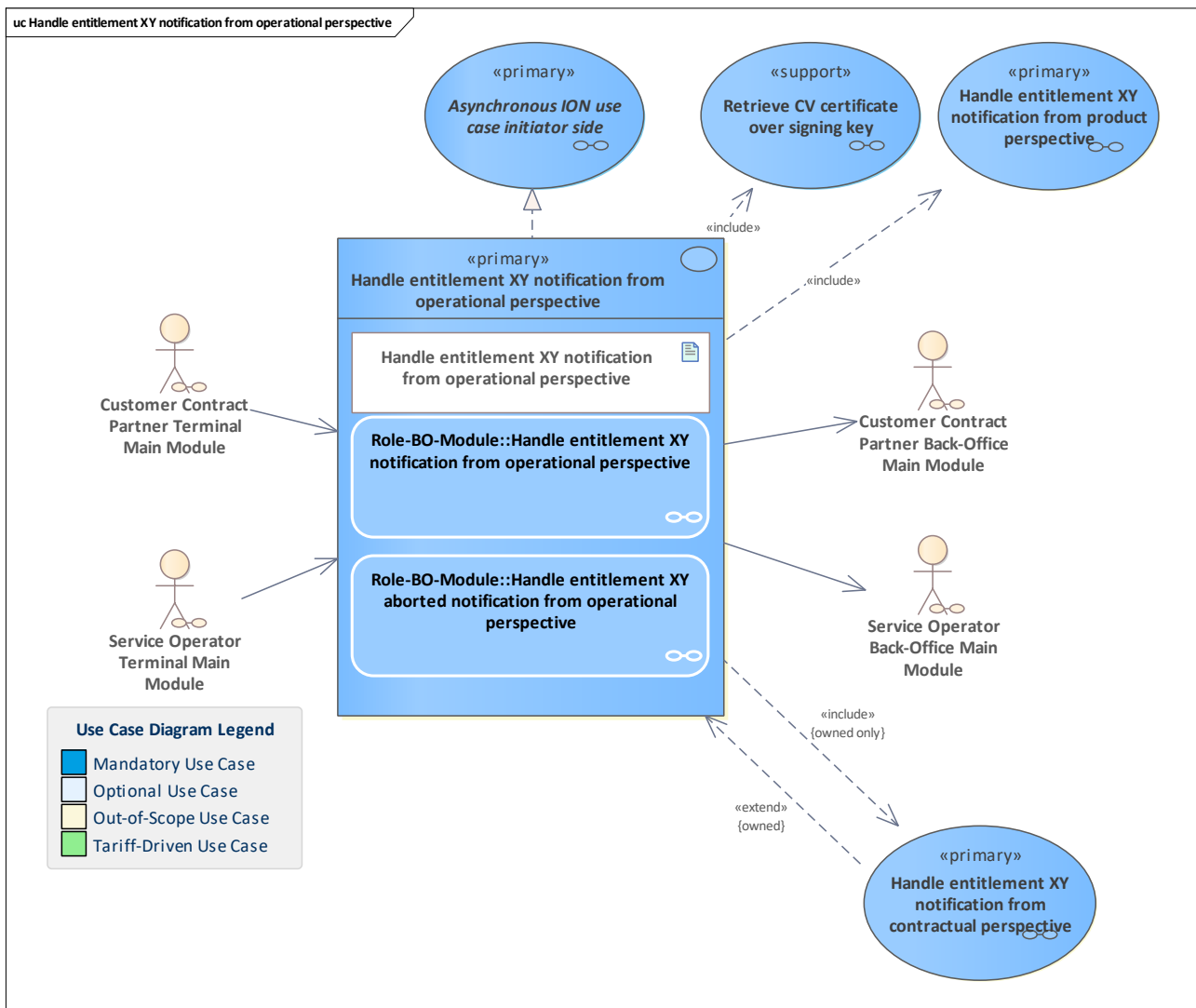


Figure 331: Handle entitlement XY notification from operational perspective

A back-office system belonging to a terminal that executed a UM action with an entitlement processes the notification about that action execution.

The processing is done from the operational perspective, focusing on the terminal-side of the action execution, i.e. logging the used SAM counter values.

Depending on the use case and the ownership of the entitlement the action was executed on, other back-office systems are informed about the action execution.

11.160 Handle entitlement XY notification from product perspective

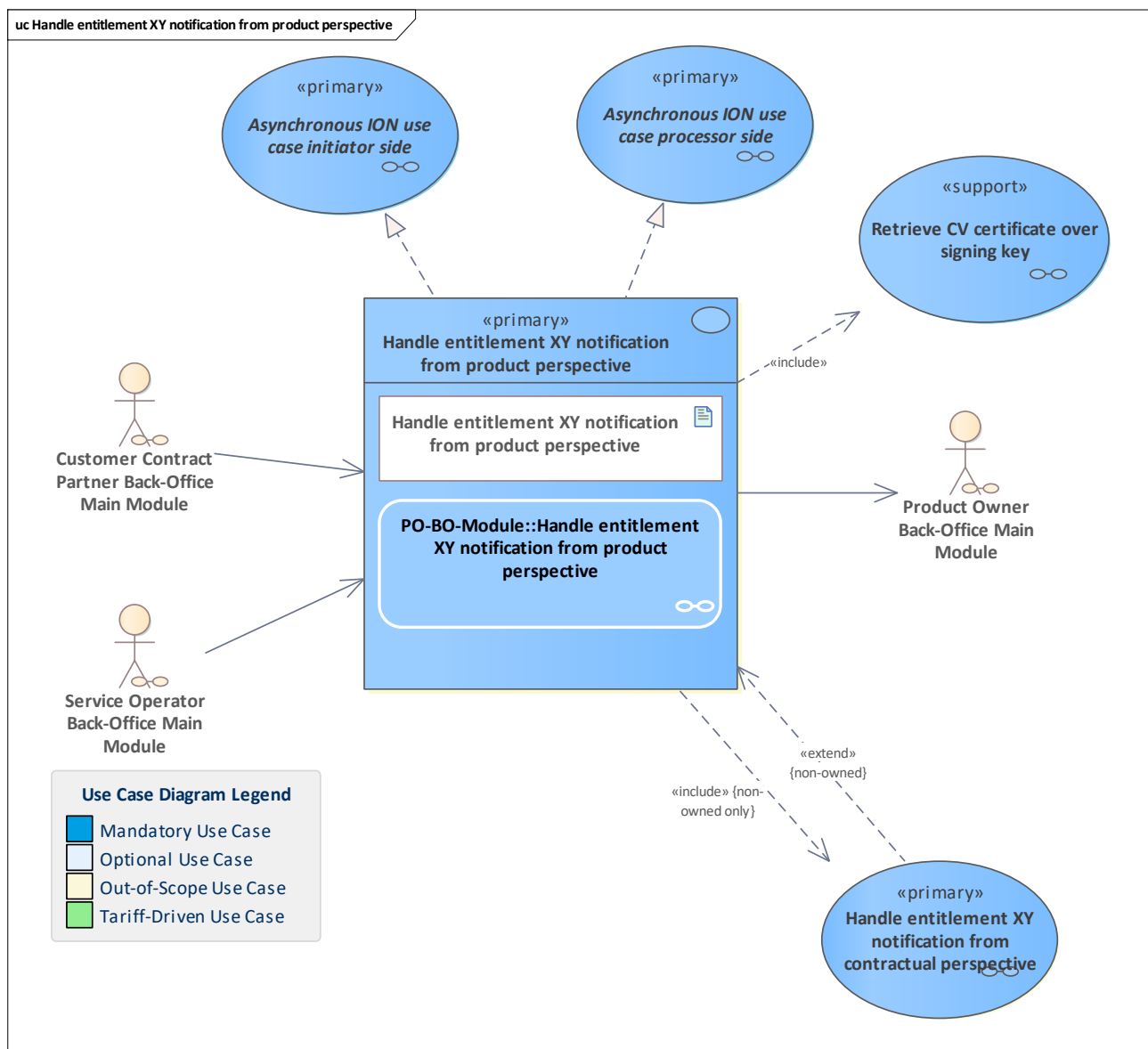


Figure 332: Handle entitlement XY notification from product perspective

A PO system processes a notification about an entitlement action executed on an entitlement that is an instance of an owned product. The processing is done from the product perspective, focusing on the entitlement lifecycle and tariff aspects.

11.161 Handle events notification

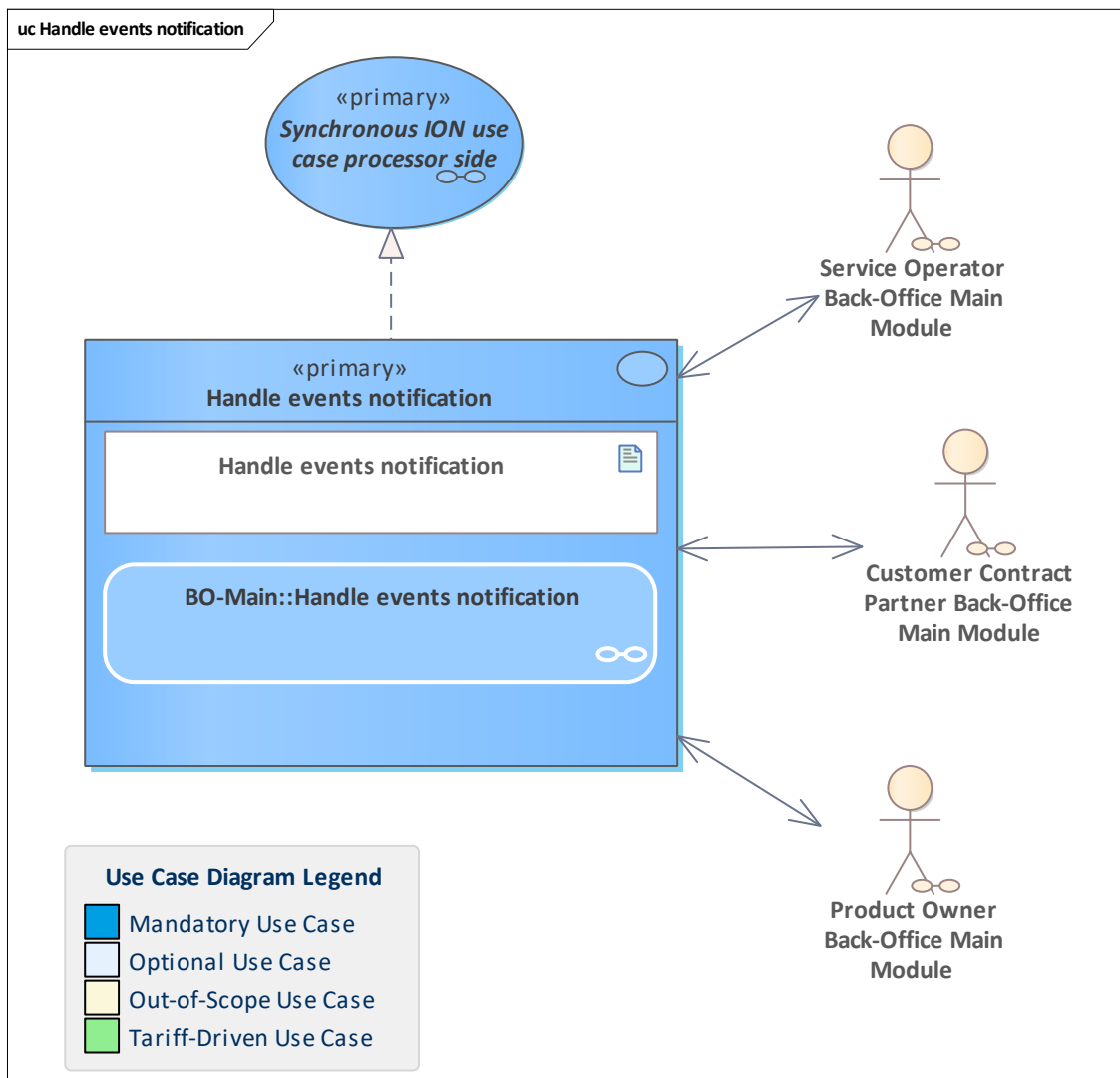


Figure 333: Handle events notification

A participant is informed about warnings.

11.162 Handle LDAP search request

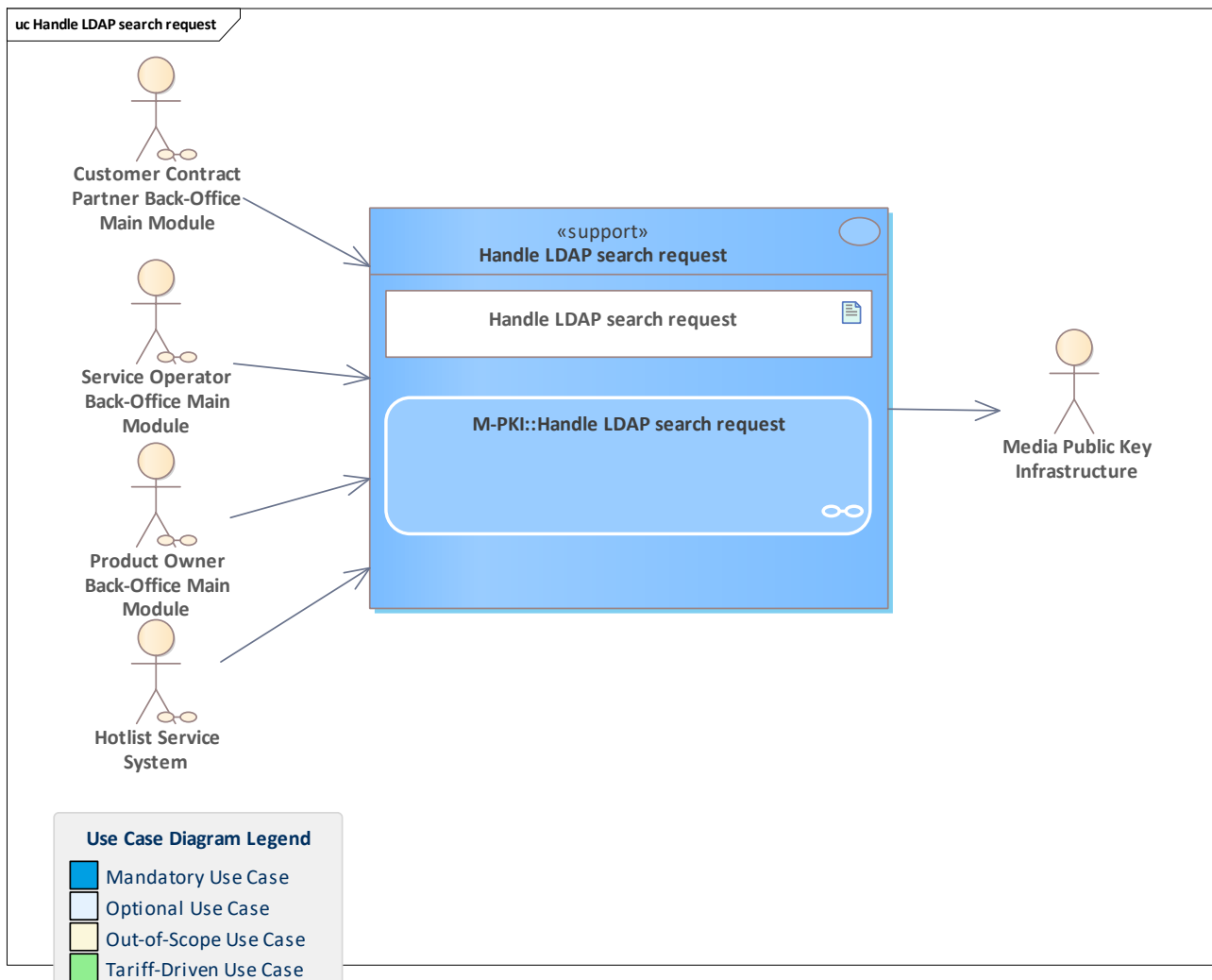


Figure 334: Handle LDAP search request

The Media PKI provides LDAP access to the CV certificates.
For technical details, see [M-PKI LDAP Documentation](#).

11.163 Handle lost user medium with application

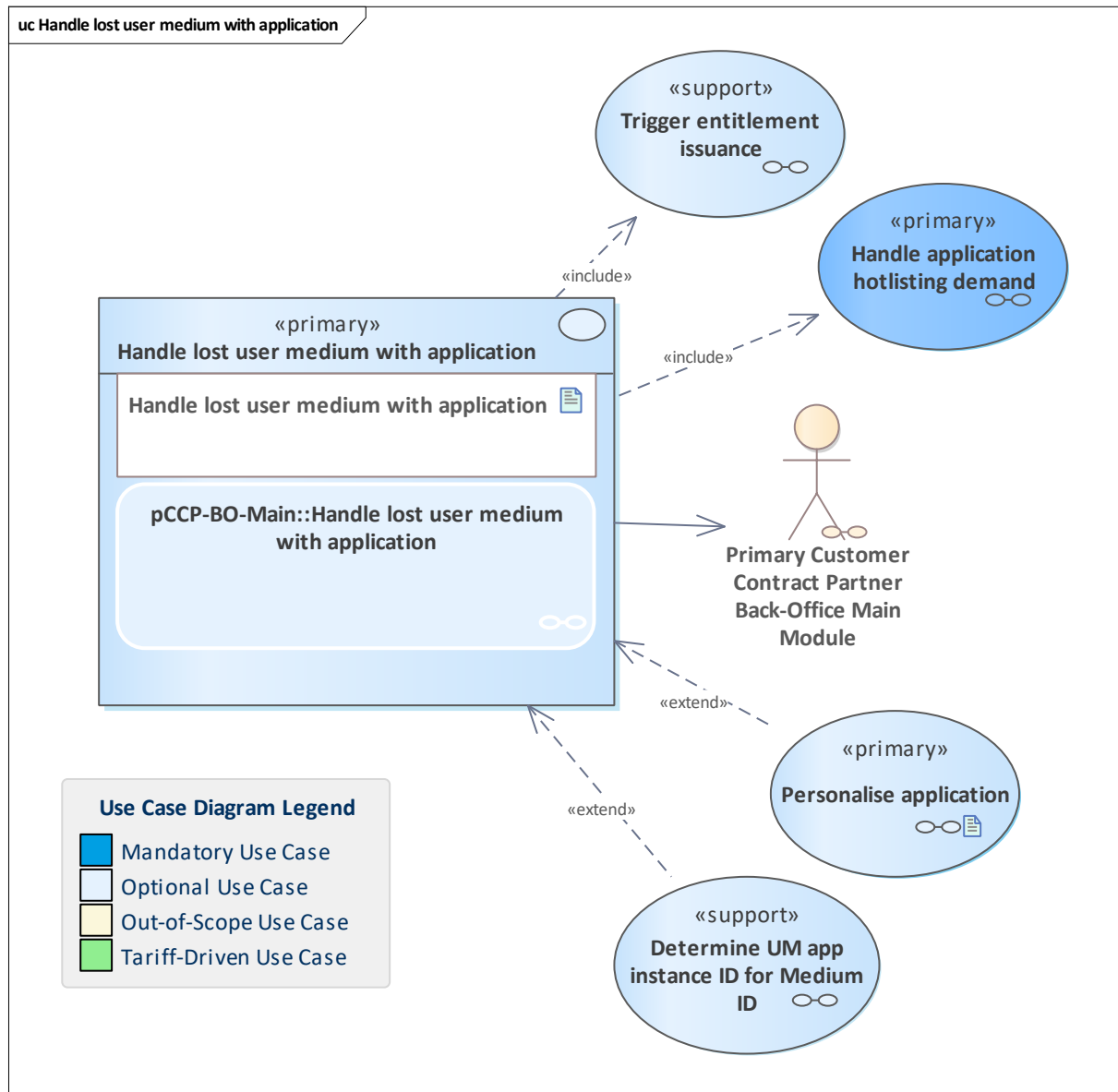


Figure 335: Handle lost user medium with application

A user medium with an application is lost. Therefore, the application must be hotlisted and entitlements on the lost user medium can be re-issued on a new user medium with application.

Please note that not yet valid entitlements are not reissued.

11.164 Handle order cancellation

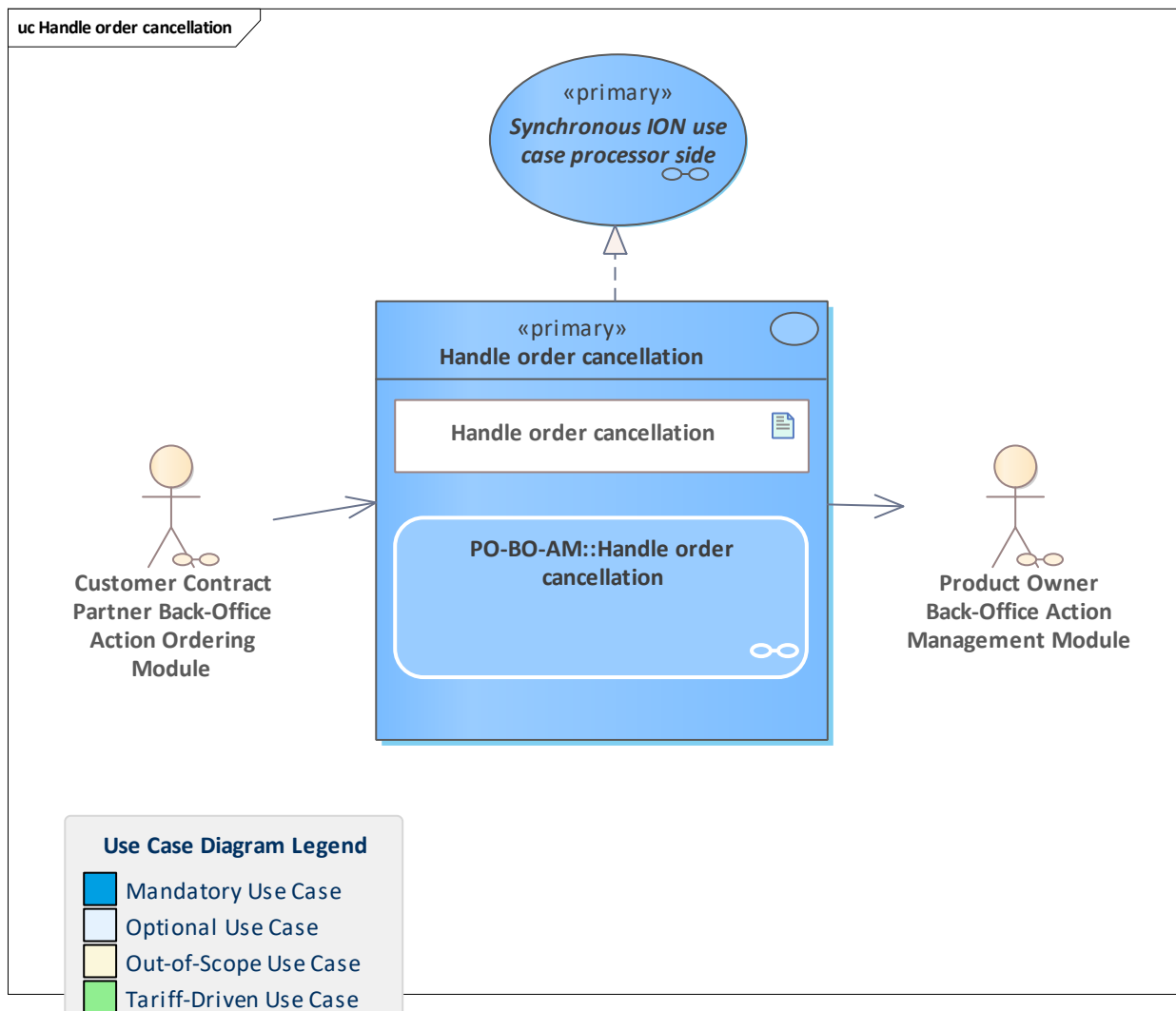


Figure 336: Handle order cancellation

The [Product Owner Back-Office Action Management Module](#) handles an order cancellation. If the cancellation passes all checks, the referenced order is marked as cancelled in the order inventory and will be removed from the next action list.

11.165 Handle order group

11.166 Handle order group

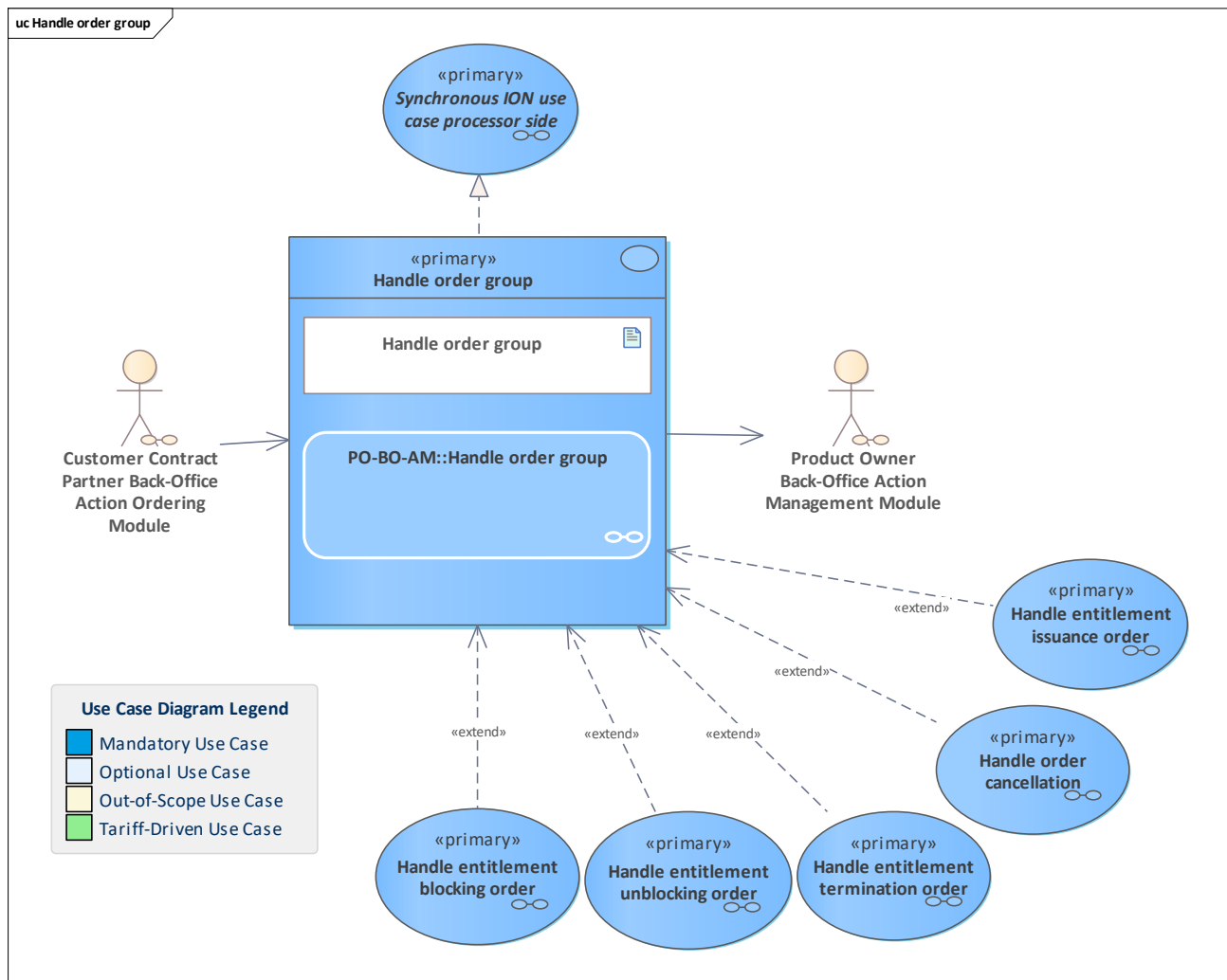


Figure 337: Handle order group

The [Product Owner Back-Office Action Management Module](#) handles an order group. Grouping the orders guarantees that either all grouped orders will be accepted or all grouped orders will be declined.

If the order group passes all checks, the contained orders are added to the order inventory and will be considered for the next action list.

11.167 Handle order obsolescence notification

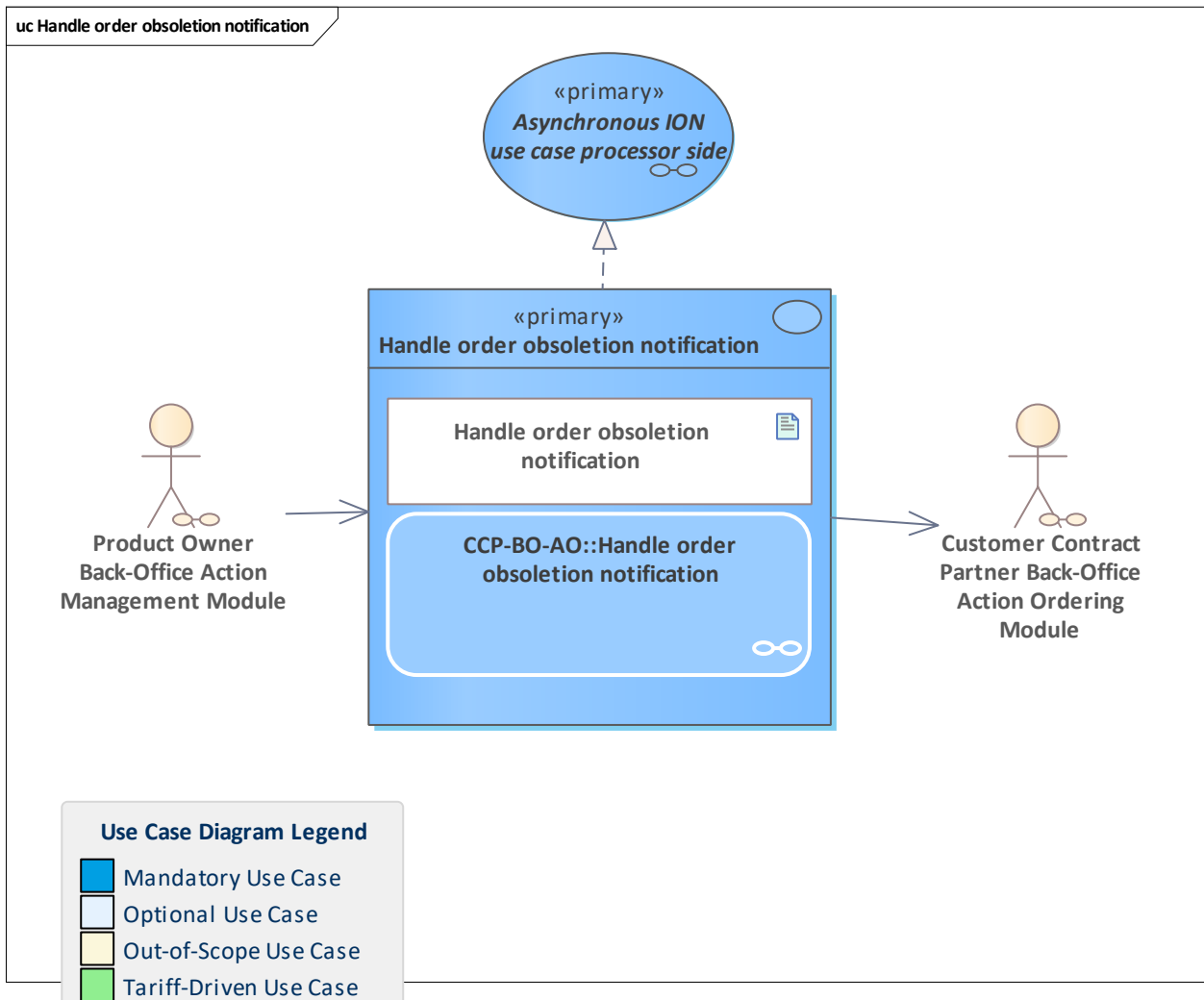


Figure 338: Handle order obsolescence notification

Handle an order obsolescence (see [Obsolete](#)) notification triggered by action list clearing (see [Check for order obsolescence](#)).

11.168 Handle ordered entitlement blocked notification from contractual perspective

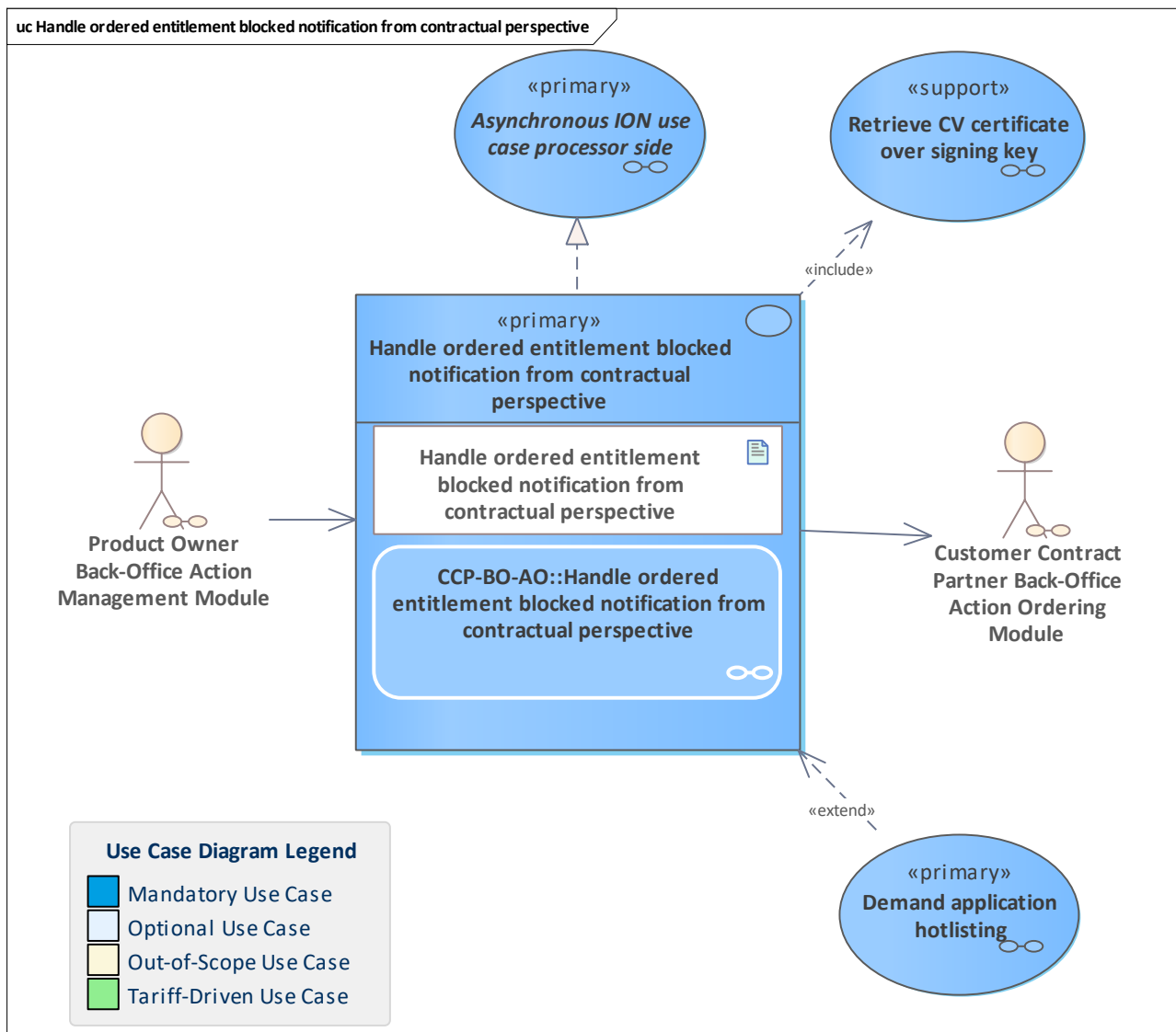


Figure 339: Handle ordered entitlement blocked notification from contractual perspective

Handle the notification about a successful execution of an entitlement blocking ordered via the action management from the contractual perspective.

The ordered entitlement blocked notification is received by the ordering CCP.

The CCP does its checks and monitoring from the contractual perspective regarding the correct execution of the blocking. In this context, the signature of the embedded attestation is verified.

11.169 Handle ordered entitlement blocked notification from operational perspective

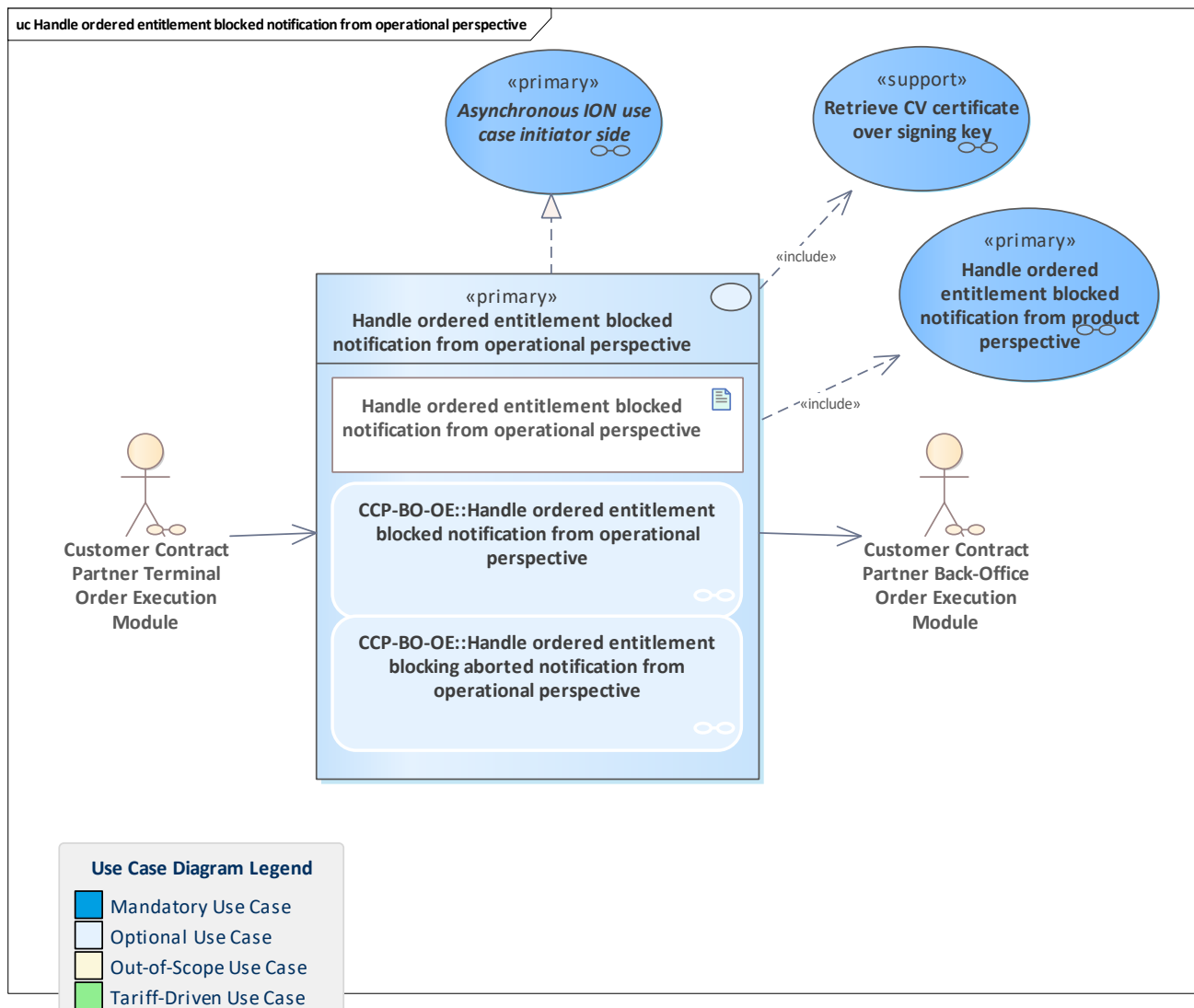


Figure 340: Handle ordered entitlement blocked notification from operational perspective

Handle the notification about a successful execution of an entitlement blocking ordered via the action management from the operational perspective.

The *ordered entitlement blocked notification* is sent by the CCP terminal to the CCP back-office system. The notification will be checked and monitored, e.g. by verifying the signature of the embedded attestation.

The notification will be sent to the PO (and the PO will forward it later to the ordering CCP) In the case action abortion, terminal and SAM action data is sent by the terminal to the back-office system. The CCP back-office system registers the abortion for internal monitoring and data consistency.

11.170 Handle ordered entitlement blocked notification from product perspective

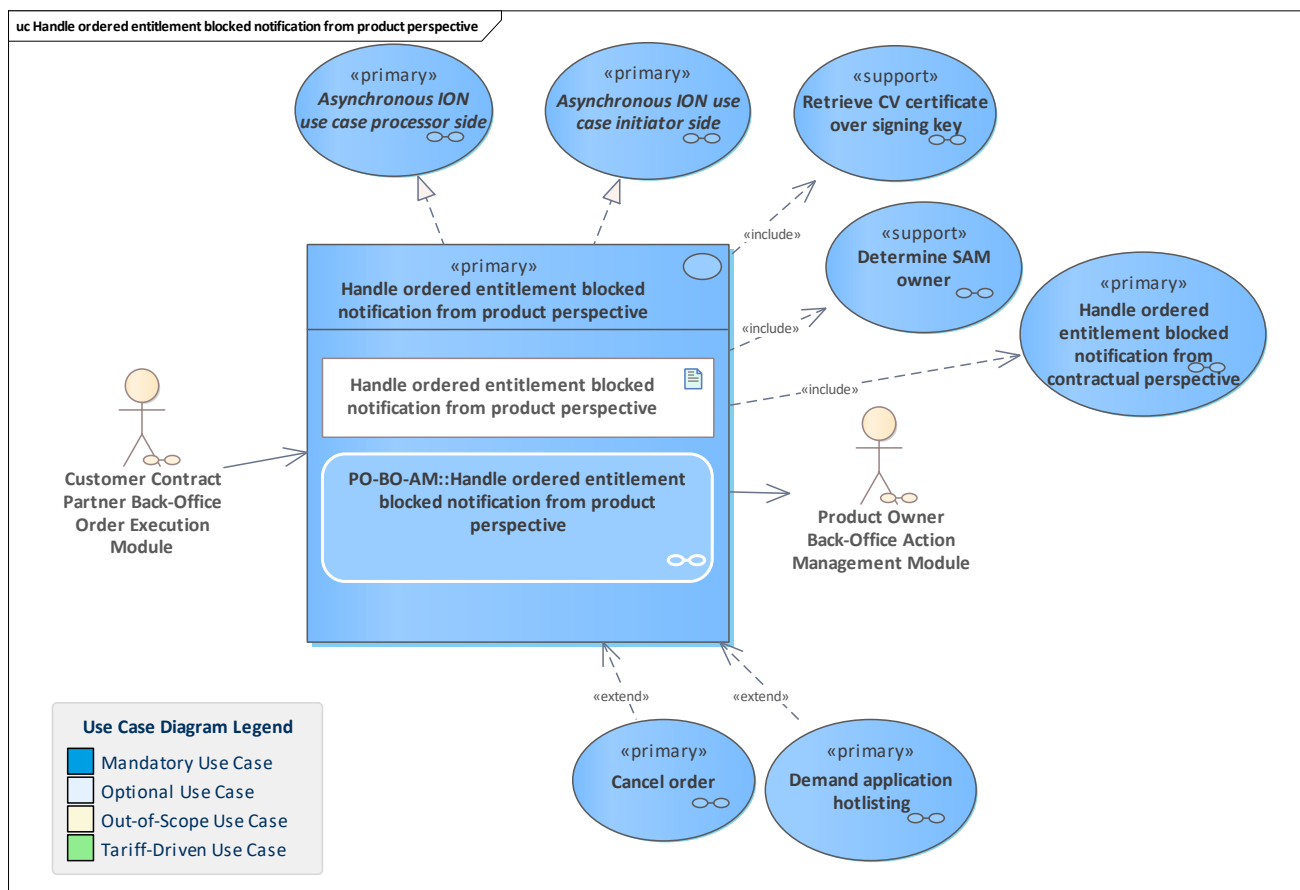


Figure 341: Handle ordered entitlement blocked notification from product perspective

Handle the notification about a successful execution of an entitlement blocking ordered via the action management from product owner perspective.

The ordered entitlement blocked notification is received by the PO.

The PO registers the entitlement blocked notification and does its checks and monitoring from the product perspective regarding the correct execution of the blocking. In this context, the signature of the embedded attestation is verified and the SAM owner of the SAM that performed the action is determined.

Finally, the notification is forwarded to the ordering CCP.

Note: in the ION context, the use case asynchronous as processor (process the notification) and an asynchronous use case as initiator due to the asynchronous call to the ordering CCP.

11.171 Handle ordered entitlement issued notification from contractual perspective

11.172 Handle ordered entitlement issued notification from contractual perspective

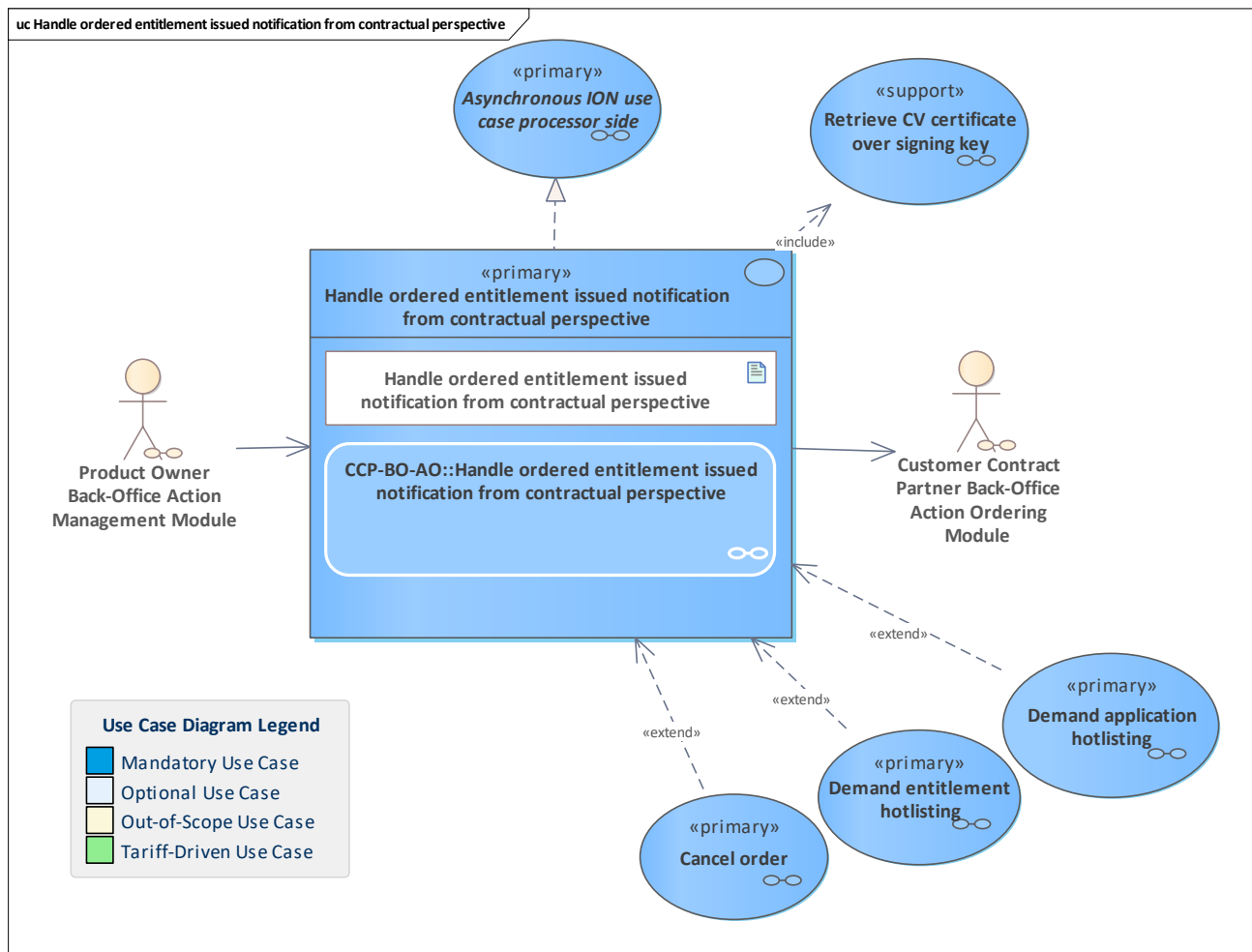


Figure 342: Handle ordered entitlement issued notification from contractual perspective

Handle the notification about a successful execution of an entitlement issuance ordered via action management from the contractual perspective.

After the monitoring, the PO forwards the notification from the executing CCP to the ordering CCP. Thus, the *ordered entitlement issuance notification* is received by the CCP that initiated the ordered action.

The CCP does its checks and monitoring from the contractual perspective regarding the correct execution of the ordered entitlement issuance. In this context, the signature of the embedded attestation is verified.

Note that the application instance ID of the user medium the entitlement was issued to can be uniquely identified via *umAppInstanceId*, which is part of the [SignedEntitlementIssuedAttestation](#) that is contained in the [OrderedEntitlementIssuedNotification](#).

11.173 Handle ordered entitlement issued notification from operational perspective

11.174 Handle ordered entitlement issued notification from operational perspective

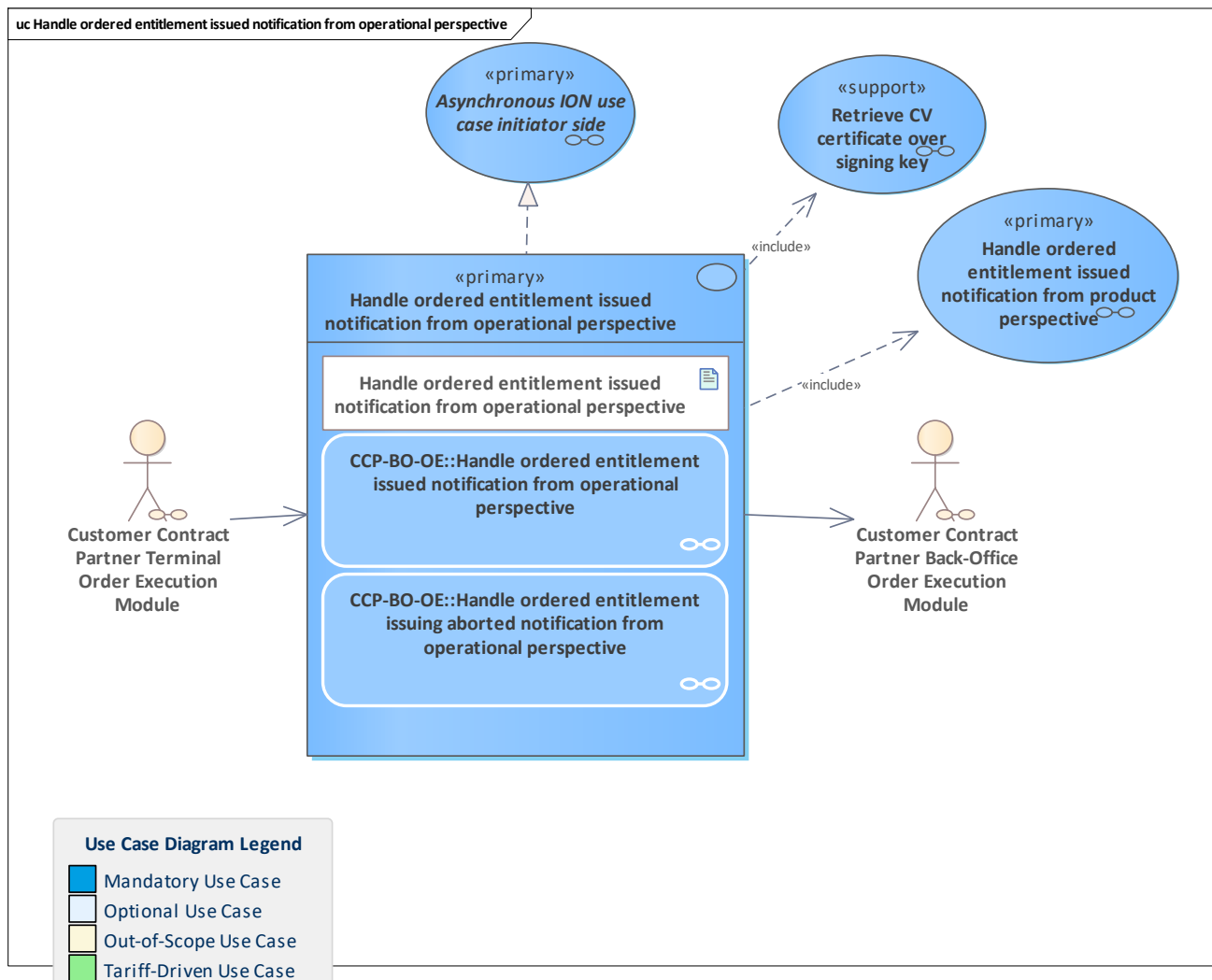


Figure 343: Handle ordered entitlement issued notification from operational perspective

Handle the notification about a successful execution of an entitlement issuance ordered via action management from the operational perspective.

The executing CCP receives the notification about an entitlement issuance caused by an ordered action and registers it.

Then it handles the notification from the operational perspective by performing checks and monitoring, e.g. verifying the signature of the issuance attestation.

Finally, the notification is forwarded to the responsible PO system.

In the case of action abortion, the notification is also sent to the PO system to announce the used SAM- and product issuance counters.

11.175 Handle ordered entitlement issued notification from product perspective

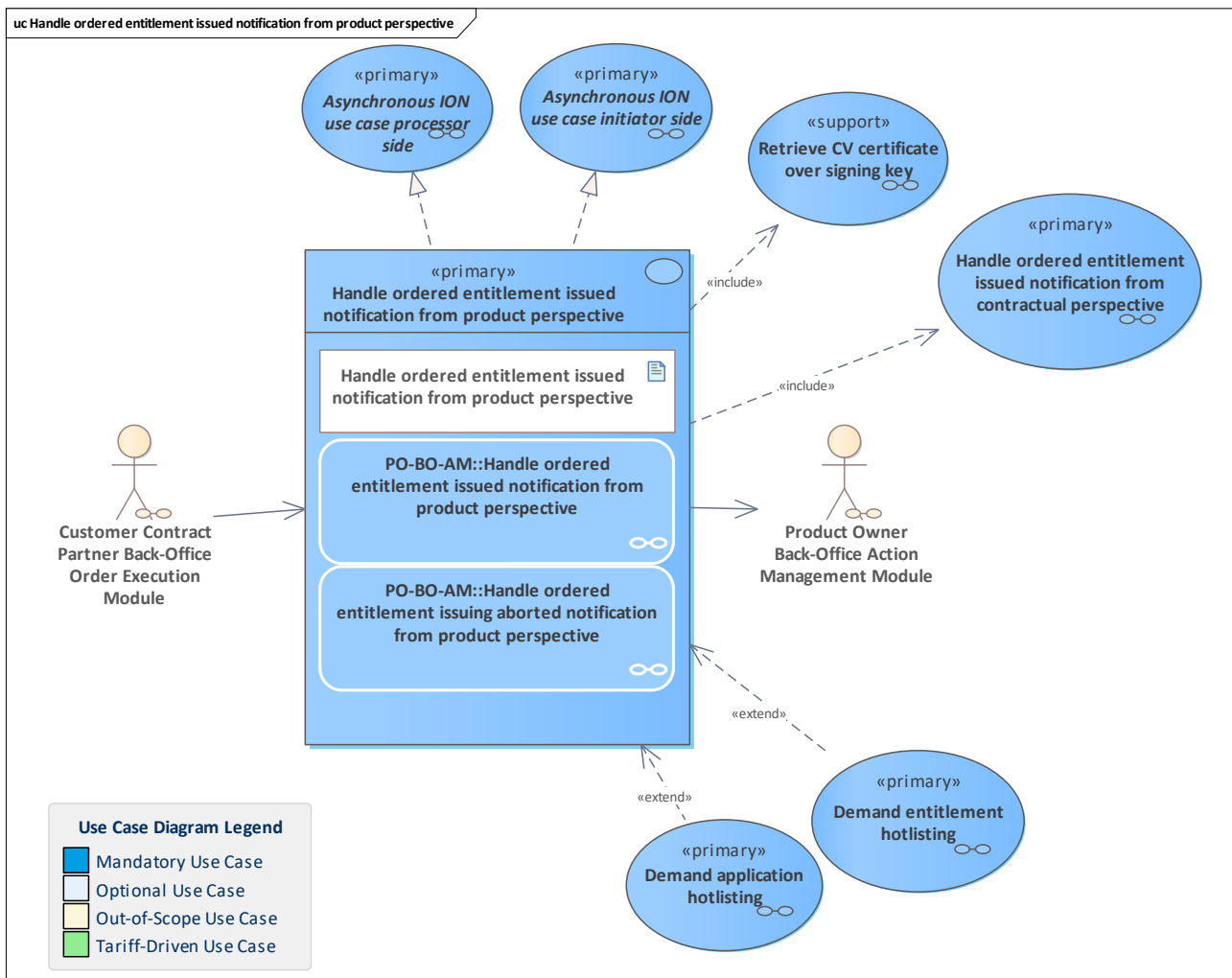


Figure 344: Handle ordered entitlement issued notification from product perspective

Handle the notification about a successful execution of an entitlement issuance ordered via action management from the product owner perspective.

The PO back-office system receives the notification about an ordered entitlement issuance and handles it from the product perspective. It registers the notification and performs checks and monitoring.

Finally, the notification is forwarded to the CCP that initiated the ordered action.

In the case of an abortion notification, the PO system has to register the SAM and product issuance counter for consistent monitoring.

Note that the application instance ID of the user medium the entitlement was issued to can be uniquely identified via *umAppInstanceId*, which is part of the [SignedEntitlementIssuedAttestation](#) that is contained in the [OrderedEntitlementIssuedNotification](#).

11.176 Handle ordered entitlement terminated notification from contractual perspective

11.177 Handle ordered entitlement terminated notification from contractual perspective

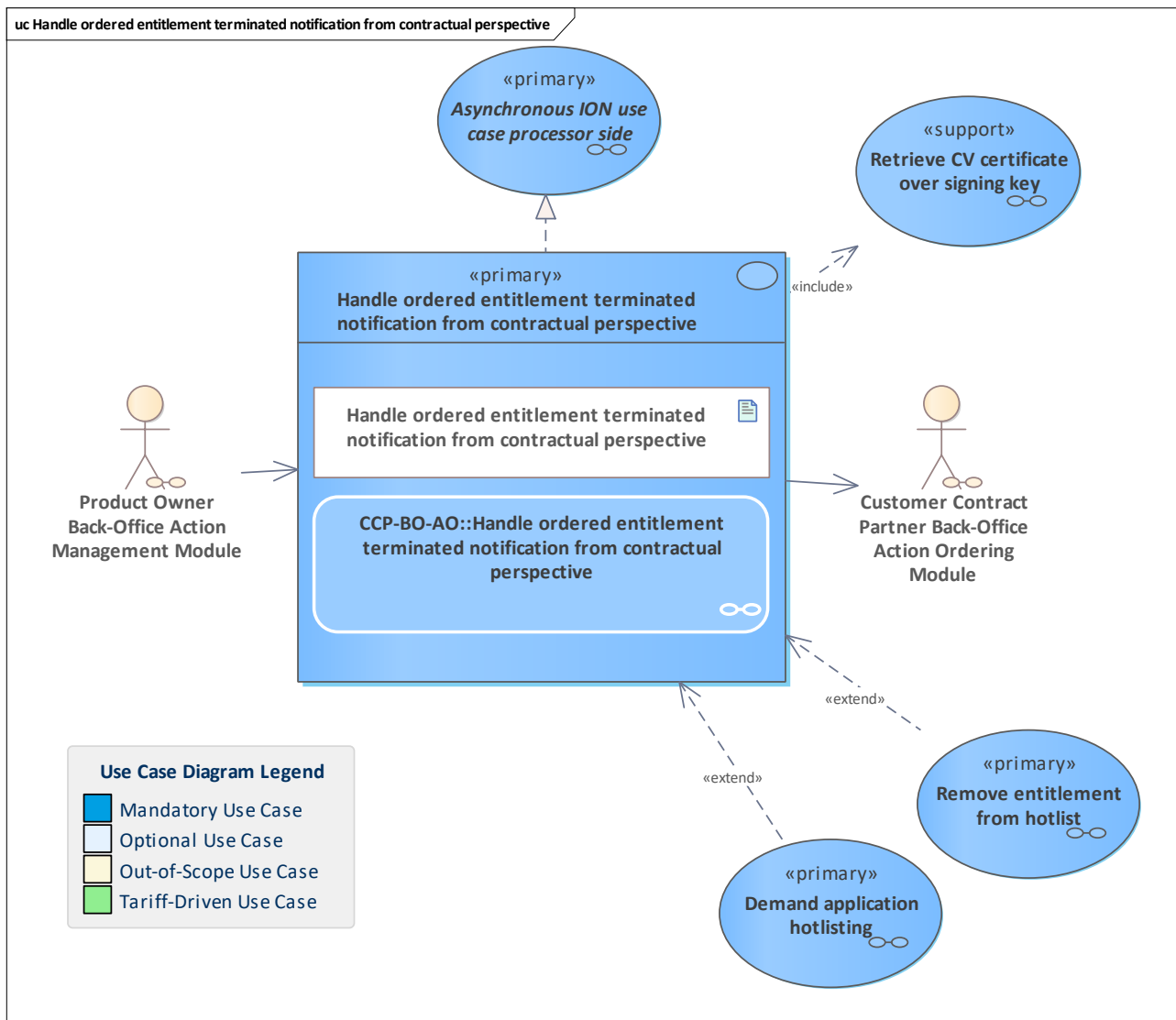


Figure 345: Handle ordered entitlement terminated notification from contractual perspective

Handle the notification about a successful execution of an entitlement termination ordered via the action management from the contractual perspective.

The ordered entitlement terminated notification is received by the ordering CCP.

The CCP does its checks and monitoring from the contractual perspective regarding the correct execution of the termination. In this context, the signature of the termination attestation is verified.

Note: this closes a potentially related user account concerning the entitlement.

If the termination was correct, the pCCP checks if the entitlement has to be removed from the hotlist.

11.178 Handle ordered entitlement terminated notification from operational perspective

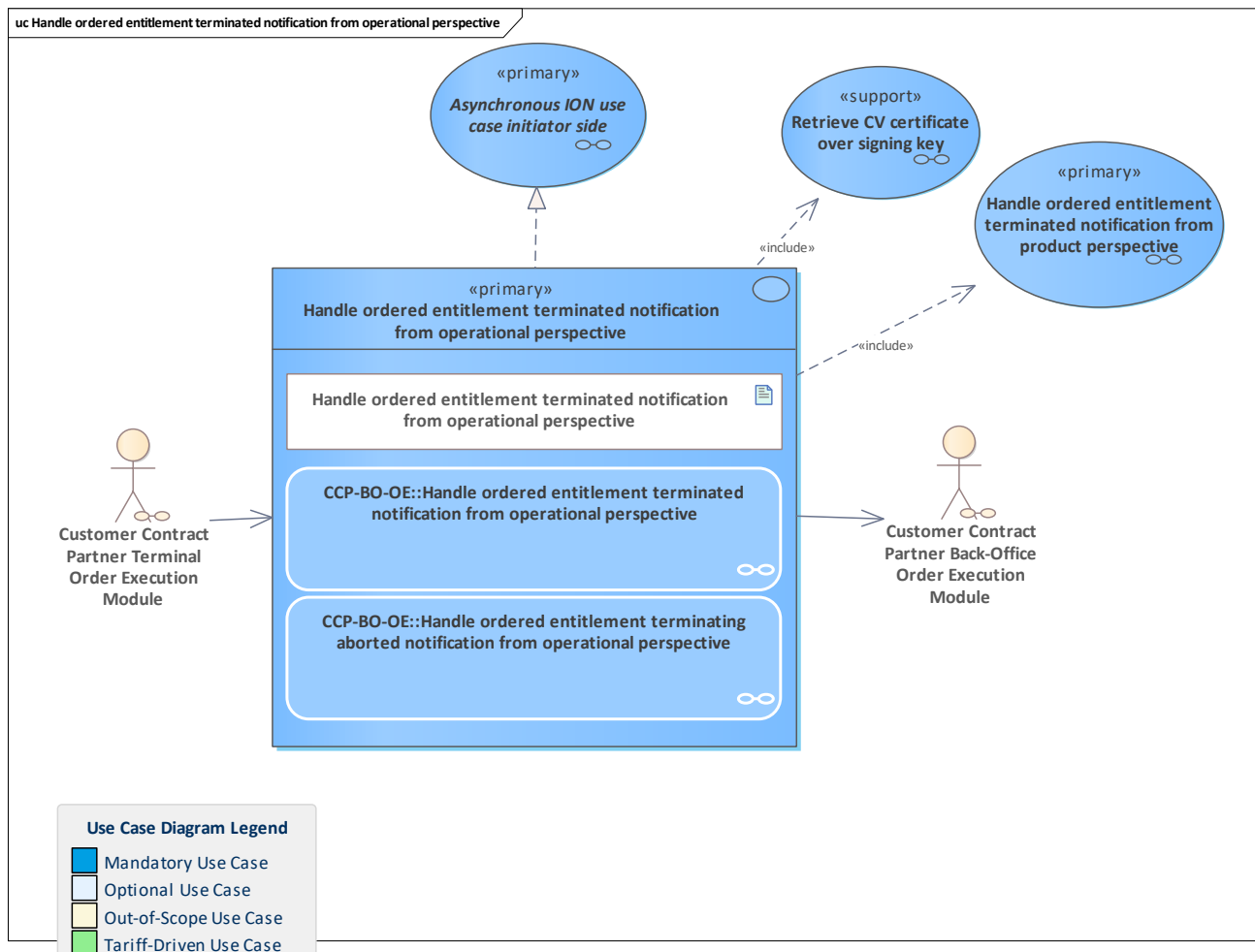


Figure 346: Handle ordered entitlement terminated notification from operational perspective

Handle the notification about a successful execution of an entitlement termination ordered via action management from the operational perspective.

The entitlement terminated notification is sent by the CCP terminal to the back-office system of the executing CCP. The notification will be checked and monitored, e.g. by verifying the signature of the embedded attestation.

The notification will be sent to the PO (and the PO will forward it later to the ordering CCP).

In the case of action abortion, terminal and SAM action data is sent by the terminal to the back-office system. The CCP back-office system registers the abortion for internal monitoring and data consistency.

11.179 Handle ordered entitlement terminated notification from product perspective

11.180 Handle ordered entitlement terminated notification from product perspective

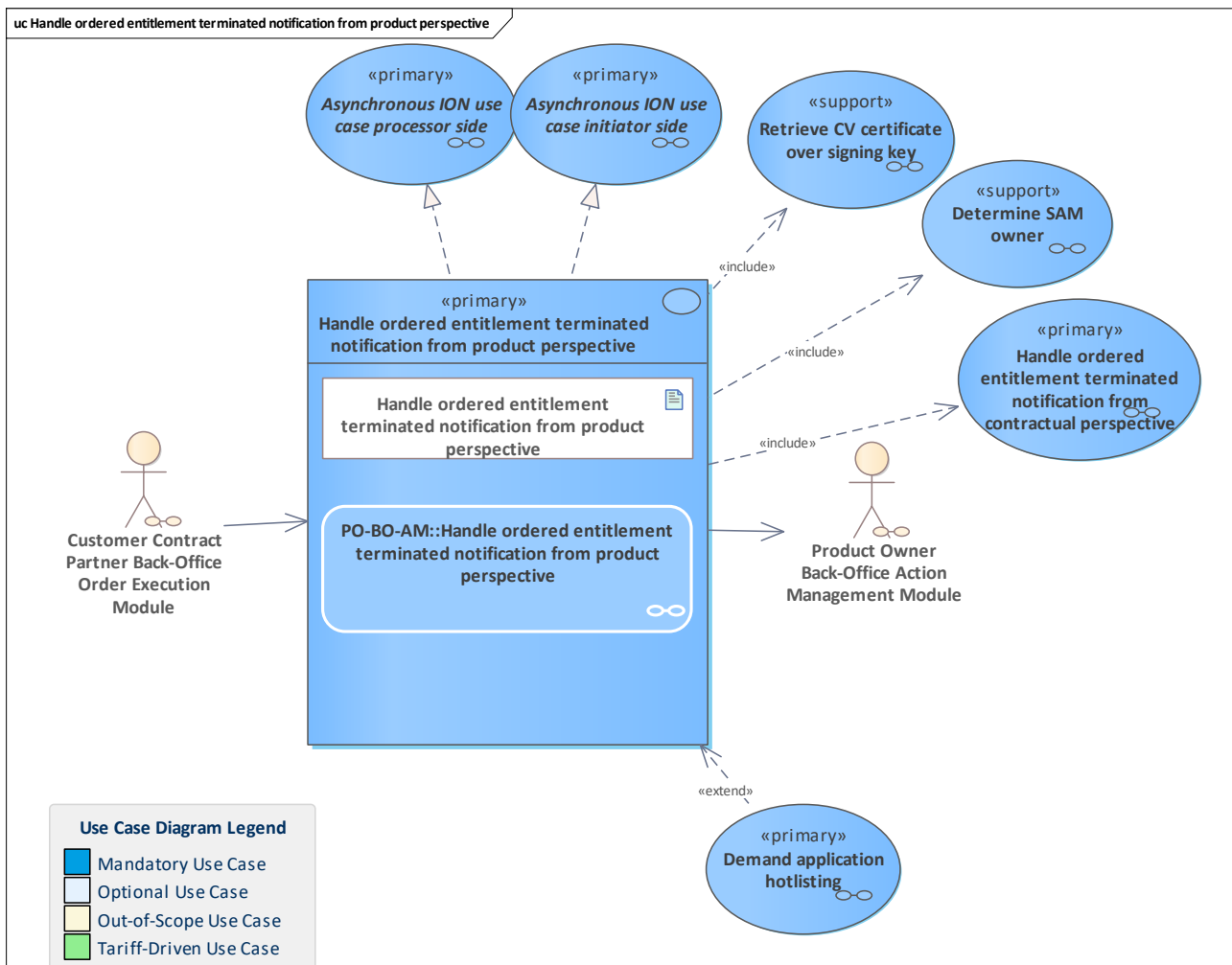


Figure 347: Handle ordered entitlement terminated notification from product perspective

Handle the notification about a successful execution of an entitlement termination ordered via action management from the product owner perspective.

The ordered entitlement terminated notification is received by the PO.

The PO registers the entitlement terminated notification and does its checks and monitoring from the product perspective regarding the correct execution of termination. In this context, the signature of the termination attestation is verified and the SAM owner of the SAM that performed the termination is determined.

Finally, the notification is forwarded to the ordering CCP.

Note: in the ION context, the use case is asynchronous as processor (process the notification) and an asynchronous use case as initiator due to the asynchronous call to the ordering CCP.

11.181 Handle ordered entitlement unblocked notification from contractual perspective

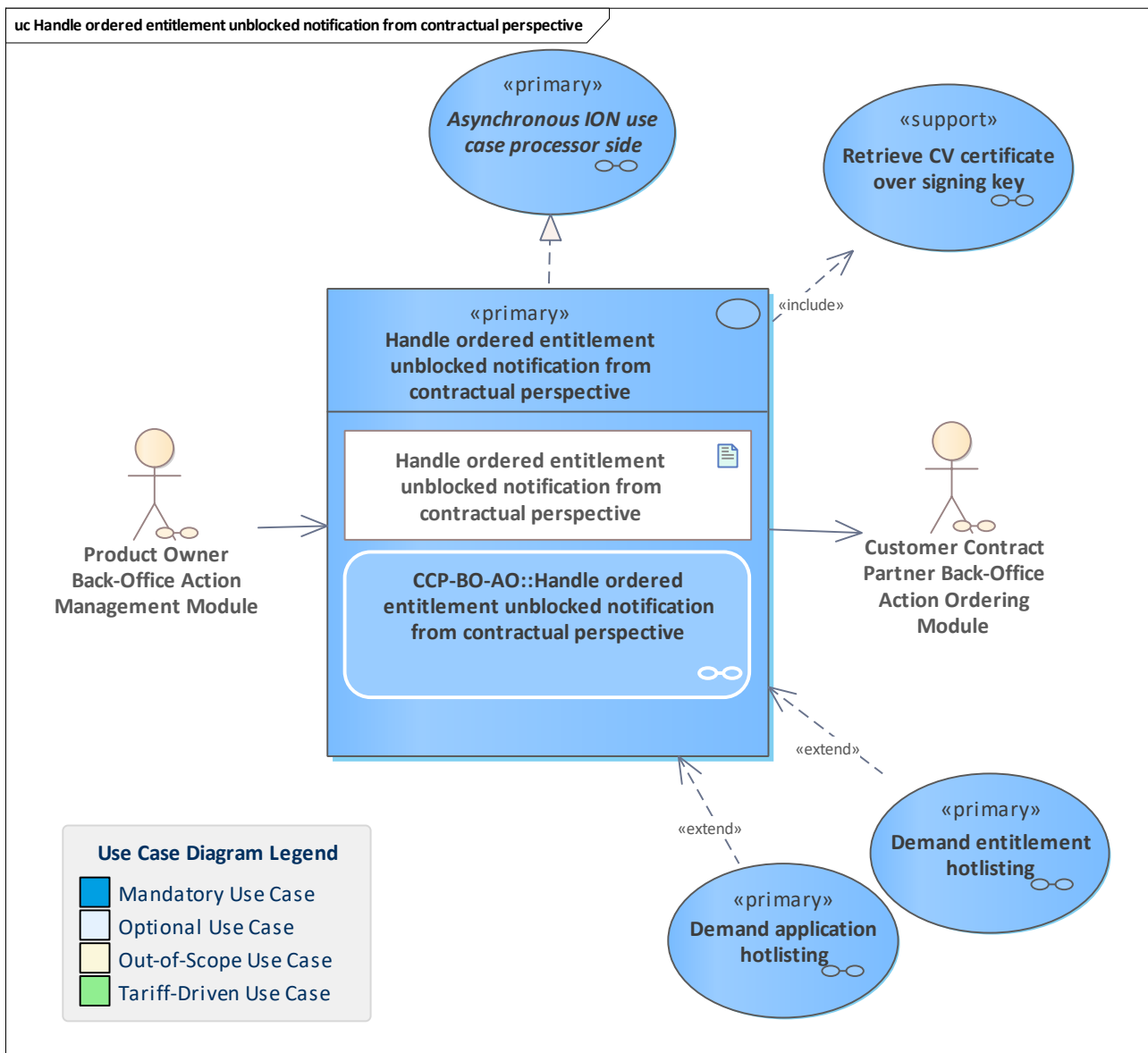


Figure 348: Handle ordered entitlement unblocked notification from contractual perspective

Handle the notification about a successful execution of an entitlement unblocking ordered via action management from the contractual perspective.

The ordered entitlement unblocked notification is received by the ordering CCP.

The CCP does its checks and monitoring from the contractual perspective regarding the correct execution of the unblocking. In this context, the signature of the embedded attestation is verified.

11.182 Handle ordered entitlement unblocked notification from operational perspective

11.183 Handle ordered entitlement unblocked notification from operational perspective

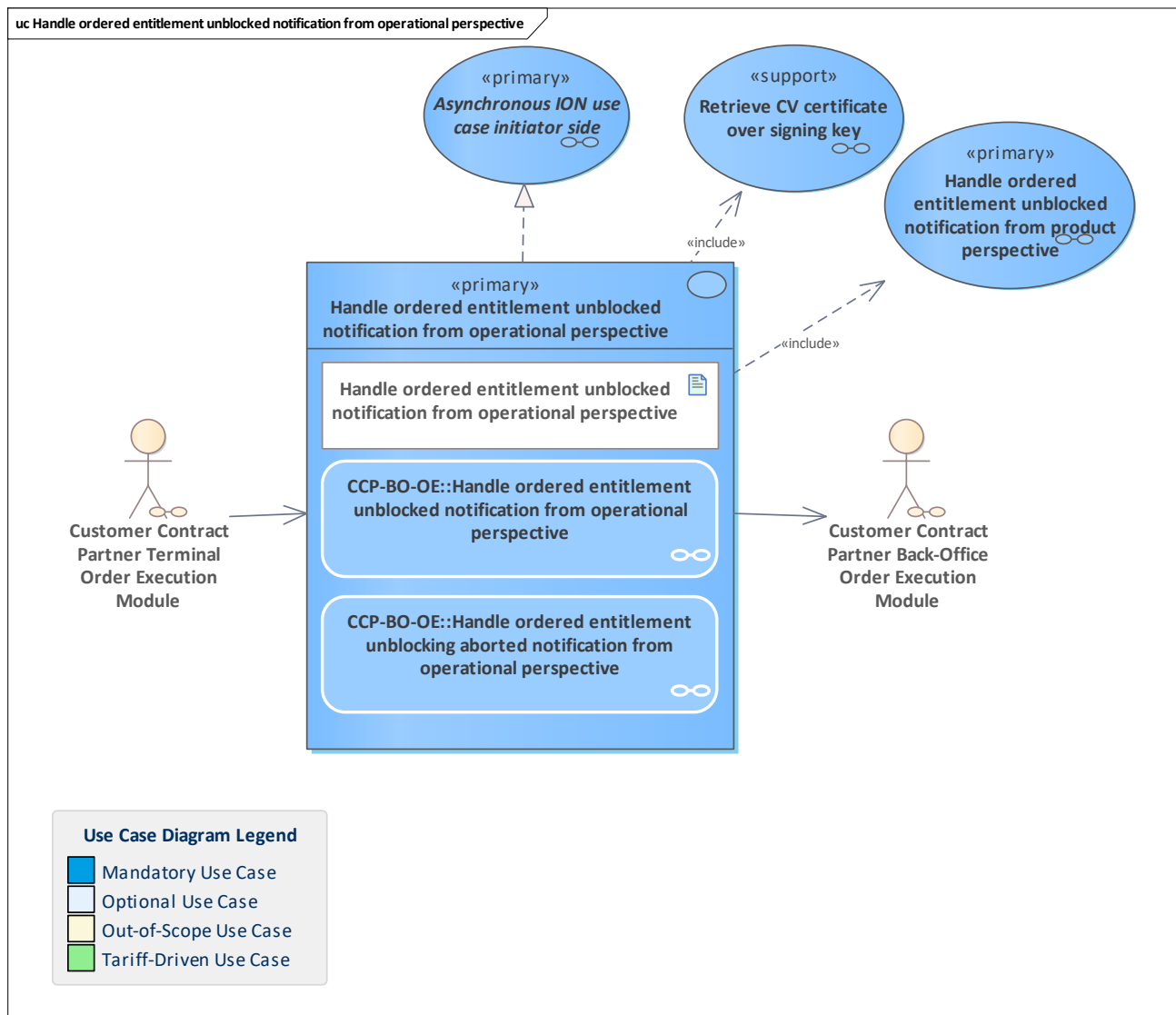


Figure 349: Handle ordered entitlement unblocked notification from operational perspective

Handle the notification about a successful execution of an entitlement unblocking ordered via action management from the operational perspective.

The ordered entitlement unblocked notification is sent by the CCP terminal to the CCP back-office system. The notification will be checked and monitored, e.g. by verifying the signature of the embedded attestation.

The notification will be sent to the PO (and the PO will forward it later to the ordering CCP).

In the case of action abortion, terminal and SAM action data is sent by the terminal to the back-office system. The CCP back-office system registers the abortion for internal monitoring and data consistency.

11.184 Handle ordered entitlement unblocked notification from product perspective

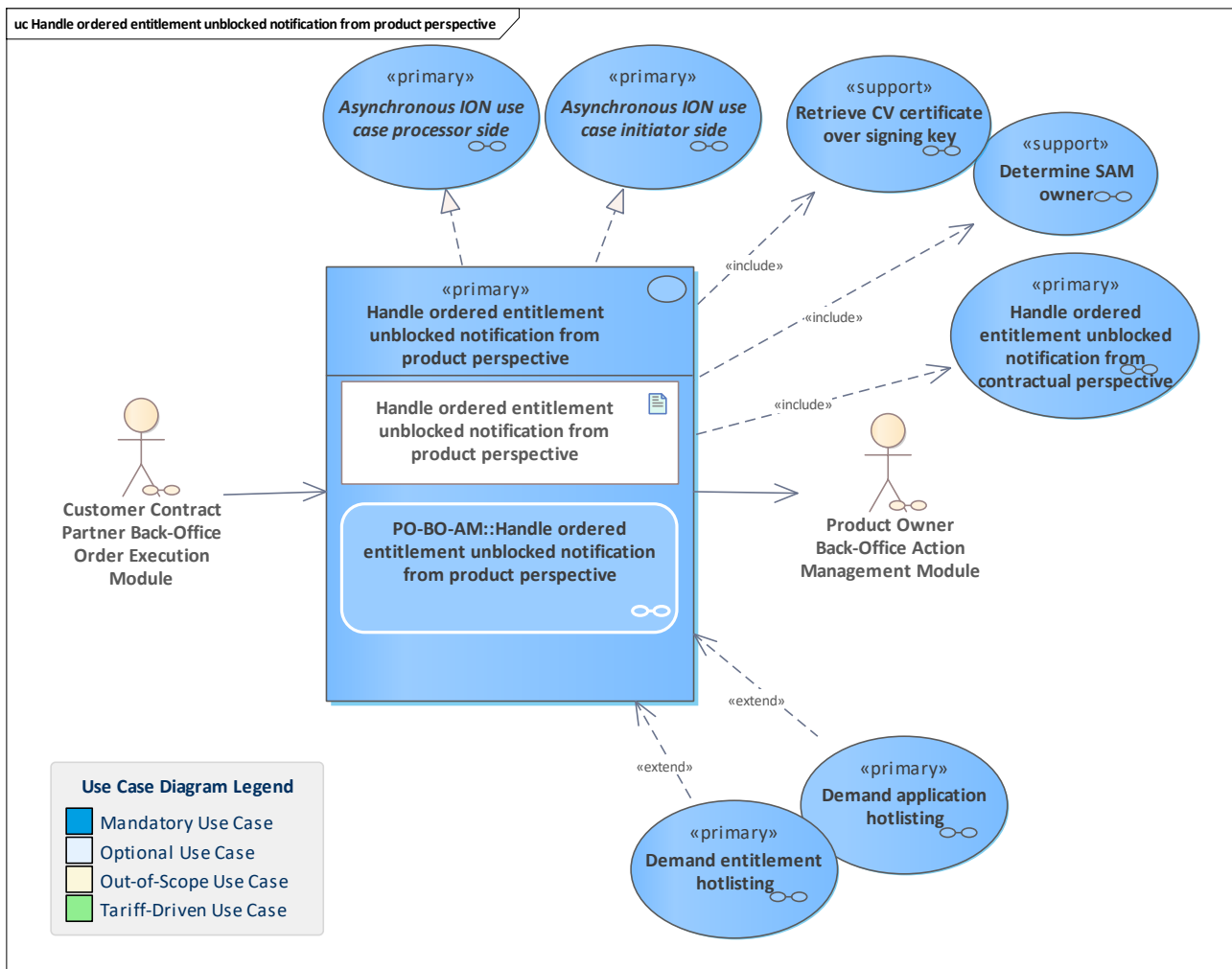


Figure 350: Handle ordered entitlement unblocked notification from product perspective

Handle the notification about a successful execution of an entitlement unblocking ordered via action management from the product owner perspective.

The ordered entitlement unblocked notification is received by the PO.

The PO registers the entitlement unblocked notification and does its checks and monitoring from the product perspective regarding the correct execution of the blocking. In this context, the signature of the embedded attestation is verified and the SAM owner of the SAM that performed the action is determined.

Finally, the notification is forwarded to the ordering CCP.

Note: in the ION context, the use case is asynchronous as processor (process the notification) and an asynchronous use case as initiator due to the asynchronous call to the ordering CCP.

11.185 Handle organisation hotlisting demand

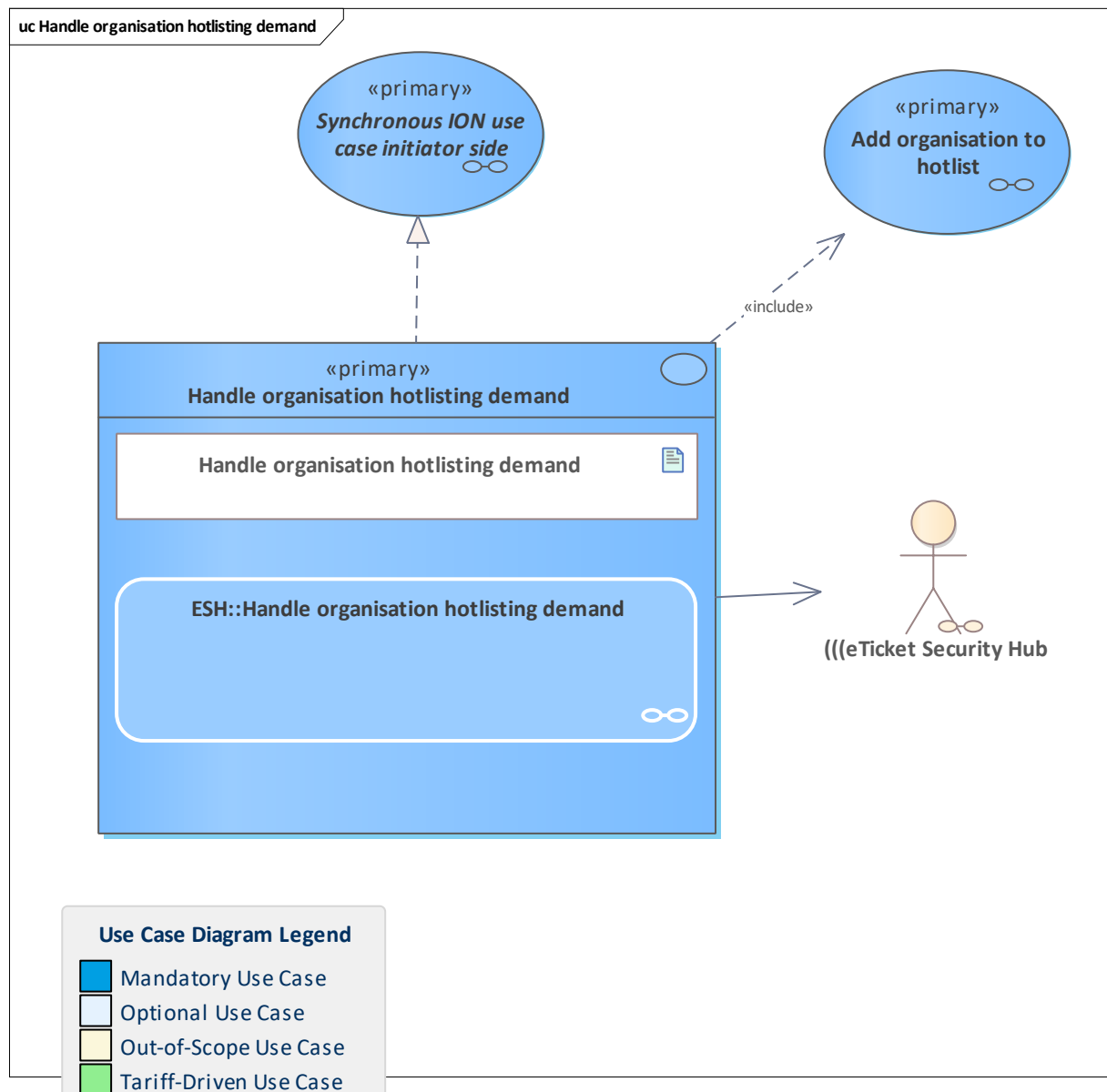


Figure 351: Handle organisation hotlisting demand

The demand for hotlisting an organisation has been received over the VDV-ETS service management or the organisation has to be hotlisted for other reasons. If the organisation must be hotlisted, then this use case can be started.

In this use case, the request for adding the organisation to the hotlist is created and sent to the hotlist service system by the [\(\(\(eTicket Security Hub](#) (ESH) of the [Scheme Manager](#). The result will be updated in the ESH.

Note: hotlisting an organisation has a huge impact on the (((etiCORE environment, the SAMs of this organisation also have to be hotlisted. All involved applications and entitlements will be blocked by a terminal if the linked organisation ID is found on the hotlist.

11.186 Handle request for product acceptance configuration list

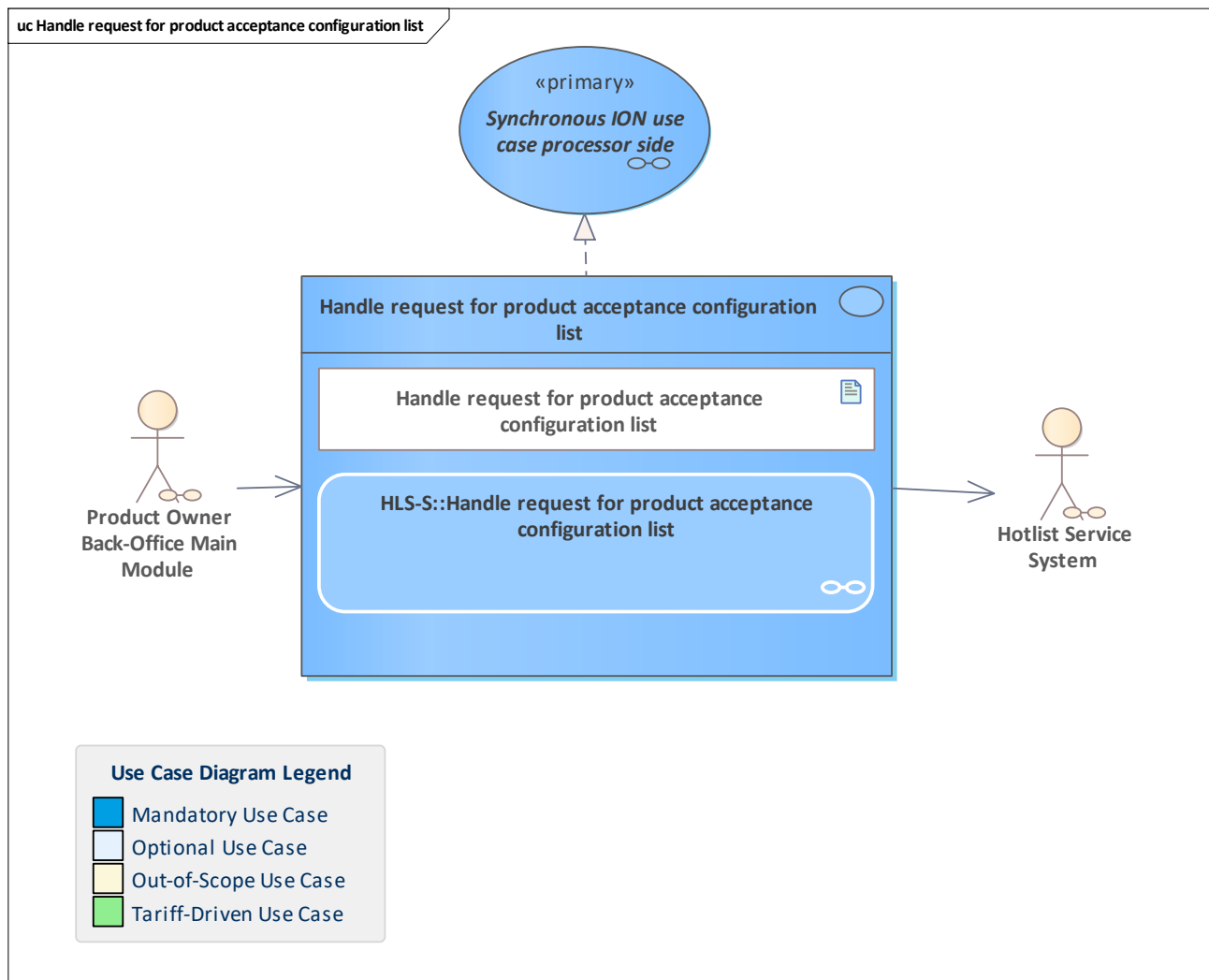


Figure 352: Handle request for product acceptance configuration list

The hotlist service system provides its product acceptance configuration as a compressed list of product acceptance entries.

11.187 Handle request to add product acceptance entry to hotlist service configuration

11.188 Handle request to add product acceptance entry to hotlist service configuration

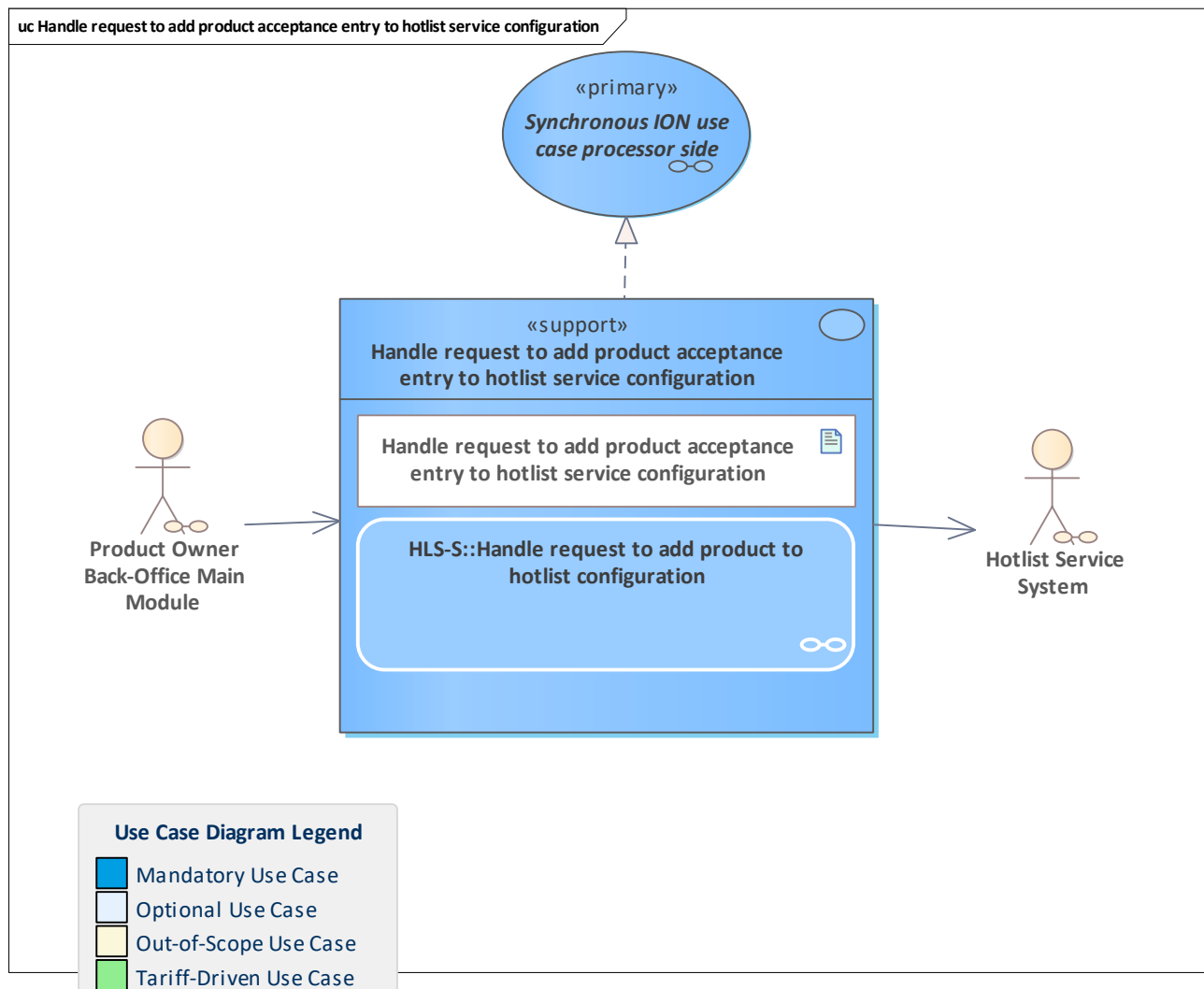


Figure 353: Handle request to add product acceptance entry to hotlist service configuration

The hotlist service system receives a request to add a product acceptance entry to its configuration. Only product owners are allowed to add an acceptance configuration for their own products. This configuration is used to filter product-based hotlist entries to only consider products that are accepted by the receiving organisation.

Please note that interoperable products are not allowed to be in the configuration list.

11.189 Handle request to determine SAM owner

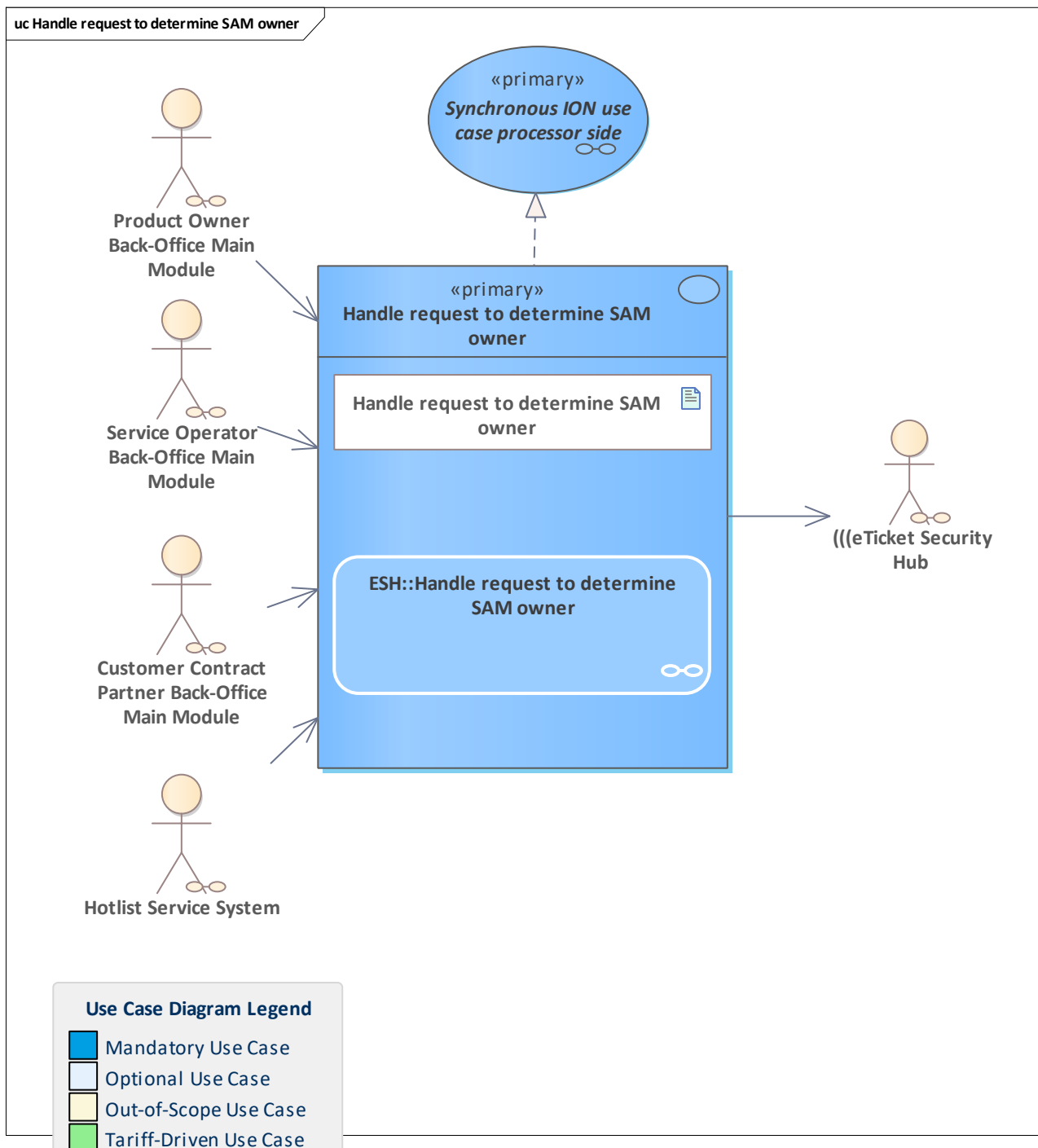


Figure 354: Handle request to determine SAM owner

Provide the organisation ID and role of the SAM owner for a given SAM ID.

11.190 Handle request to determine UM app instance ID for Medium ID

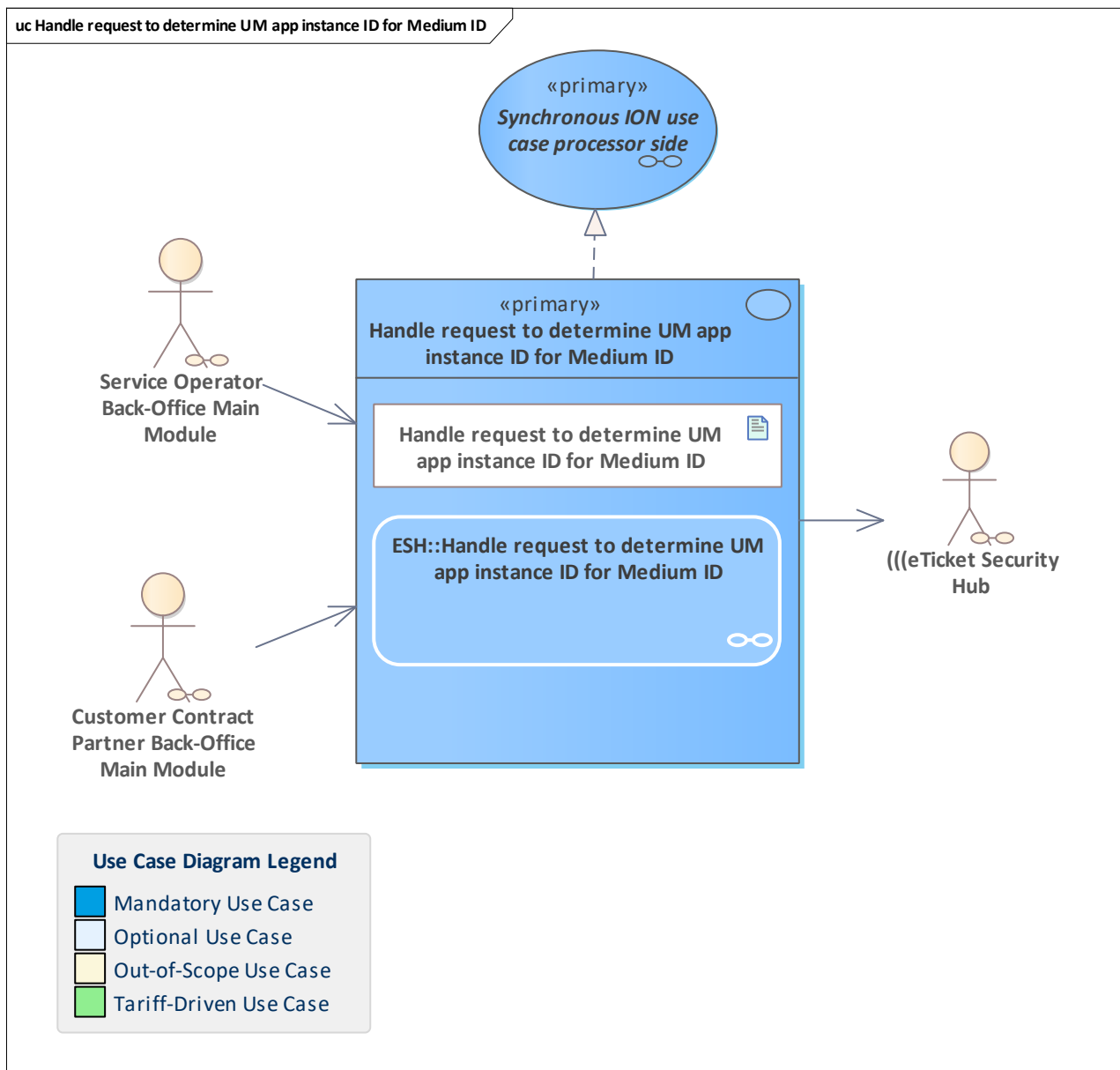


Figure 355: Handle request to determine UM app instance ID for Medium ID

Provides the user medium application instance ID for a given medium ID.

11.191 Handle request to remove product acceptance entry from hotlist service configuration

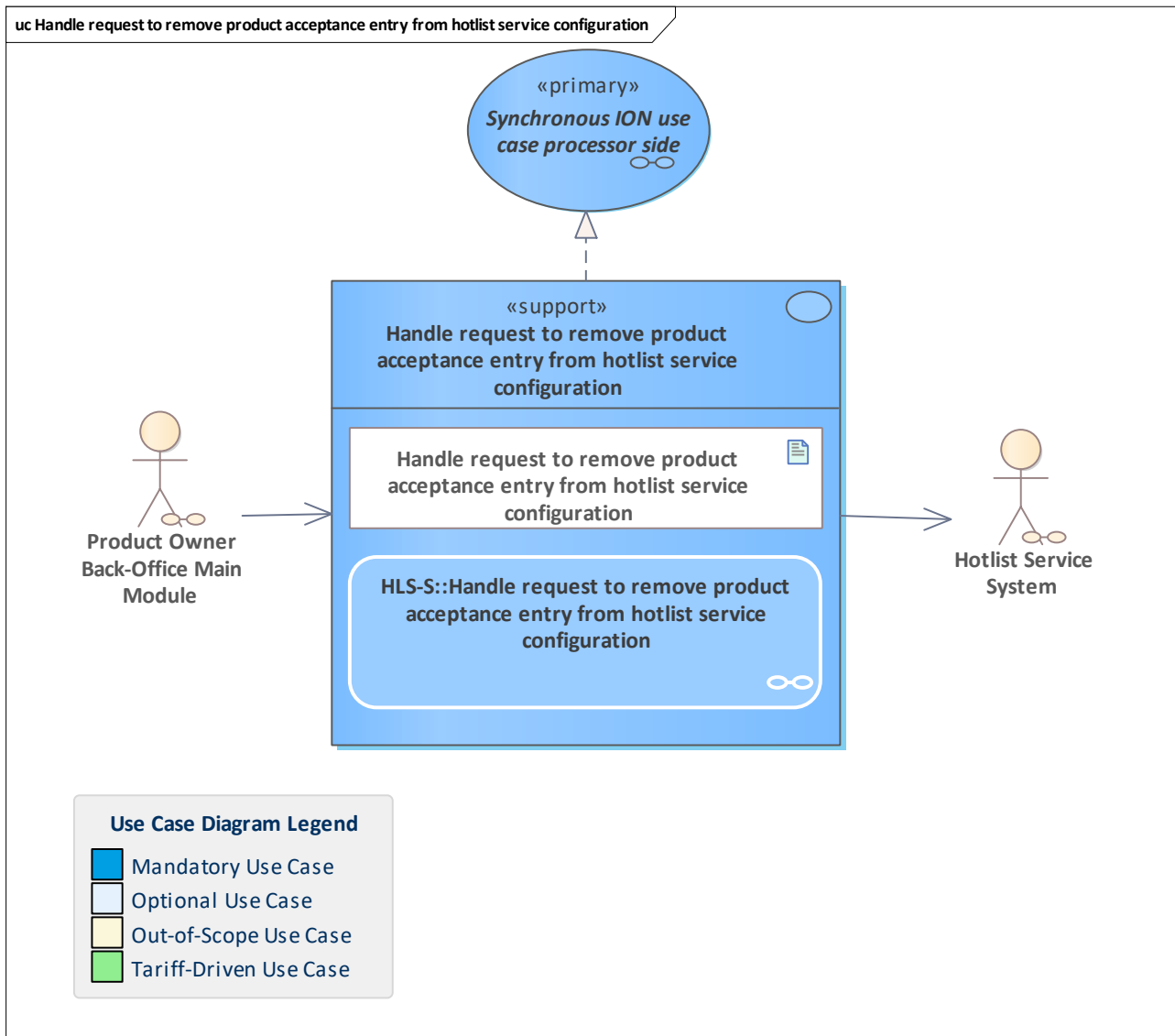


Figure 356: Handle request to remove product acceptance entry from hotlist service configuration

The hotlist service system receives a request to remove a product acceptance entry from its configuration.

Only product owners are allowed to remove an acceptance configuration for their own products. This configuration is used to filter product-based hotlist entries to only consider products that are accepted by the receiving organisation.

11.192 Handle request to remove product acceptance from participants

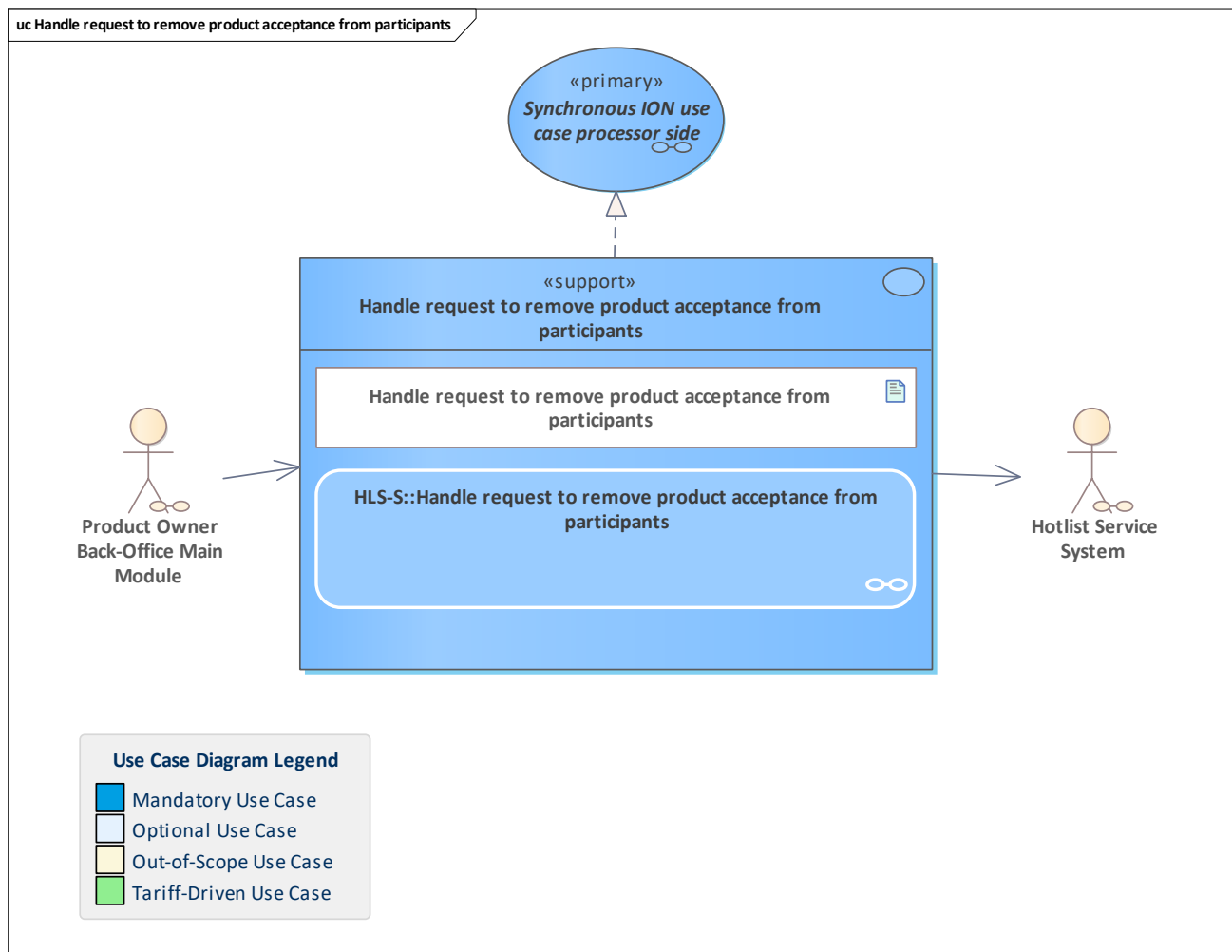


Figure 357: Handle request to remove product acceptance from participants

The hotlist service system receives a request to remove all acceptance entries for a product from its configuration for all acceptance candidates. Only product owners are allowed to remove an acceptance configuration for their own products. The request contains a product expiry date which must be taken into account when handling the request.

11.193 Handle retrieval request for action list

11.194 Handle retrieval request for action list

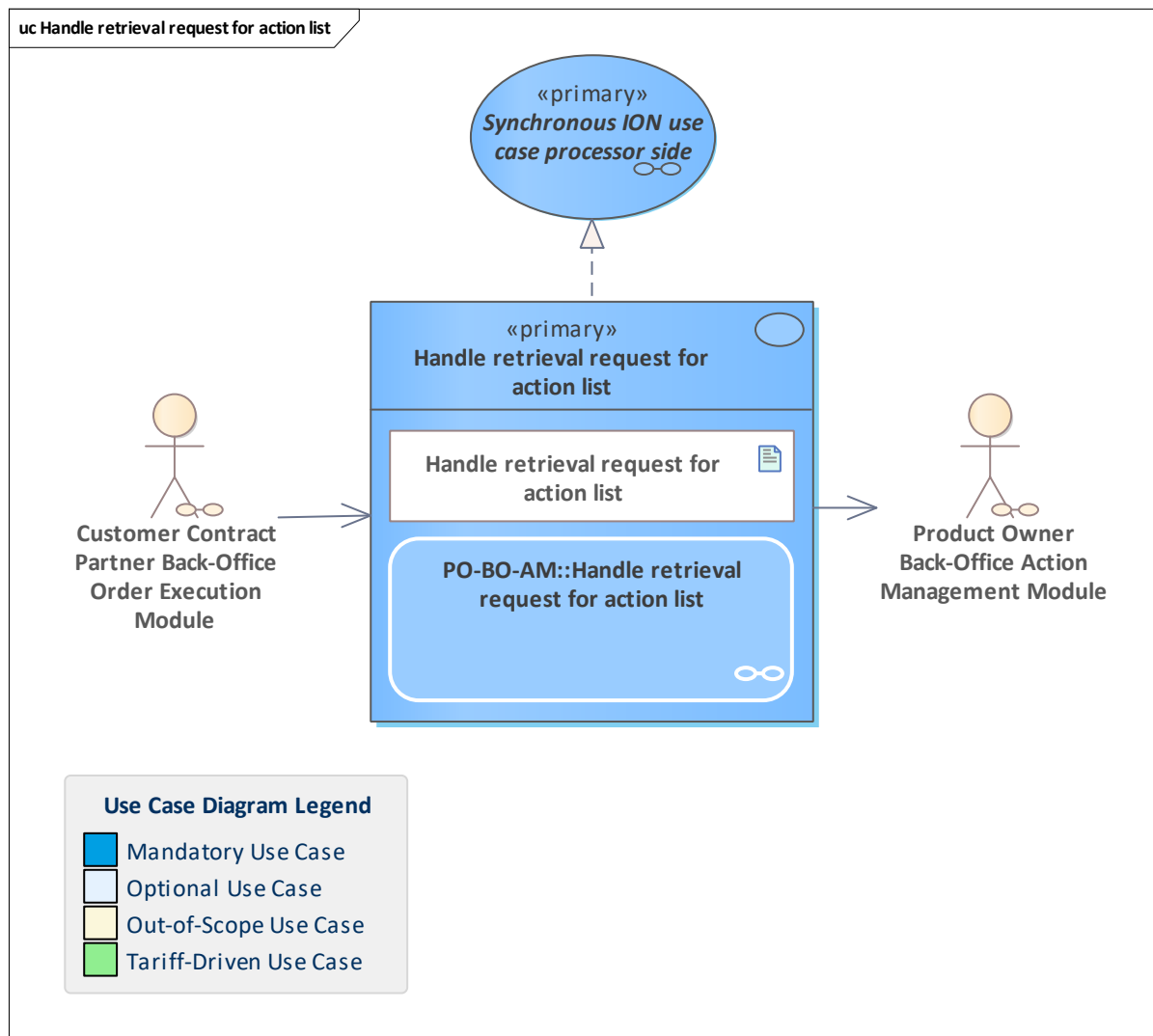


Figure 358: Handle retrieval request for action list

The [Product Owner Back-Office Action Management Module](#) provides the current full action list for the [Customer Contract Partner Back-Office Order Execution Module](#).

11.195 Handle retrieval request for application hotlist

11.196 Handle retrieval request for application hotlist

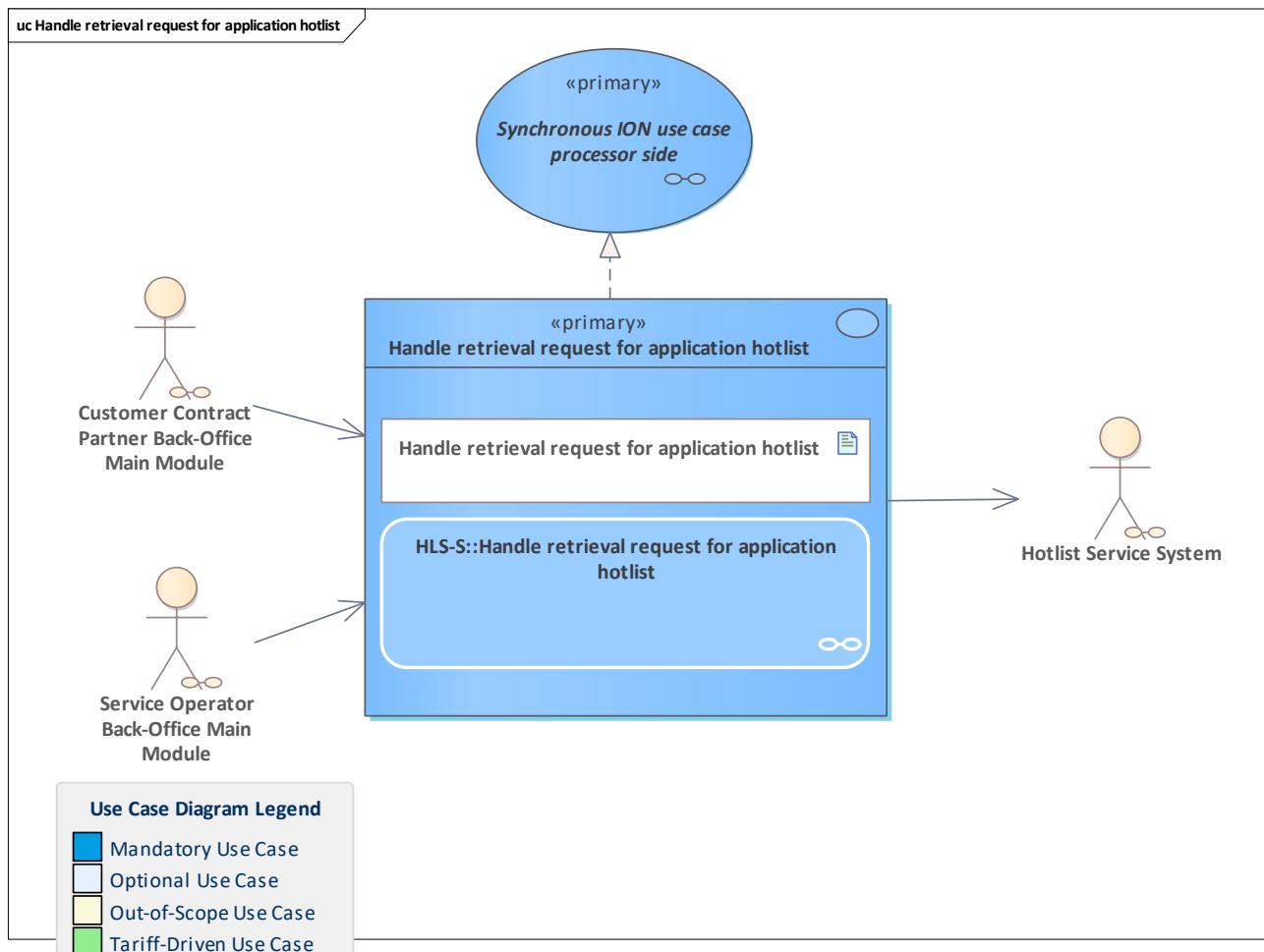


Figure 359: Handle retrieval request for application hotlist

After receiving a request for an application hotlist, the hotlist service system processes the request by running checks (see activity diagram) and creating the current application hotlist as a response.

11.197 Handle retrieval request for authentication key hotlist

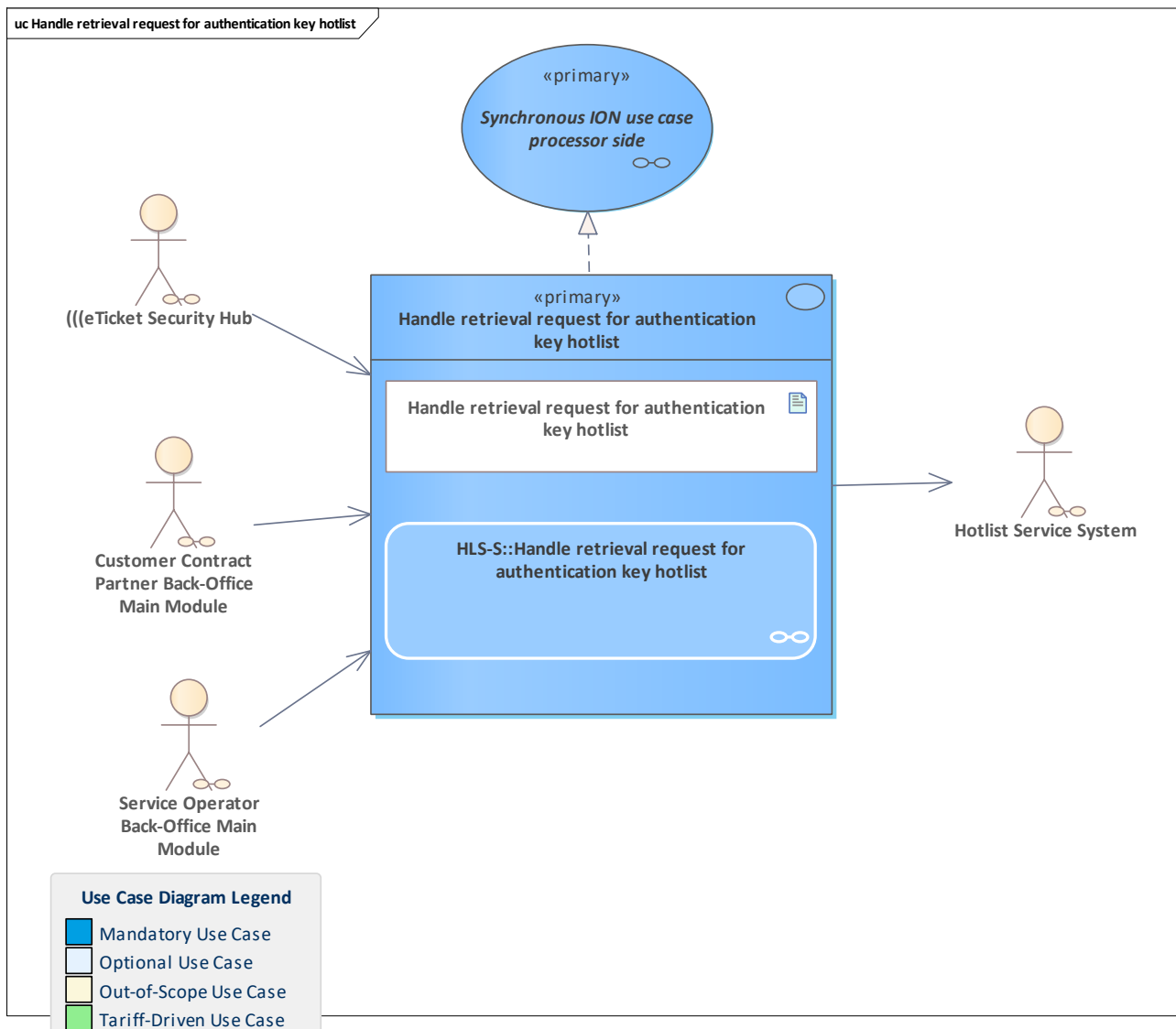


Figure 360: Handle retrieval request for authentication key hotlist

After receiving a request for an authentication key hotlist, the hotlist service system processes the request by running checks (see activity diagram) and creating the current authentication key hotlist as a response.

11.198 Handle retrieval request for entitlement hotlist

11.199 Handle retrieval request for entitlement hotlist

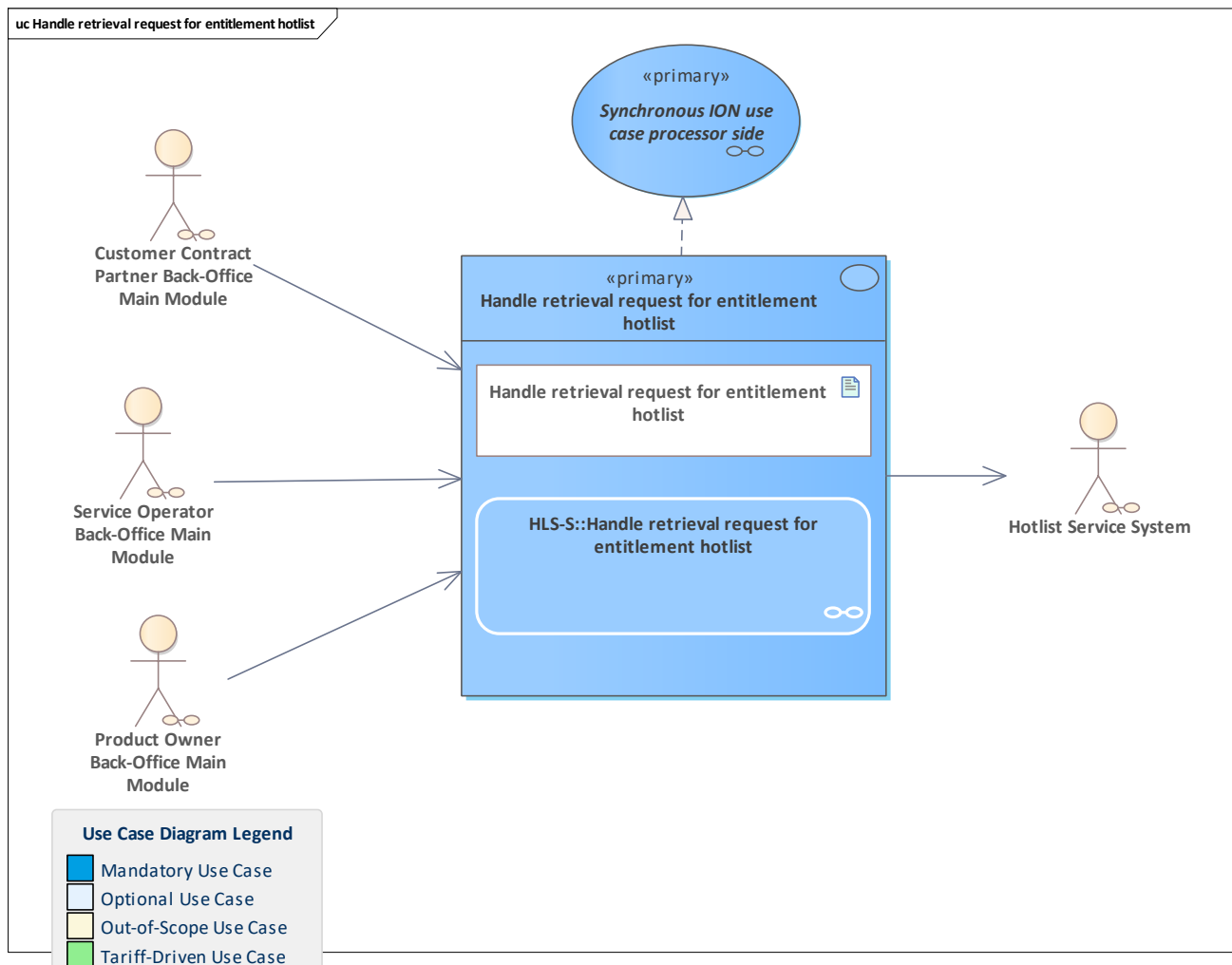


Figure 361: Handle retrieval request for entitlement hotlist

After receiving a request for an entitlement hotlist, the hotlist service system processes the request by running checks (see activity diagram) and creating the current entitlement hotlist as a response.

11.200 Handle retrieval request for entitlement hotlist with product information

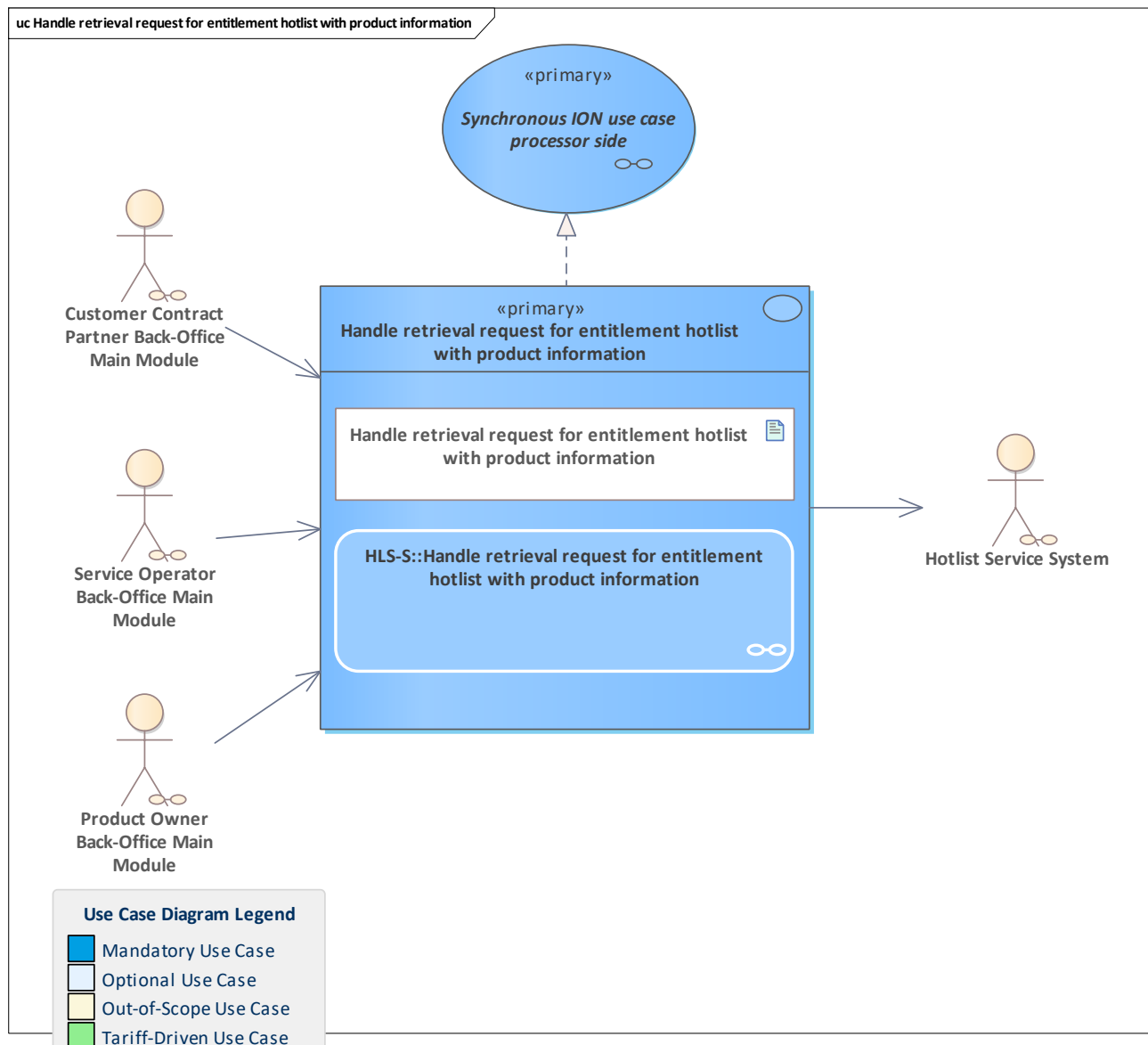


Figure 362: Handle retrieval request for entitlement hotlist with product information

After receiving a request for an entitlement hotlist with product information, the hotlist service system processes the request by running checks (see activity diagram) and creating the current entitlement hotlist with product information as a response. In contrast to [Handle retrieval request for entitlement hotlist](#), each hotlist entry contains additional information about the product (see [ProductId](#)).

11.201 Handle retrieval request for incremental action list

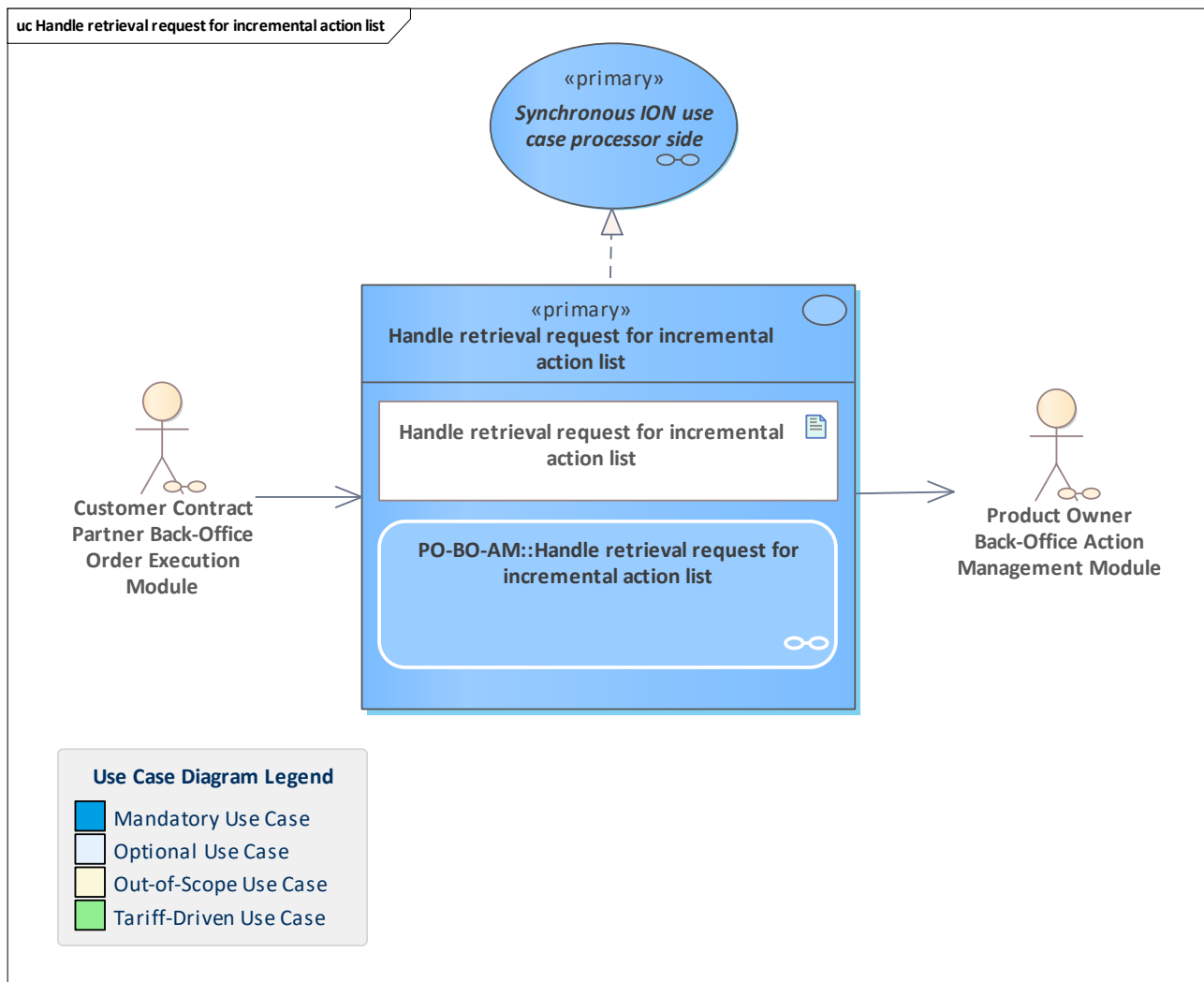


Figure 363: Handle retrieval request for incremental action list

The [Product Owner Back-Office Action Management Module](#) provides the incremental action list covering the cycles starting with the given cycle up to the latest cycle for the [Customer Contract Partner Back-Office Order Execution Module](#).

11.202 Handle retrieval request for incremental application hotlist

11.203 Handle retrieval request for incremental application hotlist

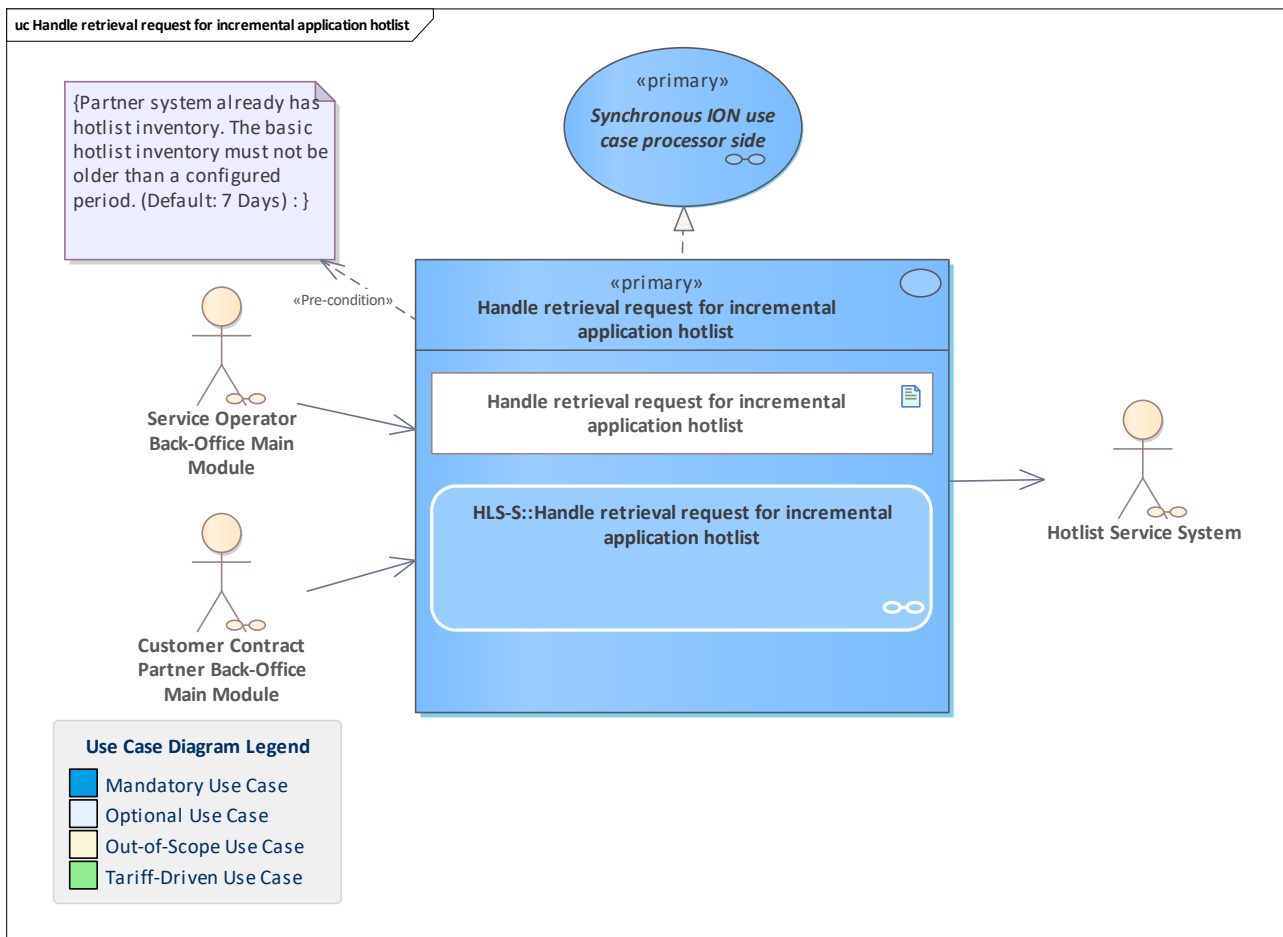


Figure 364: Handle retrieval request for incremental application hotlist

After receiving a request for an application incremental hotlist, the hotlist service system processes the request by running checks (see activity diagram) and creating the application incremental hotlist as a response, which contains changes for relevant applications from the given cycle number until the current cycle.

Note: For performance reasons, incremental hotlists may contain additions and removals that cancel each other out. Thus, the increments need to be applied in the order they appear on the list.

11.204 Handle retrieval request for incremental entitlement hotlist

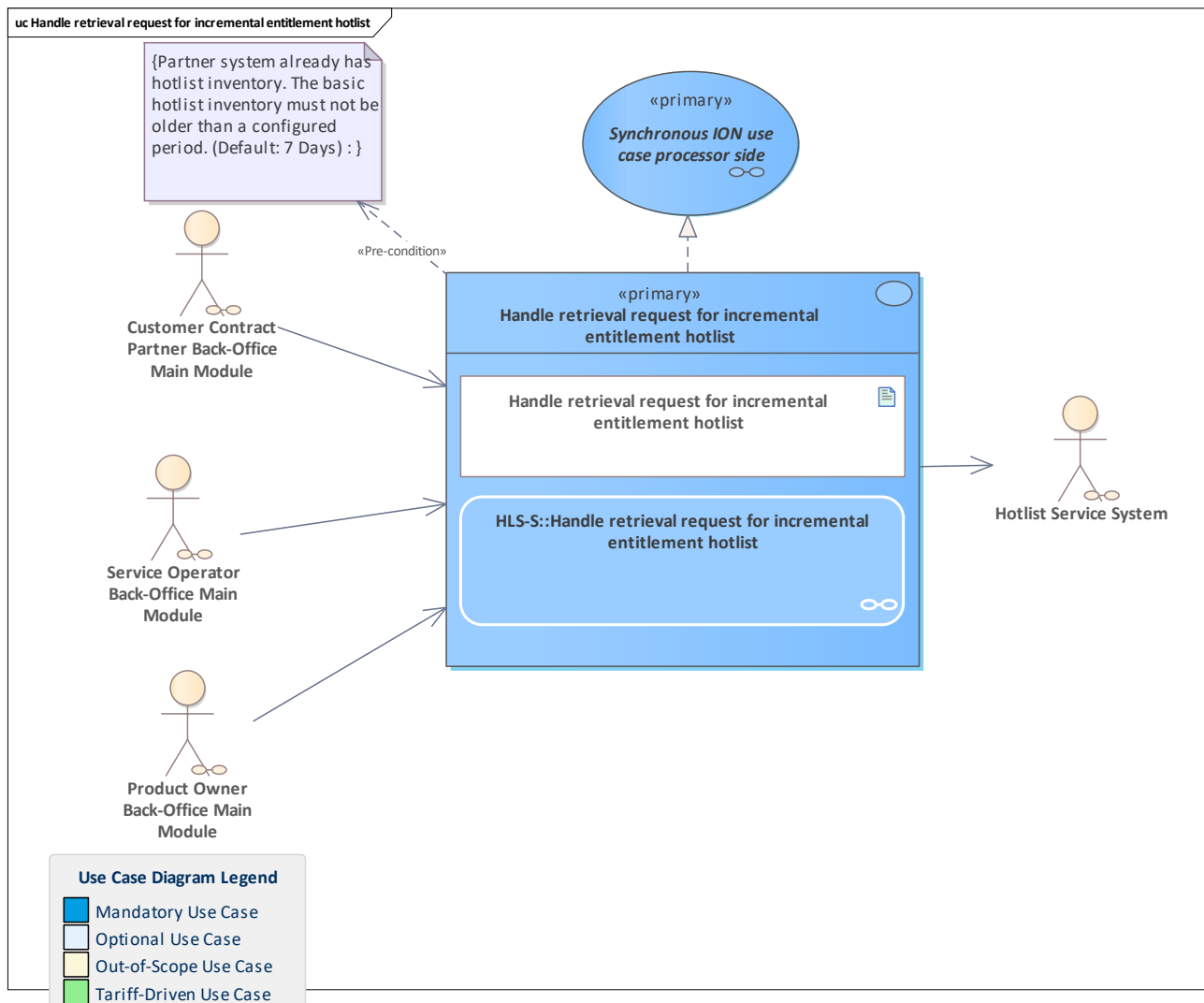


Figure 365: Handle retrieval request for incremental entitlement hotlist

After receiving a request for an entitlement incremental hotlist, the hotlist service system processes the request by running checks (see activity diagram) and creating the incremental entitlement hotlist as a response, which contains changes for relevant entitlements from the given cycle number until the current cycle.

Note: For performance reasons, incremental hotlists may contain additions and removals that cancel each other out. Thus, the increments need to be applied in the order they appear on the list.

11.205 Handle retrieval request for organisation hotlist

11.206 Handle retrieval request for organisation hotlist

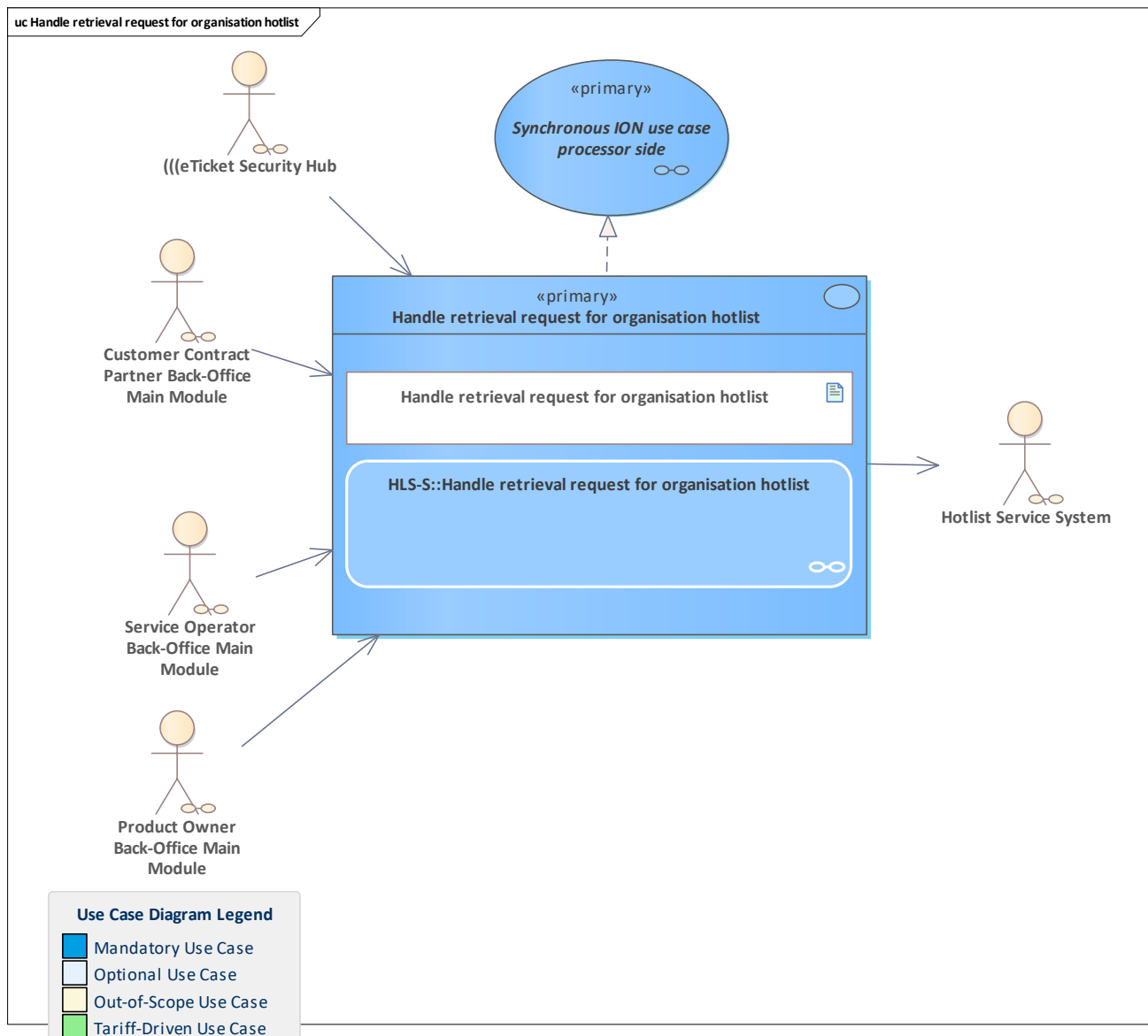


Figure 366: Handle retrieval request for organisation hotlist

After receiving a request for an organisation hotlist, the hotlist service system processes the request by running checks (see activity diagram) and creating the current organisation hotlist as a response.

11.207 Handle retrieval request for SAM hotlist

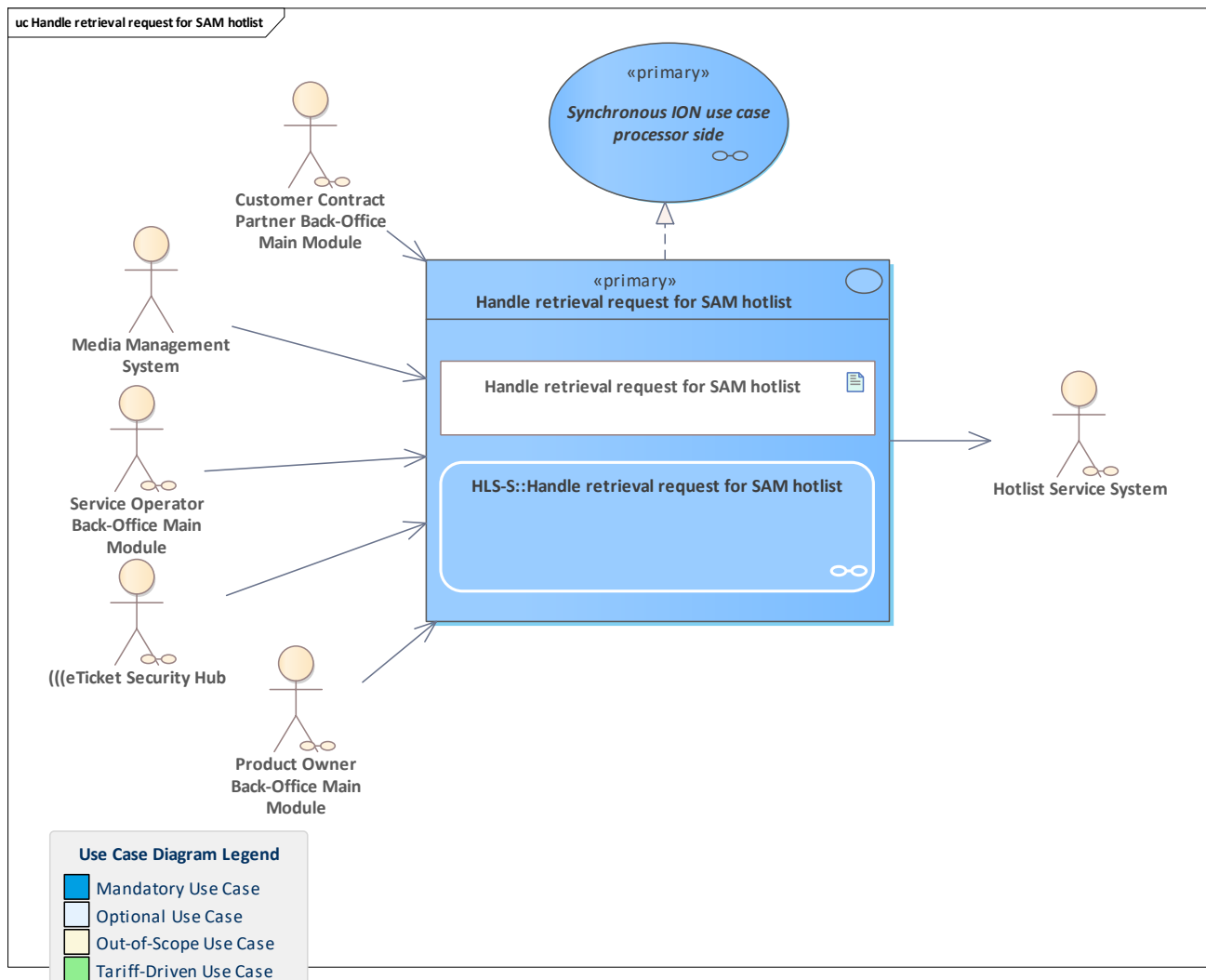


Figure 367: Handle retrieval request for SAM hotlist

After receiving a request for a SAM hotlist, the hotlist service system processes the request by running checks (see activity diagram) and creating the current SAM hotlist as a response.

11.208 Handle retrieval request for unclaimed list information

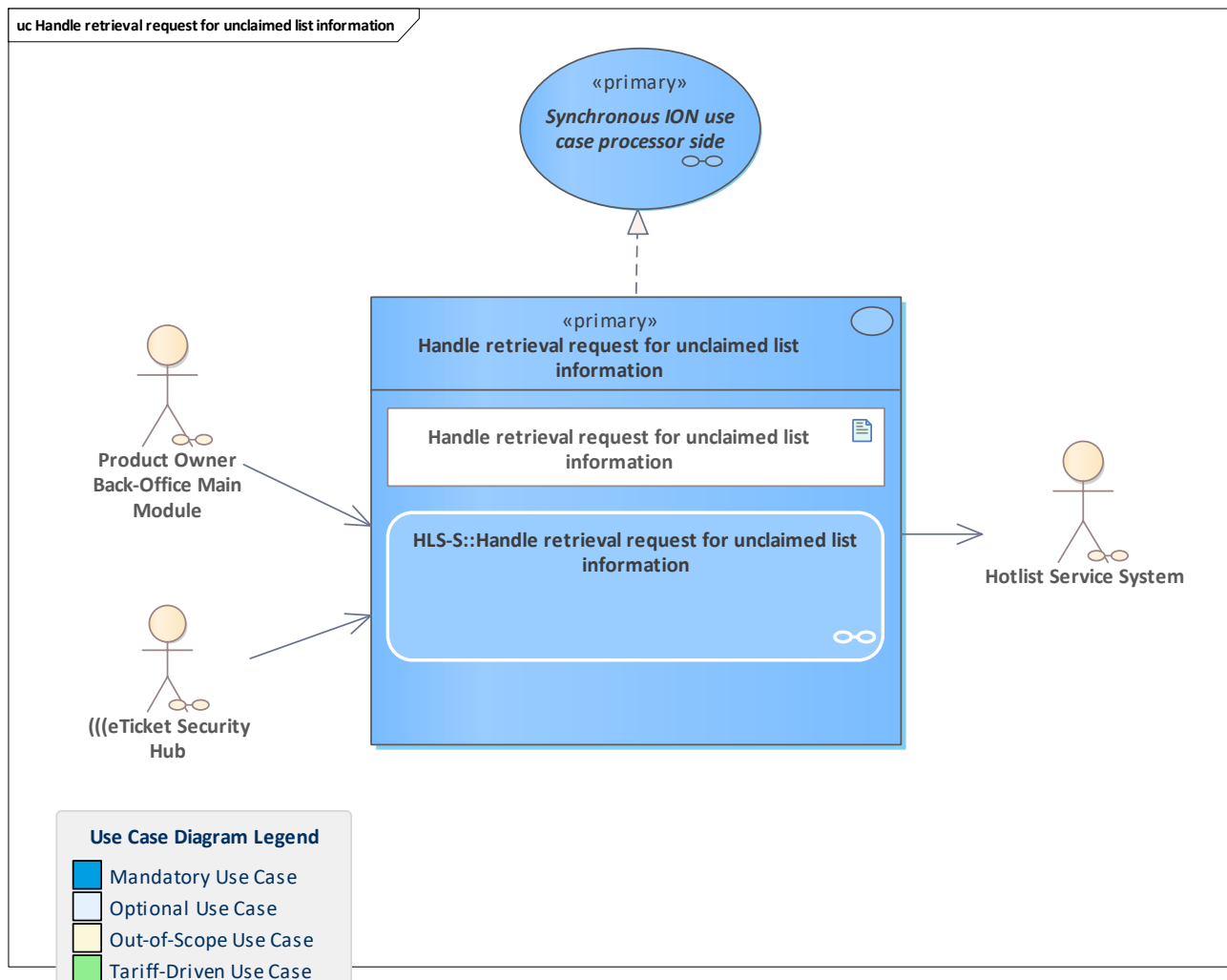


Figure 368: Handle retrieval request for unclaimed list information

The hotlist service system creates and sends the unclaimed list information to the requesting PO.

The hotlist service system gathers the information for all related acceptance candidates of the PO's products.

For each candidate, the period of the passed list cycle until the current list cycle is examined. All unclaimed list information per cycle per candidate is returned.

The list is empty if all acceptance candidates have collected their lists regularly.

11.209 Handle revocation for application hotlisting demand

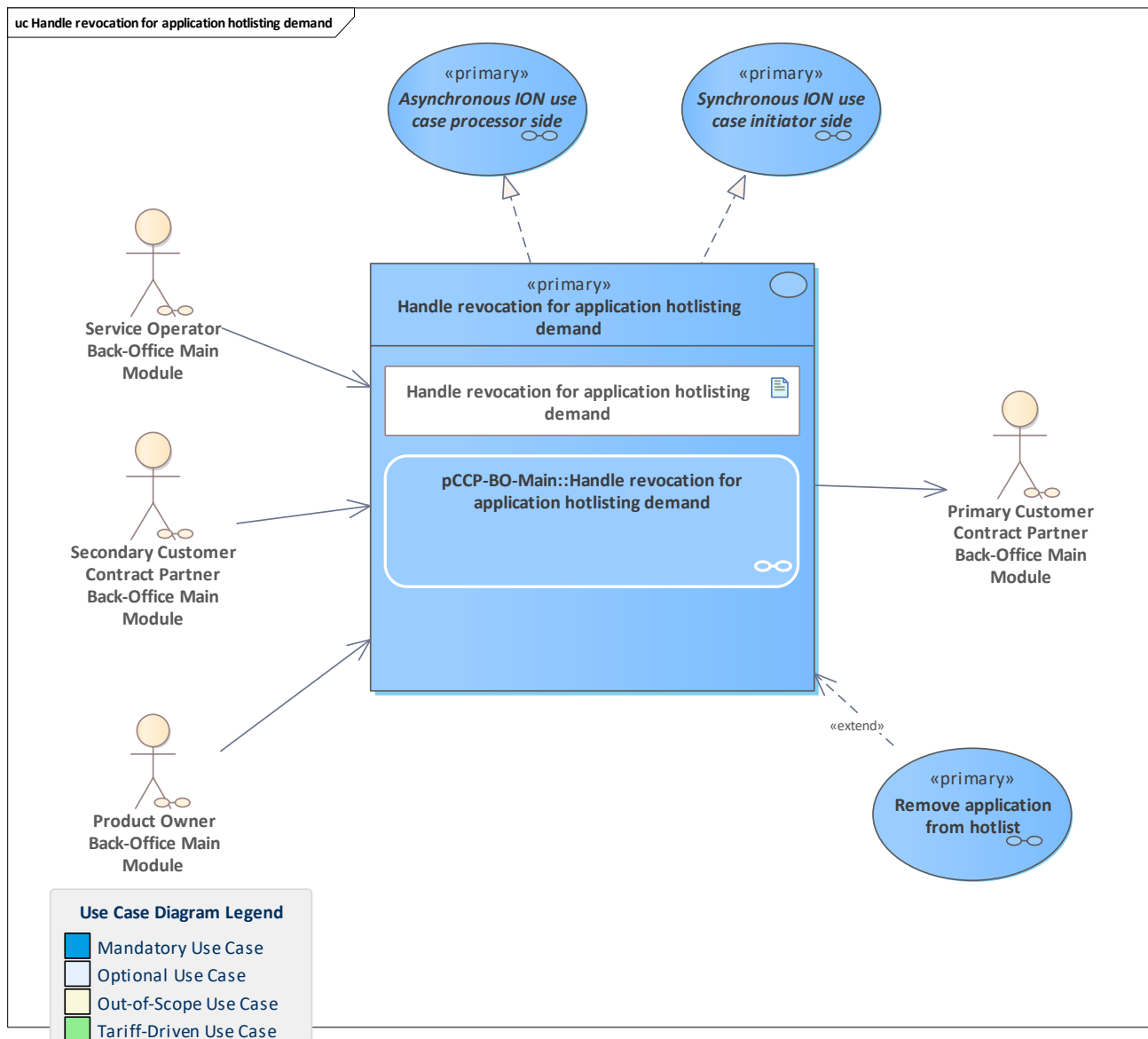


Figure 369: Handle revocation for application hotlisting demand

The pCCP receives a revocation request for the application hotlisting demand.
The pCCP checks the requests and its information in its application management.
If the result is positive and there is no current hotlisting reason, the application will be removed from the hotlist.

Note: if the pCCP is a new application owner of an existing application instance, the pCCP must have previously transferred the information about old hotlisting demands (if any), especially the ION message references.

11.210 Handle revocation for authentication key hotlisting demand

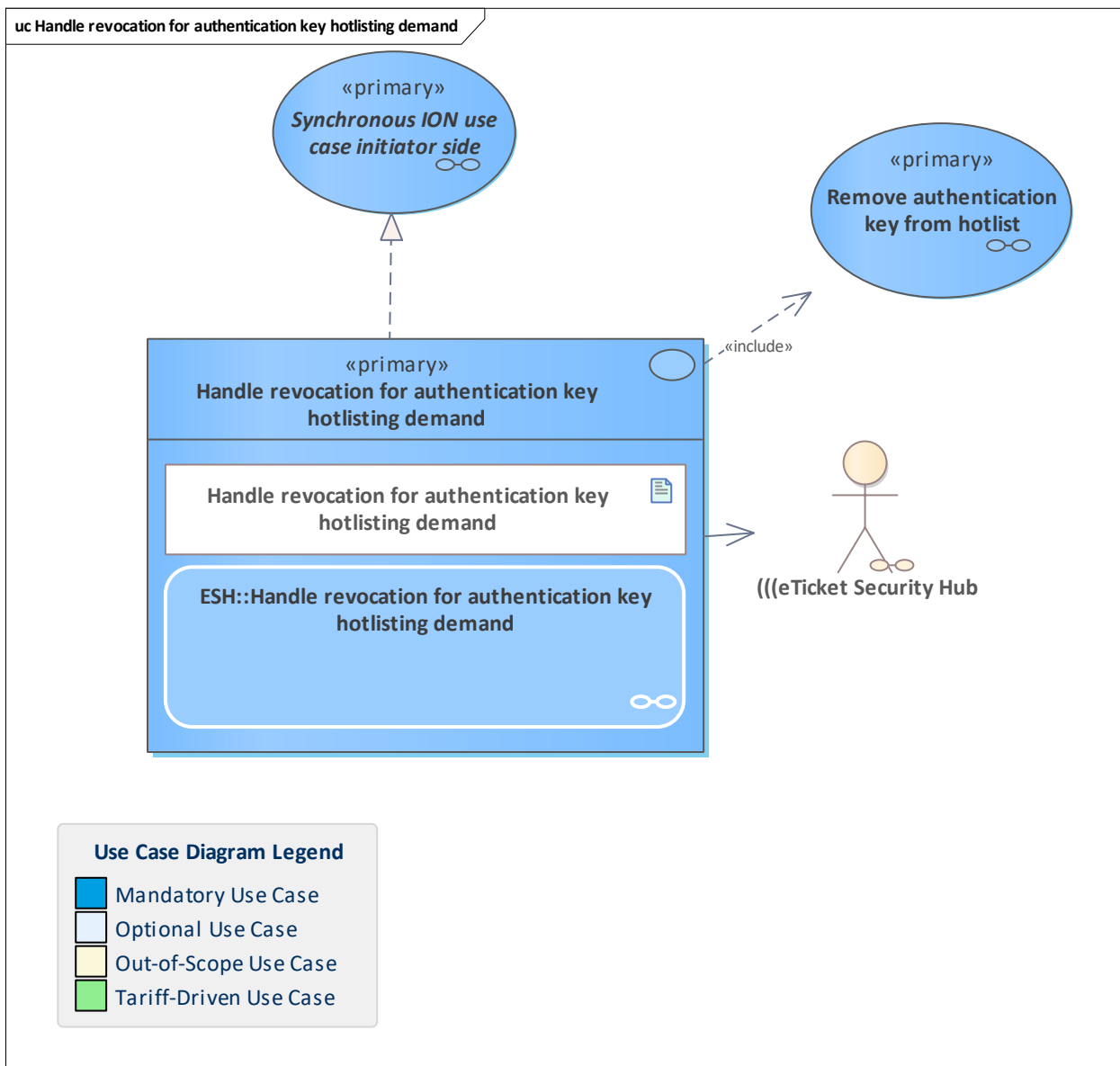


Figure 370: Handle revocation for authentication key hotlisting demand

Rare use case only for the scheme manager's ESH.

Authorised staff can remove an authentication key from the hotlist. This is only needed, if a new generation of user media or SAMs comes into play so that old authentication keys which are no longer placed on any UM or SAM can be removed.

For this target, a request is sent to the hotlist service system to remove a certain authentication key from the hotlist.

11.211 Handle revocation for entitlement hotlisting demand

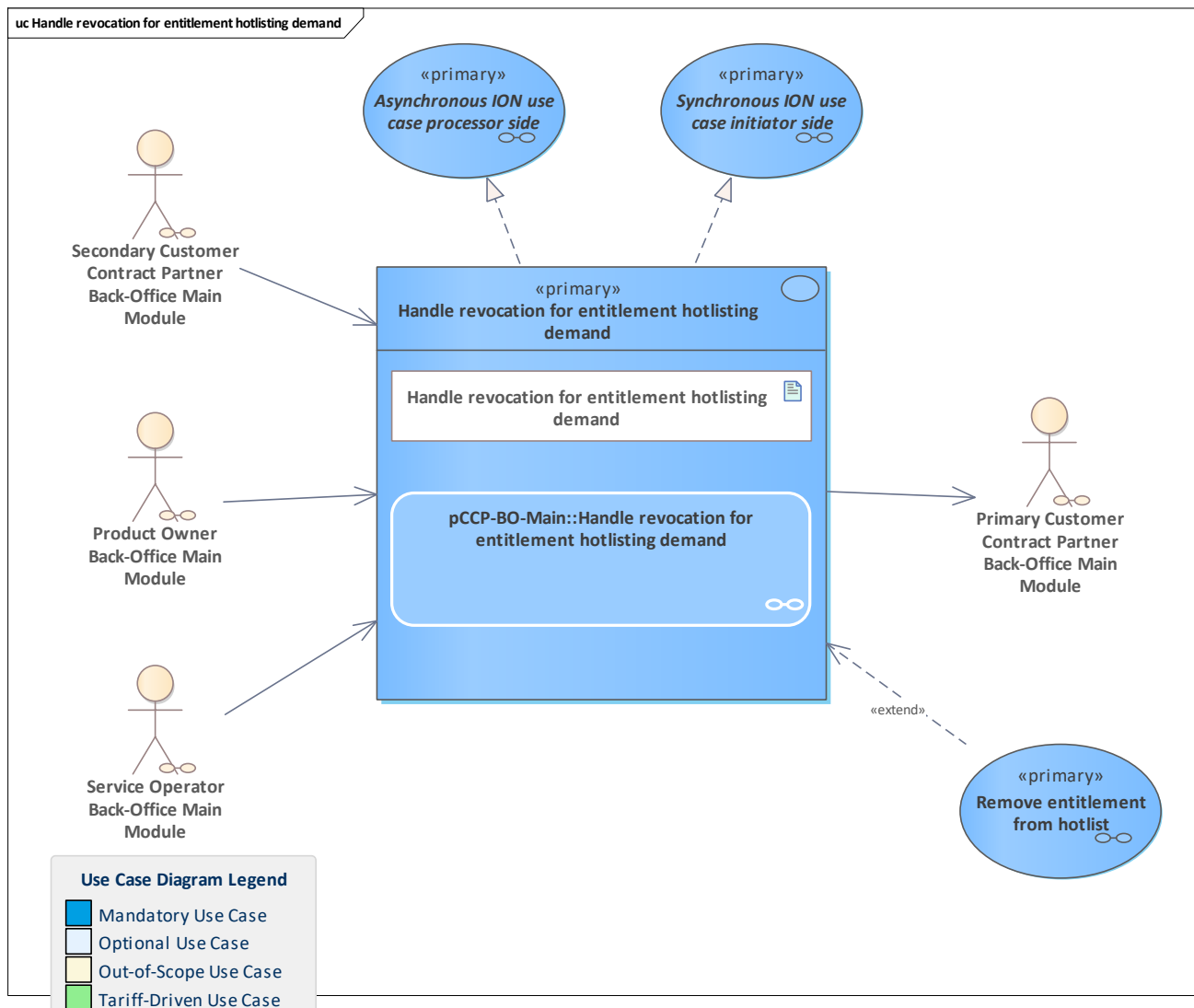


Figure 371: Handle revocation for entitlement hotlisting demand

The pCCP receives a revocation request for the entitlement hotlisting demand. The pCCP checks the requests and its information in its entitlement management. If the result is positive and there is no current hotlisting reason, the entitlement will be removed from the hotlist.

11.212 Handle revocation for organisation hotlisting demand

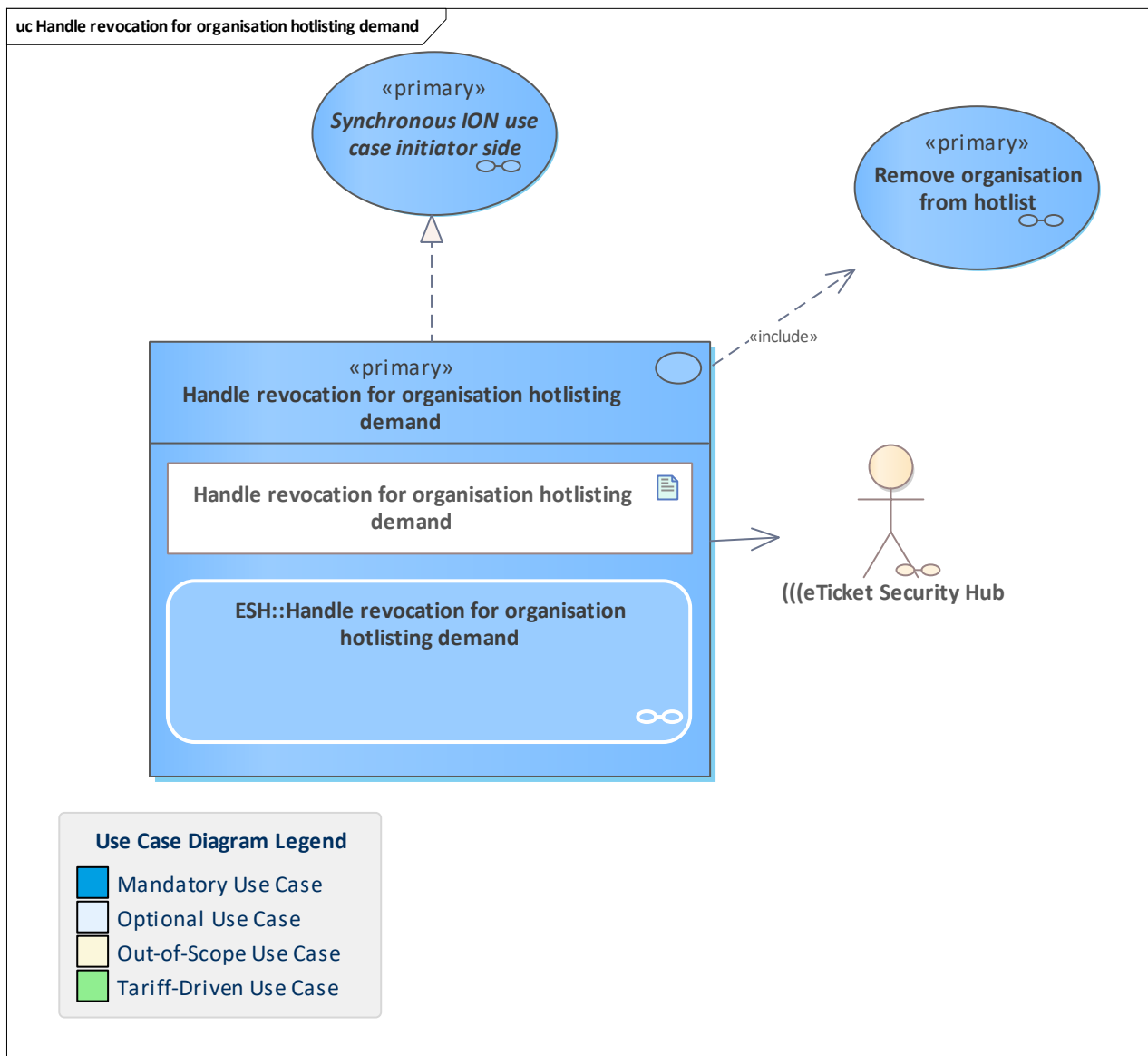


Figure 372: Handle revocation for organisation hotlisting demand

For certain reasons, there may be no need to keep an organisation on the organisation hotlist (out of scope).

In this rare use case, the Scheme Manager requests via the ESH to remove the organisation from the organisation hotlist.

11.213 Handle revocation for SAM hotlisting demand

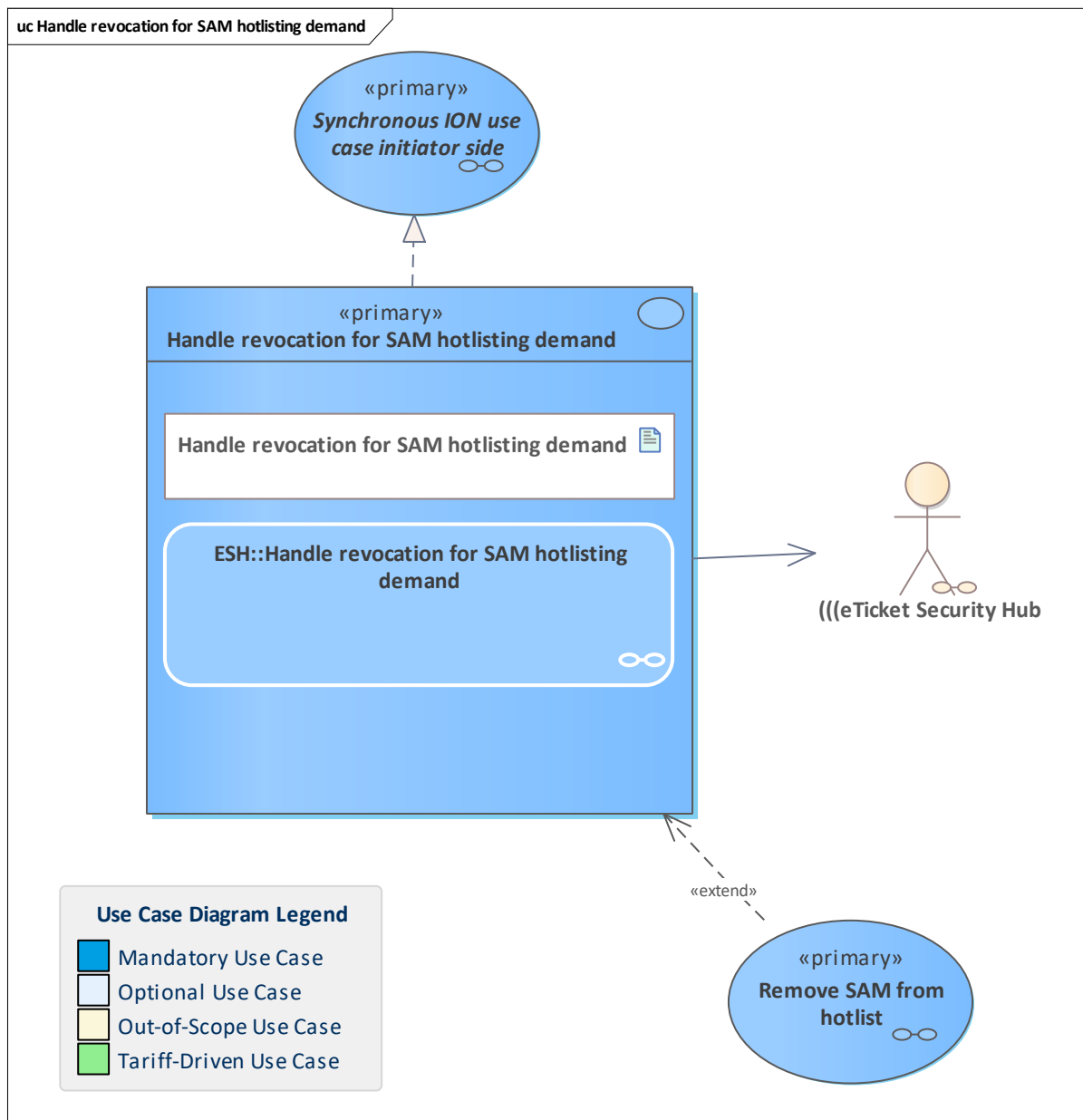


Figure 373: Handle revocation for SAM hotlisting demand

In this rare use case, the scheme manager's ESH requests to remove the SAM from the hotlist, as a result of internal checks (e.g. SAM is more than 10 years on the hotlist or the SAM owner has provided proof of scrapping).

11.214 Handle SAM hotlisting demand

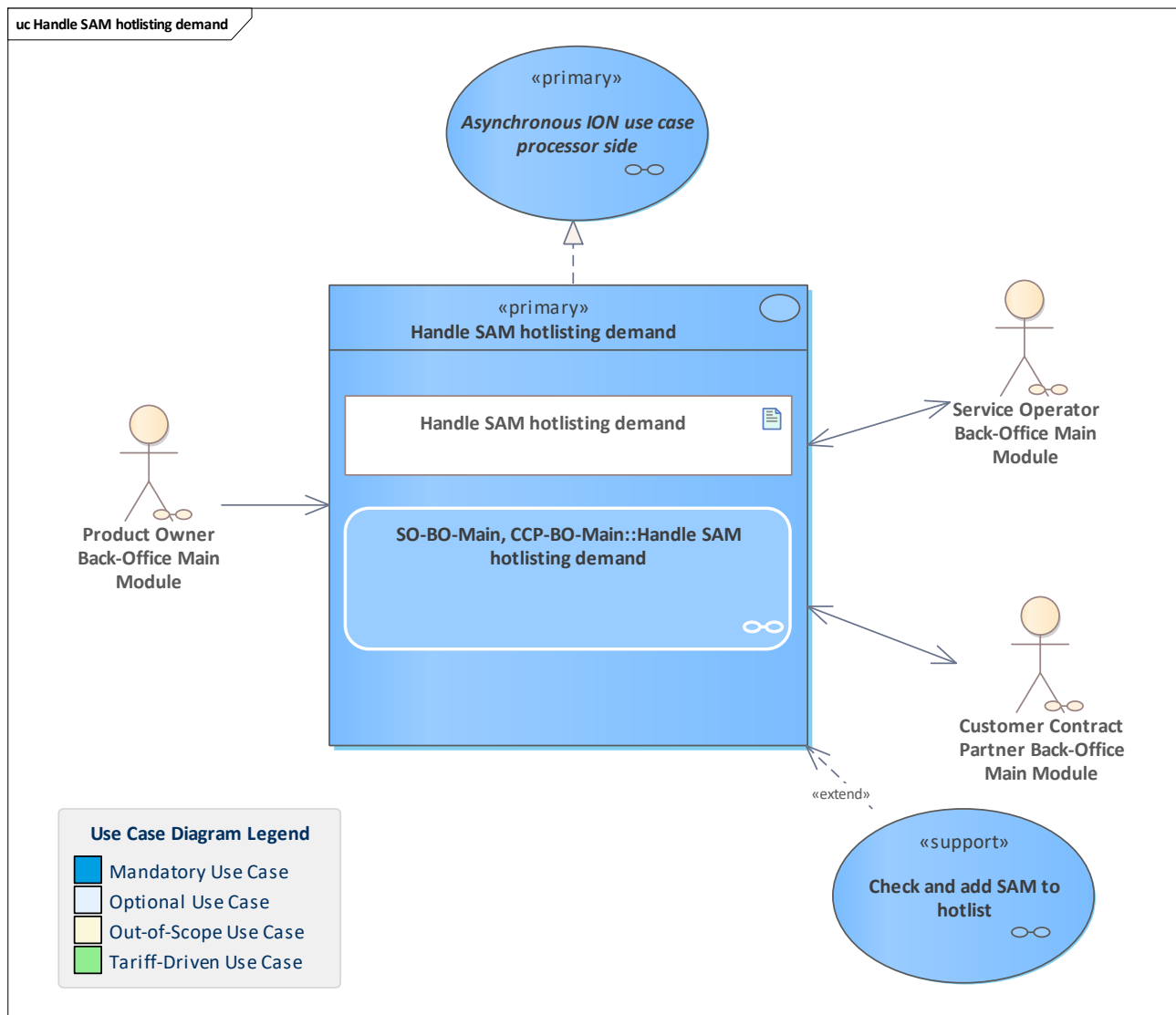


Figure 374: Handle SAM hotlisting demand

The owner (SO or CCP) of a SAM to be hotlisted checks the hotlisting demand. If the result of the check is such that a hotlisting is required, it will inform the hotlist service system to have the SAM added to the SAM hotlist. Otherwise, the demand will be rejected and there is no need to communicate with the hotlist service system.

Please note that if there is more than one demand for hotlisting the same SAM, this has to be considered in the check for a required hotlisting but will not result in an exception. Especially for the monitoring of a third-party system, it must be possible to demand hotlisting even if the same object was demanded for hotlisting in the past.

Please note that the scheme manager is allowed to hotlist SAMs, for example in case of a hotlisted organisation or as an escalation instance.

11.215 Handle static entitlement inspected notification from contractual perspective

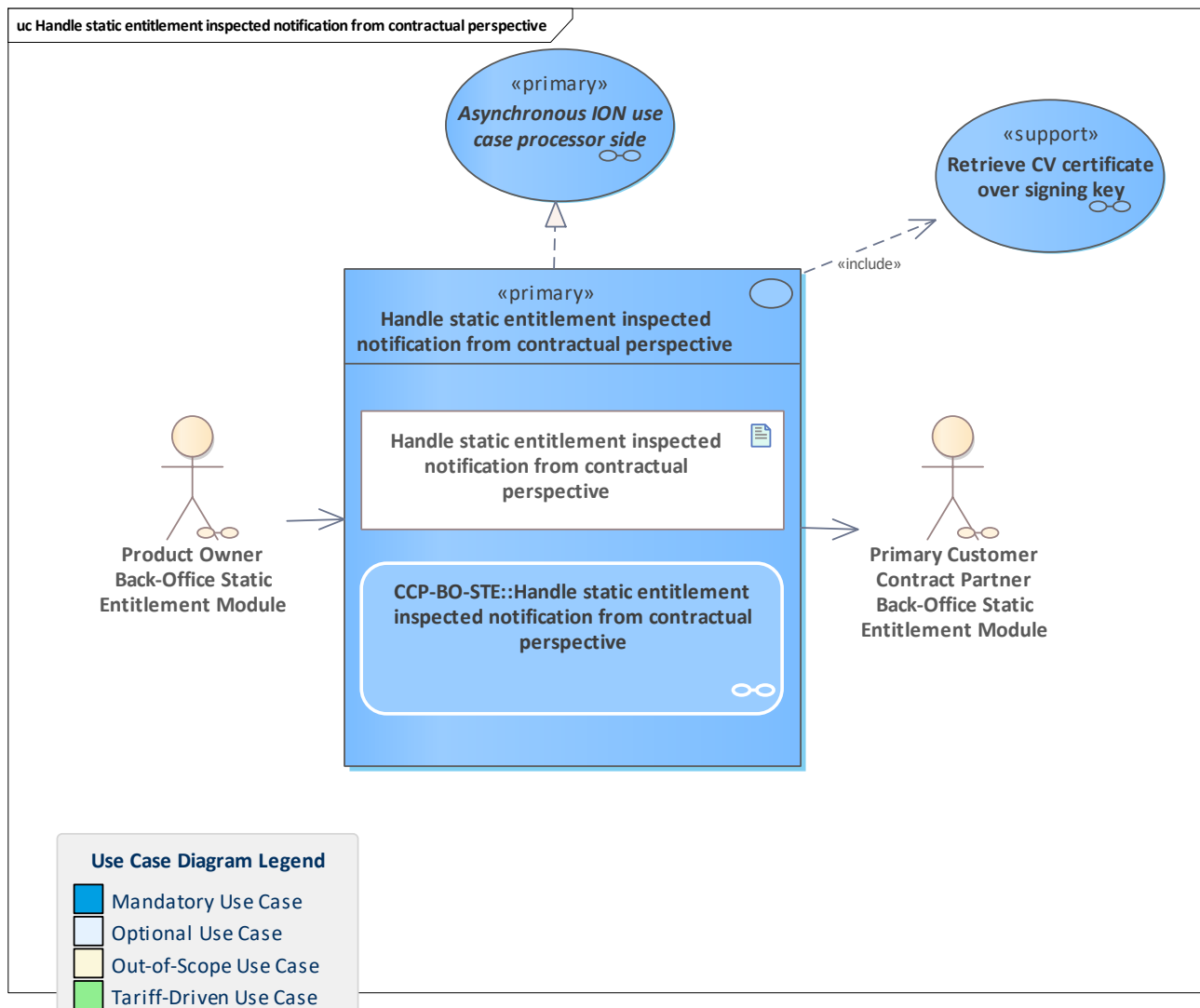


Figure 375: Handle static entitlement inspected notification from contractual perspective

This use case describes the processing of the notification of an inspected static entitlement in the pCCP back-office system.

The pCCP receives the notification from the PO system, registers it and does its contractual monitoring checks.

11.216 Handle static entitlement inspected notification from operational perspective

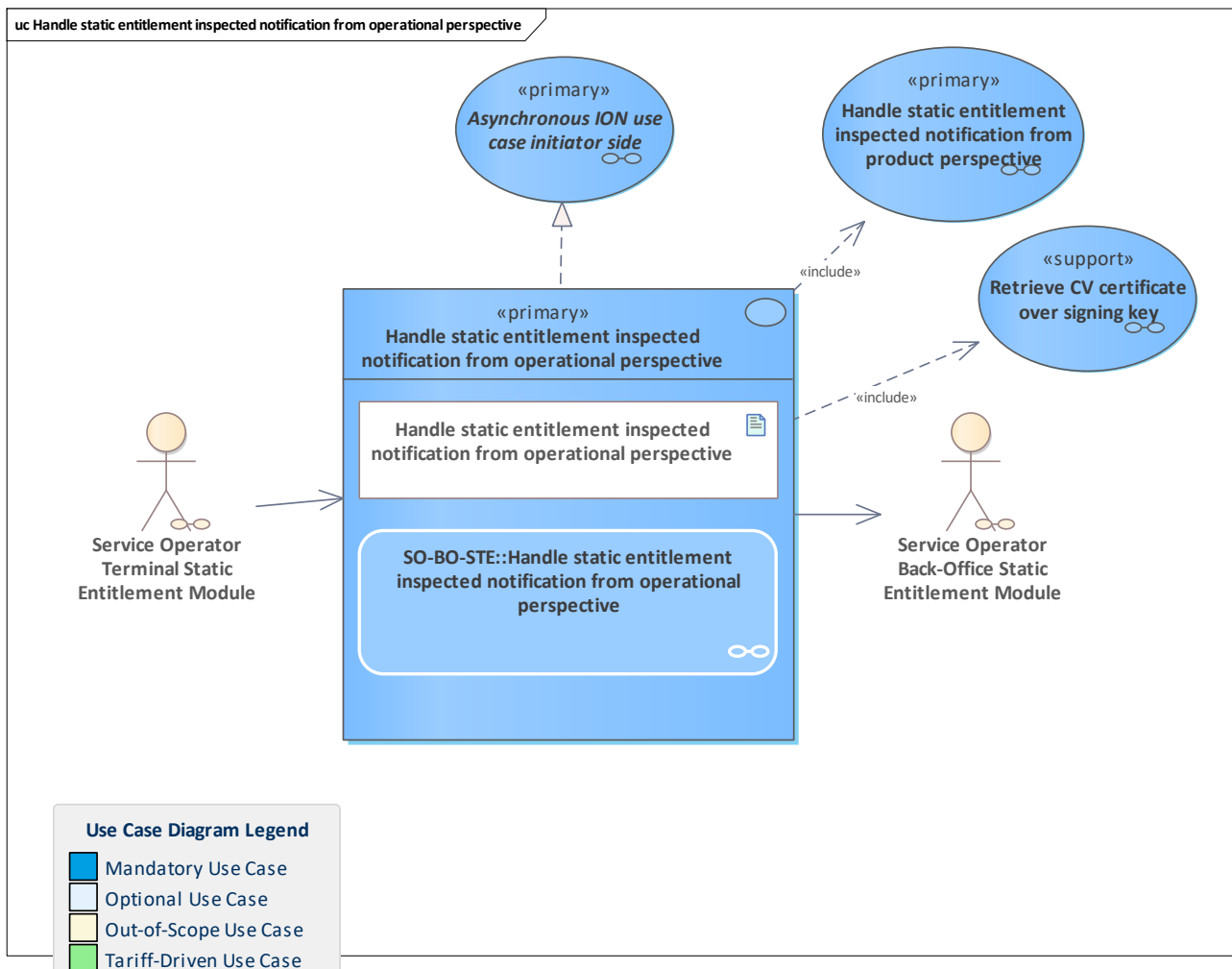


Figure 376: Handle static entitlement inspected notification from operational perspective

Use case for the SO back-office system with static entitlement extension. The notification from the terminal is received and checked from the operational perspective, including the SAM signature verification of the static entitlement. Finally, the notification is sent to the PO back-office system. This can be done either directly with a single message or in a scheduled process that sends a list of notifications.

11.217 Handle static entitlement inspected notification from product perspective

11.218 Handle static entitlement inspected notification from product perspective

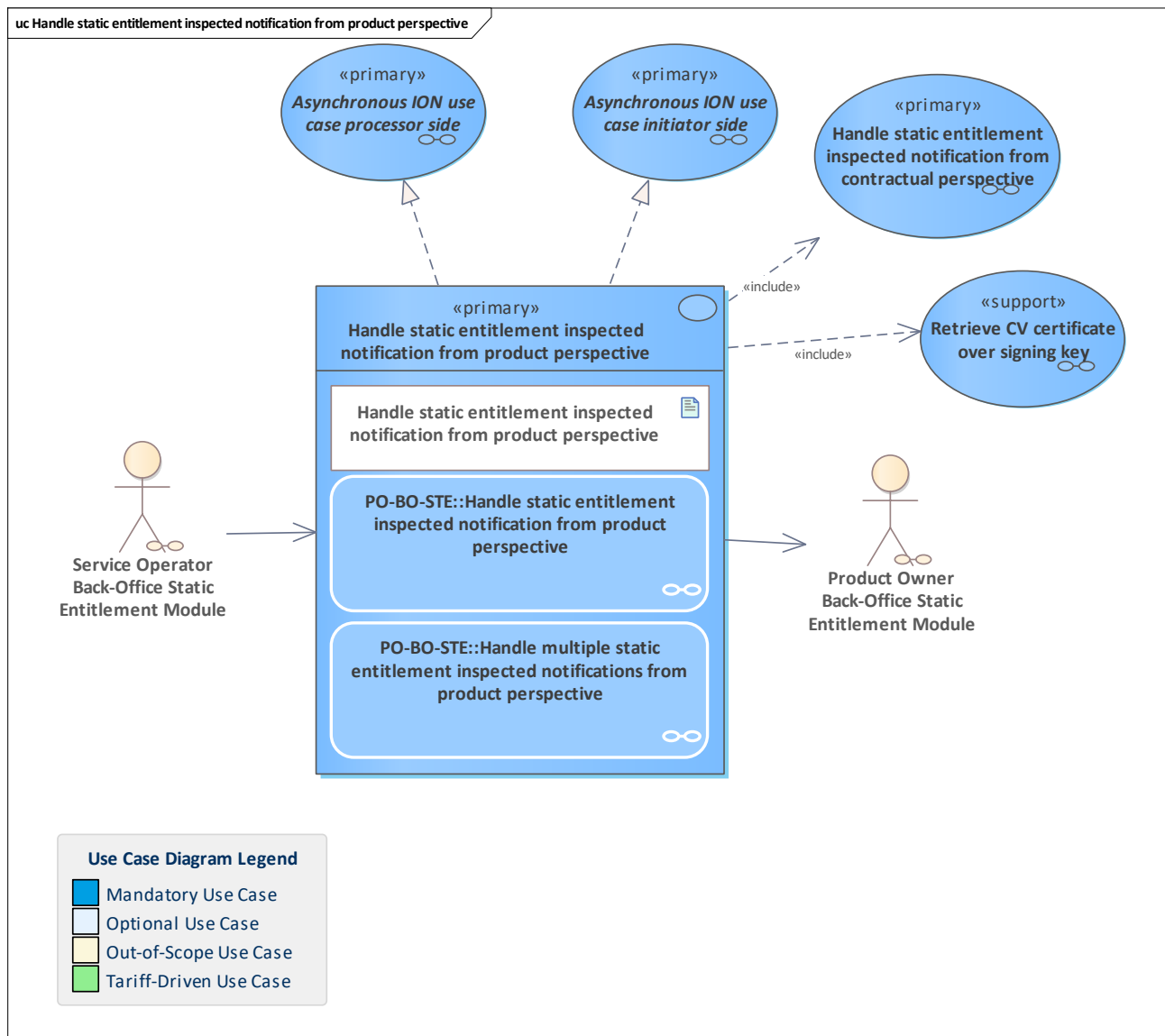


Figure 377: Handle static entitlement inspected notification from product perspective

This use case describes the processing of the notification of an inspected static entitlement in the PO back-office system.

The SO sends the notification, the PO registers it and does its monitoring checks. After these checks, the notification is forwarded to the CCP back-office system.

Furthermore, static entitlement inspected notifications can be collected in the SO system and then sent as a list in a scheduled process to the PO.

11.219 Handle static entitlement issued notification from contractual perspective

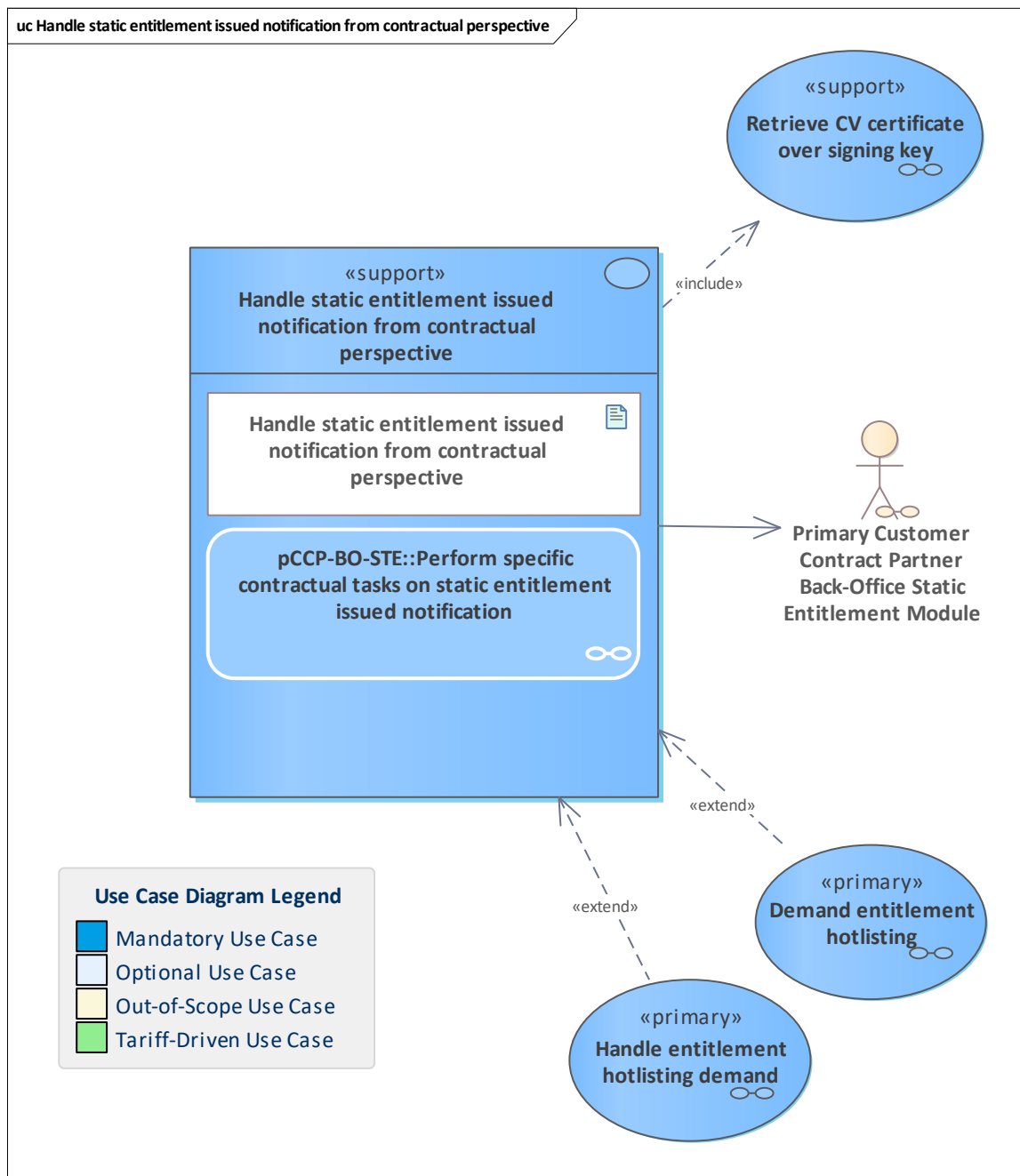


Figure 378: Handle static entitlement issued notification from contractual perspective

Handle a notification about an issuance of an owned static entitlement from the contractual perspective.

All needed checks and monitoring are performed.

11.220 Handle static entitlement issued notification from operational perspective

11.221 Handle static entitlement issued notification from operational perspective

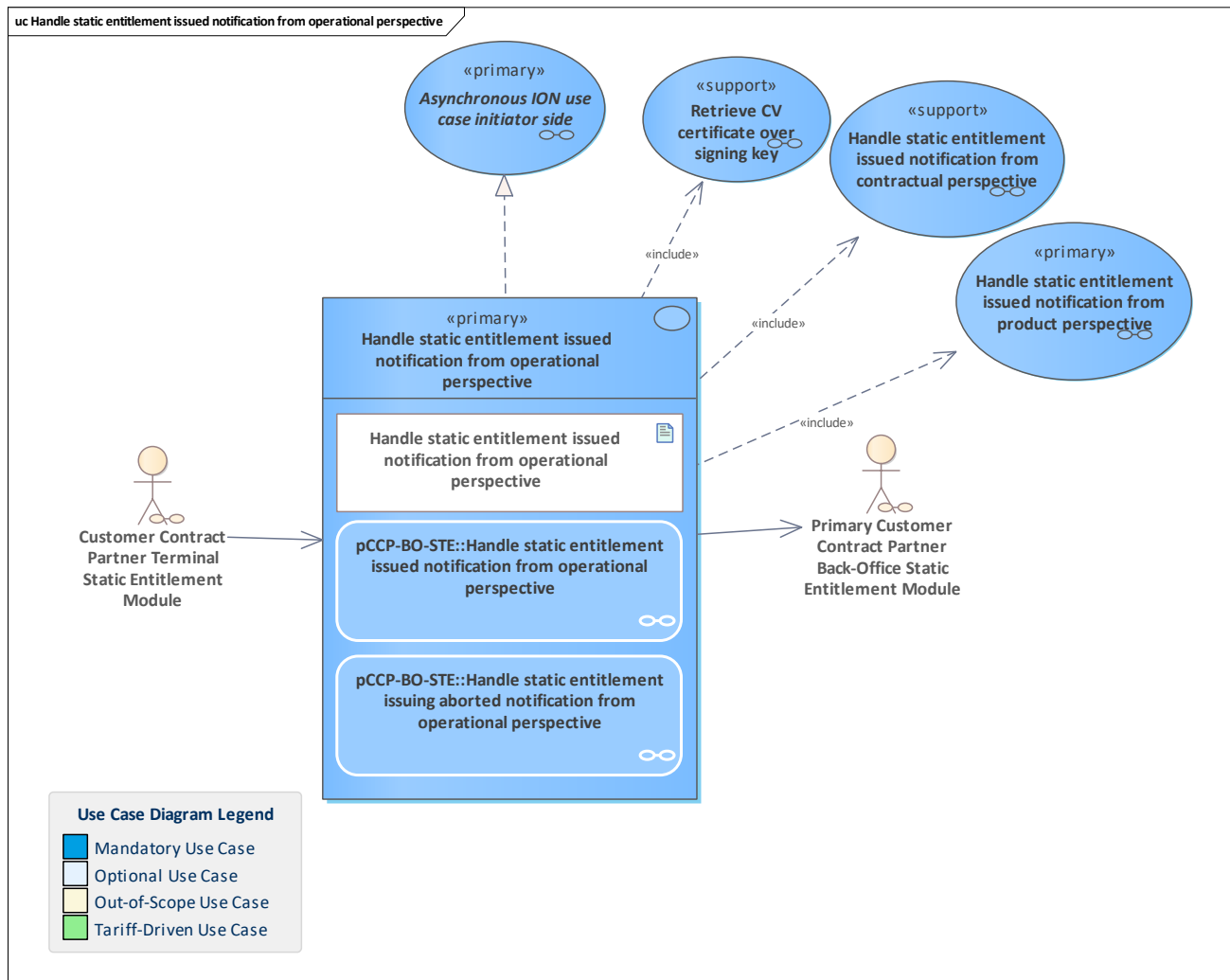


Figure 379: Handle static entitlement issued notification from operational perspective

Handle a notification about a static entitlement issuance from the operational perspective. The static entitlement extension of the CCP terminal sends the notification about the issuance of a static entitlement to the responsible pCCP back-office system with static entitlement extension.

The pCCP system does its operational checks and monitoring, such as the SAM signature verification of the static entitlement.

Then, the notification is sent to the responsible PO system with a static entitlement extension.

This can be done either via a single message or in a scheduled process as a message list.

Since it is the pCCP, the system does the contractual checks and monitoring, too.

This use case also handles a possible abortion of the entitlement issuance. Also in this case, the registration and forwarding of the notification to the PO system is important due to the SAM and product issuance counters, to keep the monitoring consistent.

11.222 Handle static entitlement issued notification from product perspective

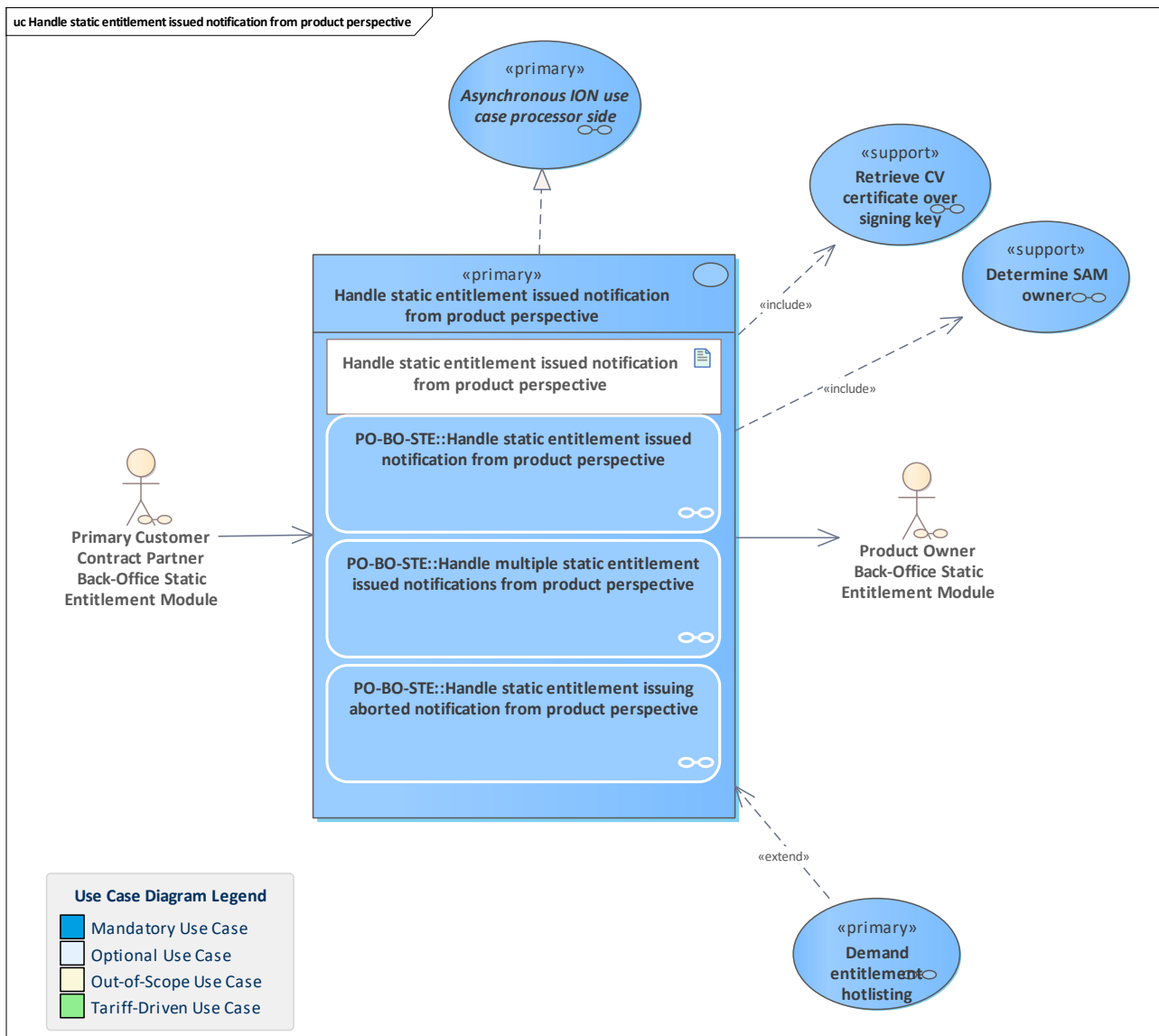


Figure 380: Handle static entitlement issued notification from product perspective

Handle a notification about a static entitlement issuance from the product owner perspective. The PO back-office system with a static entitlement extension receives the message coming from the pCCP.

The message can be either a single notification or a notification list.

The PO registers the notification(s) and does its checks and monitoring from the product owner perspective.

11.223 Handle static entitlement terminated notification from contractual perspective

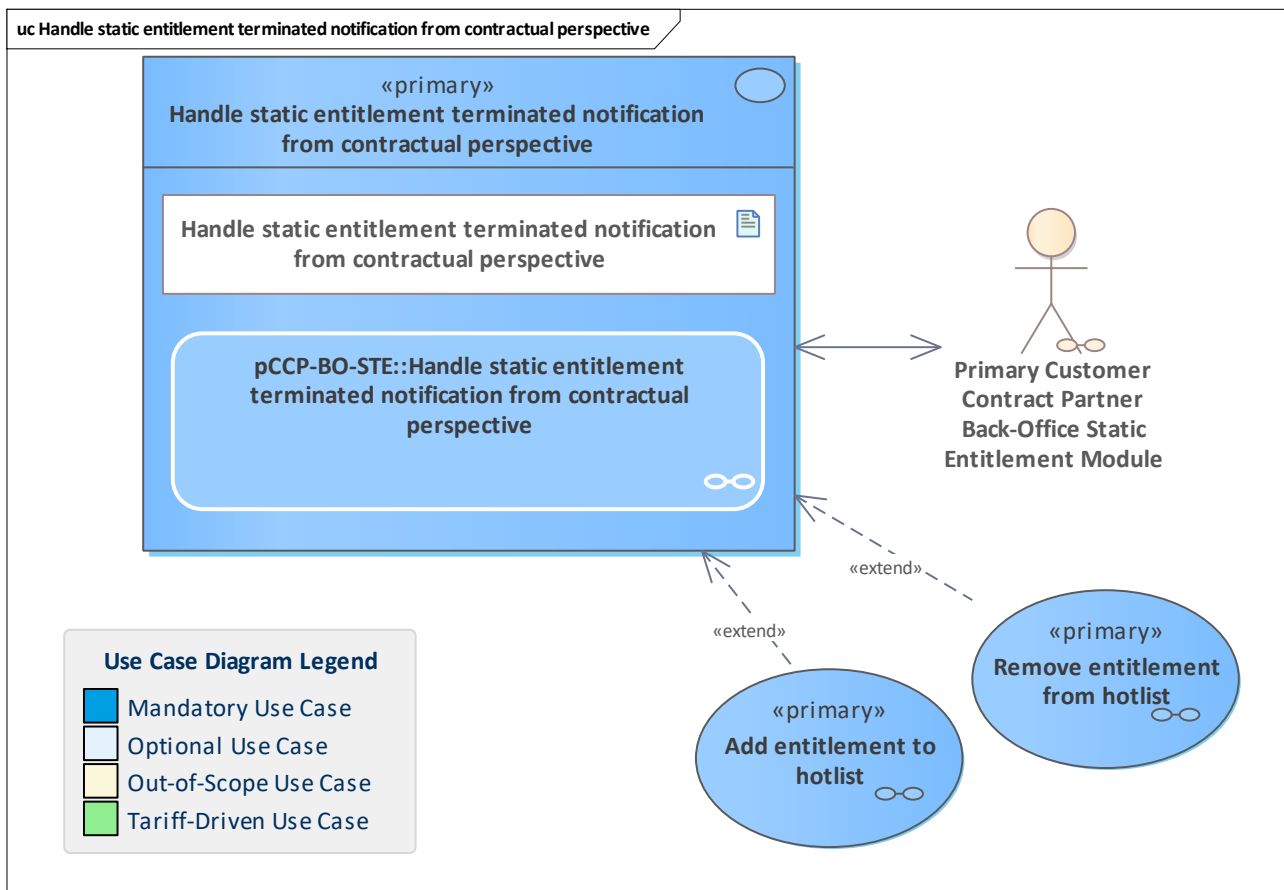


Figure 381: Handle static entitlement terminated notification from contractual perspective

The notification of a terminated static entitlement is handled from the contractual perspective. The pCCP back-office system with static entitlement extension does its contractual checks and monitoring.

Note: this closes a potentially related user account concerning the static entitlement.

The static entitlement must be hotlisted (if it was not already hotlisted) to avoid illegal usage of the static entitlement.

Note: For a temporarily hotlisted static entitlement, the hotlist entry must be changed to a non-temporarily hotlist entry by removing it and adding it again with a new hotlist expiry date to the hotlist.

Note: the contractual handling of the static entitlement termination is done by the [Primary Customer Contract Partner Back-Office Static Entitlement Module](#); the hotlist activities are done by the [Primary Customer Contract Partner Back-Office Main Module](#).

11.224 Handle static entitlement terminated notification from operational perspective

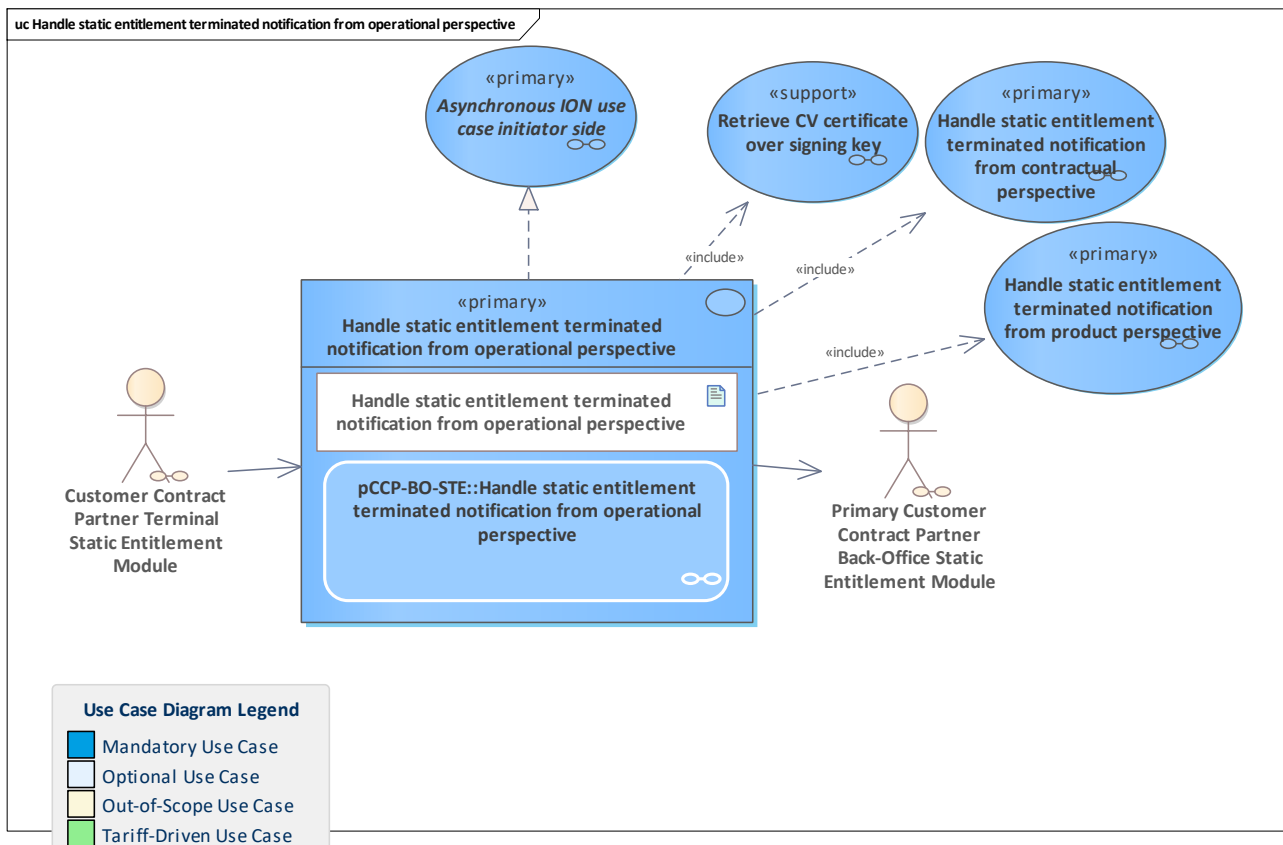


Figure 382: Handle static entitlement terminated notification from operational perspective

Handle a static entitlement terminated notification from the operational perspective. The entitlement terminated notification is sent by the CCP terminal to the back-office system of the CCP with a static entitlement extension. The notification will be checked and monitored. Only the pCCP can terminate its own static entitlement:

- the notification will be sent to the PO
- the pCCP does its contractual checks and monitoring

11.225 Handle static entitlement terminated notification from product perspective

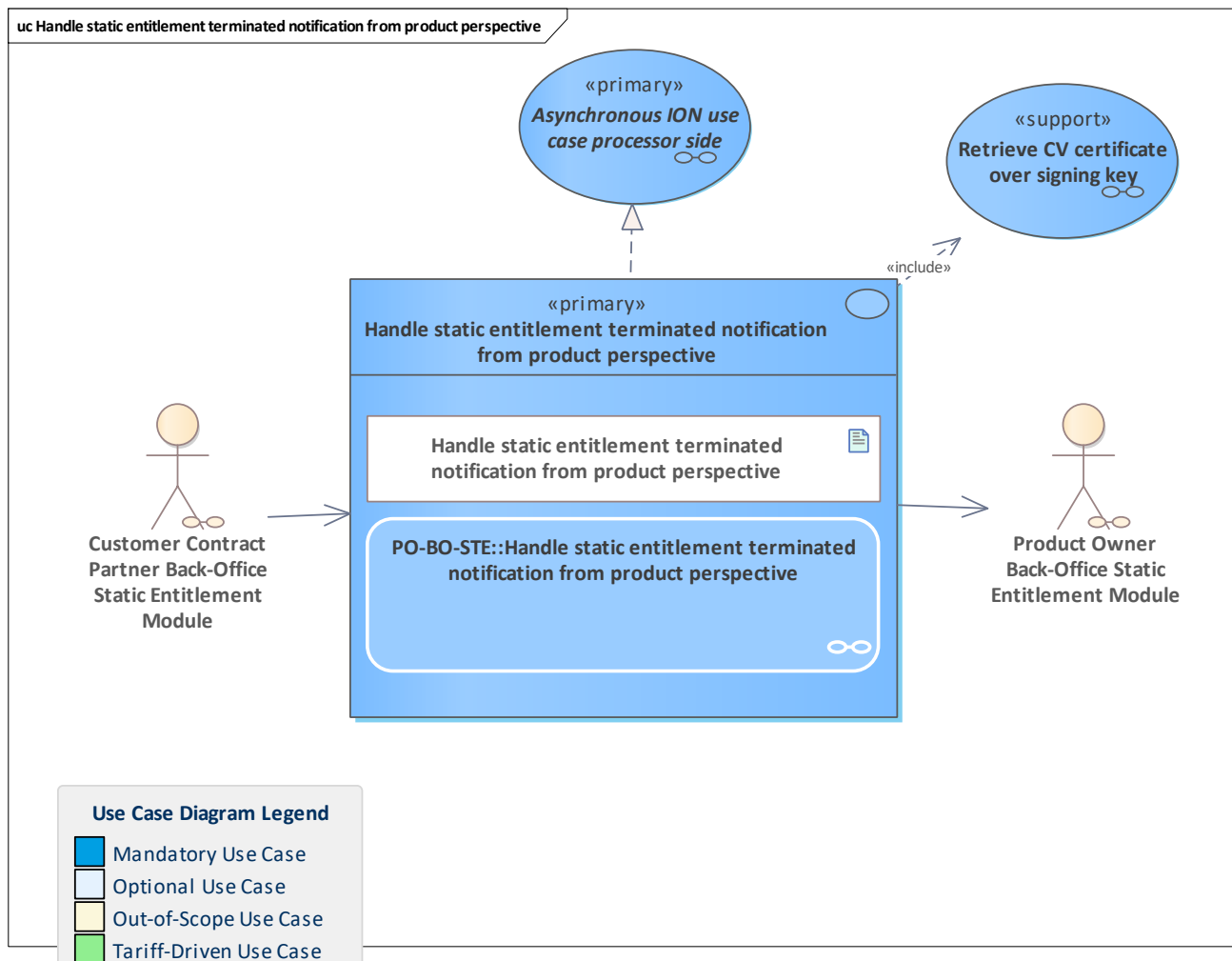


Figure 383: Handle static entitlement terminated notification from product perspective

Static entitlement terminated notification is handled from the product owner perspective. The static entitlement terminated notification is received by the PO. The PO registers static entitlement terminated notification and does its checks and monitoring from the product owner perspective.

11.226 Handle stored-value payment method credited notification from contractual perspective

11.227 Handle stored-value payment method credited notification from contractual perspective

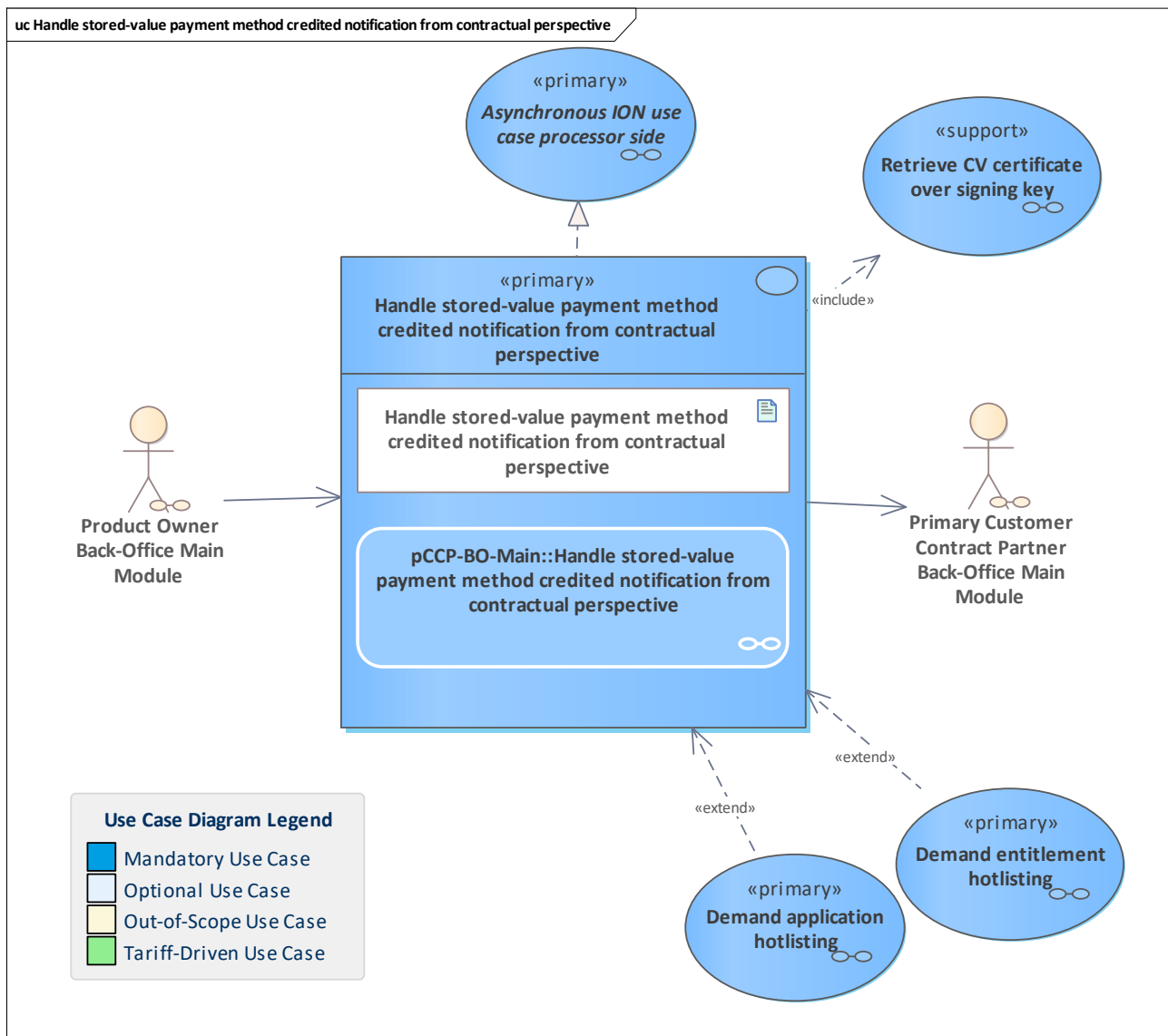


Figure 384: Handle stored-value payment method credited notification from contractual perspective

Handle a stored-value payment method credited notification from the contractual perspective. The pCCP does its contractual checks and monitoring.

Note: if the pCCP itself performed the credit action, this use case takes inside the use case [Handle stored-value payment method credited notification from operational perspective](#) and is not called by the PO.

In this case, this use case is not an [Asynchronous ION use case processor side](#).

11.228 Handle stored-value payment method credited notification from operational perspective

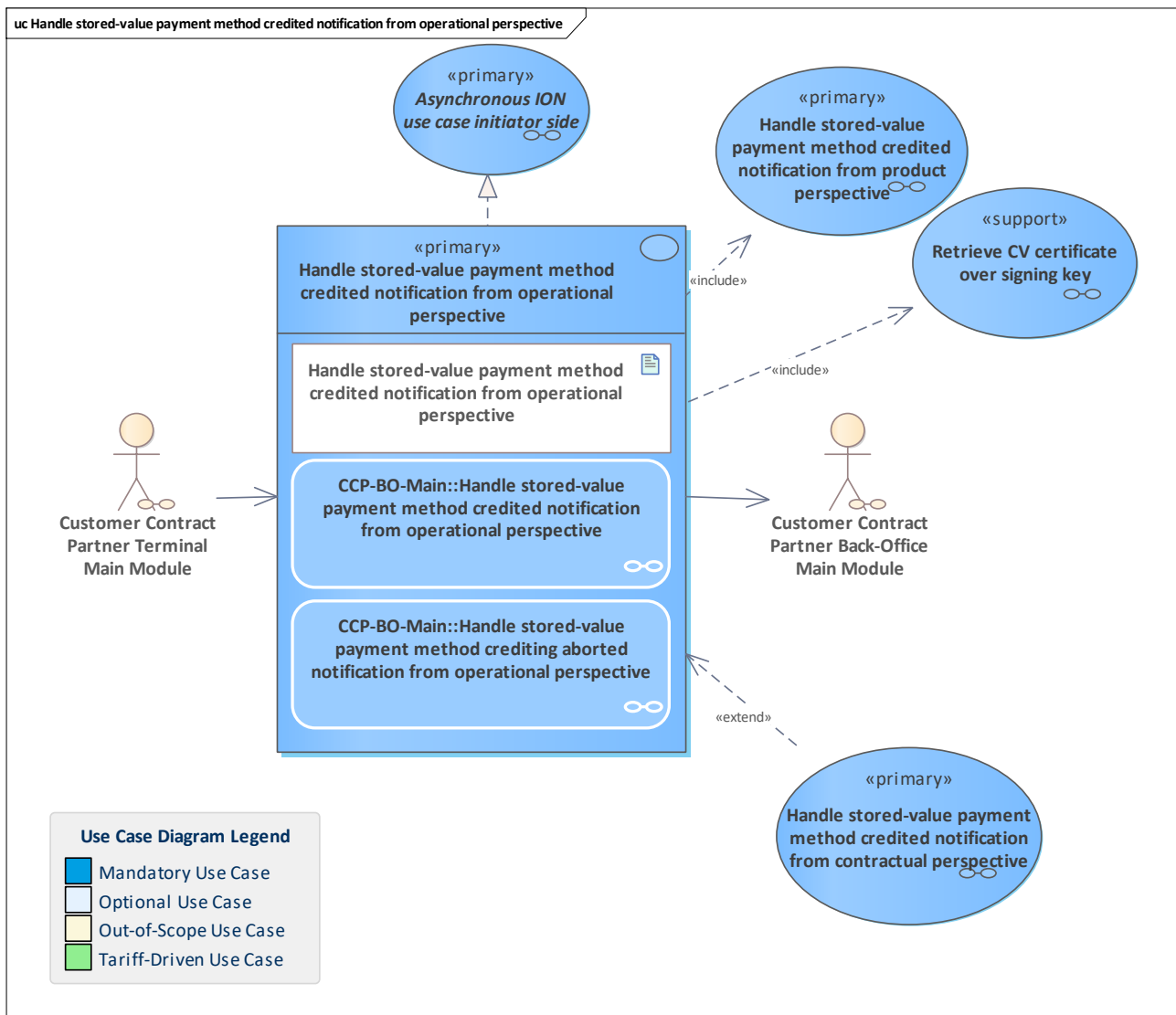


Figure 385: Handle stored-value payment method credited notification from operational perspective

Handle a stored-value payment method credited notification from the operational perspective. The CCP back-office system receives the notification about the credit transaction of a stored-value payment method and registers it.

Then it handles the notification from the operational perspective by performing checks and monitoring, e.g. verifying the signature of the credit action attestation.

Then, the notification is forwarded to the responsible PO system.

If the current CCP is the pCCP, it can also do the contractual checks and monitoring directly.

In case of an abortion, the notification is registered for consistent monitoring. Since no counters are affected which are important to the PO, in the case of a transaction abortion, the notification is not forwarded.

11.229 Handle stored-value payment method credited notification from product perspective

11.230 Handle stored-value payment method credited notification from product perspective

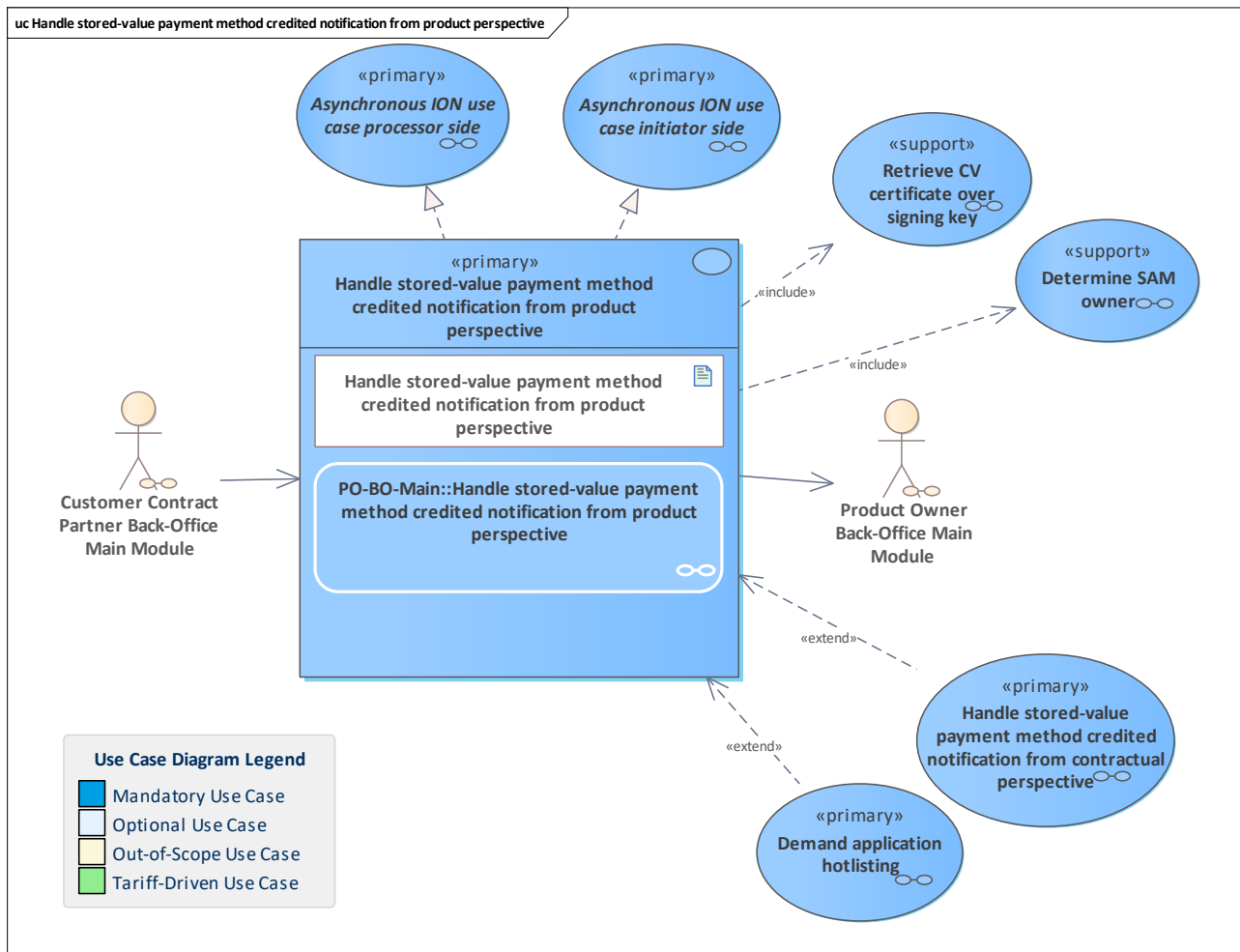


Figure 386: Handle stored-value payment method credited notification from product perspective

Handle a stored-value payment method credited notification from the product owner perspective.

The PO back-office system receives the notification from the CCP system and registers the notification about a performed credit with a stored-value payment method.

It does the checks and monitoring from the product owner perspective.

If the sending CCP was not the pCCP, the notification is forwarded to the responsible pCCP system. Then, this use case is also an [Asynchronous ION use case initiator side](#).

11.231 Handle stored-value payment method debited notification from contractual perspective

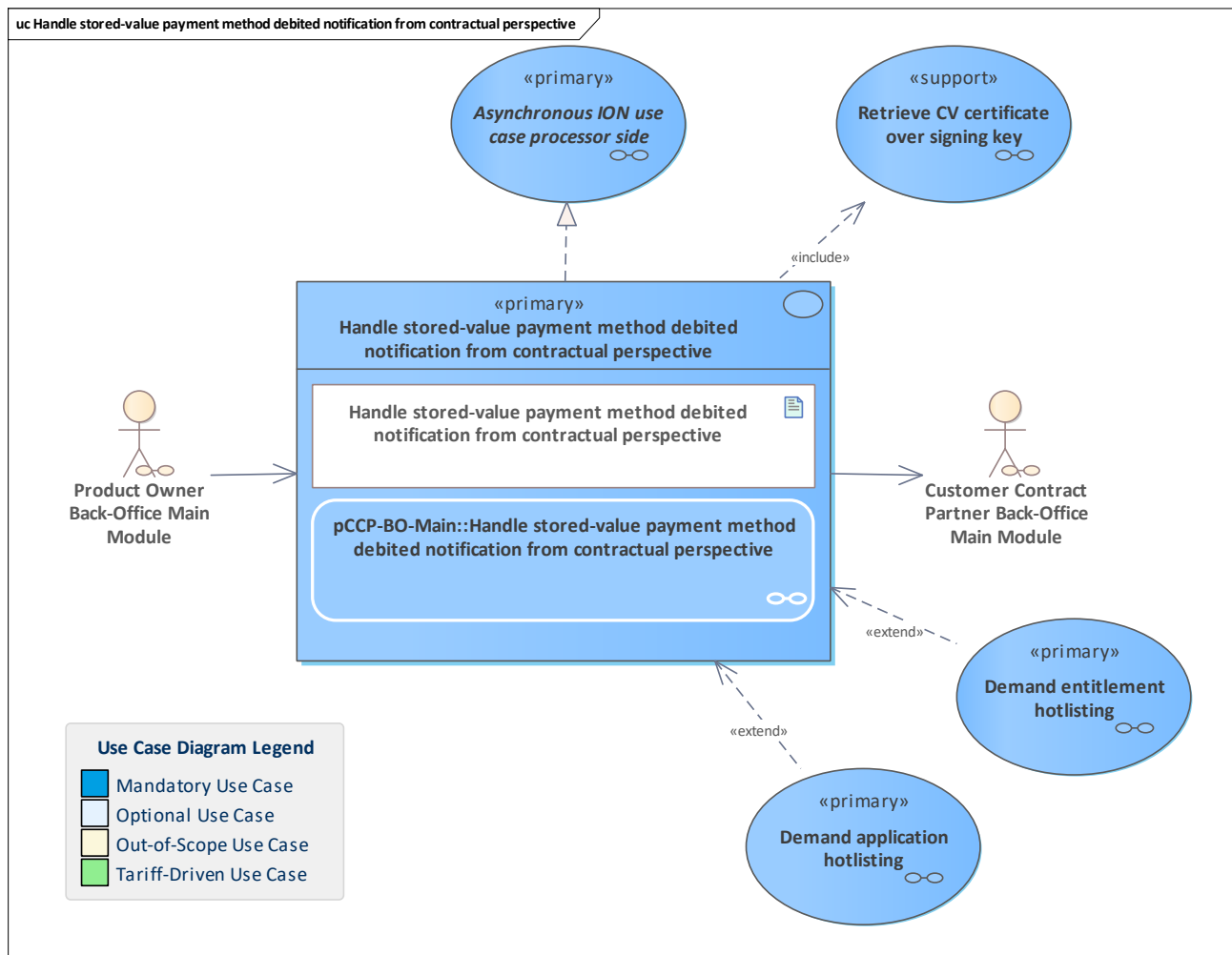


Figure 387: Handle stored-value payment method debited notification from contractual perspective

Handle a notification about a stored-value payment method debiting from the contractual perspective.

The pCCP does its contractual checks and monitoring.

Note: if the pCCP itself performed the debit action, this use case takes place inside the use case [Handle stored-value payment method debited notification from operational perspective](#) and is not called by the PO.

In this case, this use case is not an [Asynchronous ION use case processor side](#).

11.232 Handle stored-value payment method debited notification from operational perspective

11.233 Handle stored-value payment method debited notification from operational perspective

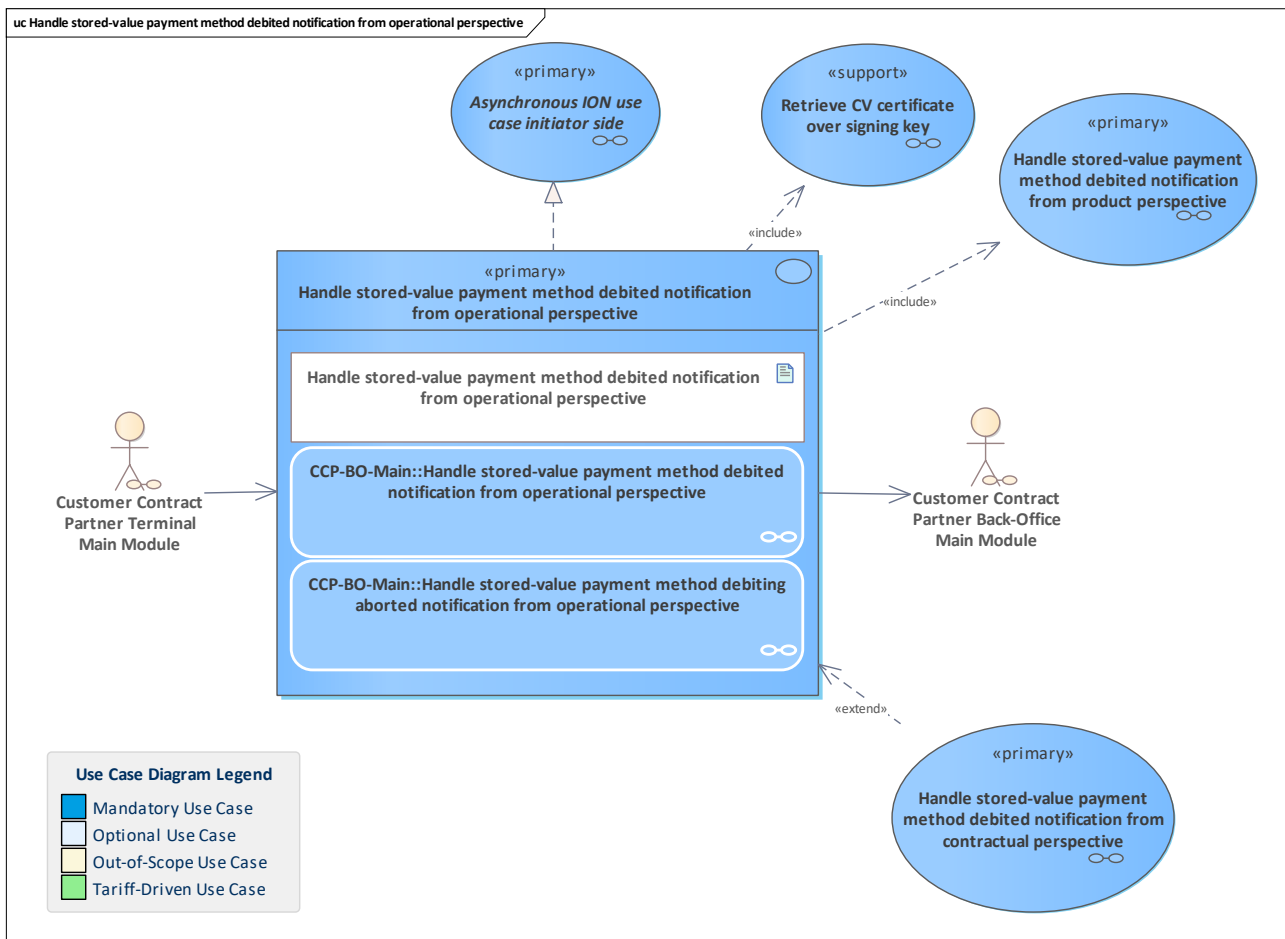


Figure 388: Handle stored-value payment method debited notification from operational perspective

Handle a notification about a stored-value payment method debiting from the operational perspective.

The CCP back-office system receives the notification about the debit transaction of a stored-value payment method and registers it.

Then it handles the notification from the operational perspective by performing checks and monitoring, e.g. verifying the signature of the debit transaction attestation.

Then, the notification is forwarded to the responsible PO system.

If the current CCP is the pCCP, it can also do the contractual checks and monitoring directly.

In case of a transaction abortion, the notification is registered for consistent monitoring. Since no counters are affected which are important to the PO, the notification is not forwarded.

11.234 Handle stored-value payment method debited notification from product perspective

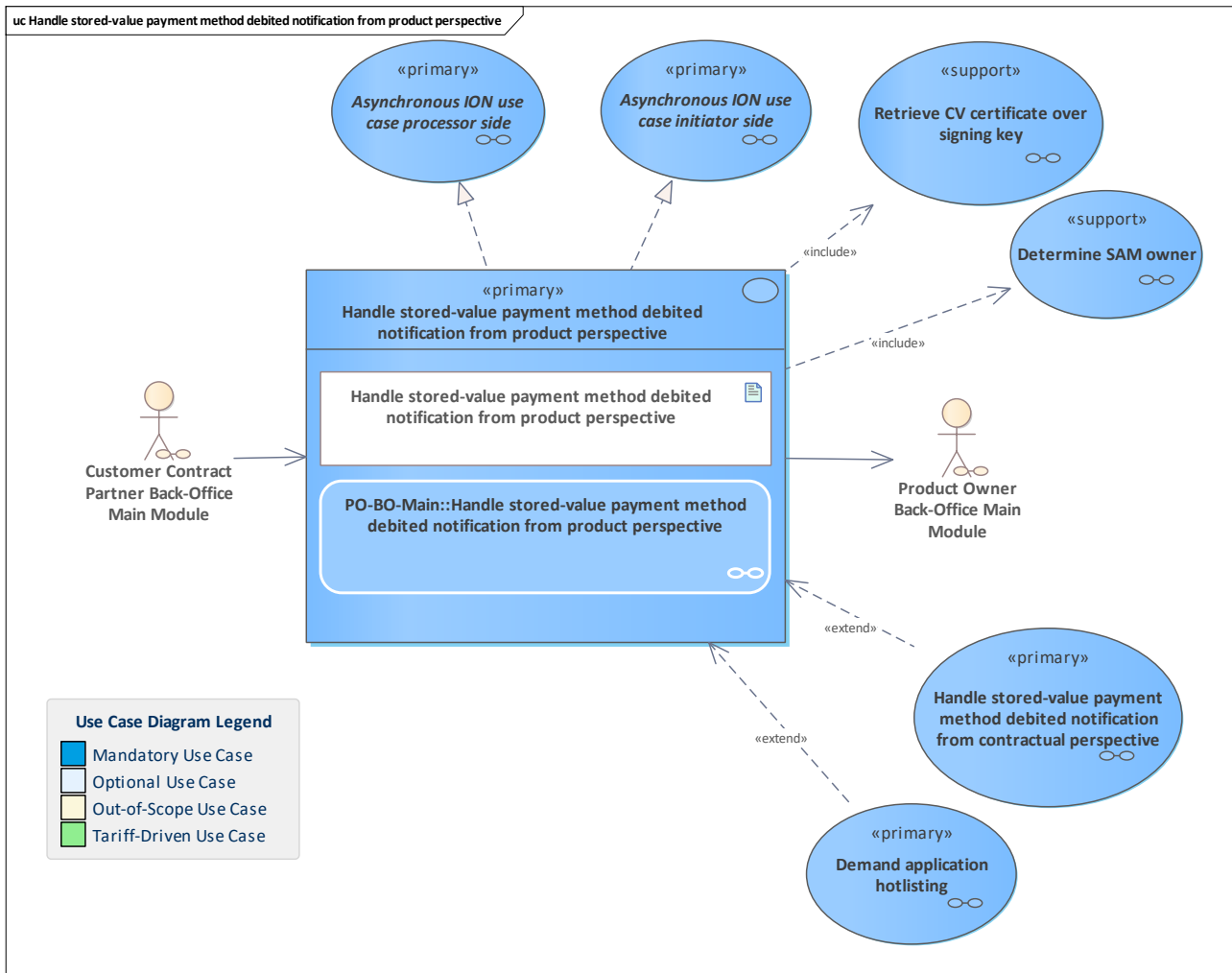


Figure 389: Handle stored-value payment method debited notification from product perspective

Handle a notification about a stored-value payment method debiting from the product owner perspective.

The PO back-office system receives from the CCP system and registers the notification about a performed debit with a stored-value payment method.

It does the checks and monitoring from the product owner perspective.

If the sending CCP was not the pCCP, the notification is forwarded to the responsible pCCP system.

11.235 Handle stored-value payment method recharged notification from contractual perspective

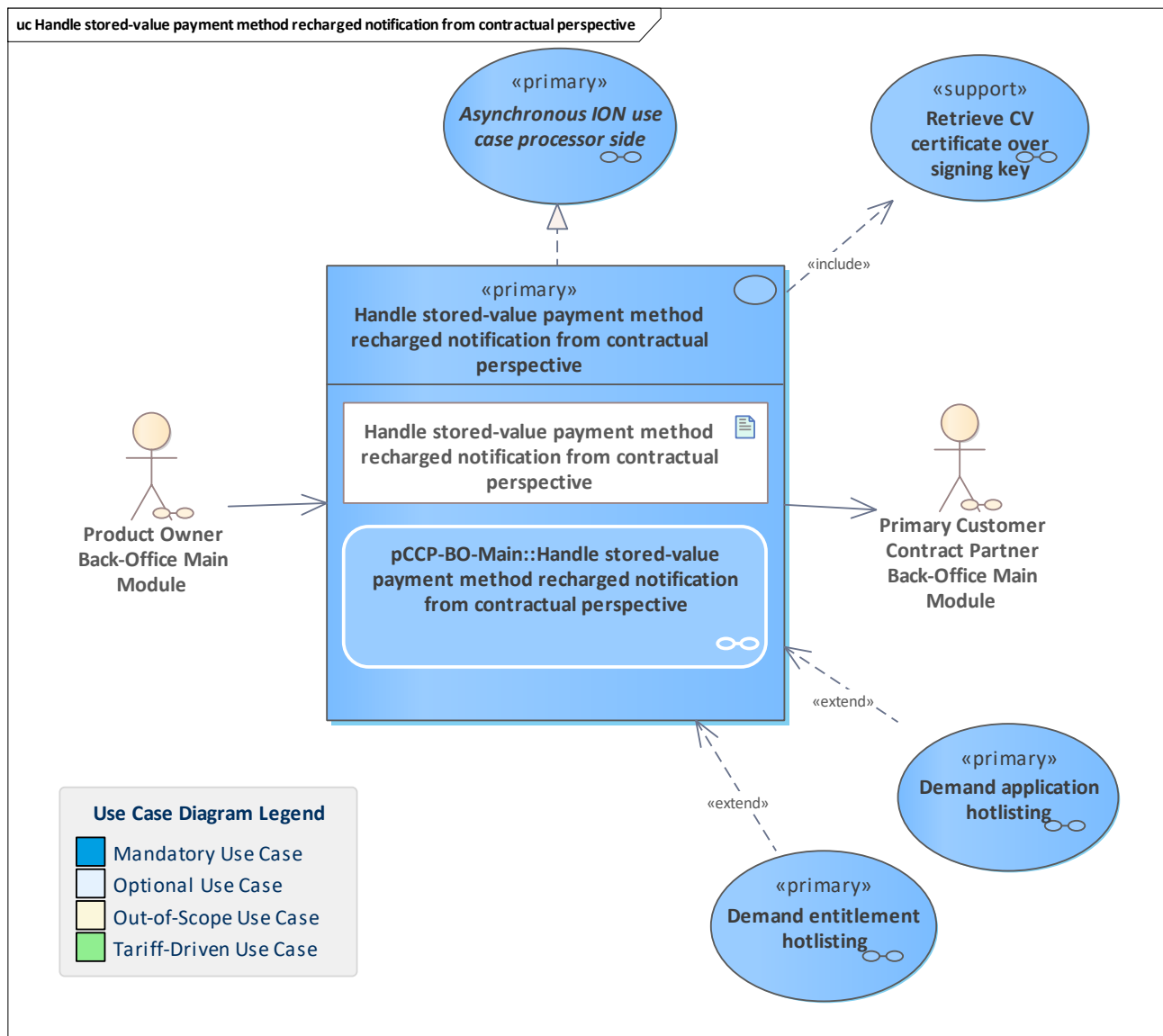


Figure 390: Handle stored-value payment method recharged notification from contractual perspective

Handle a stored-value payment method recharged notification from the contractual perspective. The pCCP does its contractual checks and monitoring.

Note: if the pCCP itself performed the recharge action, this use case takes place inside the use case [Handle stored-value payment method recharged notification from operational perspective](#) and is not called by the PO.

In this case, this use case is not an [Asynchronous ION use case processor side](#).

11.236 Handle stored-value payment method recharged notification from operational perspective

11.237 Handle stored-value payment method recharged notification from operational perspective

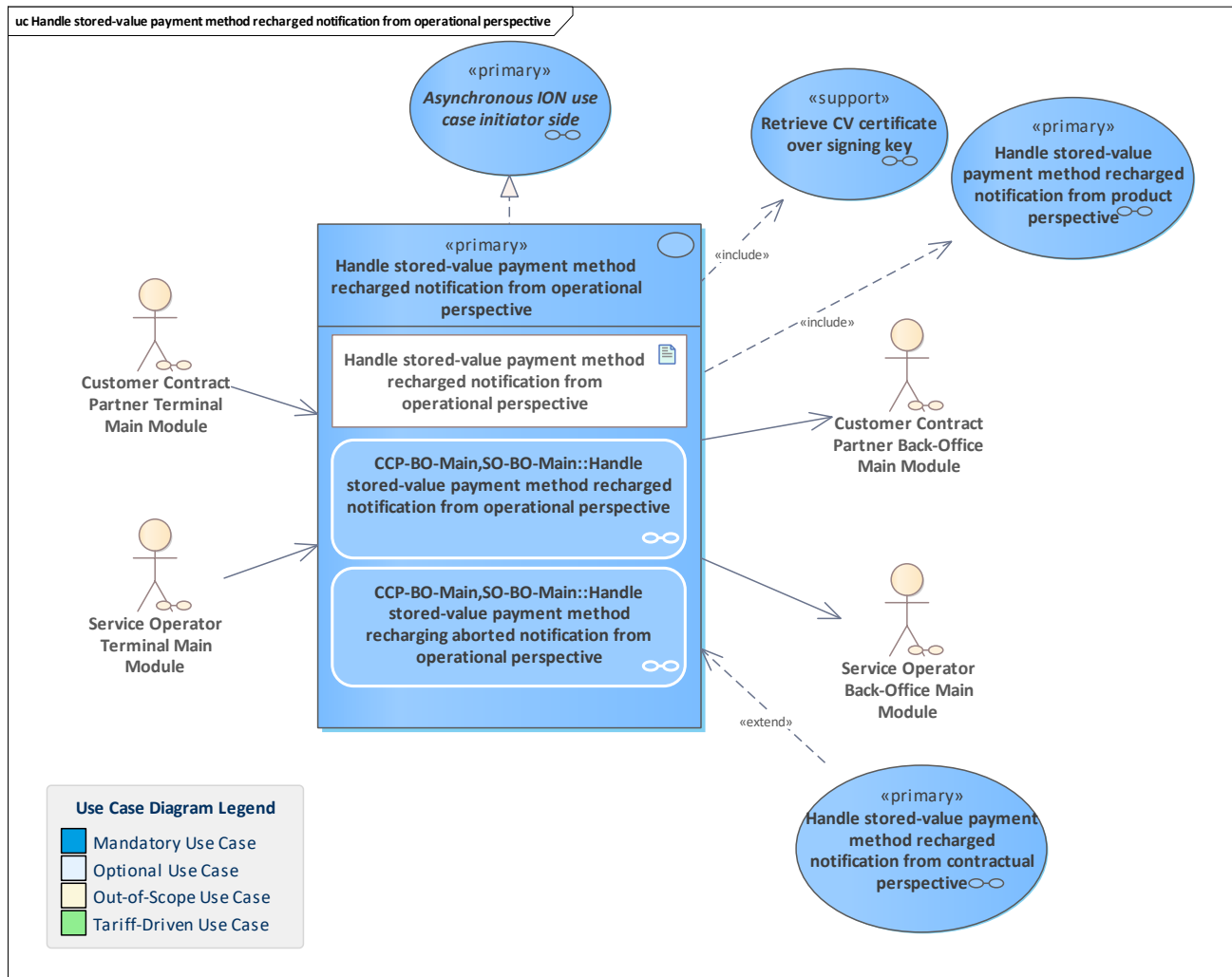


Figure 391: Handle stored-value payment method recharged notification from operational perspective

Handle a stored-value payment method recharged notification from the operational perspective. The CCP back-office system receives the notification about the recharge action of a stored-value payment method and registers it.

Then it handles the notification from the operational perspective by performing checks and monitoring, e.g. verifying the signature of the recharge action attestation.

Then, the notification is forwarded to the responsible PO system.

If the current CCP is the pCCP, it can also do the contractual checks and monitoring directly.

In the case of a transaction abortion, the notification is registered for consistent monitoring.

Since no counters are affected which are important to the PO, the notification is not forwarded.

11.238 Handle stored-value payment method recharged notification from product perspective

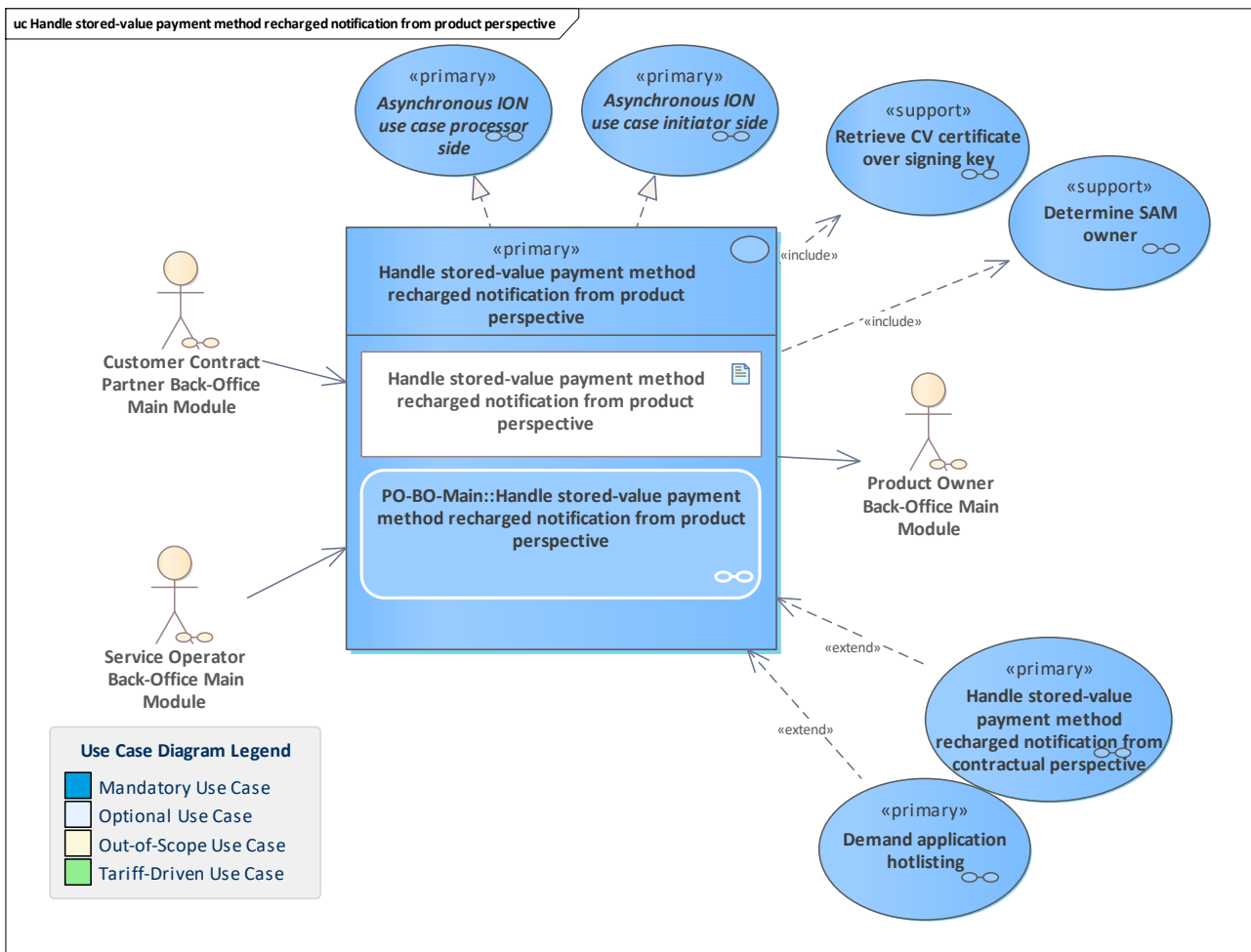


Figure 392: Handle stored-value payment method recharged notification from product perspective

Handle a stored-value payment method recharged notification from the product owner perspective.

The PO back-office system receives from the CCP system and registers the notification about a performed recharge with a stored-value payment method.

It does the checks and monitoring from the product owner perspective.

If the sending CCP was not the pCCP, the notification is forwarded to the responsible pCCP system.

11.239 Handle stored-value payment method reimbursed notification from contractual perspective

11.240 Handle stored-value payment method reimbursed notification from contractual perspective

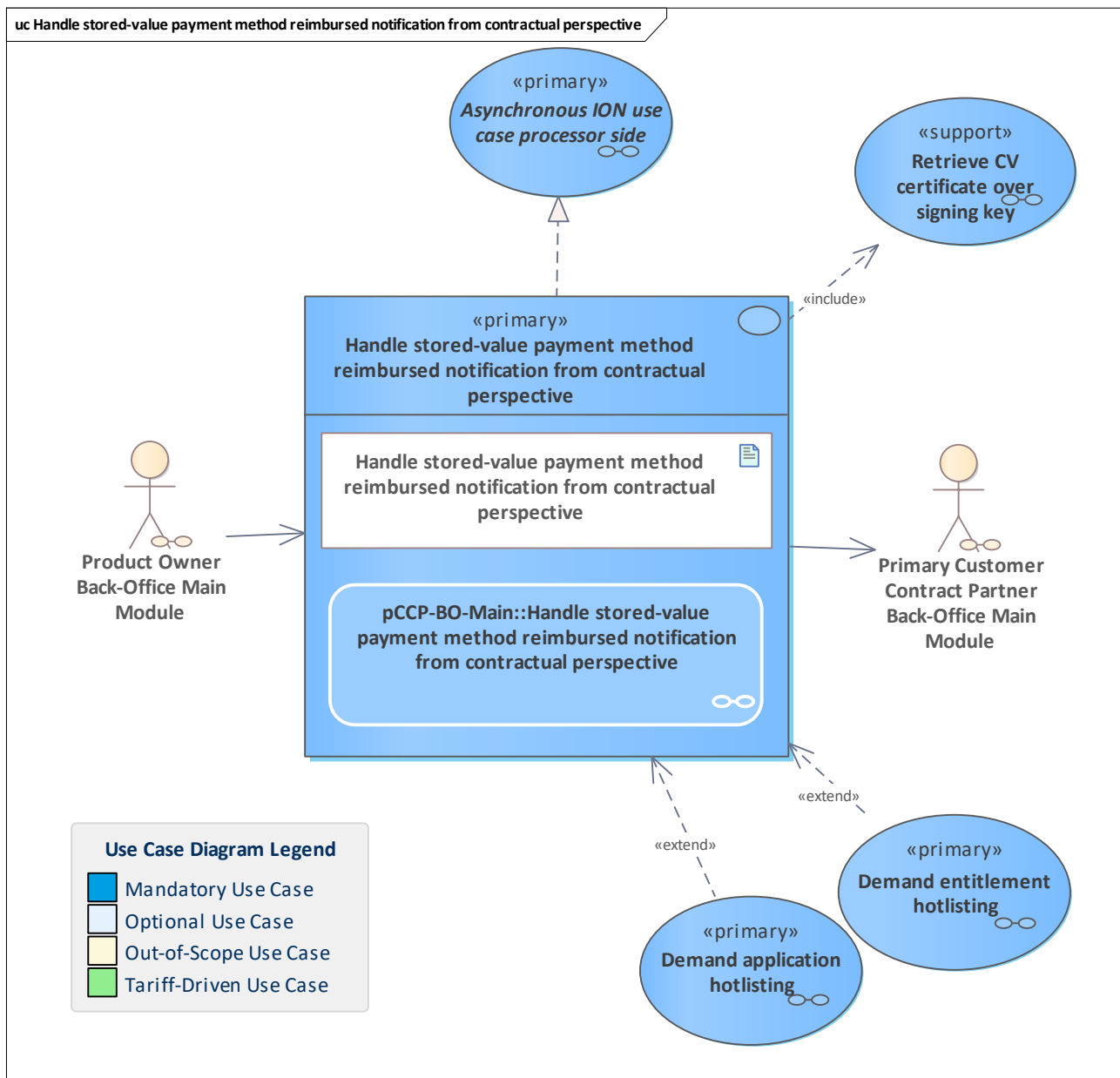


Figure 393: Handle stored-value payment method reimbursed notification from contractual perspective

Handle a stored-value payment method reimbursed notification from the contractual perspective.

The pCCP does its contractual checks and monitoring.

Note: if the pCCP itself performed the reimburse action, this use case takes place inside the use case [Handle stored-value payment method reimbursed notification from operational perspective](#) and is not called by the PO.

In this case, this use case is not an [Asynchronous ION use case processor side](#).

11.241 Handle stored-value payment method reimbursed notification from operational perspective

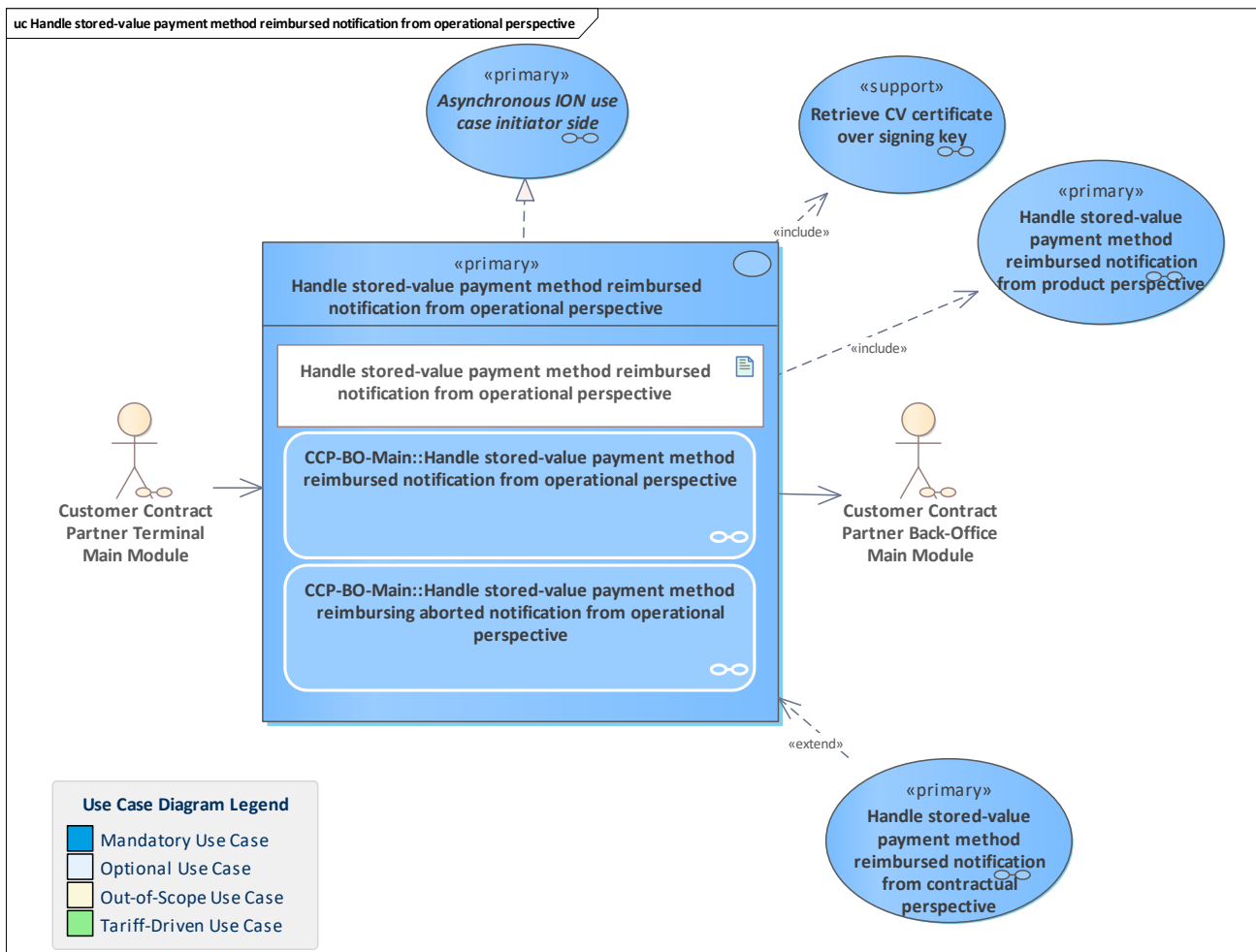


Figure 394: Handle stored-value payment method reimbursed notification from operational perspective

Handle a stored-value payment method reimbursed notification from the operational perspective.

The CCP back-office system receives the notification about the reimburse action of an stored-value payment method and registers it.

Then it handles the notification from the operational perspective by performing checks and monitoring, e.g. verifying the signature of the reimburse action attestation.

Then, the notification is forwarded to the responsible PO system.

If the current CCP is the pCCP, it can also do the contractual checks and monitoring directly.

In the case of a transaction abortion, the notification is registered for consistent monitoring.

Since no counters are affected which are important to the PO, the notification is not forwarded.

11.242 Handle stored-value payment method reimbursed notification from product perspective

11.243 Handle stored-value payment method reimbursed notification from product perspective

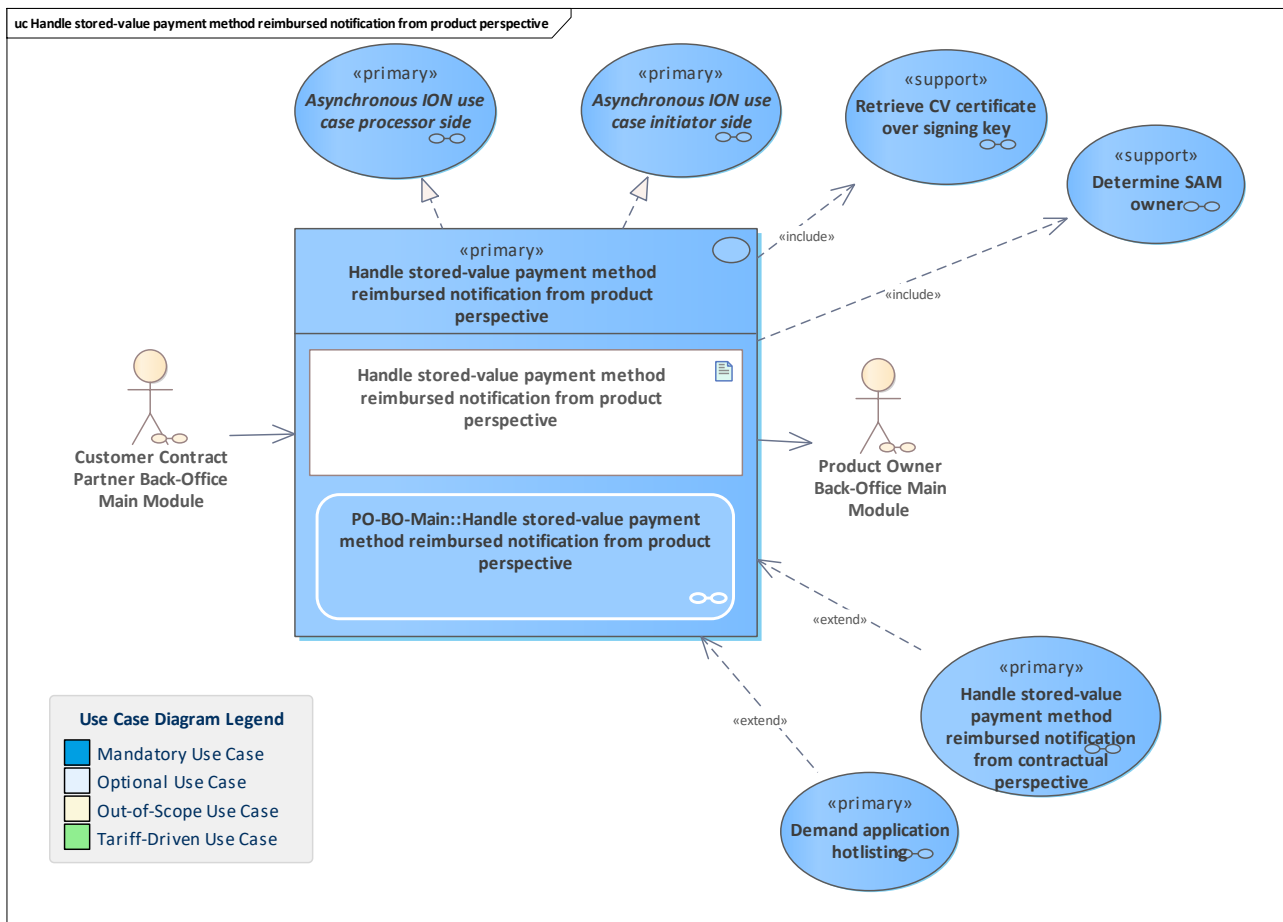


Figure 395: Handle stored-value payment method reimbursed notification from product perspective

Handle a stored-value payment method reimbursed notification from the product owner perspective.

The PO back-office system receives from the CCP system and registers the notification about a performed reimburse with a stored-value payment method.

It does the checks and monitoring from the product owner perspective.

If the sending CCP was not the pCCP, the notification is forwarded to the responsible pCCP system.

11.244 Handle user tariff parameters changed notification from contractual perspective

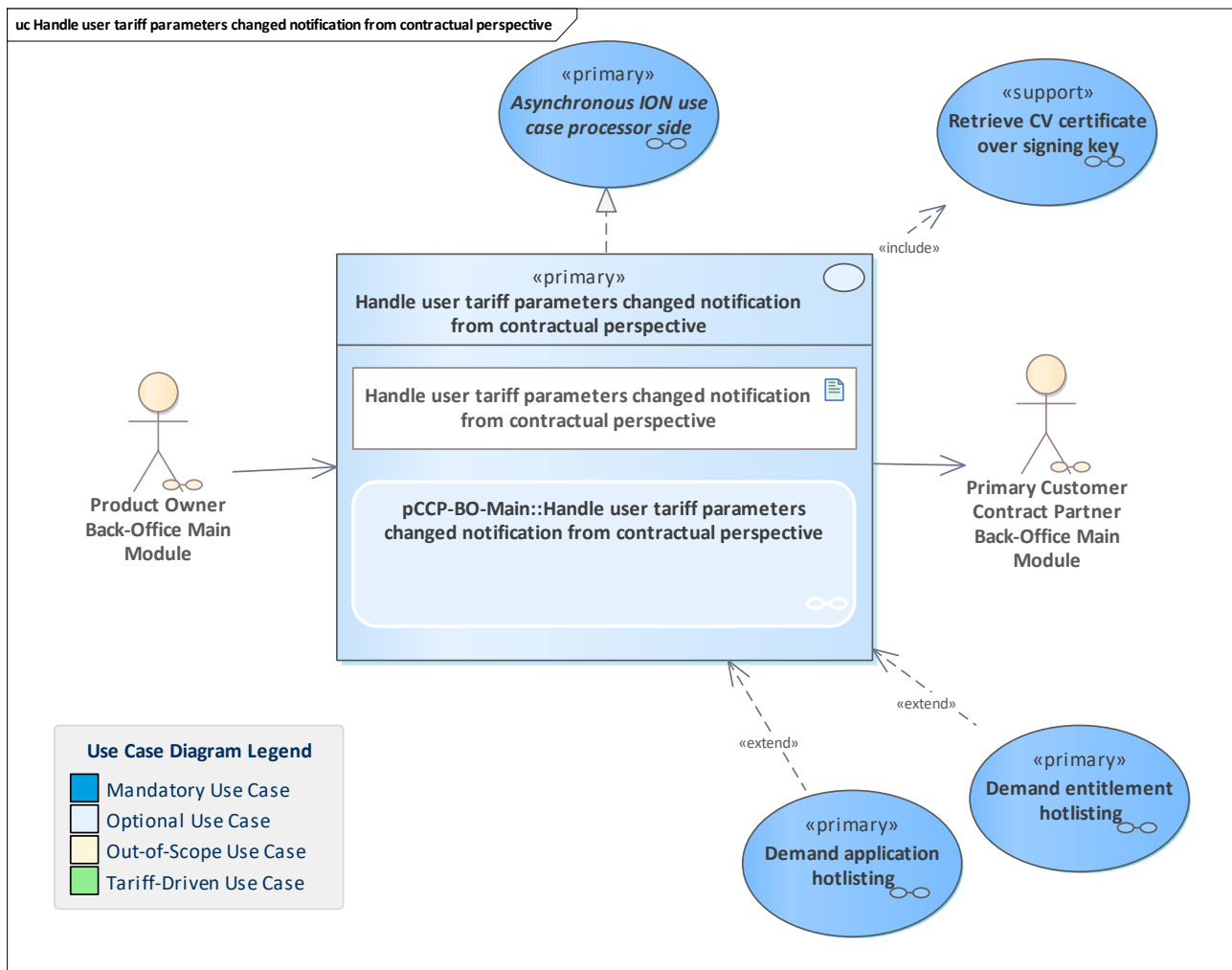


Figure 396: Handle user tariff parameters changed notification from contractual perspective

Handling a notification regarding changed user tariff parameters from the contractual perspective of the pCCP.

The entitlement changed user tariff parameters notification is received by the pCCP. The pCCP does its checks and monitoring from the contractual perspective regarding the correct execution. In this context, the signature of the embedded attestation is verified.

Note: if the pCCP itself performed the changing of the user tariff parameters, this use case takes place inside the use case [Handle user tariff parameters changed notification from operational perspective](#) and is not called by the PO. In this case, this use case is not an [Asynchronous ION use case processor side](#).

11.245 Handle user tariff parameters changed notification from operational perspective

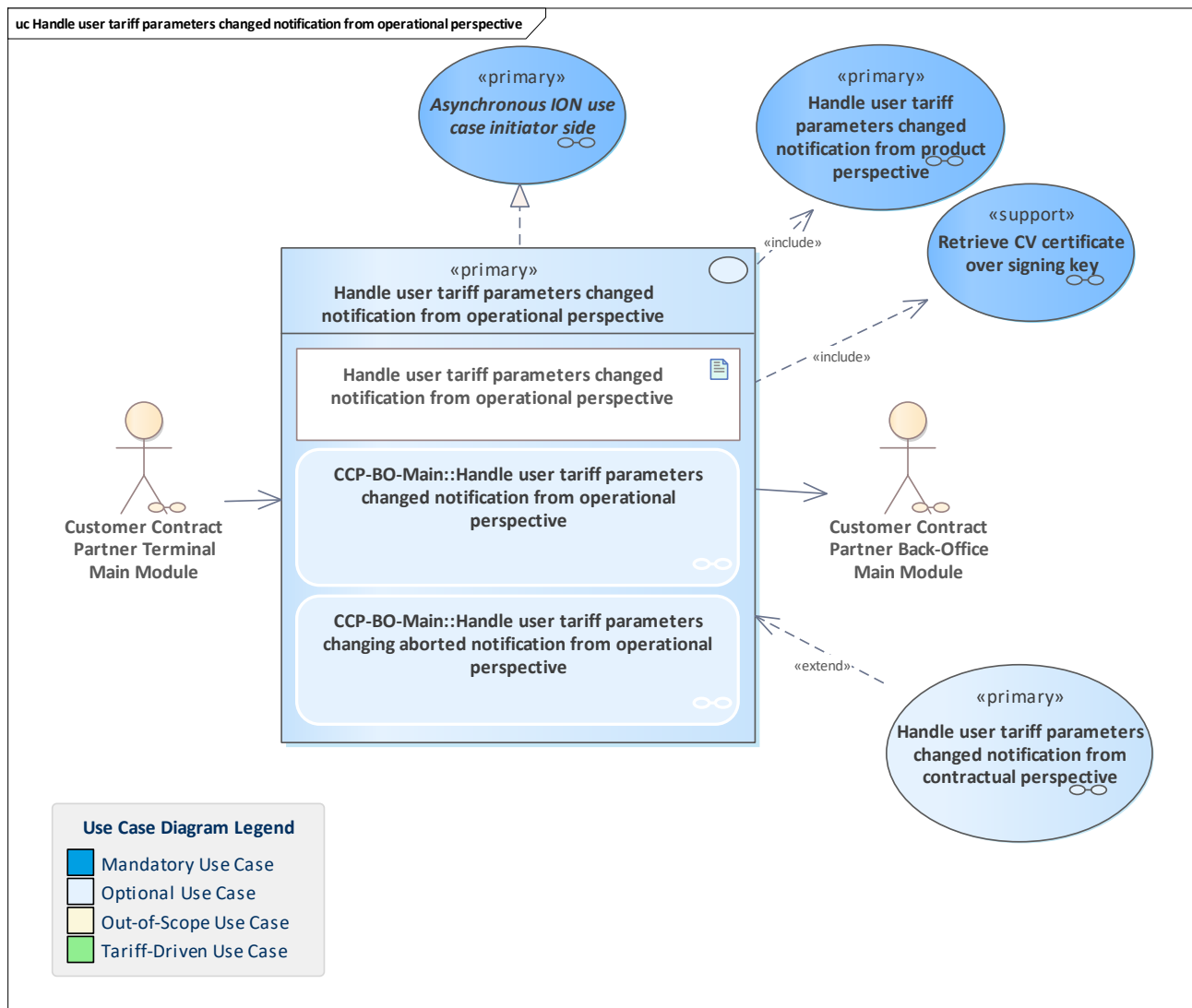


Figure 397: Handle user tariff parameters changed notification from operational perspective

Handle notification regarding changed user tariff parameters of an entitlement from the operational perspective.

The changed user tariff parameters notification is sent by the CCP terminal to the CCP back-office system. The notification will be checked and monitored, e.g. by verifying the signature of the embedded attestation.

If the pCCP has changed user tariff parameters for its own entitlement:

- the notification will be sent to the PO
- the pCCP does its contractual checks and monitoring

If an sCCP has changed user tariff parameters for the entitlement:

- the notification will be sent to the PO (and the PO will forward it later to the pCCP)

In the case of action abortion, terminal and SAM action data is sent by the terminal to the back-office system. The CCP back-office system registers the abortion for internal monitoring and data consistency.

11.246 Handle user tariff parameters changed notification from product perspective

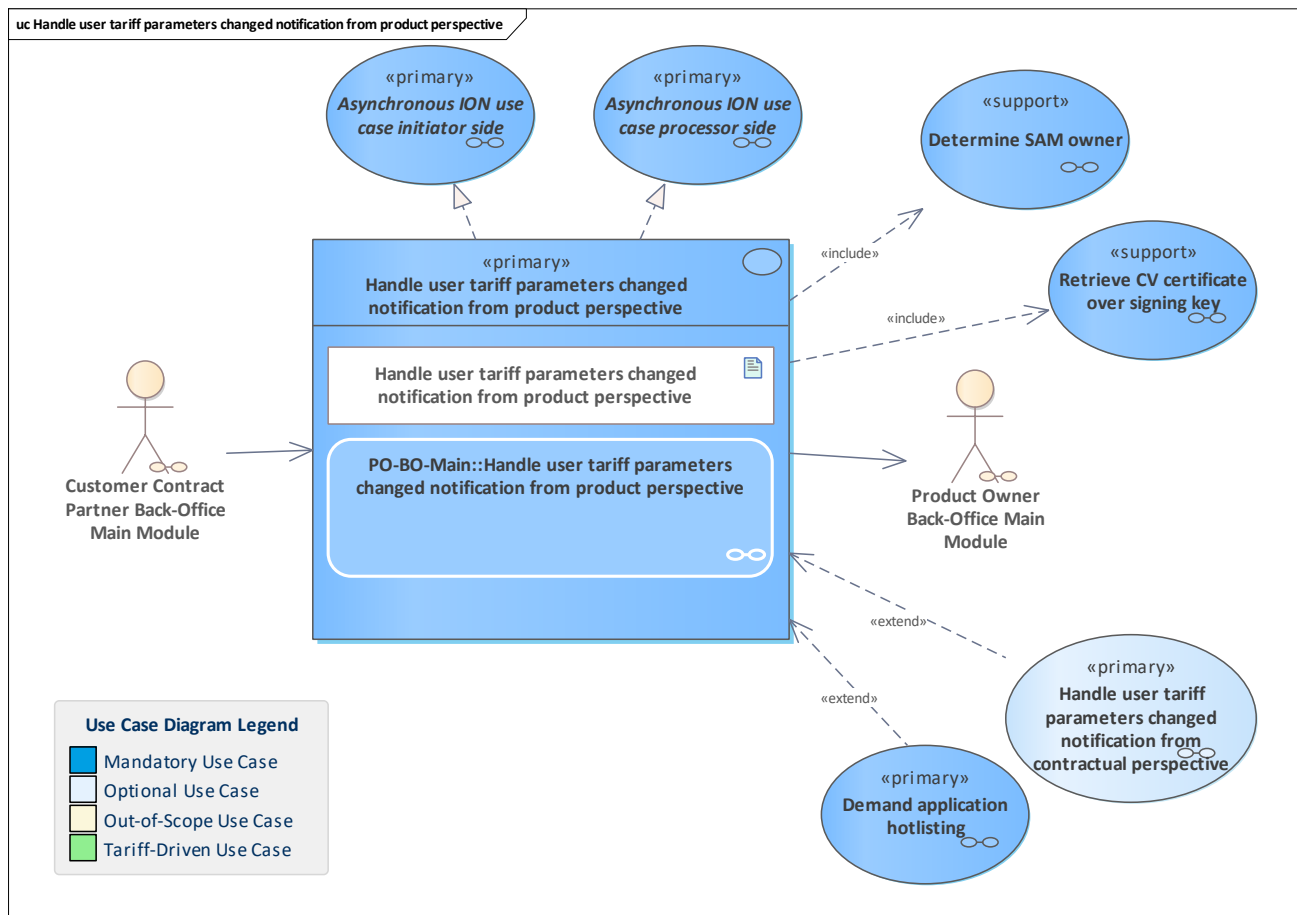


Figure 398: Handle user tariff parameters changed notification from product perspective

Handle an changed user tariff parameters notification from the product owner perspective. The entitlement changed user tariff parameters notification is received by the PO. The PO registers the changed user tariff parameters notification and does its checks and monitoring from the product perspective regarding the correct execution. In this context, the signature of the embedded attestation is verified and the SAM owner of the SAM that performed the changing of the user tariff parameters is determined.

- if the sender is a sCCP: forward the notification to the pCCP
- if the sender is the pCCP: do not forward the notification. In this case, the current use case is not an asynchronous ION use case as an initiator.

Note: in the ION context, the use case is asynchronous as processor (process the notification) and an asynchronous use case as initiator due to the asynchronous call to the pCCP.

11.247 Handle verification request for action list updated via increments

11.248 Handle verification request for action list updated via increments

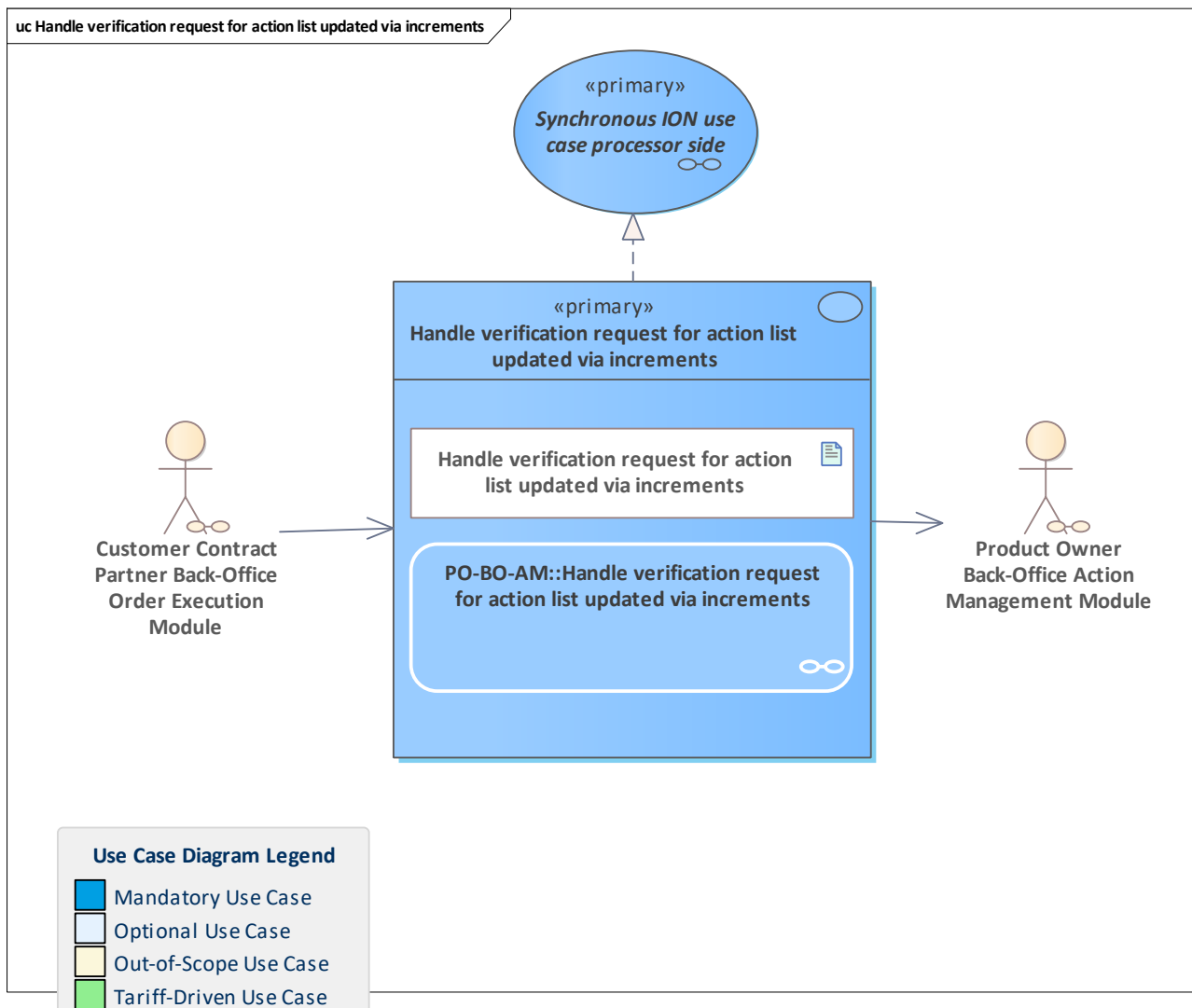


Figure 399: Handle verification request for action list updated via increments

The [Product Owner Back-Office Action Management Module](#) verifies that the given checksum matches the calculated one of the total list for the given cycle.

May be used after an incremental action list was incorporated into a system's inventory of action list entries to verify that this inventory is consistent with the full list that would have been provided by this module.

See also [Checksum calculation for hotlist and action list verification](#) and [Example calculation for an action list inventory](#).

11.249 Handle verification request for application hotlist updated via increments

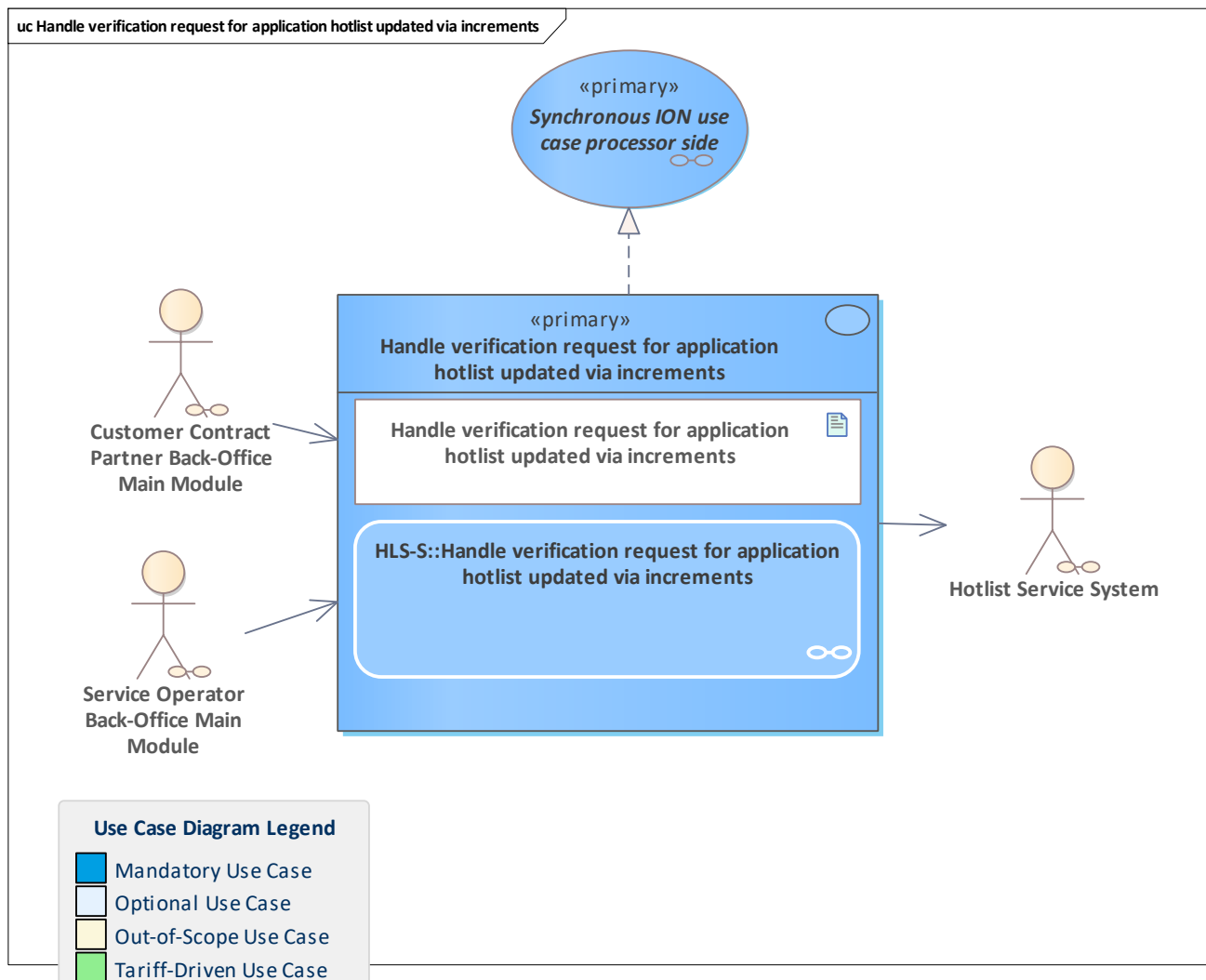


Figure 400: Handle verification request for application hotlist updated via increments

The hotlist service system (HLS-S) retrieves an application hotlist list cycle reference together with the caller's computed checksum in the input and computes the check sum over the full application hotlist that is referenced by list cycle in the request. The HLS-S asserts the equality of the internally and externally computed checksum and reports the result back to the requestor. See also [Checksum calculation for hotlist and action list verification](#) and [Example calculation for an application hotlist inventory](#).

11.250 Handle verification request for entitlement hotlist updated via increments

11.251 Handle verification request for entitlement hotlist updated via increments

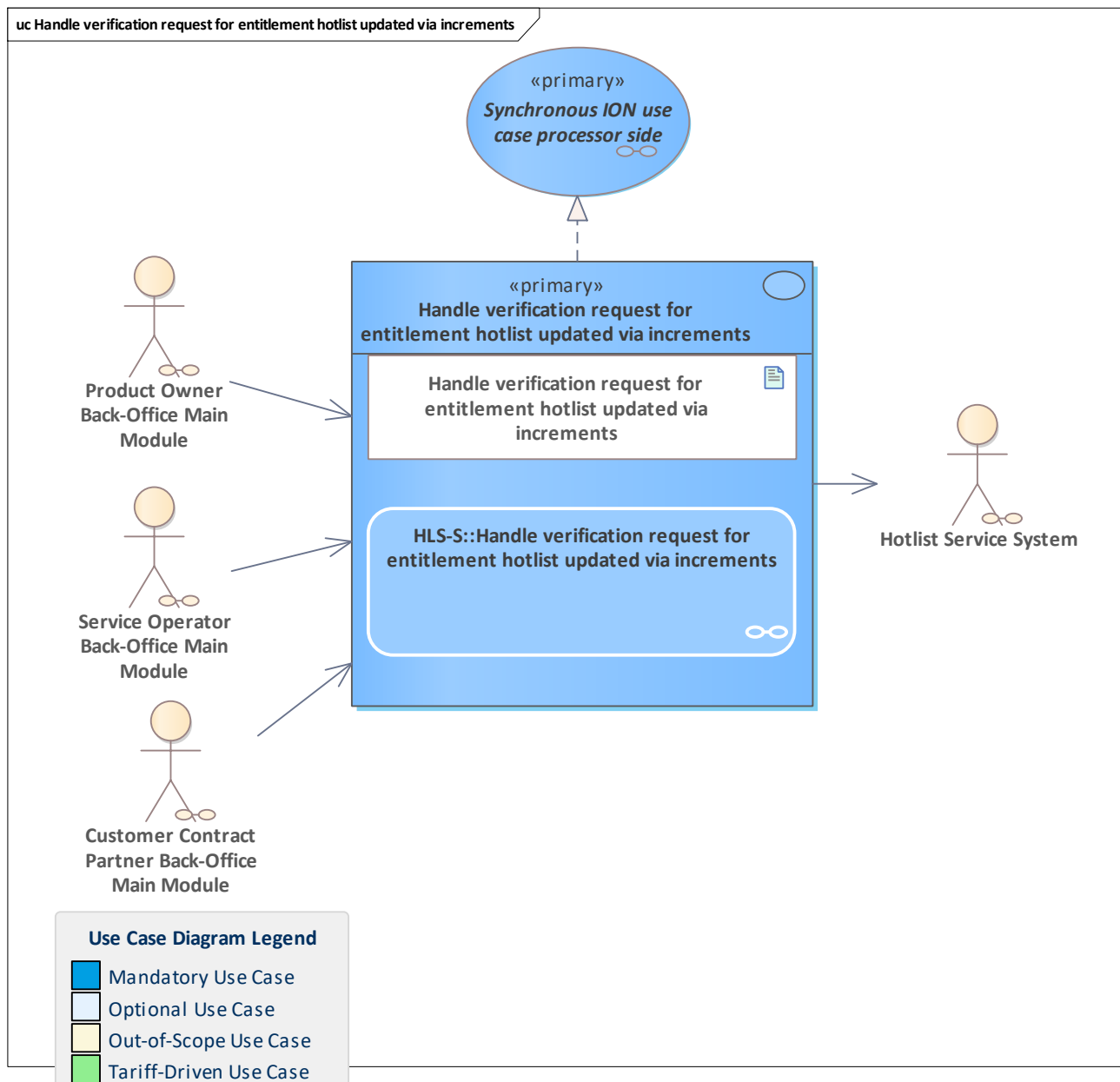


Figure 401: Handle verification request for entitlement hotlist updated via increments

The HLS-S retrieves an entitlement hotlist reference together with the caller's computed checksum in the input and computes the check sum over the full entitlement hotlist that belongs to the reference of the request.

The HLS-S asserts the equality of the internally and externally computed checksum and reports the result back to the requestor. See also [Checksum calculation for hotlist and action list verification](#) and [Example calculation for an entitlement hotlist inventory](#).

11.252 Initialise password

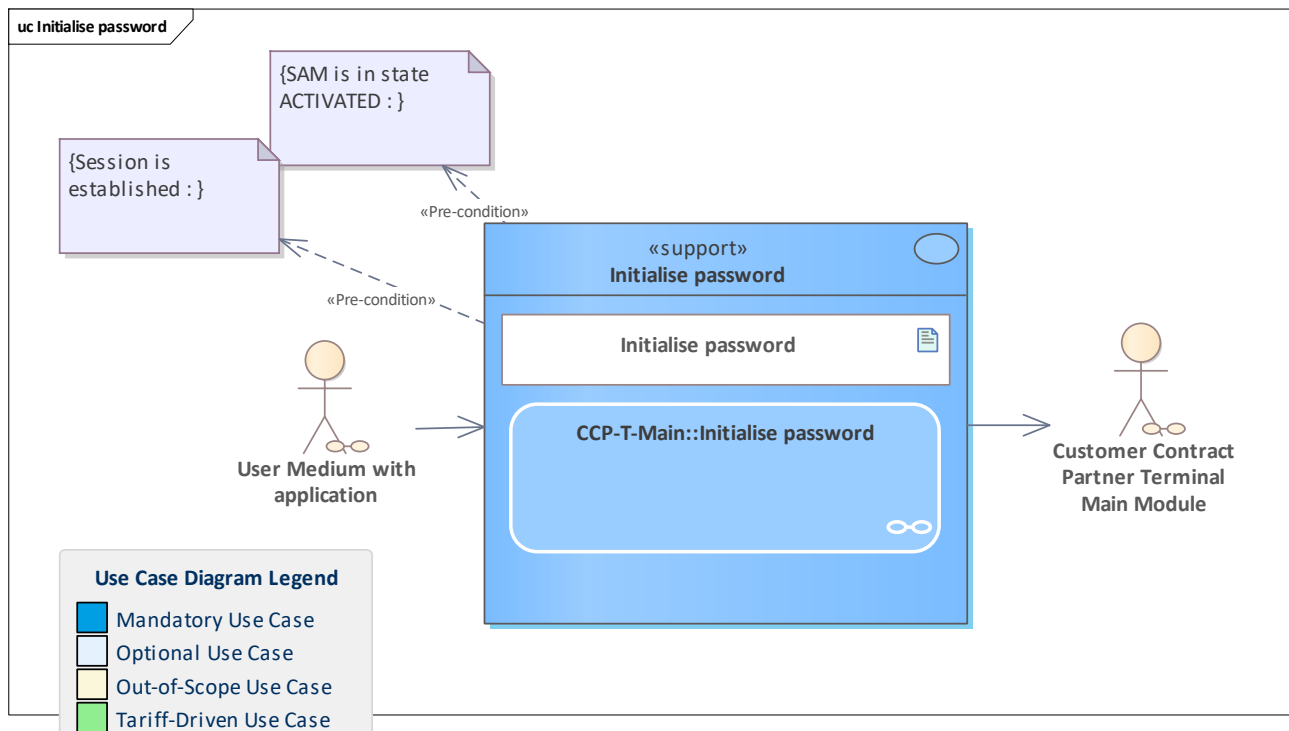


Figure 402: Initialise password

The terminal initialises the password for the user medium application.

11.253 Initialise User Medium with application for customer

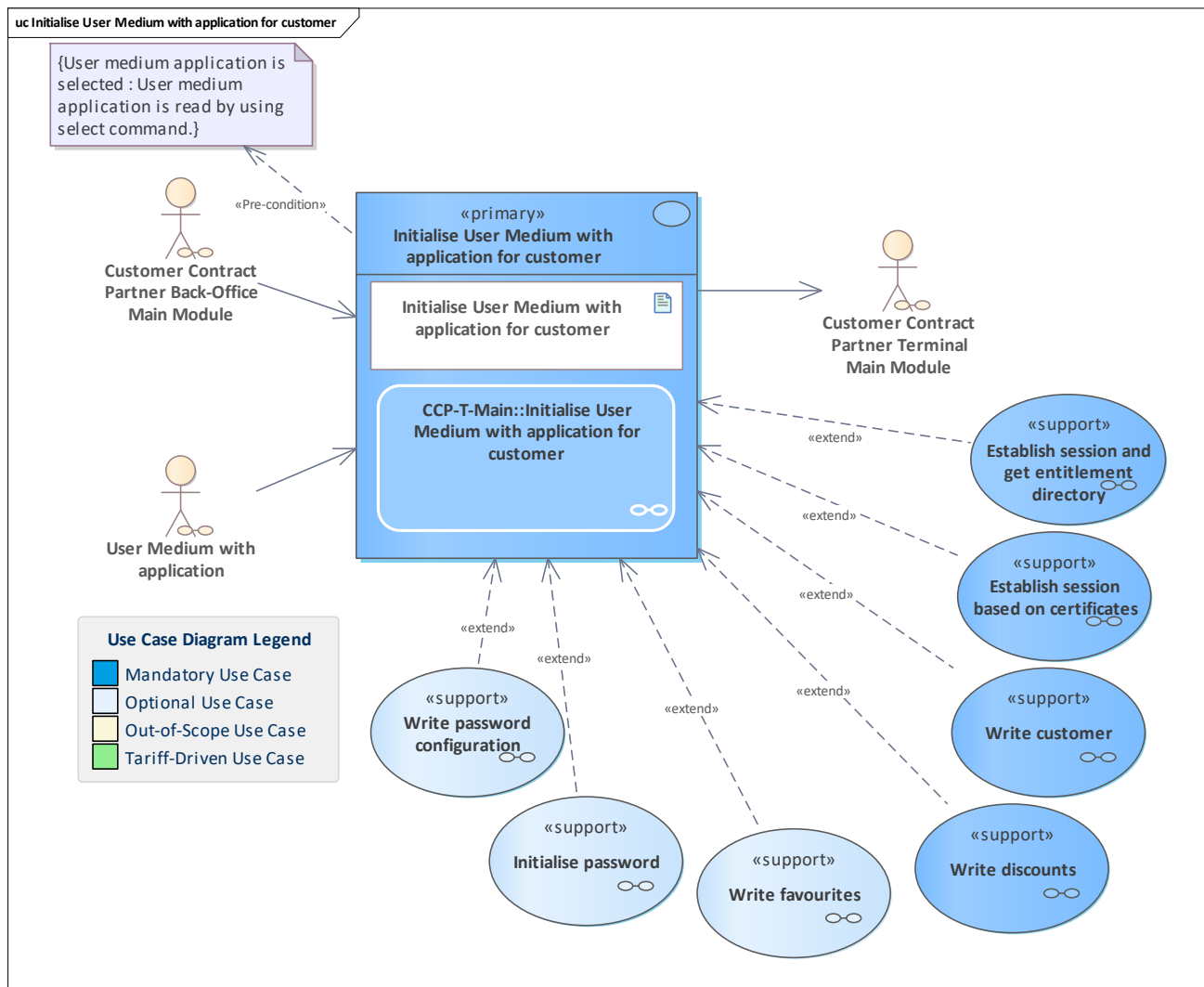


Figure 403: Initialise User Medium with application for customer

This process may write customer-related information as needed to a fully configured user medium application:

- customer
- discounts
- favourites
- password configuration
- the initial password

Note: if the password or its configuration shall be set, the session needs to be established based on certificates, otherwise a symmetric key-based session establishment suffices.

11.254 Inspect user medium with application

11.255 Inspect user medium with application

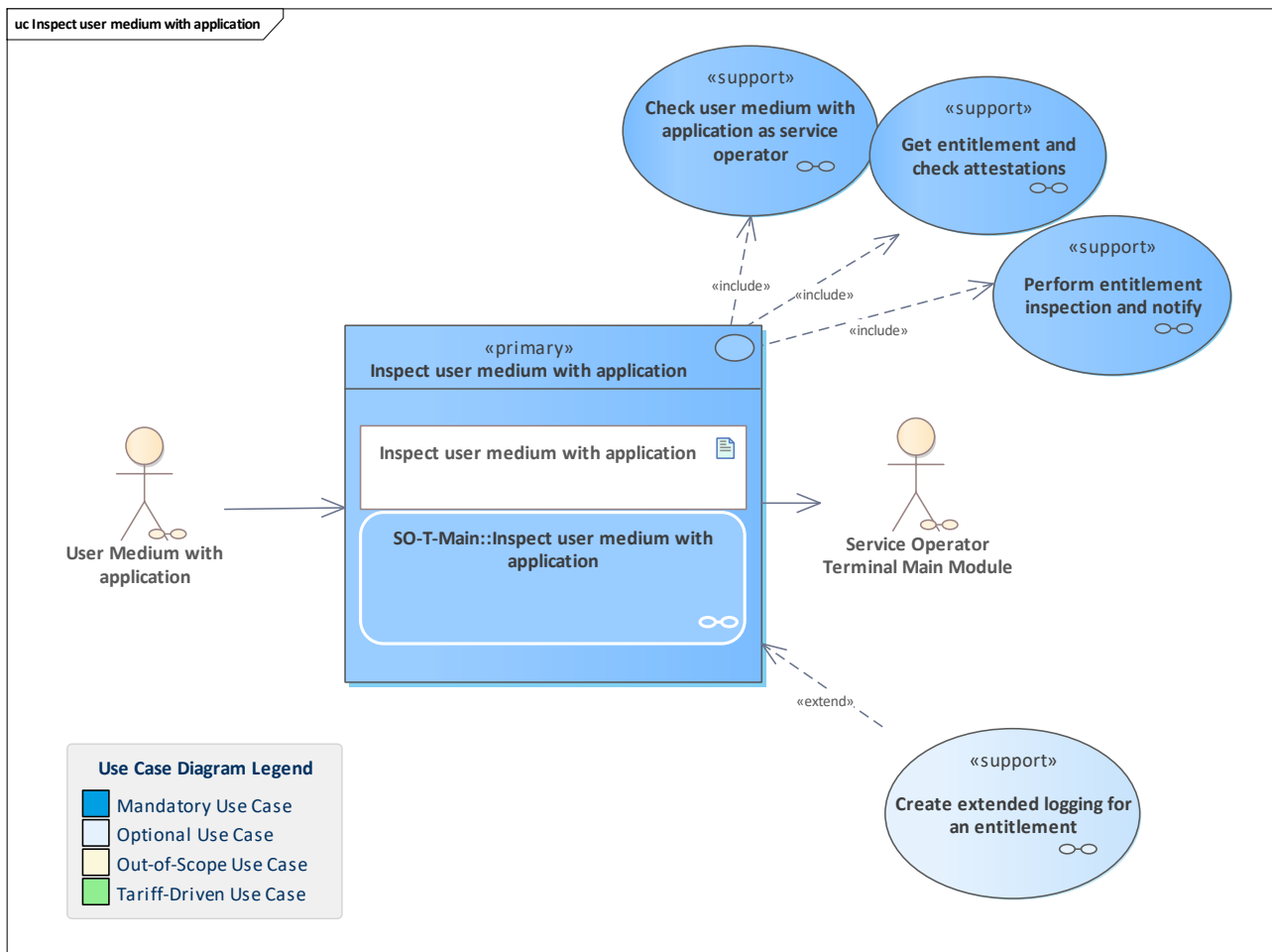


Figure 404: Inspect user medium with application

This primary use case describes the verification of the existence and validity of an entitlement/application.

The terminal reads the application and entitlements on a user medium and inspects them based on the predefined requirements.

An inspection attestation is created by the terminal for each read entitlement and the involved actors are notified about the inspection attestation.

Meanwhile, extended logging is created in case of e.g. invalid application/entitlement for monitoring purposes.

Please note that if after filtering there are not any available entitlements left for a check, the customer is not entitled to travel. Modelling that there might not be any available entitlements can complicate the model. For the sake of simplicity, it is not modelled.

11.256 Inspect user medium without application

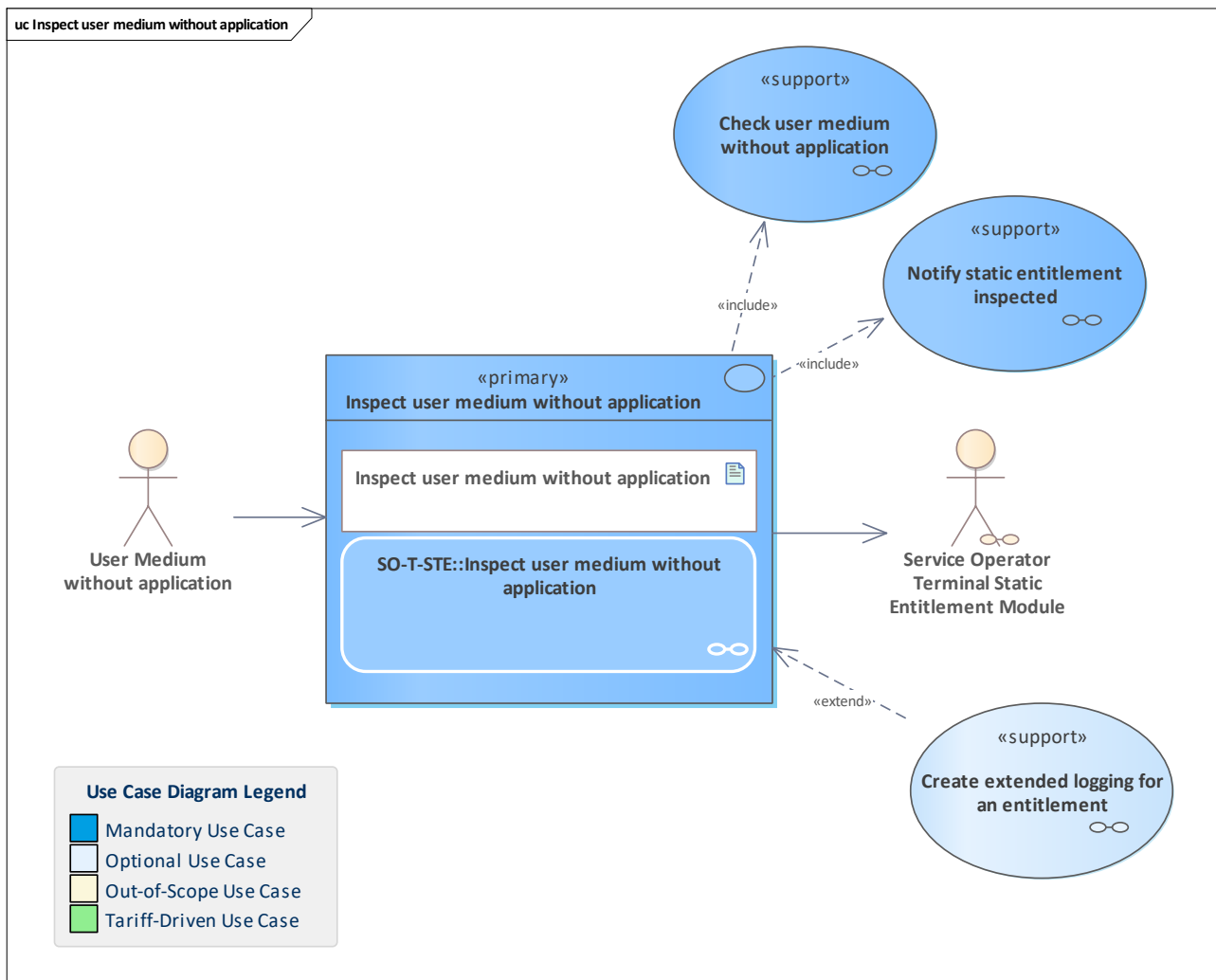


Figure 405: Inspect user medium without application

Use case to inspect one or more static entitlements in an SO terminal with a static entitlement extension. The entitlement(s) are checked for validity and the result is notified to the responsible SO back-office system with static entitlement extension.

11.257 Issue entitlement

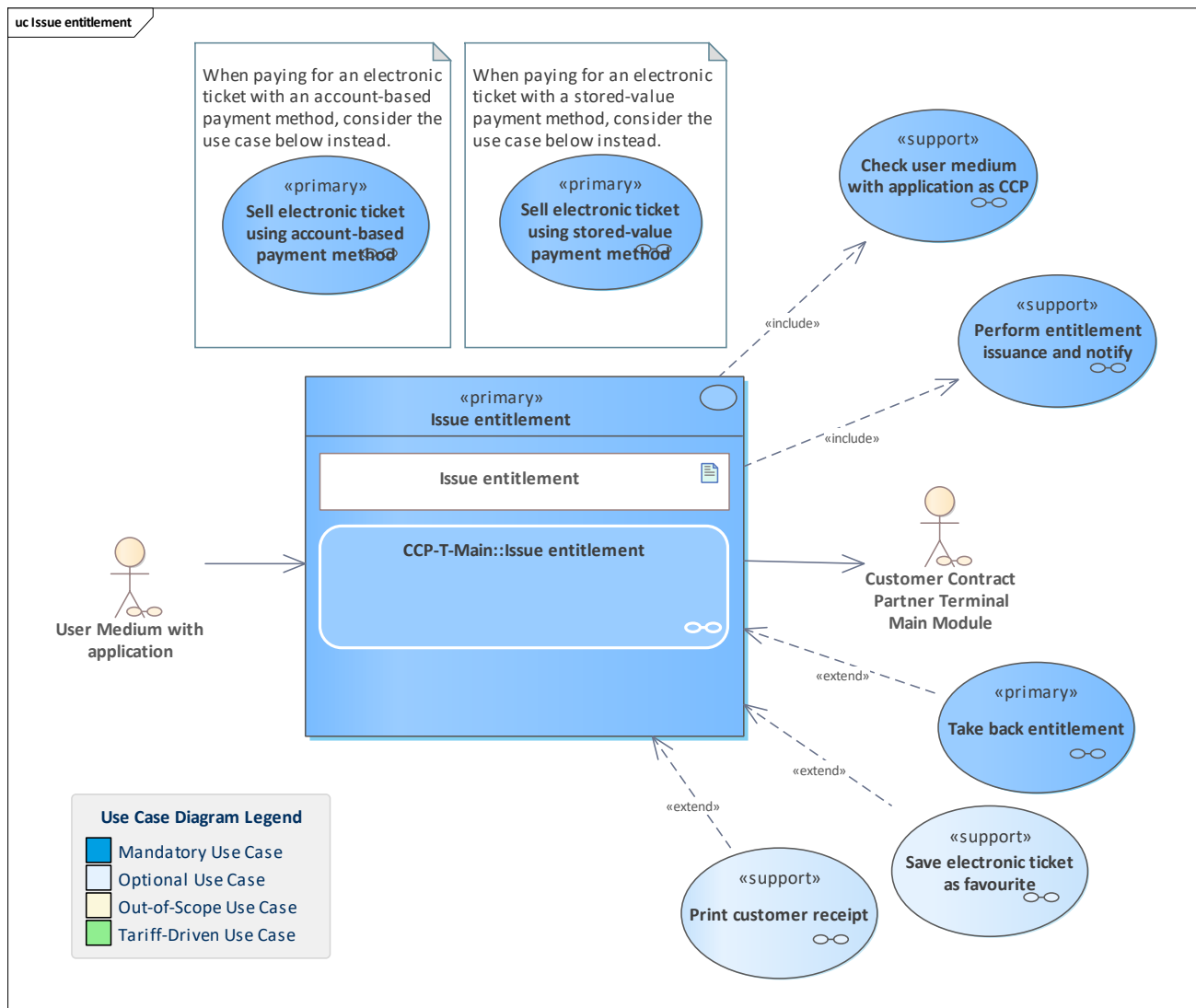


Figure 406: Issue entitlement

An entitlement is issued to a user medium. This can be an electronic ticket or a payment method.

Note: if a stored-value payment method is issued and the current balance is not equal to zero, the payment means involved in the implicit recharging must match the ones given in the product parameters and may additionally be given as part of the entitlement issued metadata.

11.258 Issue entitlement triggered by action order

11.259 Issue entitlement triggered by action order

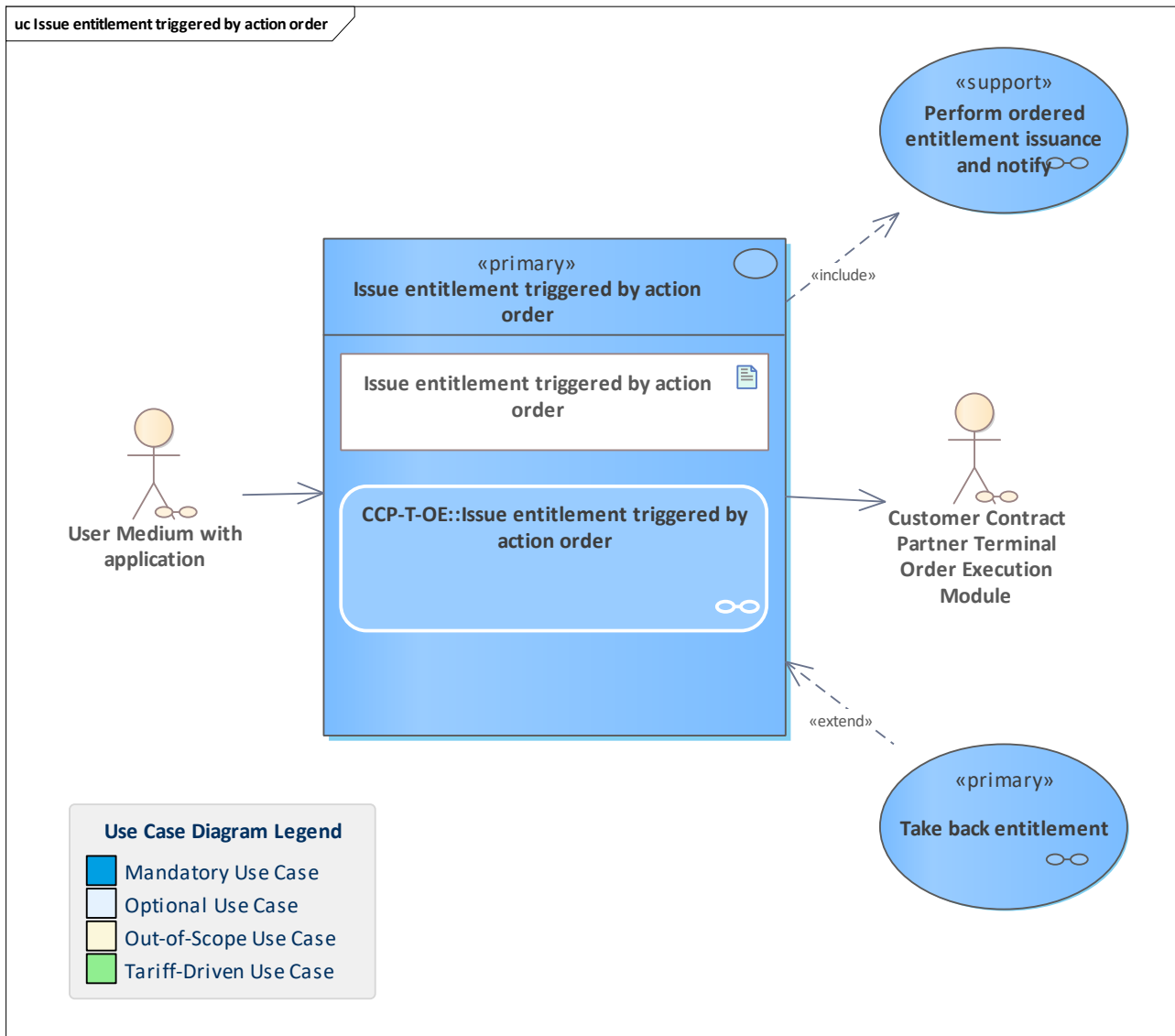


Figure 407: Issue entitlement triggered by action order

An entitlement issuance order potentially relevant for a given user medium is checked regarding the need to execute it and is executed, if necessary. The issuance process is similar to the regular issuance process without action management. Additionally, the order ID is written into the issued entitlement to avoid further issuance attempts in other terminals. The terminal does a lookup in the action list for the application instance ID. To avoid a duplicate issuance, the user medium has to be examined, if an entitlement with the same order ID already exists (which was written previously into the entitlement).

11.260 Issue static entitlement

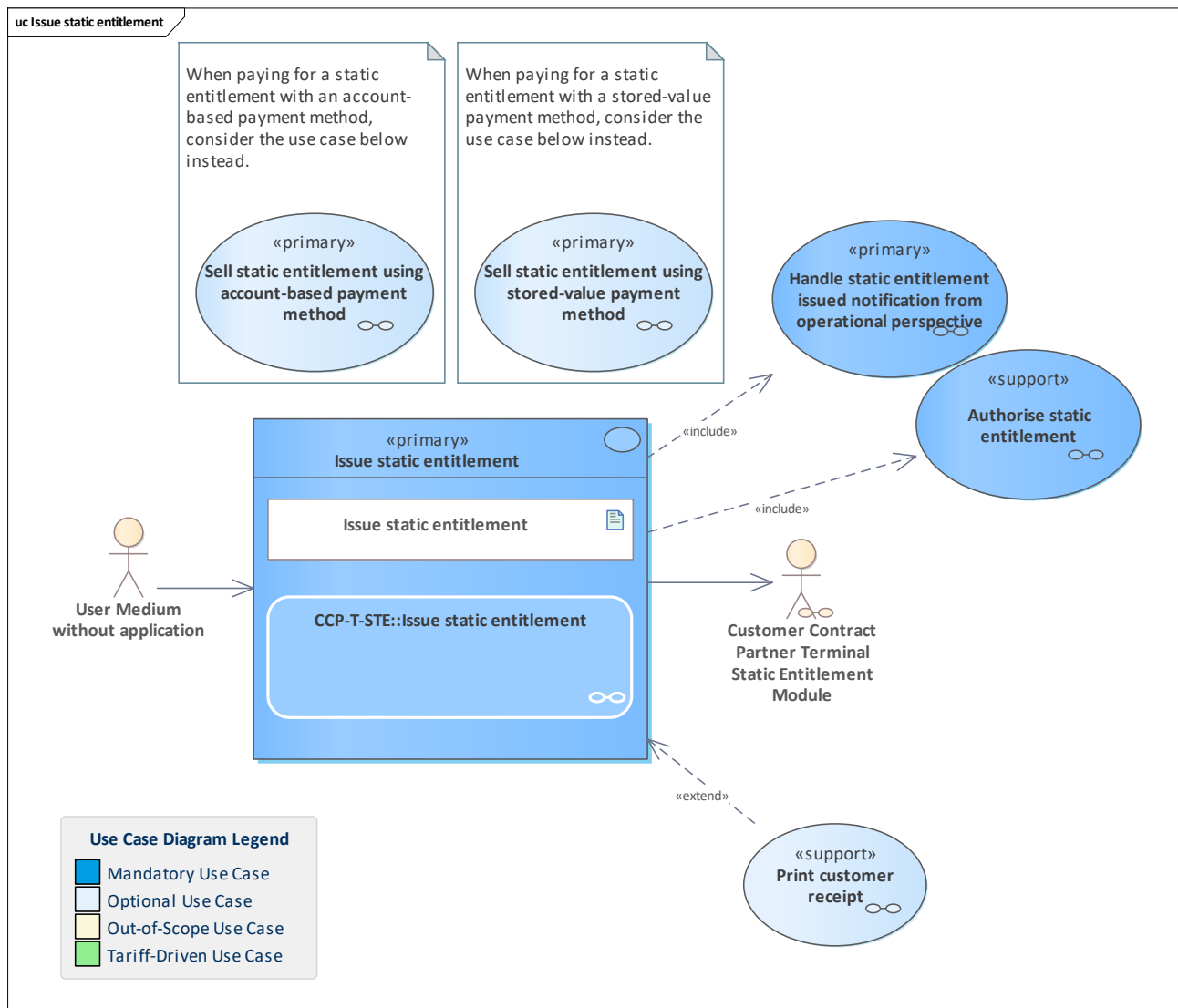


Figure 408: Issue static entitlement

Issue a static entitlement and notify the back-office system.

Note: this use case only issues a single static electronic ticket. In case multiple such issuances should be performed in the same context, the [StaticEntitlements](#) structures can be merged by combining the contained [StaticEntitlementData](#) objects, thus deduplicating the SAM certificate. This merging can either happen before delivering the data objects to the target (e.g. if it is a printer) or afterwards (e.g. if it is a smartphone). The back-office systems receive their notifications for every single static electronic ticket (i.e. [StaticEntitlementData](#) object), however.

11.261 Log defective user medium with application

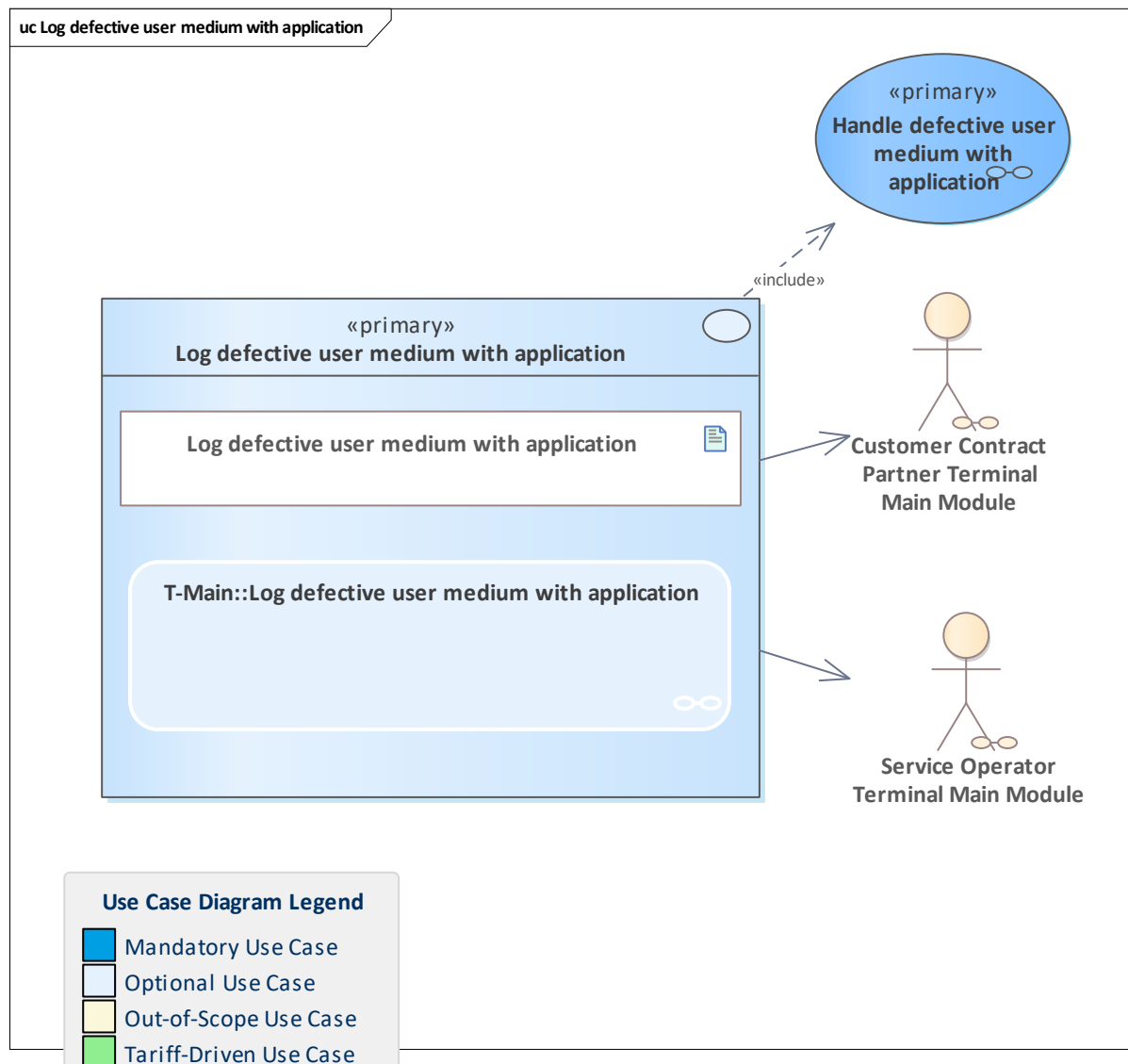


Figure 409: Log defective user medium with application

If a user medium with an application is defective, the terminal action data and medium ID are registered and sent to the back-office system.

The recording of the media ID must be supported by person-operated terminals in such a way that the organisation ID of the card manufacturer which individualises cards in the region can be pre-set in the terminal. It must also be possible to manually enter another organisation ID or alphanumeric characters (possibly the case for foreign media), rather than use the pre-set organisation ID. The pre-set ID must, therefore, be able to be overwritten!

Separate entry fields are recommended for the string before the first hyphen (registration authority as organisation ID) and the string after the first hyphen (subject number as an integer, including the checksum digit, separated by a hyphen).

The terminals must ensure error-tolerant input processing of the media ID, with or without full stops or hyphens.

The last digit of the media ID can be a checksum digit, which is calculated according to the Luhn algorithm (calculation using the media ID without the last printed (entered) digit). The checksum digit is intended to prevent input errors during entry.

It must be possible to complete the entry, even if the checksum digit is displayed as being incorrect.

11.262 Monitor SAMs from operational perspective

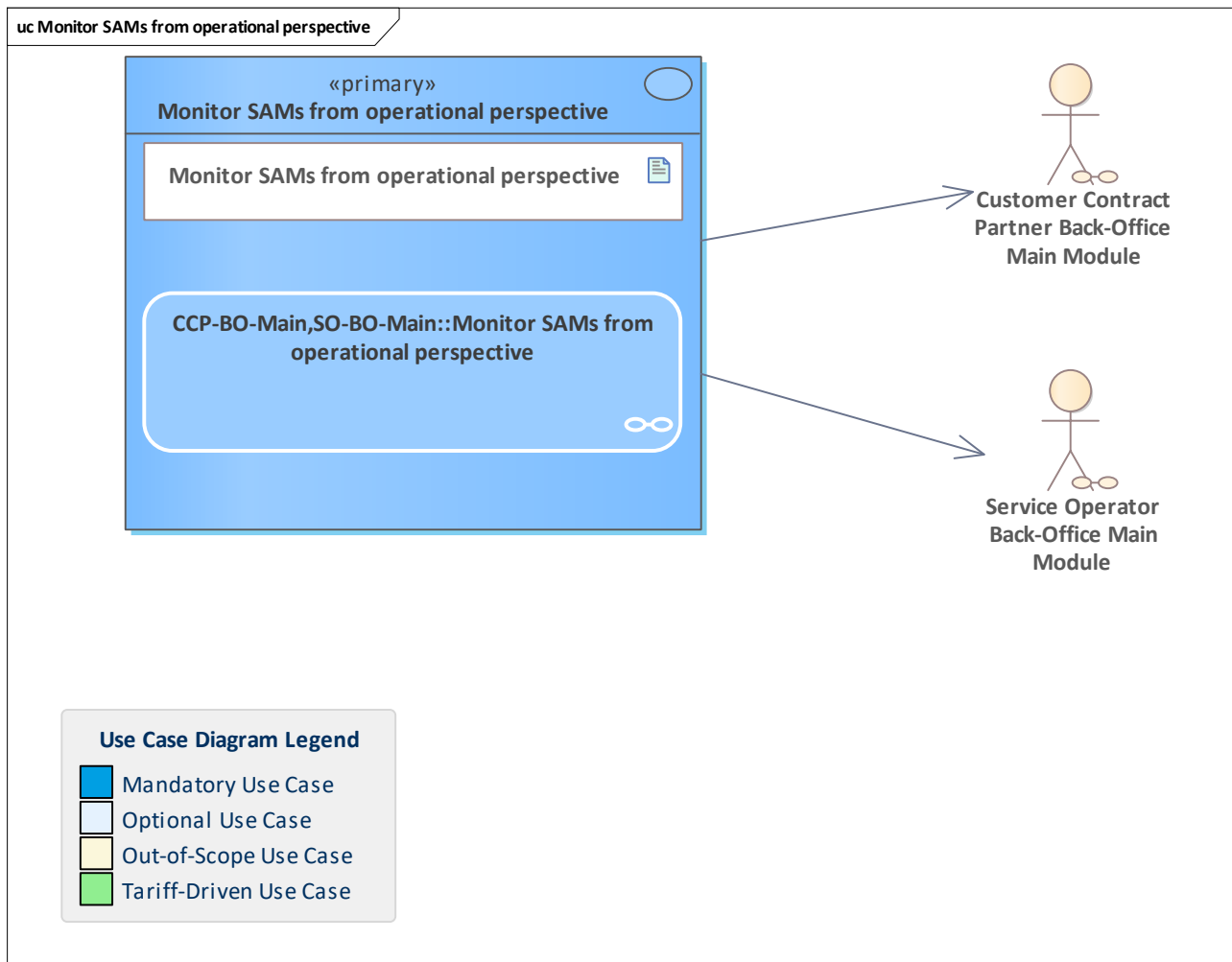


Figure 410: Monitor SAMs from operational perspective

The own SAMs need to be monitored from the operational perspective. All entitlement issuances and action authorisations have to be documented using notifications.

11.263 Monitor SAMs from product perspective

11.264 Monitor SAMs from product perspective

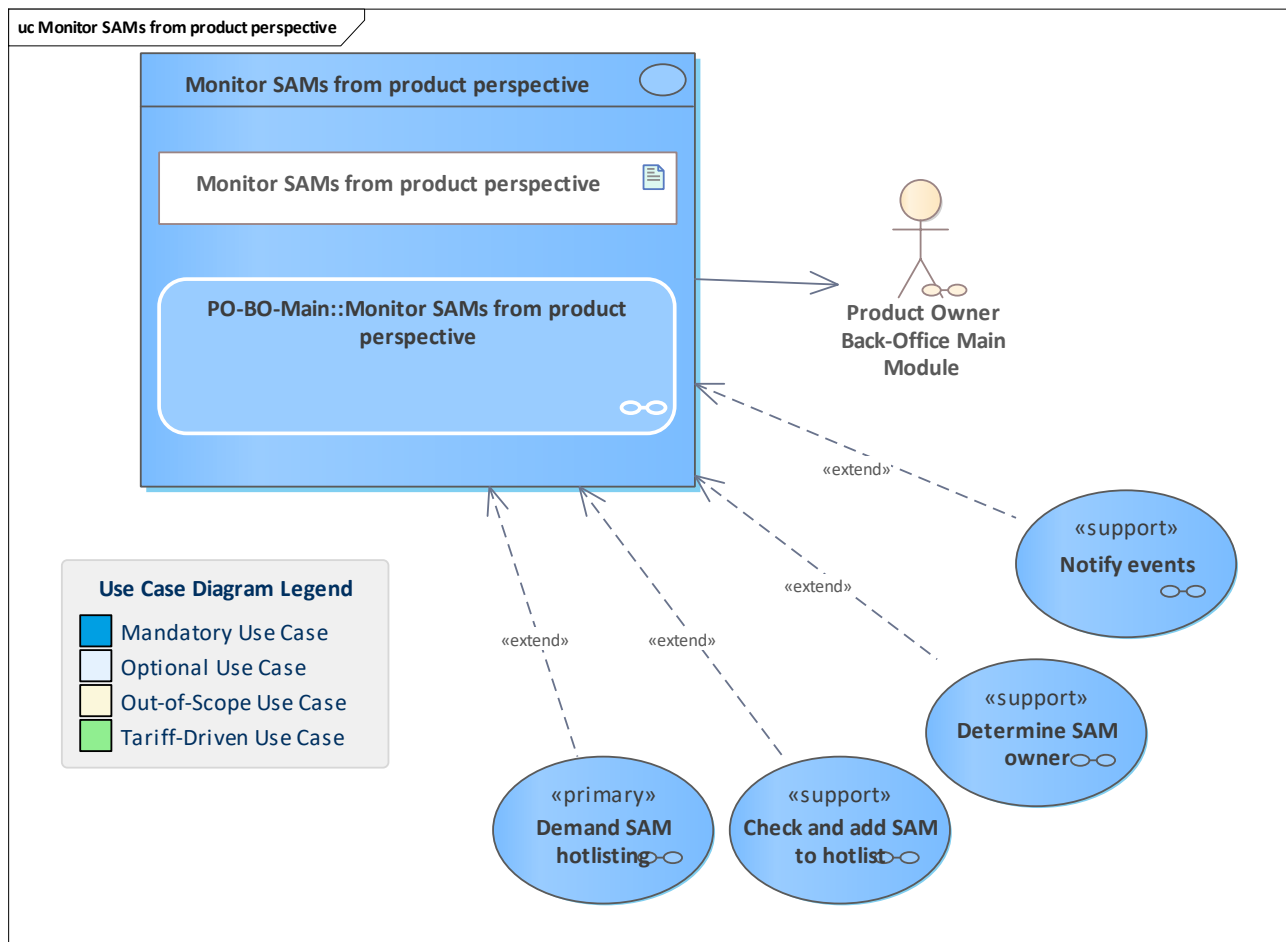


Figure 411: Monitor SAMs from product perspective

The product owner has to monitor the usage of the product owner tokens configured into SAMs. Every issue found shall only be reported once.

11.265 Notify events

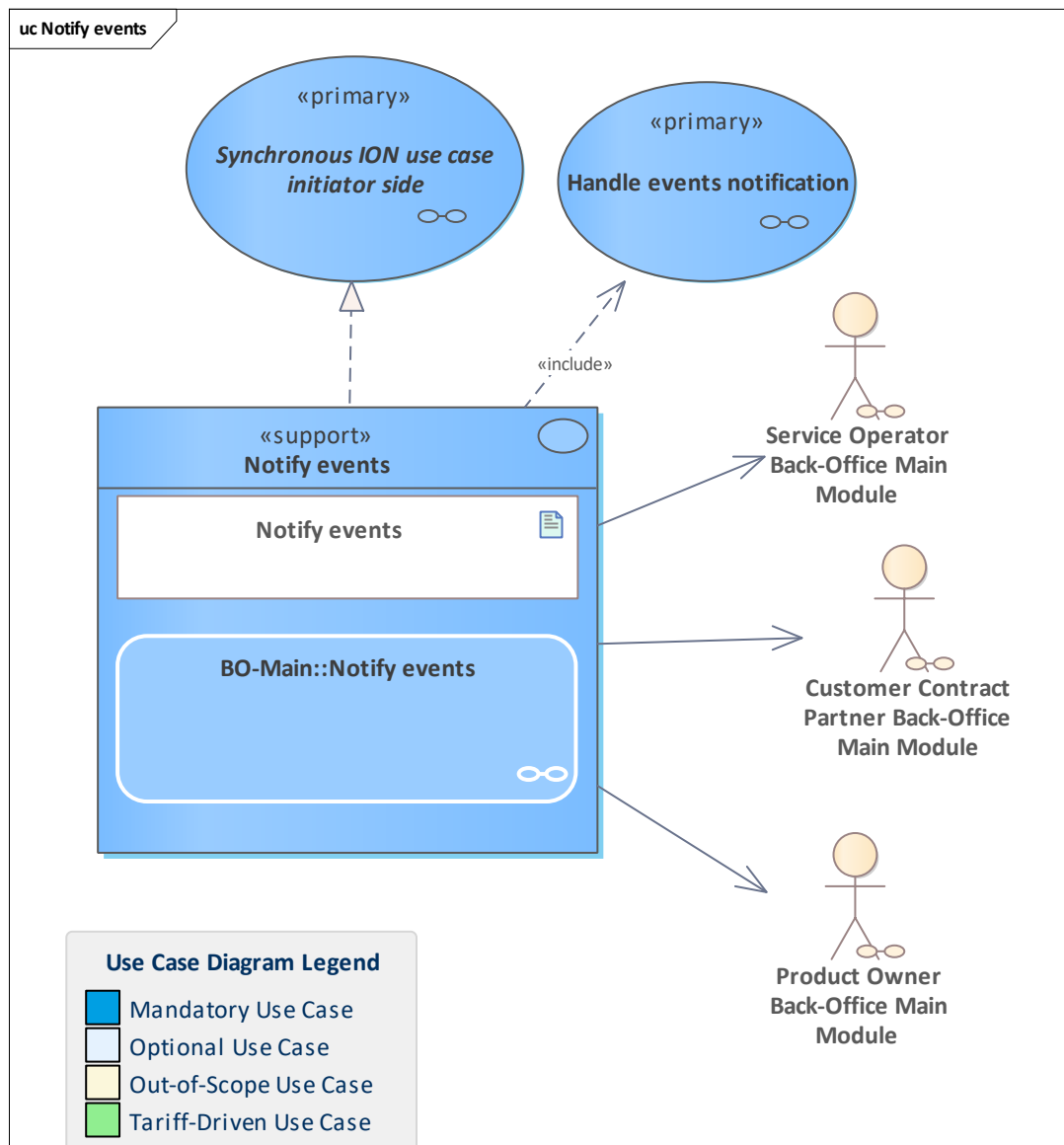


Figure 412: Notify events

Notify a participant about warnings which occurred during the downstream monitoring or similar.

11.266 Notify static entitlement inspected

11.267 Notify static entitlement inspected

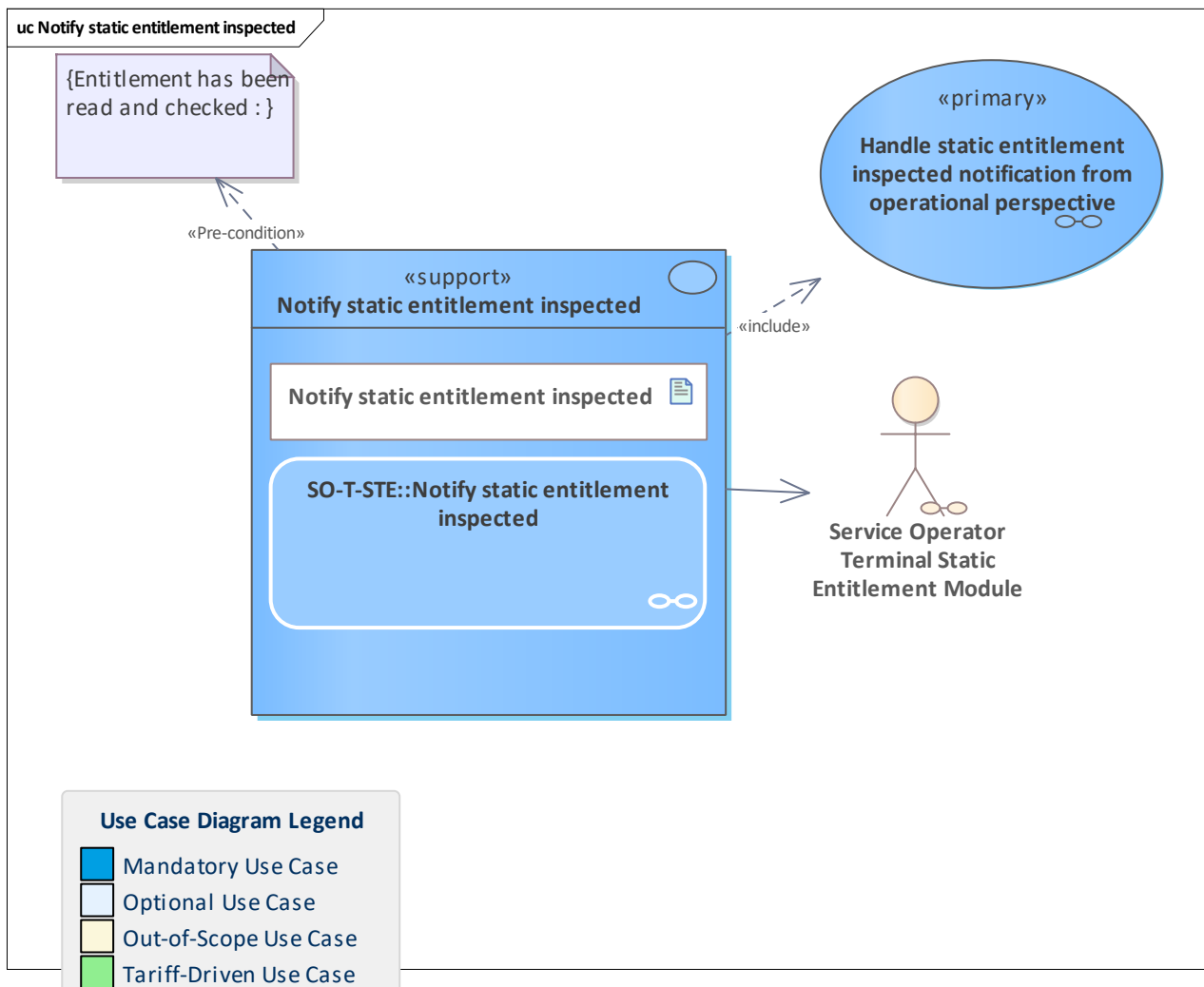


Figure 413: Notify static entitlement inspected

The terminal action data and static entitlement are compiled to a terminal notification. The terminal notification is sent to the SO back-office system.

11.268 Notify static entitlement terminated

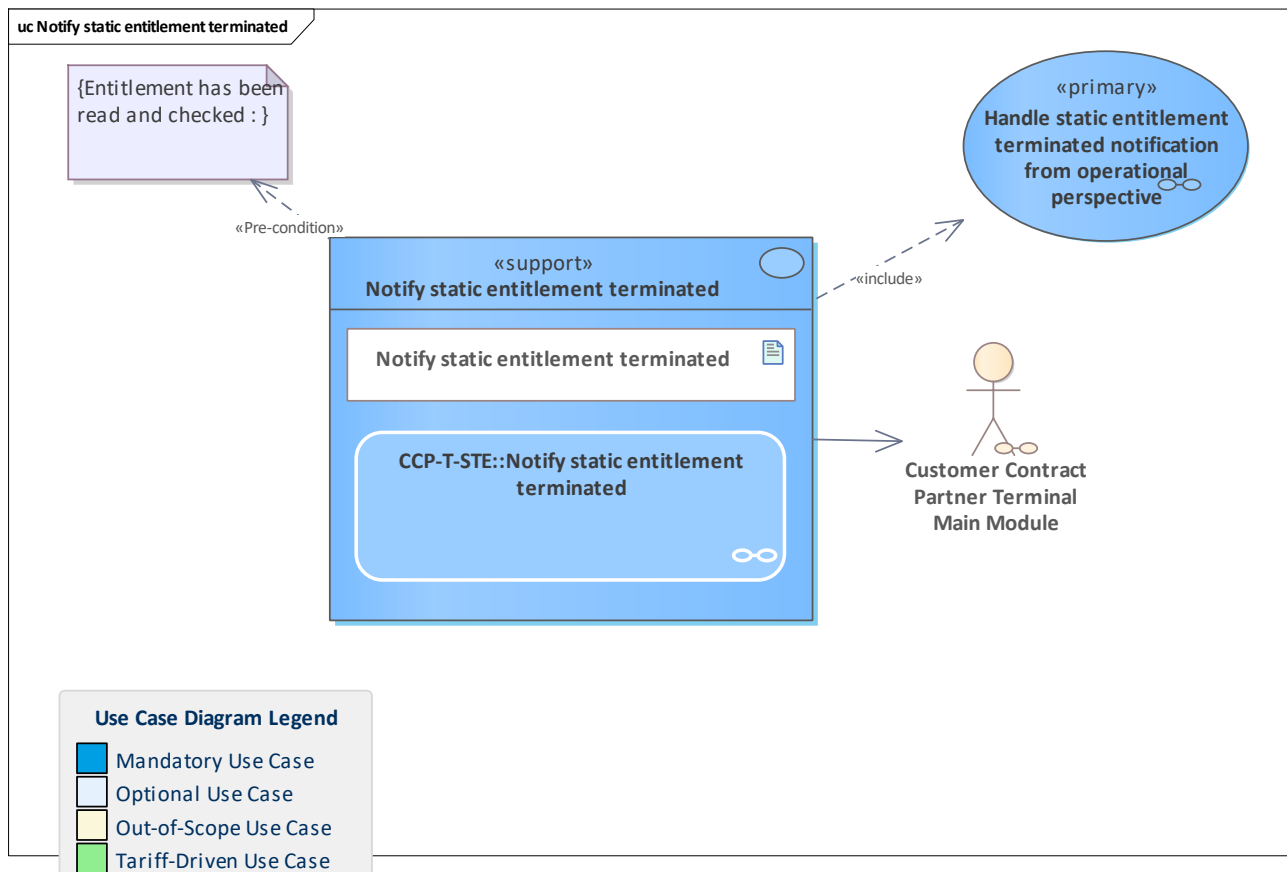


Figure 414: Notify static entitlement terminated

The termination of a static entitlement is notified by the CCP terminal with static entitlement extension to the responsible CCP back-office system.

11.269 Order entitlement blocking

11.270 Order entitlement blocking

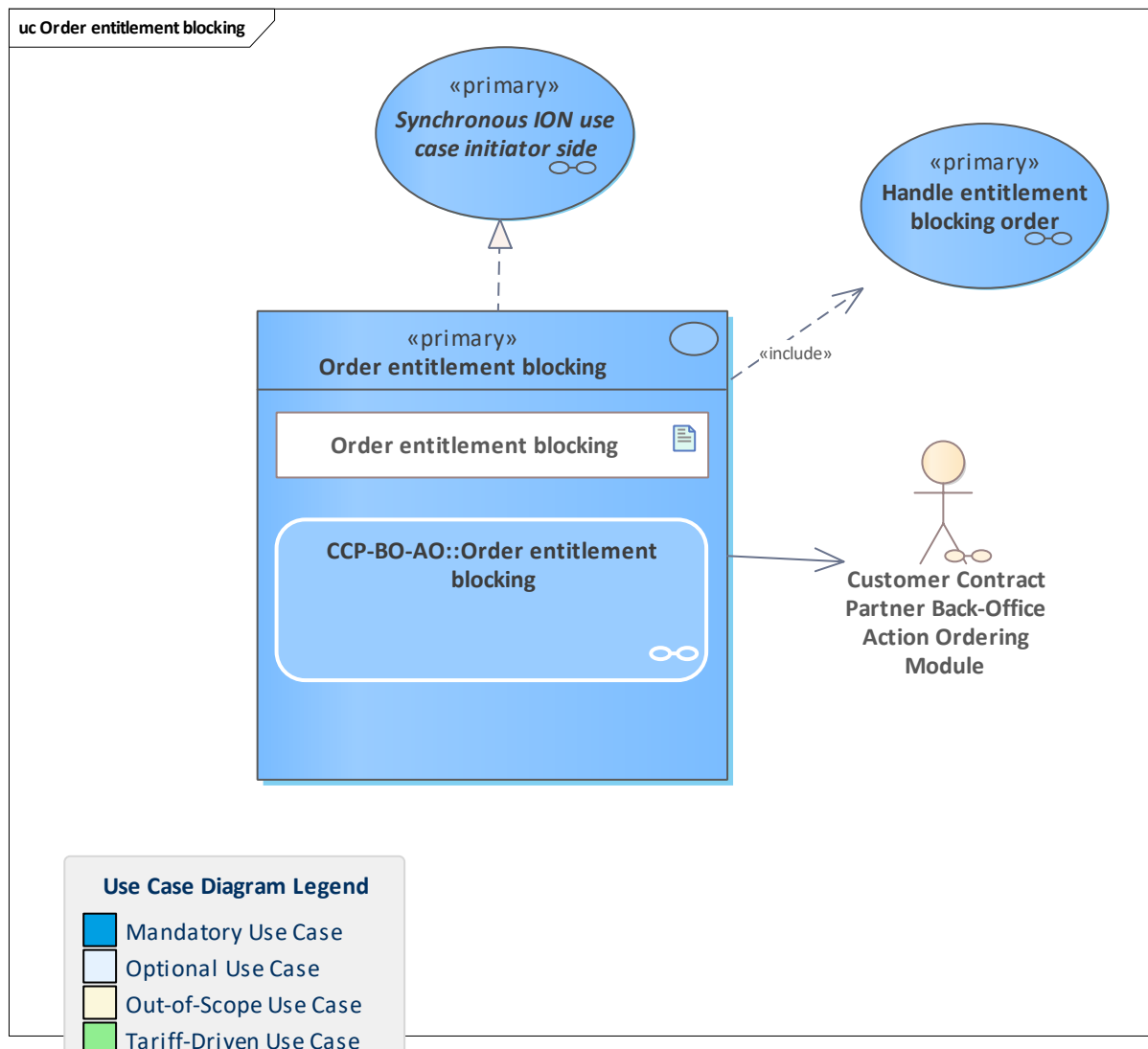


Figure 415: Order entitlement blocking

Using ordered action execution, the ordering CCP orders the blocking of an entitlement that might already have been issued, but whose issuance notification has not yet reached its owner system. Therefore, the blocking order is based on the order ID of the issuance order, since the entitlement ID is not available/known yet.

As soon as the ordered entitlement issued notification reaches the ordering system, the blocking order should be cancelled (if still active) and replaced by a regular hotlist entry.

11.271 Order entitlement issuance

11.272 Order entitlement issuance

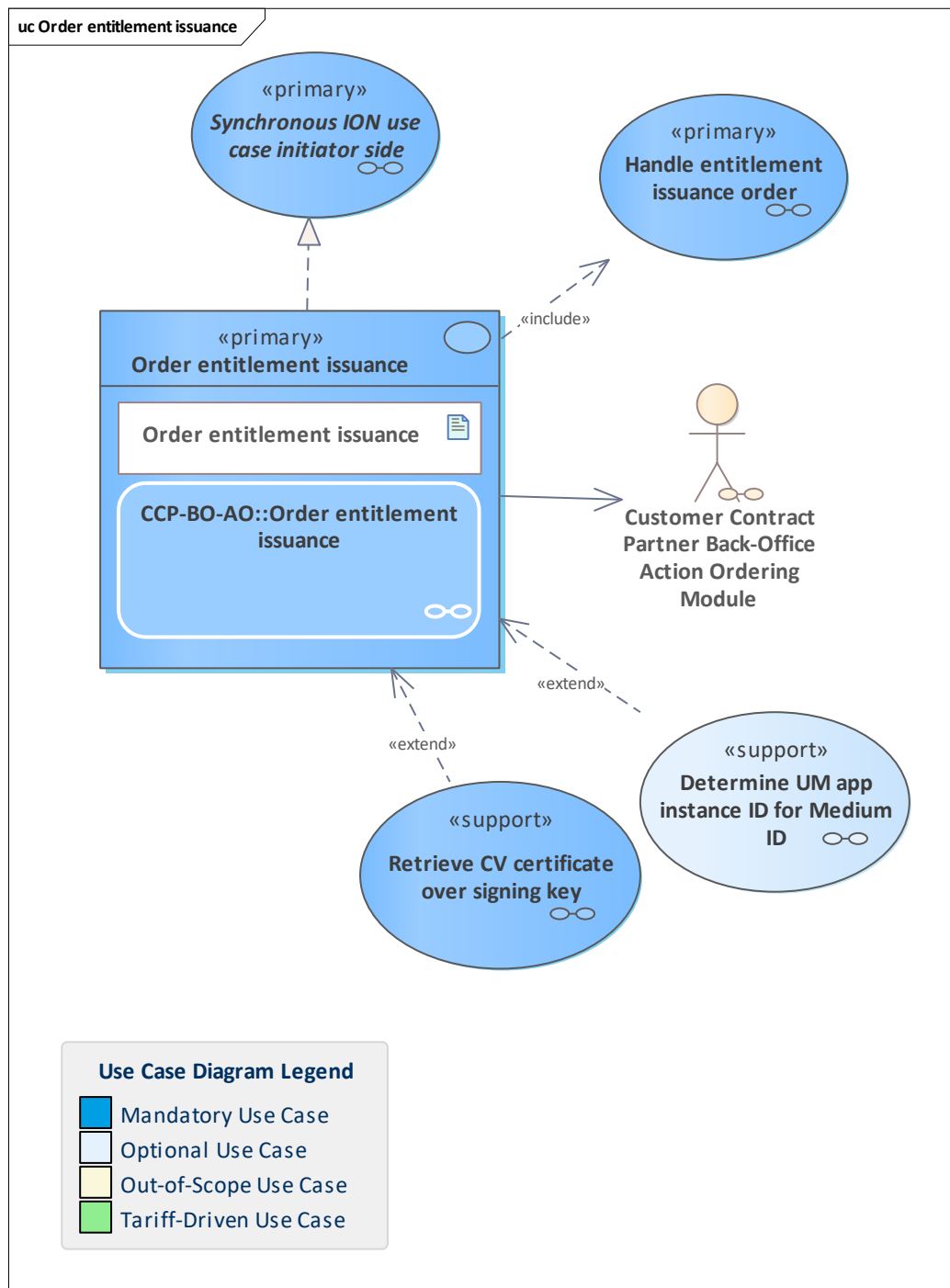


Figure 416: Order entitlement issuance

The ordering CCP orders an entitlement issuance using ordered action execution.

11.273 Order entitlement termination

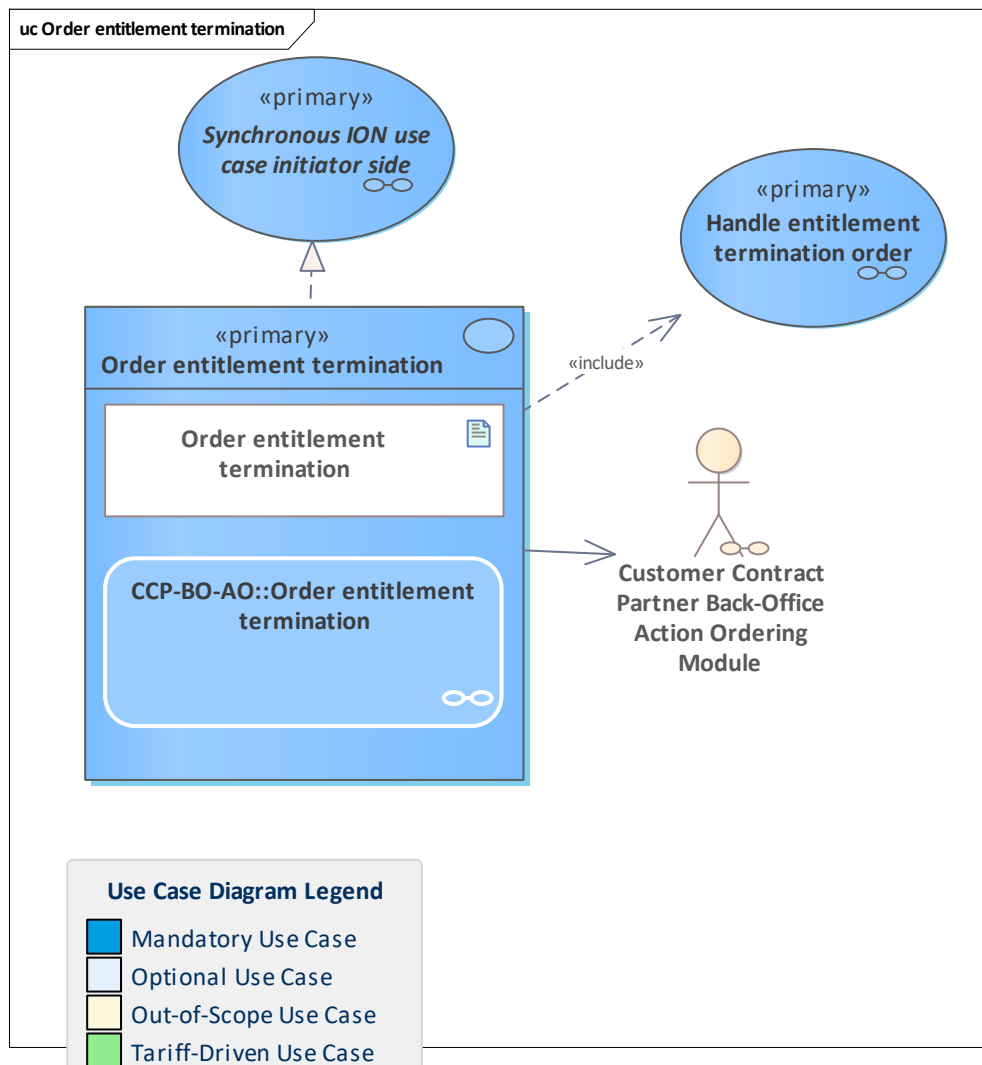


Figure 417: Order entitlement termination

The ordering CCP orders an entitlement termination.
The termination may only be ordered after the corresponding issuance notification (resulting from a previous order) has been handled.

11.274 Order entitlement unblocking

11.275 Order entitlement unblocking

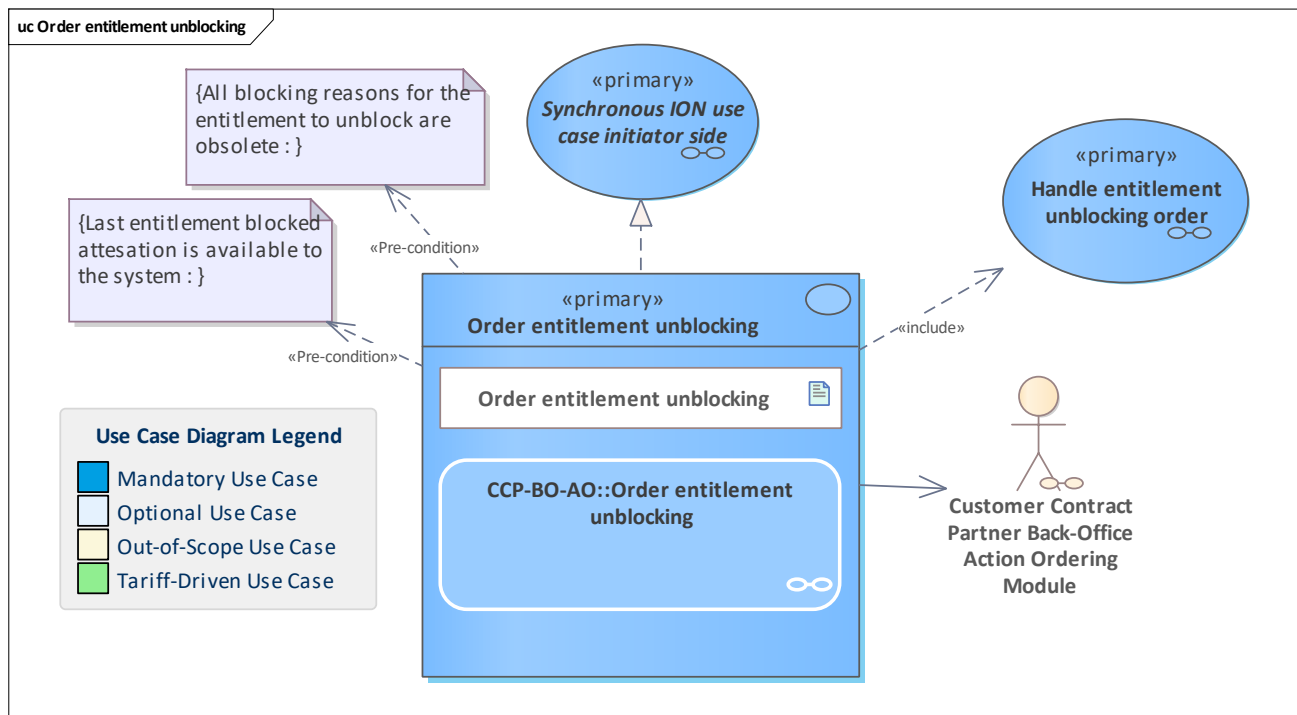


Figure 418: Order entitlement unblocking

The ordering CCP orders an entitlement unblocking.
The unblocking may only be ordered after the corresponding blocked notification (resulting from a previous order) has been handled.

11.276 Order group

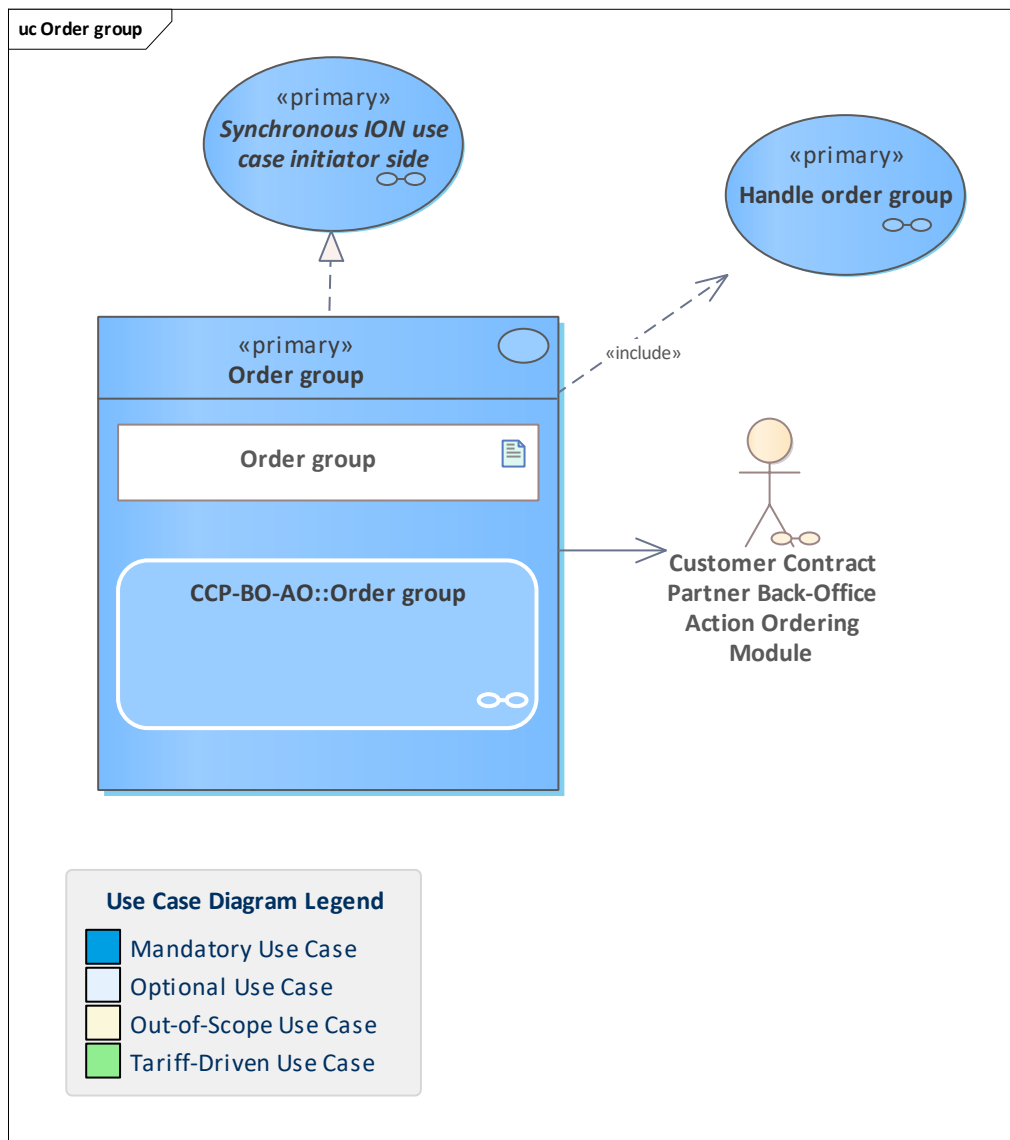


Figure 419: Order group

The ordering CCP sends a set of orders to the [Product Owner Back-Office Action Management Module](#), which in turn will either accept all of them or reject all of them.

The orders contained in the set have to target the same UM app instance ID and must share the same group ID.

This process can be used, e.g., to change the product parameters of an entitlement at a future date X. For that purpose, the following orders would be created:

- A termination order for the current entitlement
- An issuance order for an entitlement identical to the current one, but with an expiration time of X
- An issuance order for an entitlement identical to the current one, but with adjusted product parameters and effective time X

For all of these orders, the group ID would be set to the same, unique value, thus tying them together. Using the 'Order group' use case guarantees the atomicity of the change, thus ensuring that the customer is not left without an entitlement.

11.277 Perform account-based payment method crediting and notify

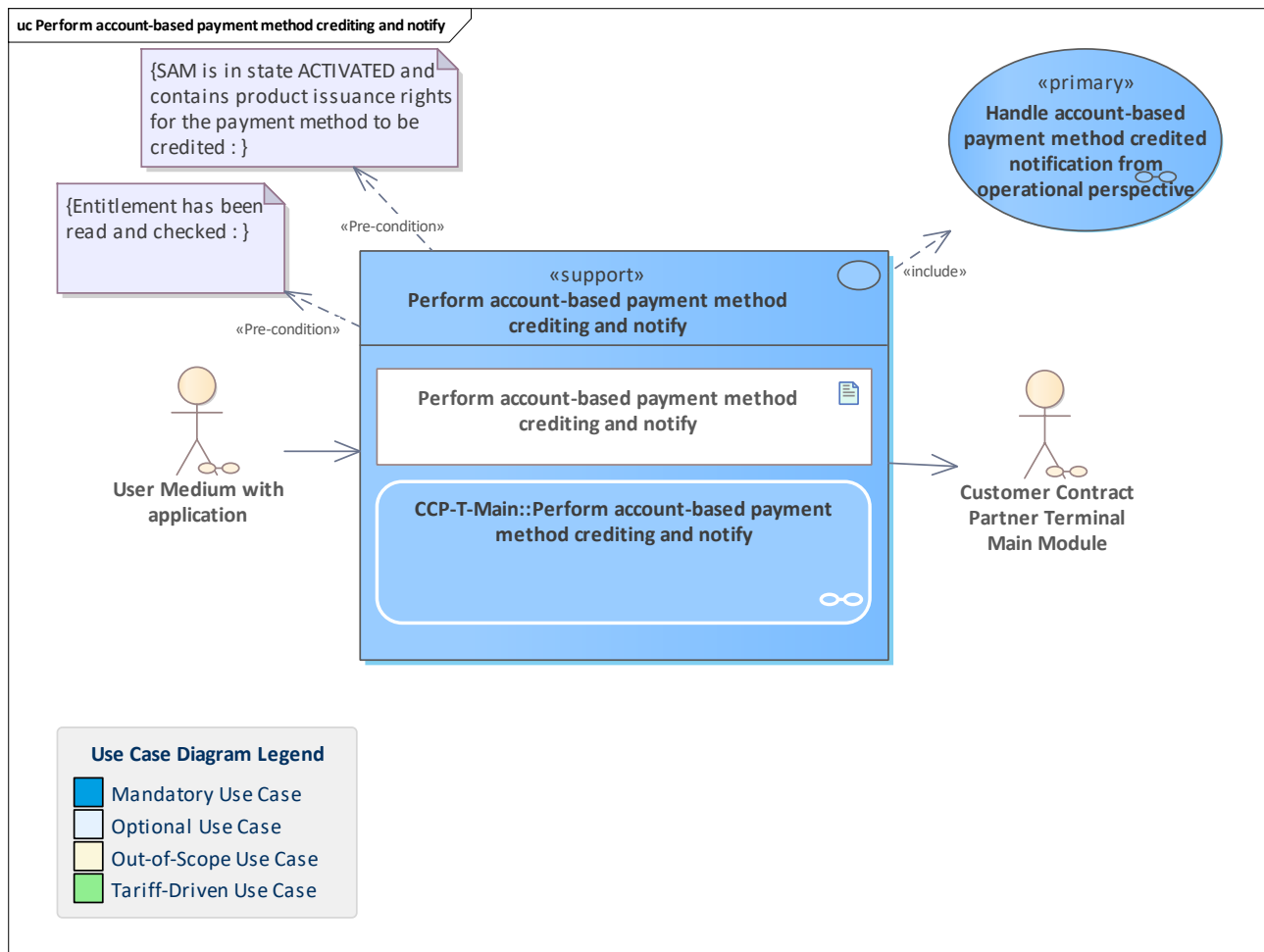


Figure 420: Perform account-based payment method crediting and notify

The CCP terminal performs a transaction to credit an account-based payment method on a user medium with an application and notifies the CCP back-office system about the crediting transaction.

If a transaction is aborted, the CCP back-office system is also notified for consistent monitoring data.

11.278 Perform account-based payment method debiting and notify

11.279 Perform account-based payment method debiting and notify

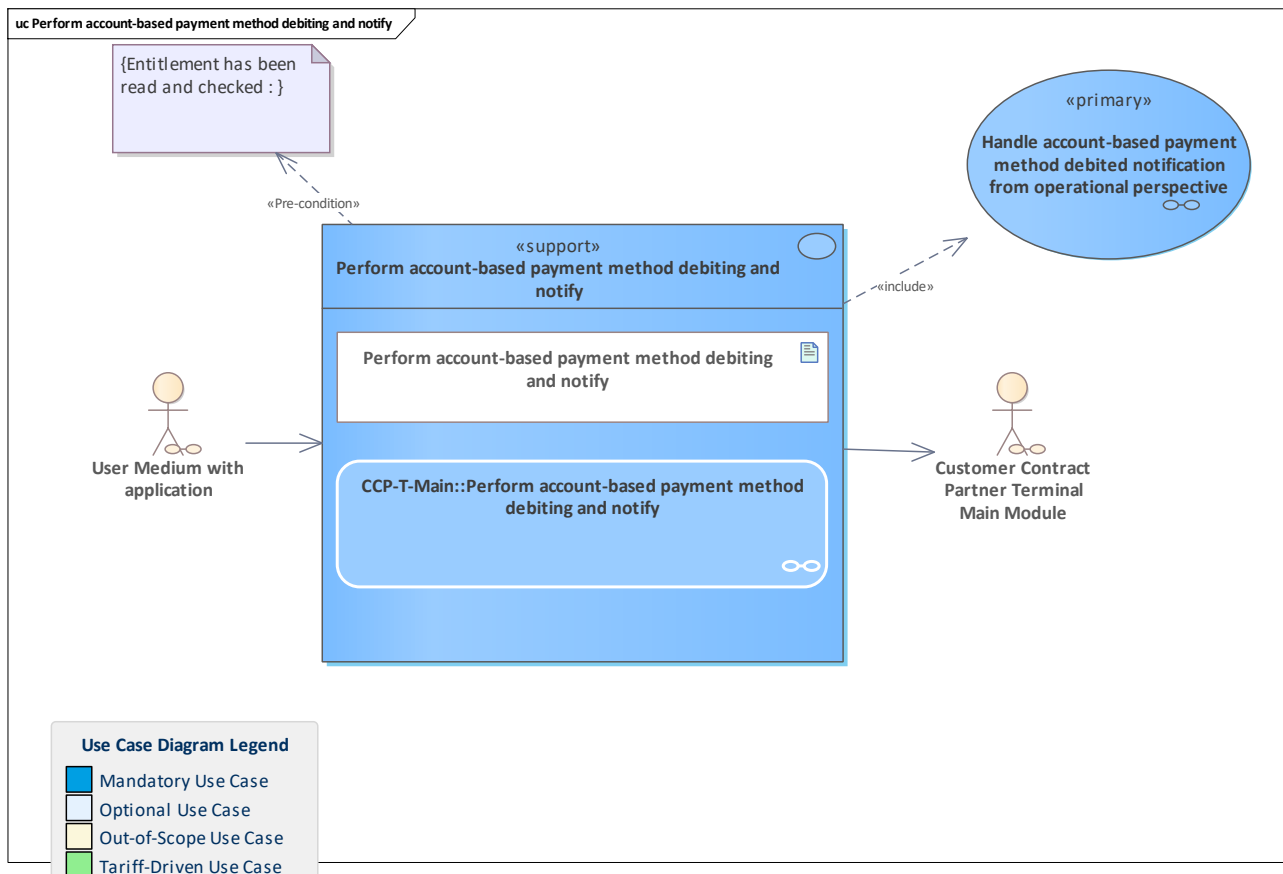


Figure 421: Perform account-based payment method debiting and notify

The CCP terminal performs a transaction to debit an account-based payment method on a user medium with an application and notifies the CCP back-office system about the debiting transaction.

If a transaction is aborted, the CCP back-office system is also notified for consistent monitoring data.

11.280 Perform application blocking and notify

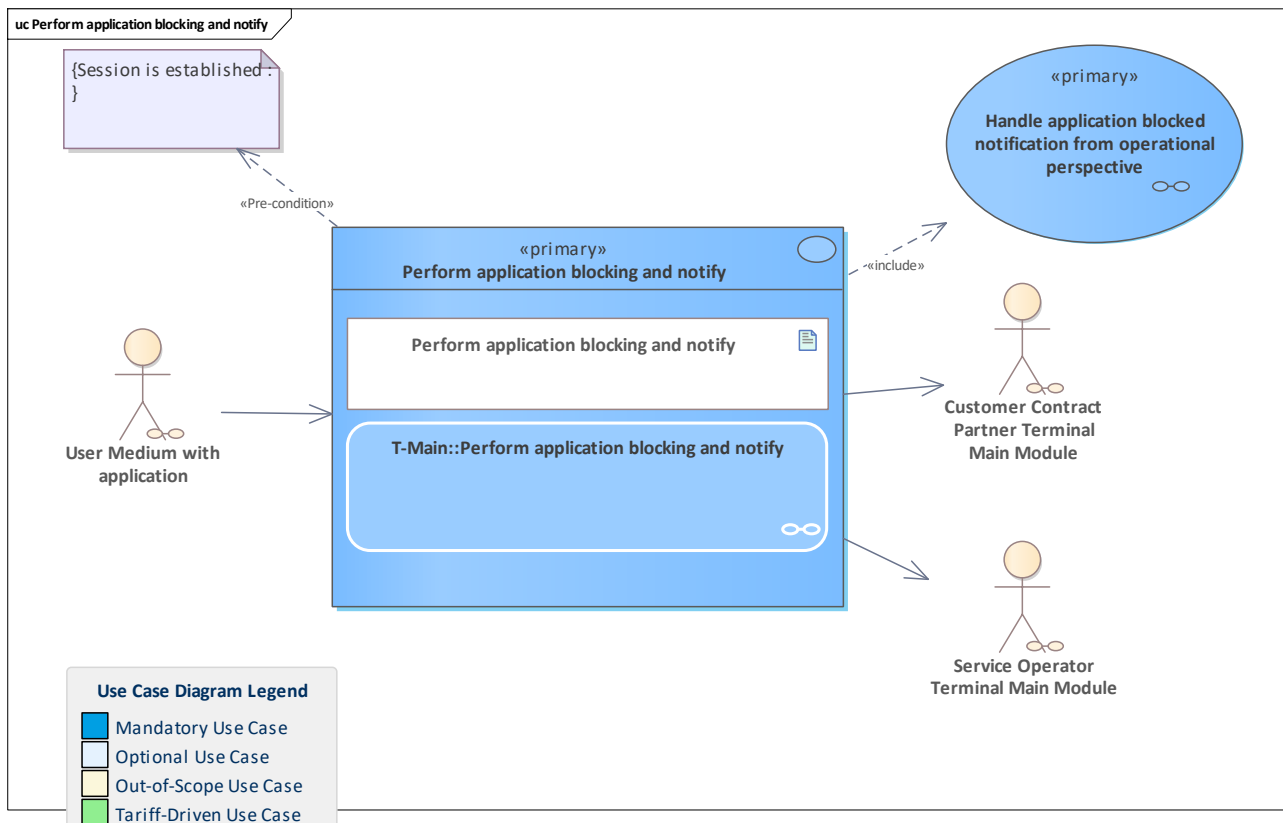


Figure 422: Perform application blocking and notify

Use case for a CCP or SO terminal. A user medium application is detected either in the application hotlist or a relevant entry is found in the SAM hotlist or the organisation hotlist. Therefore, the user medium application must be blocked physically by switching its state. The terminal performs this action and notifies the back-office system of the terminal operator (CCP or SO).

If the transaction is aborted, this has also to be notified to the terminal operator (CCP or SO).

11.281 Perform application termination and notify

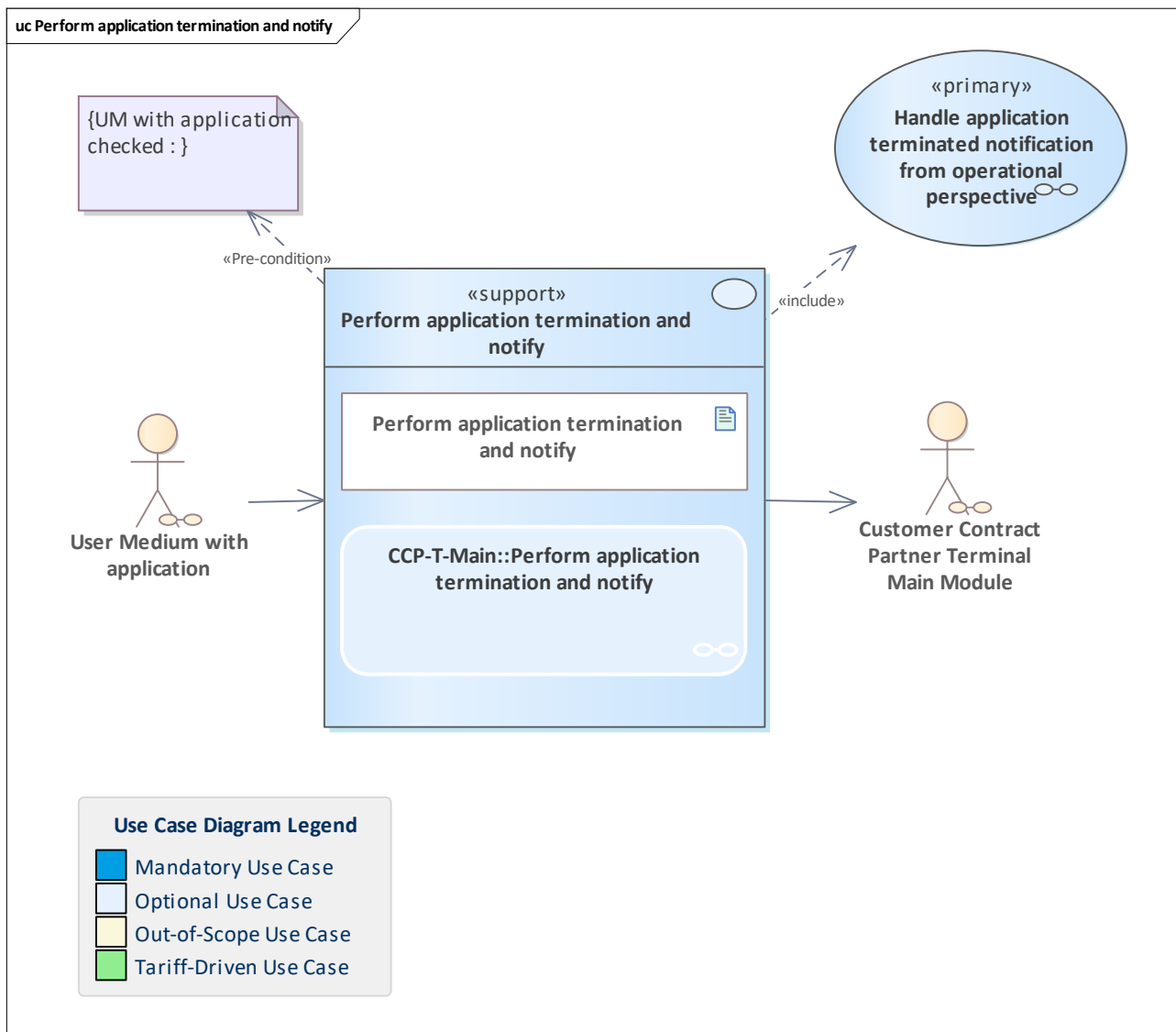


Figure 423: Perform application termination and notify

Perform the transaction to terminate a user medium application and notify interested parties.

Note: the application termination action is automatically committed by the user medium application after execution. Thus, no timeout scenario has to be handled. Instead, for application terminations an answer by the user medium that can not be validated by the SAM may indicate something similar to a timeout scenario: the termination may be permanent, but no attestation about it can be provided.

11.282 Perform application unblocking and notify

11.283 Perform application unblocking and notify

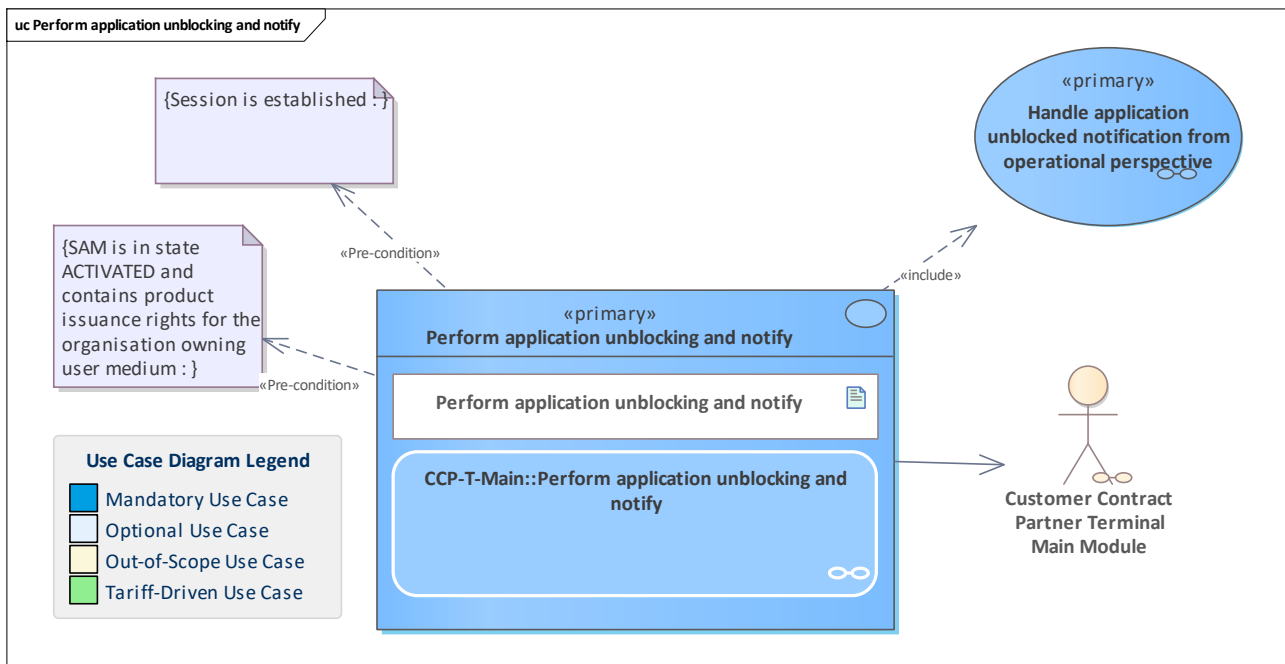


Figure 424: Perform application unblocking and notify

The pCCP decides to unblock the application. The terminal runs the transaction to unblock the user medium application and informs the back-office system about the result.

11.284 Perform application XY and notify

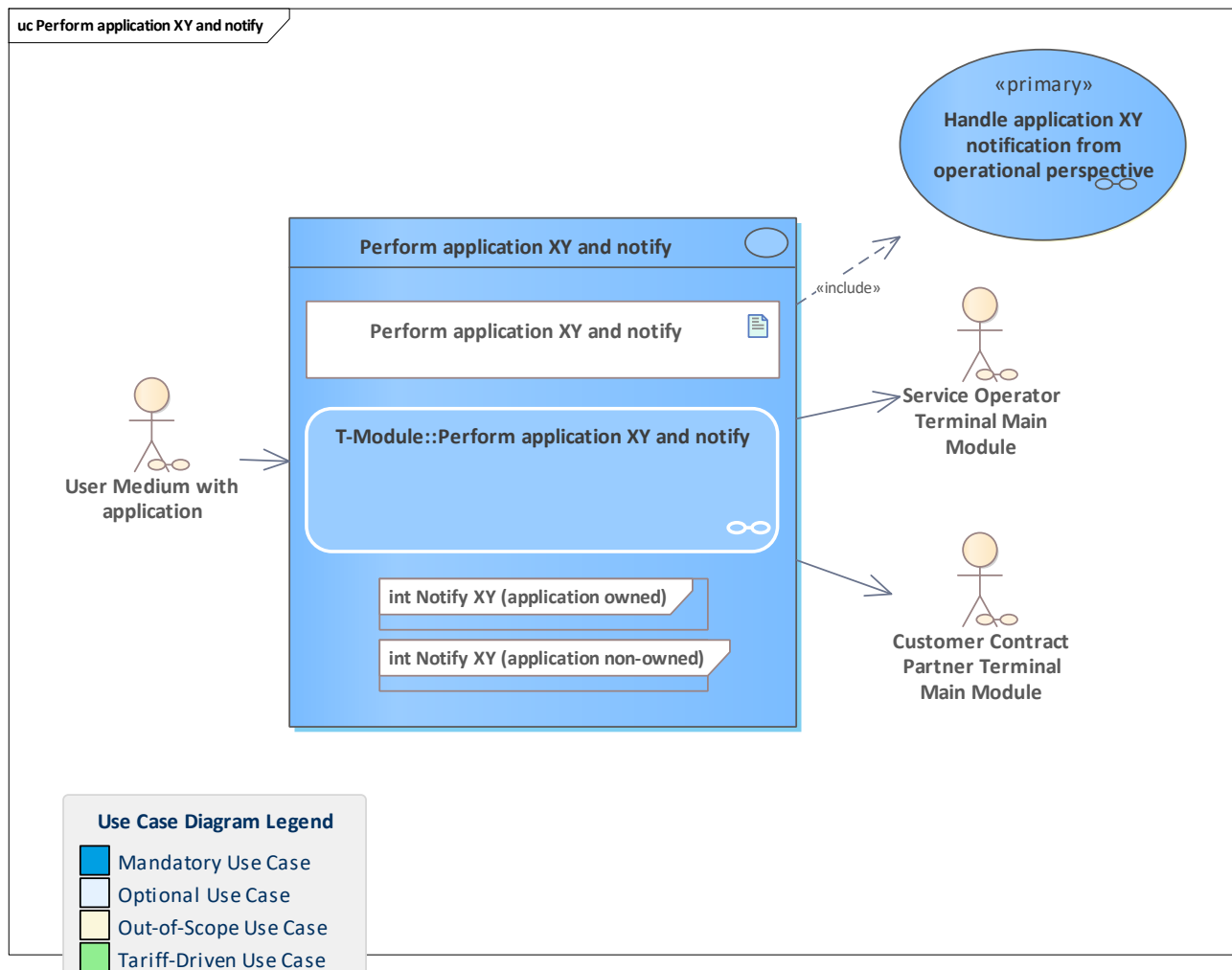


Figure 425: Perform application XY and notify

The UM application action XY is performed by the terminal in conjunction with a SAM and the back-office system corresponding to the terminal is notified about the action execution.

Combines the transaction including the action XY and the notification processes triggered by this. Shows the interaction between the UM transaction and the notification processes. The timeout warning is generated in this diagram if needed.

The business preconditions to perform the transaction (e.g. determining the actual need to block an application) are already satisfied before the process shown in this diagram begins.

11.285 Perform check in and notify

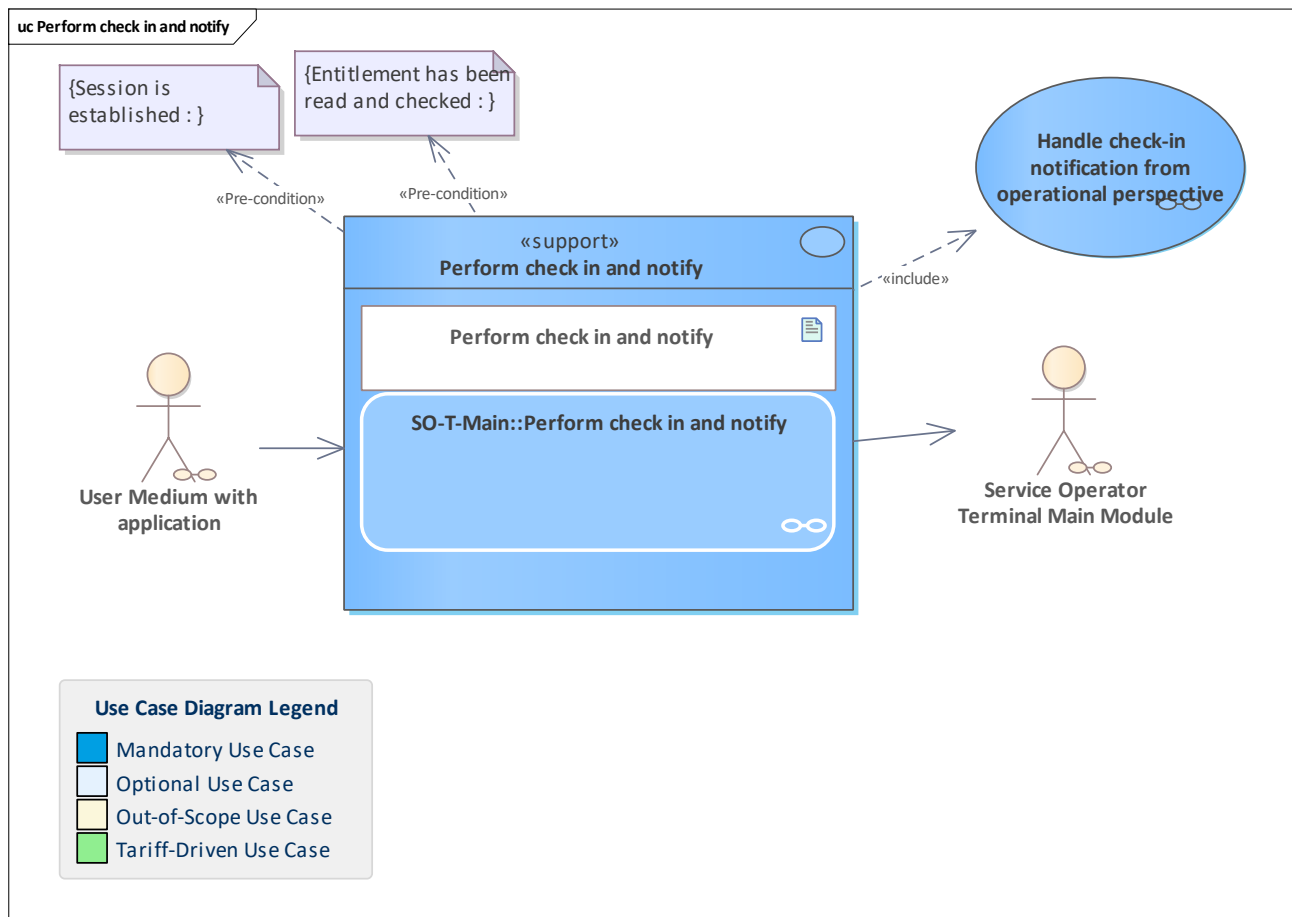


Figure 426: Perform check in and notify

Supporting use case performed by the SO terminal with CICO extension. The primary use case is [Record entitlement within CICO system](#).

The check-in attestation is created and stored on the user medium. The check-in notification based on the attestation is created by the terminal and sent to the responsible SO back-office system.

A detected transaction abortion is also sent to the SO system for consistent monitoring data.

11.286 Perform check out and notify

11.287 Perform check out and notify

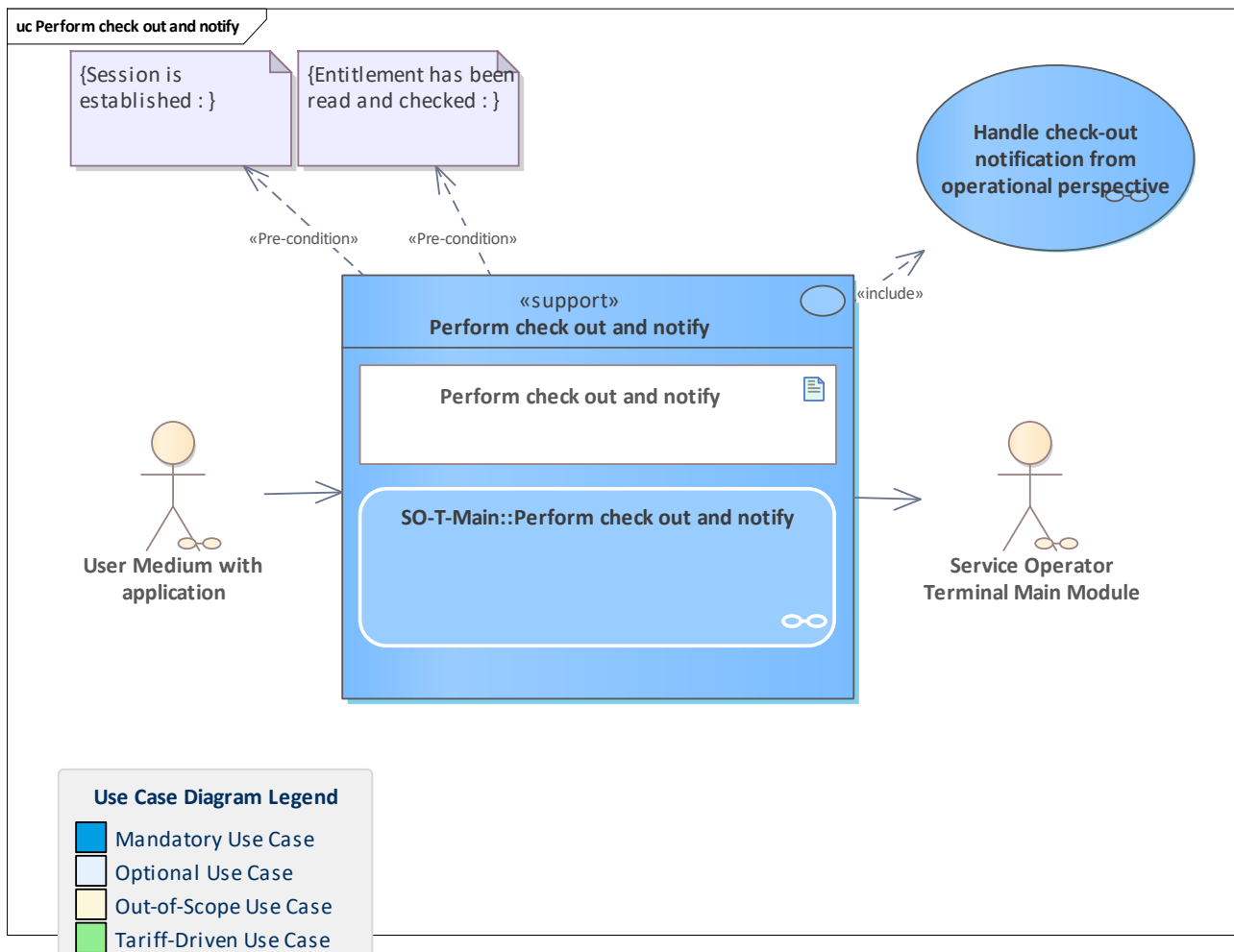


Figure 427: Perform check out and notify

Supporting use case performed by the SO terminal with CICO extension. The primary use case is [Record entitlement within CICO system](#).

The check-out attestation is created and stored on the user medium. The check-out notification based on the attestation is created by the terminal and sent to the responsible SO back-office system.

A detected transaction abortion is also sent to the SO system for consistent monitoring data.

11.288 Perform entitlement blocking and notify

11.289 Perform entitlement blocking and notify

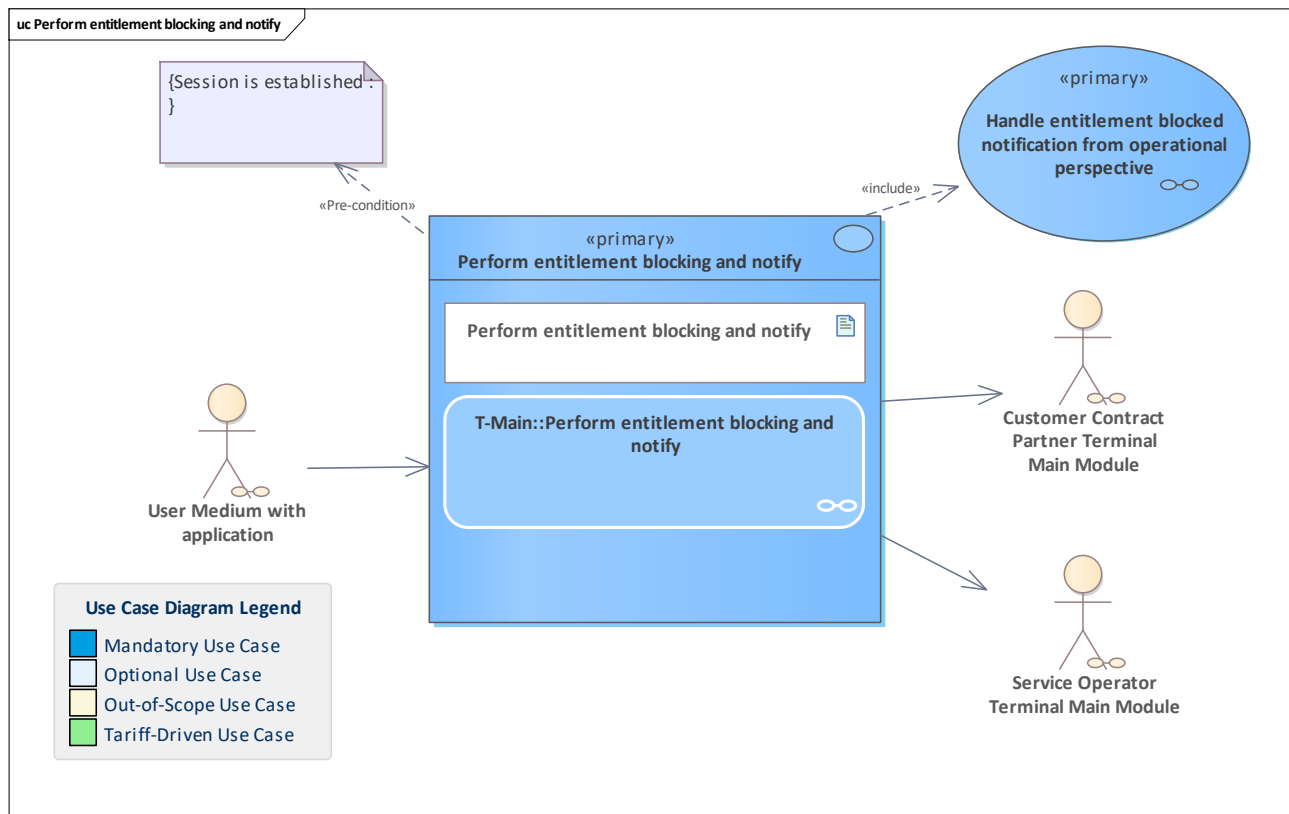


Figure 428: Perform entitlement blocking and notify

Use case for a CCP or SO terminal.

An entitlement on a user medium with an application is detected either in the entitlement hotlist or a relevant entry is found in the SAM hotlist or the organisation hotlist. Therefore, the entitlement must be blocked physically by switching its state.

The terminal performs this action and notifies the back-office system of the terminal operator (CCP or SO).

If the transaction is aborted, this has also to be notified to the terminal operator (CCP or SO).

11.290 Perform entitlement inspection and notify

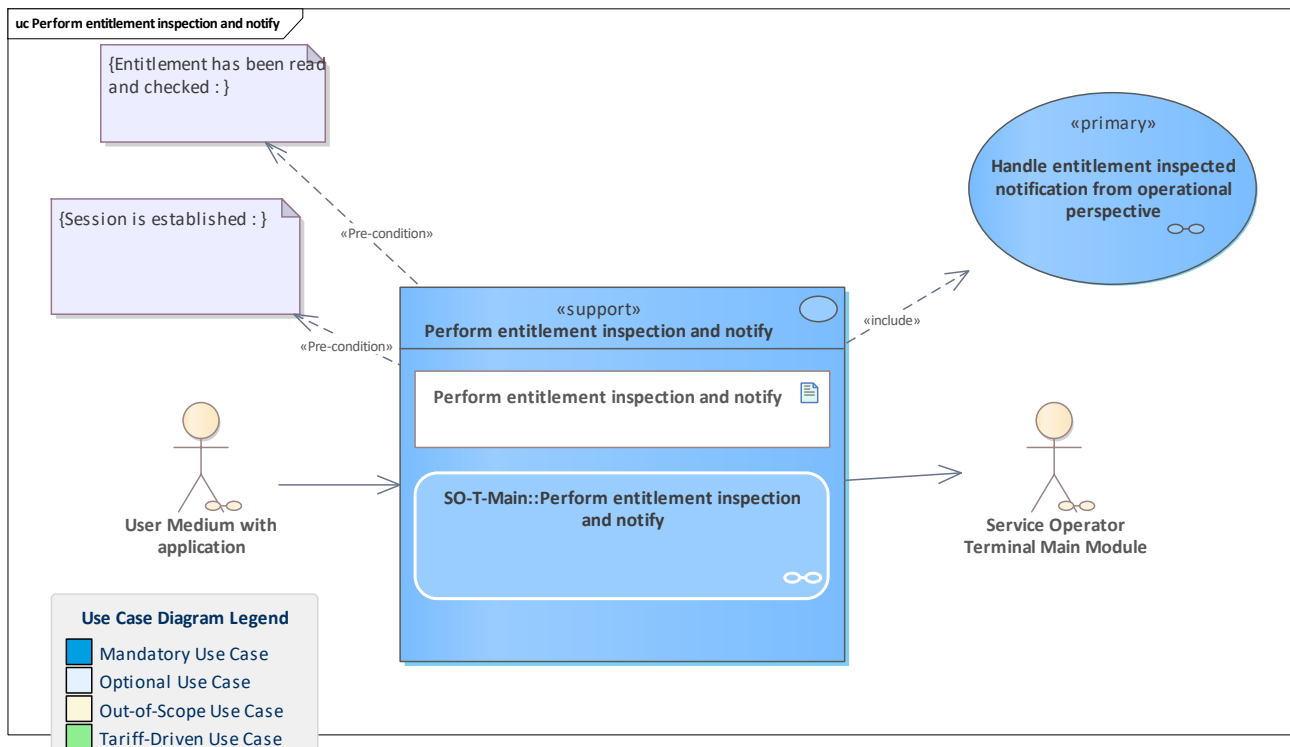


Figure 429: Perform entitlement inspection and notify

After an entitlement has been read in the inspection process, an *entitlement inspected attestation* is created.

The attestation will **not** be written on the user medium but only embedded in the notification for the back-office systems.

PO and CCP will be informed about this action. The notification will be checked from product (PO) and contractual (CCP) perspectives.

11.291 Perform entitlement issuance and notify

11.292 Perform entitlement issuance and notify

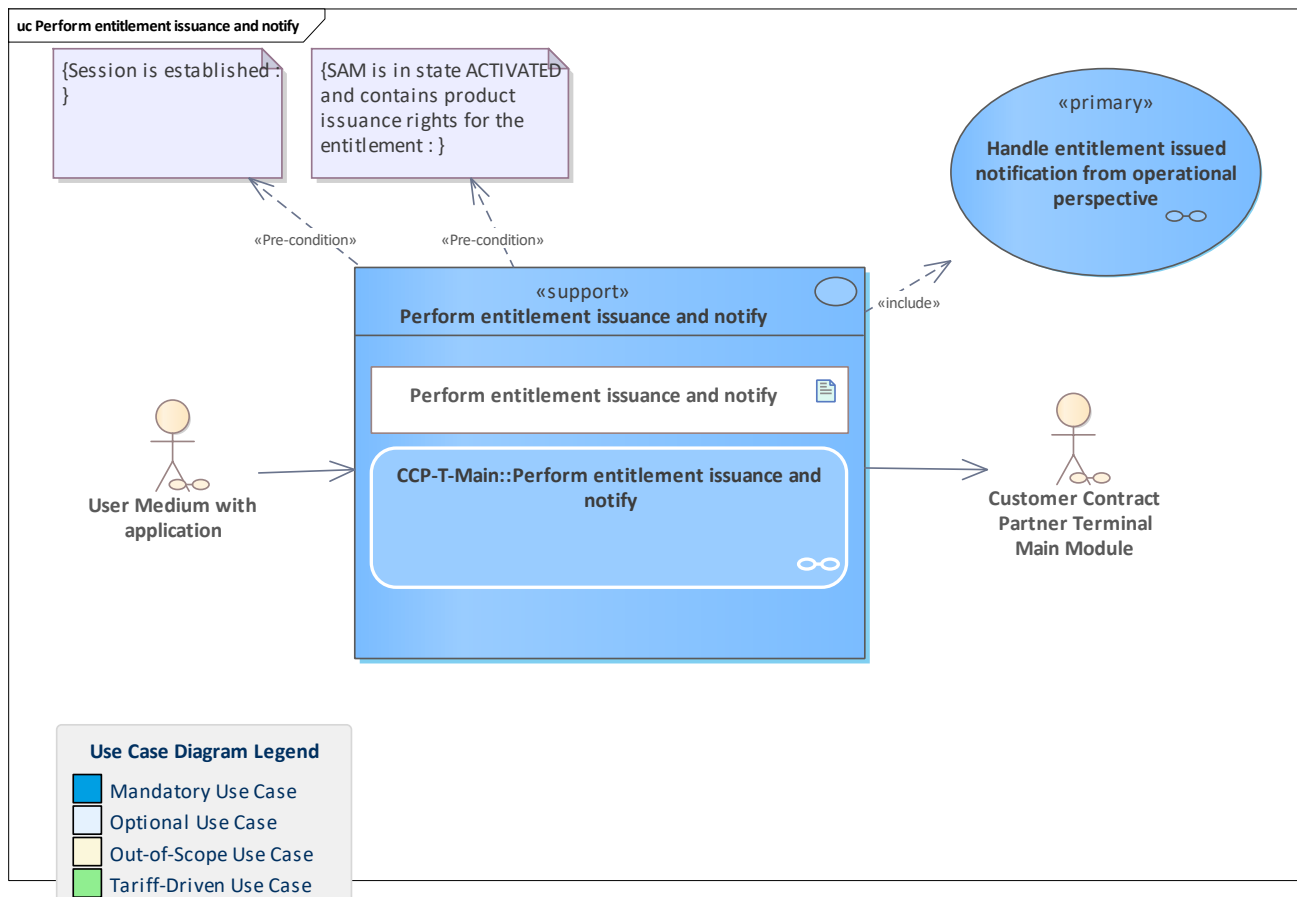


Figure 430: Perform entitlement issuance and notify

The CCP terminal performs a transaction to issue an entitlement to a user medium with application.

The CCP back-office system is notified about the issuance.

If the transaction is aborted, the CCP back-office system is also informed.

11.293 Perform entitlement termination and notify

11.294 Perform entitlement termination and notify

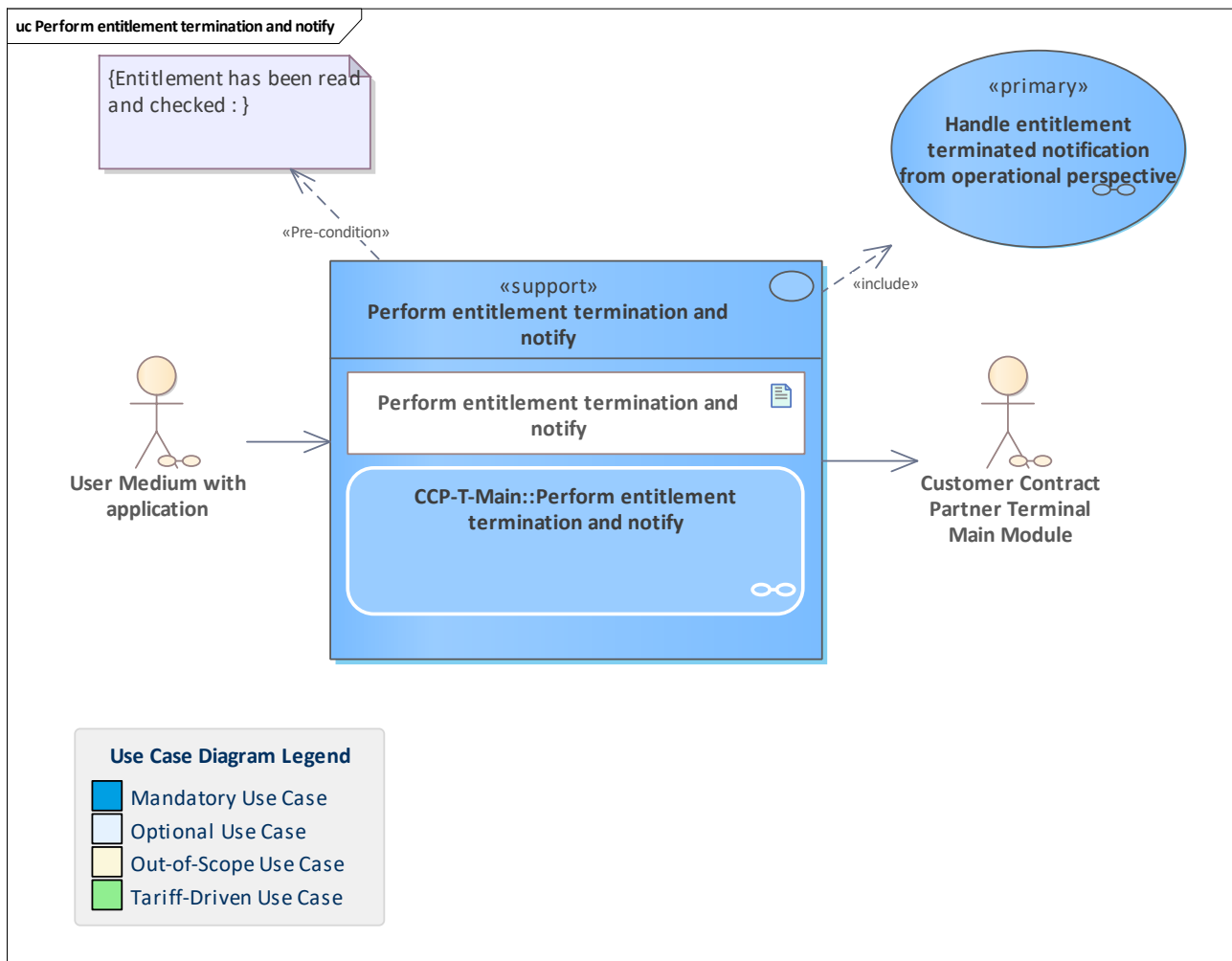


Figure 431: Perform entitlement termination and notify

Perform the transaction to terminate an entitlement in the CCP terminal and notify the responsible CCP back-office system (same CCP as for the terminal).
If the transaction is aborted, the back-office system is also notified for consistent monitoring data.

11.295 Perform entitlement unblocking and notify

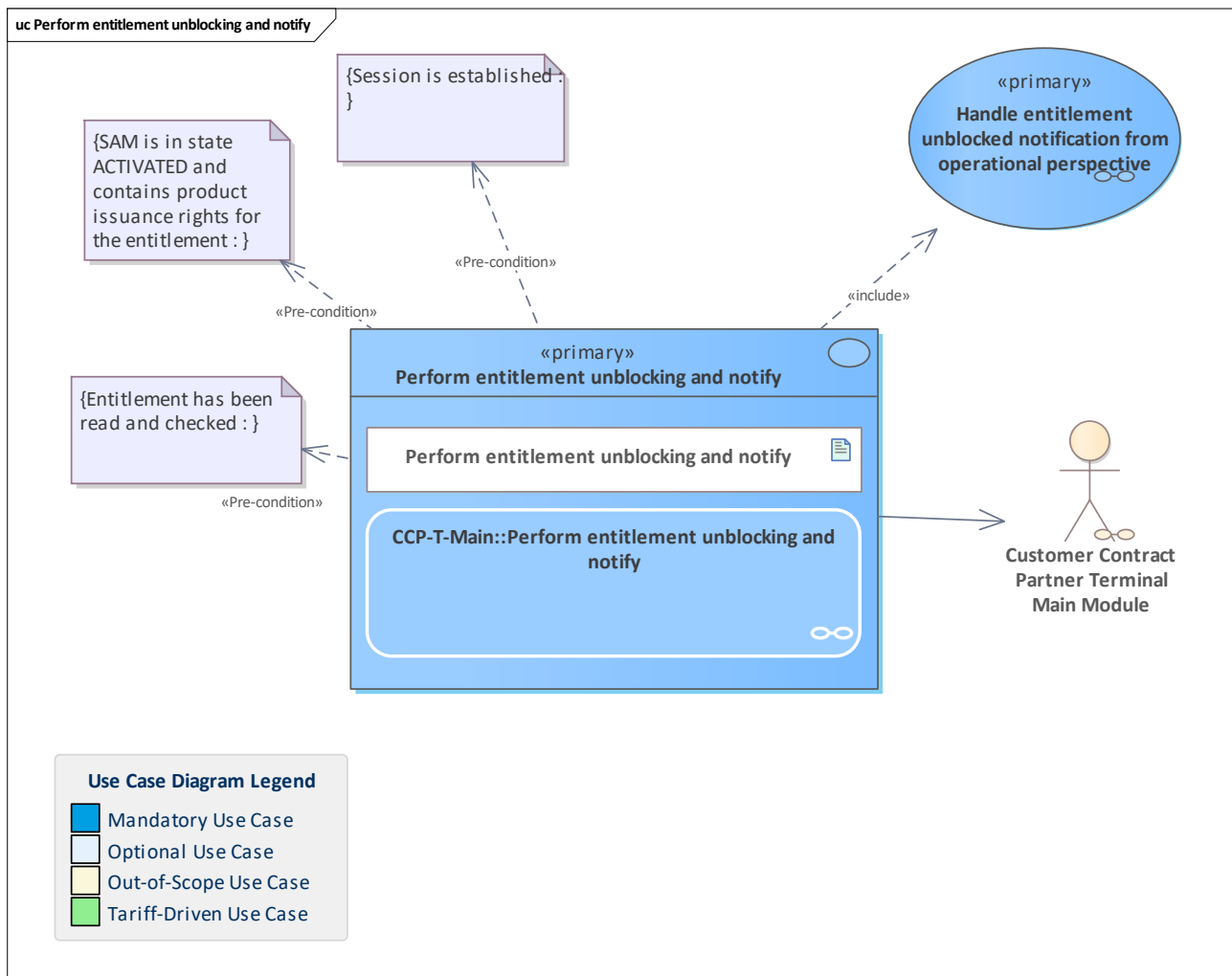


Figure 432: Perform entitlement unblocking and notify

The terminal runs the transaction to unblock the entitlement in question and informs the back-office system about the result.

If the transaction is aborted, the back-office system is also notified for consistent monitoring data.

Note: only the pCCP is allowed to unblock its owned entitlements.

11.296 Perform entitlement validation and notify

11.297 Perform entitlement validation and notify

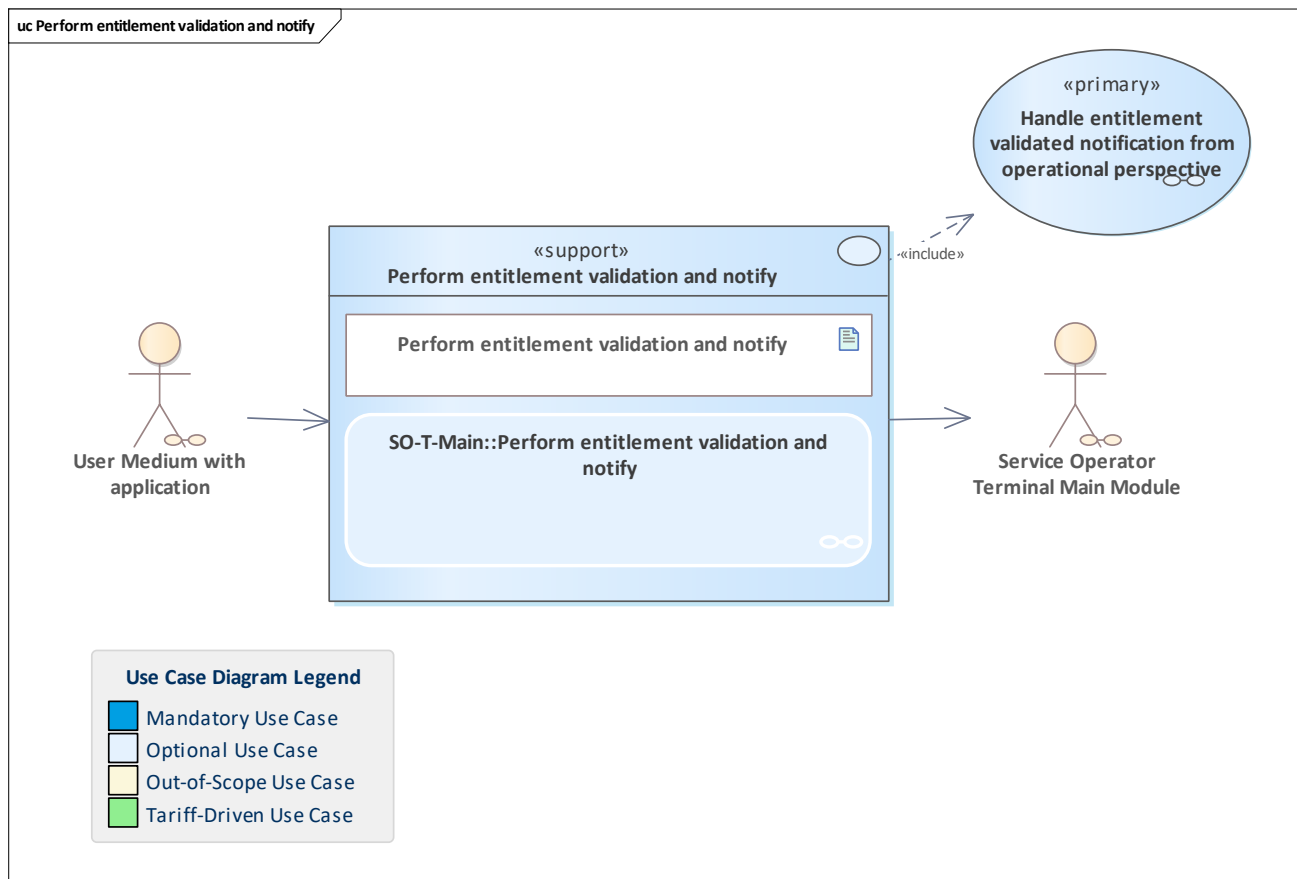


Figure 433: Perform entitlement validation and notify

The entitlement validation attestation is created by the SO terminal. The entitlement validation notification from the terminal is created based on attestation and sent to the SO back-office system.

If the transaction is aborted, the back-office system is also notified for consistent monitoring data.

11.298 Perform entitlement XY and notify

11.299 Perform entitlement XY and notify

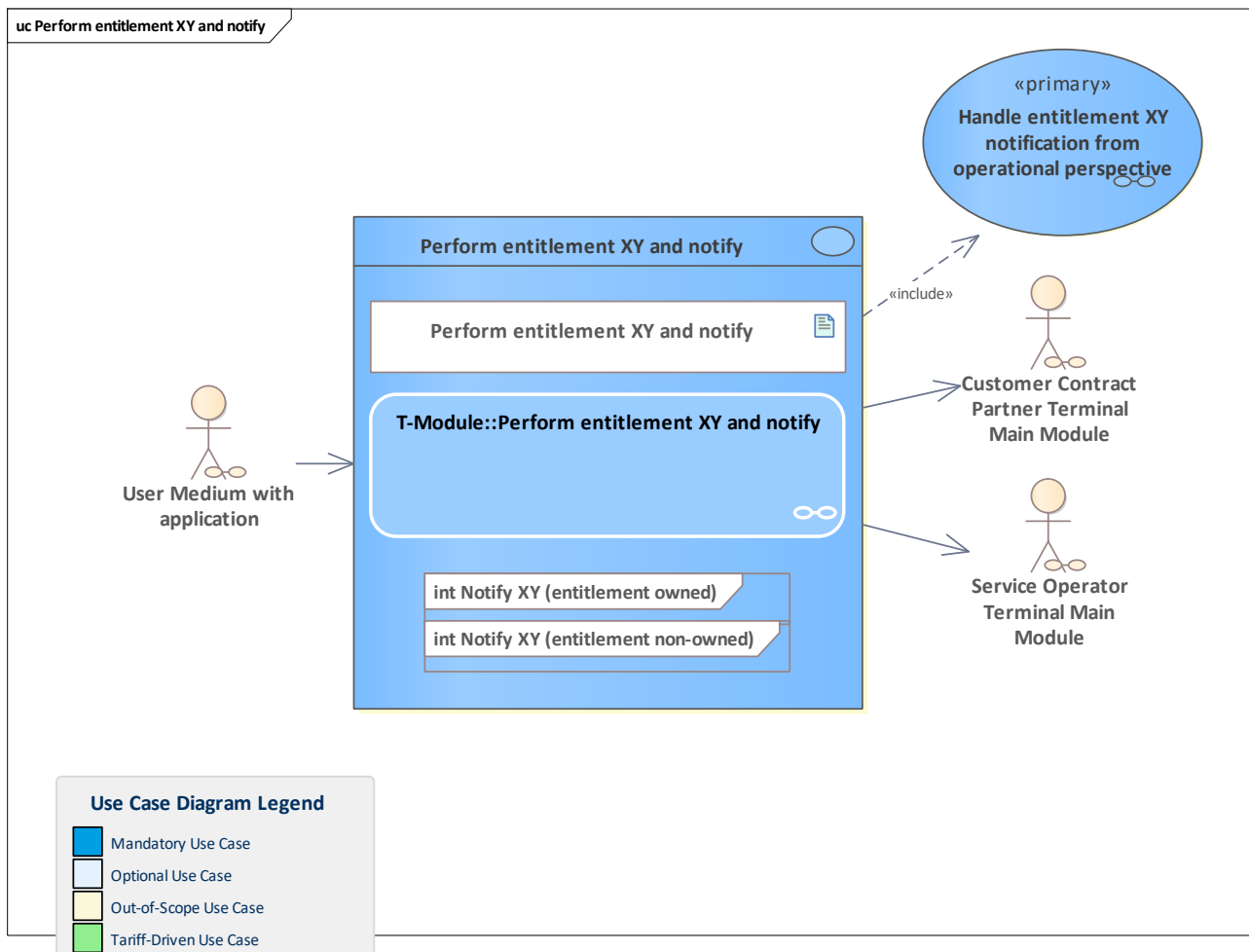


Figure 434: Perform entitlement XY and notify

The user medium-based entitlement action XY is performed by the terminal in conjunction with a SAM and the back-office system corresponding to the terminal is notified about the action execution.

Combines the transaction including the action XY and the notification processes triggered by this. Shows the interaction between the UM transaction and the notification processes. The timeout warning is generated in the corresponding diagram if needed.

The business preconditions to perform the transaction (e.g. determining the actual need to block an entitlement or making sure that a stored-value payment method is sufficiently charged) are already satisfied before the process shown in this diagram begins.

Note:

Executing orders in the context of ordered action management involves the same UM operations as outside of that scope, but leads to different notification processes (partly belonging to different notification categories). In this case, only one diagram "Perform transaction to XY" may be needed, which is used in two diagrams "T-XYZ::Perform XY and notify" combining it with different notification activities (i.e. the ordered and the regular variant).

11.300 Perform ordered entitlement blocking and notify

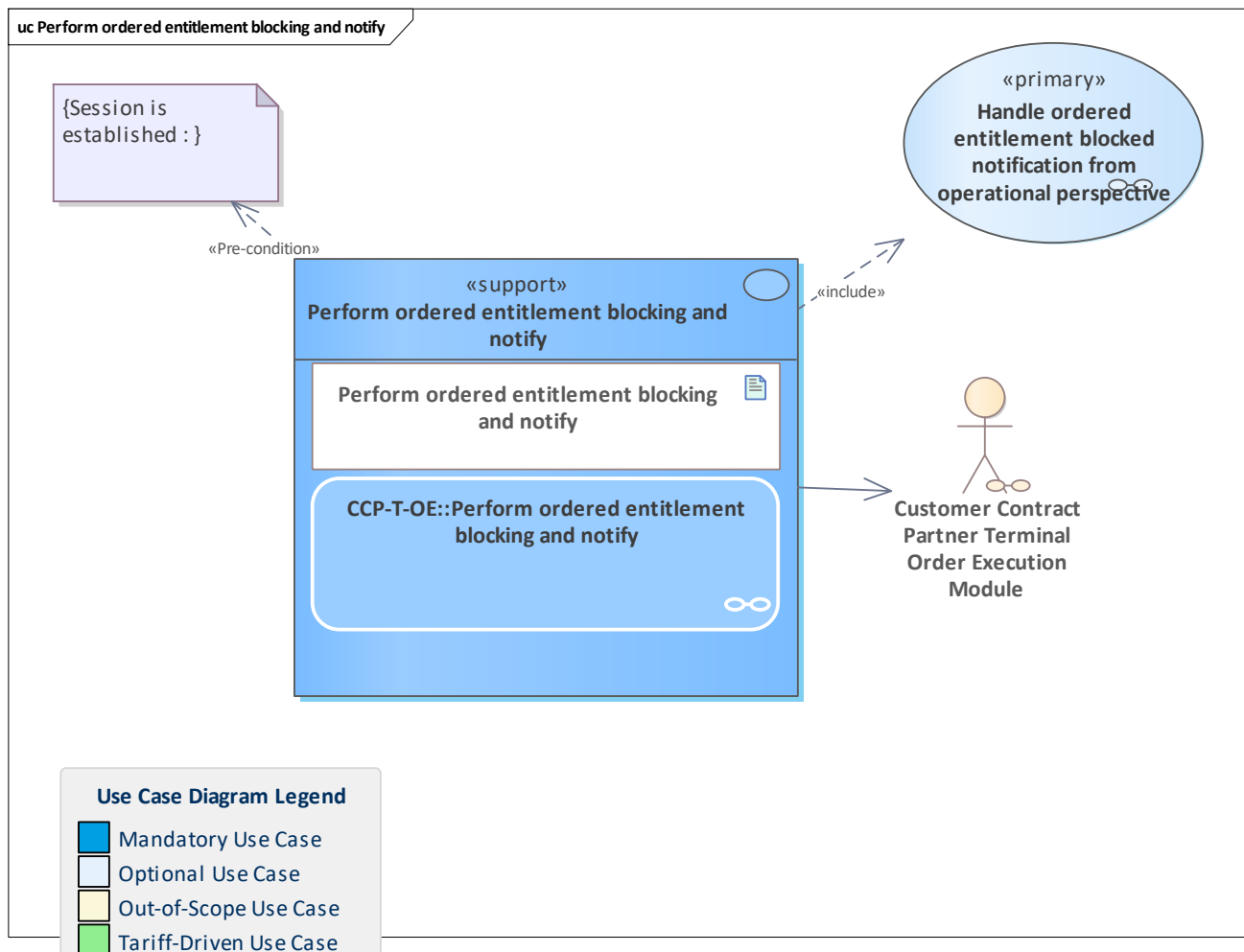


Figure 435: Perform ordered entitlement blocking and notify

Execute an entitlement blocking order and notify the own back-office system about it. If the transaction is aborted, the CCP back-office system is also notified for consistent monitoring data.

11.301 Perform ordered entitlement issuance and notify

11.302 Perform ordered entitlement issuance and notify

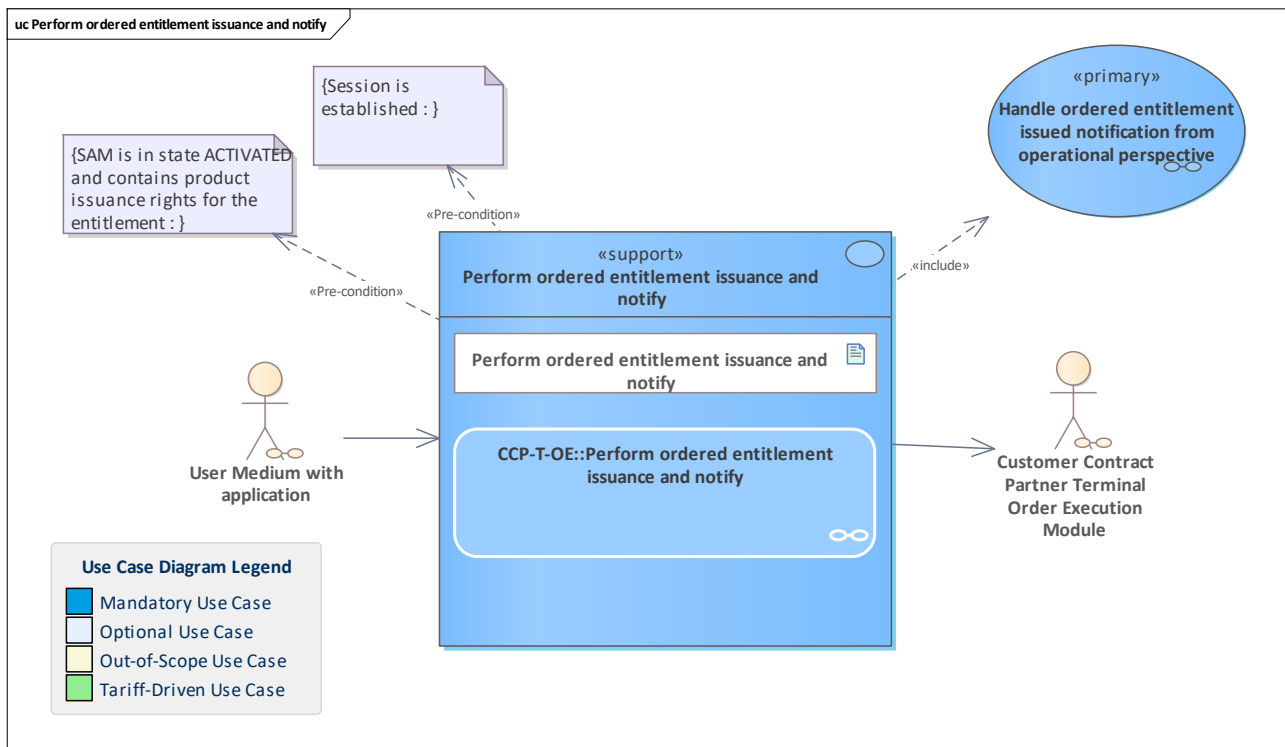


Figure 436: Perform ordered entitlement issuance and notify

The CCP terminal with an action management extension executes an entitlement issuance order and notifies the own back-office system about it.
If the transaction is aborted, the CCP back-office system is also notified for consistent monitoring data.

11.303 Perform ordered entitlement termination and notify

11.304 Perform ordered entitlement termination and notify

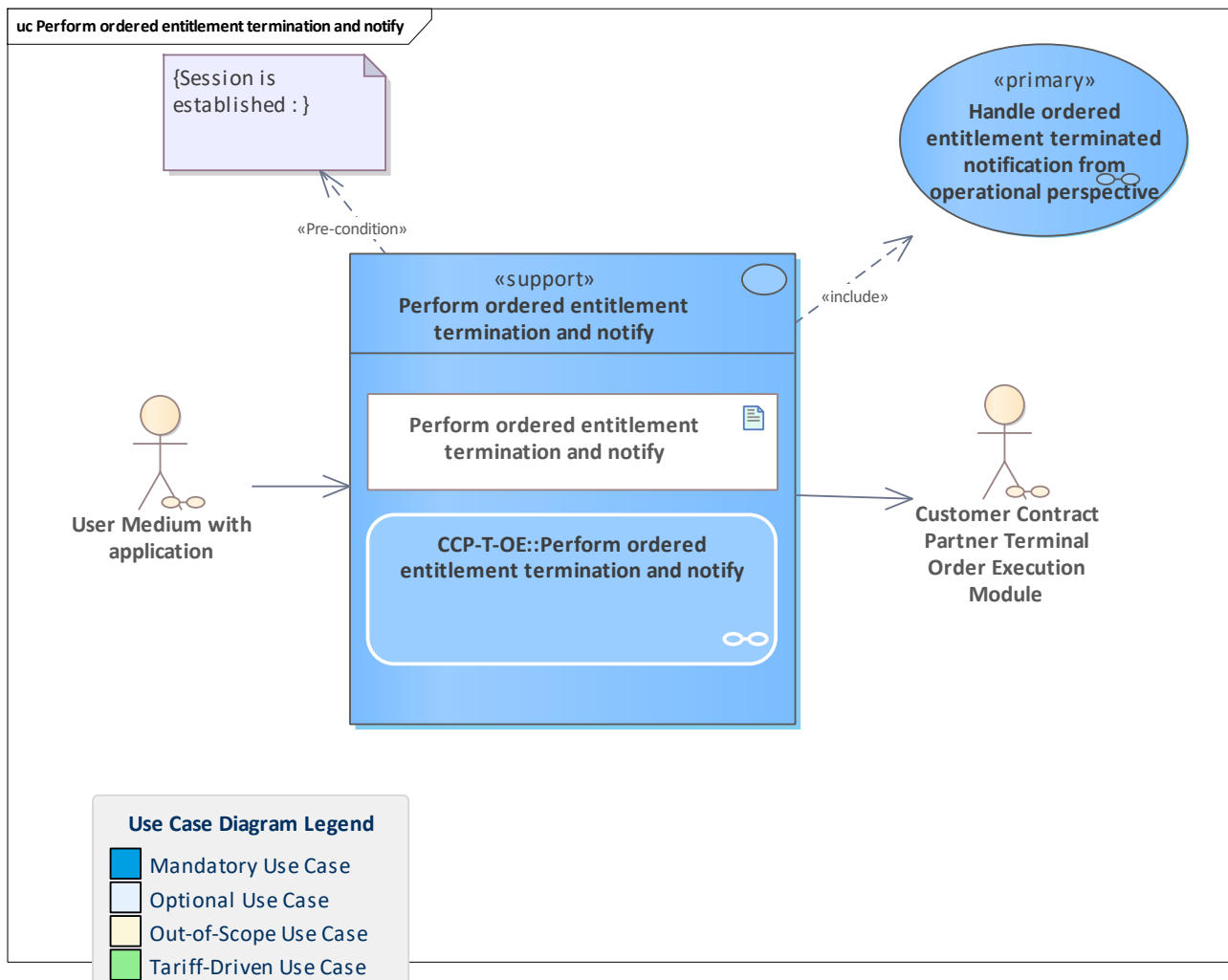


Figure 437: Perform ordered entitlement termination and notify

Execute an entitlement termination order in the CCP terminal with an action management extension and notify the own back-office system about it.
If the transaction is aborted, the CCP back-office system is also notified for consistent monitoring data.

11.305 Perform ordered entitlement unblocking and notify

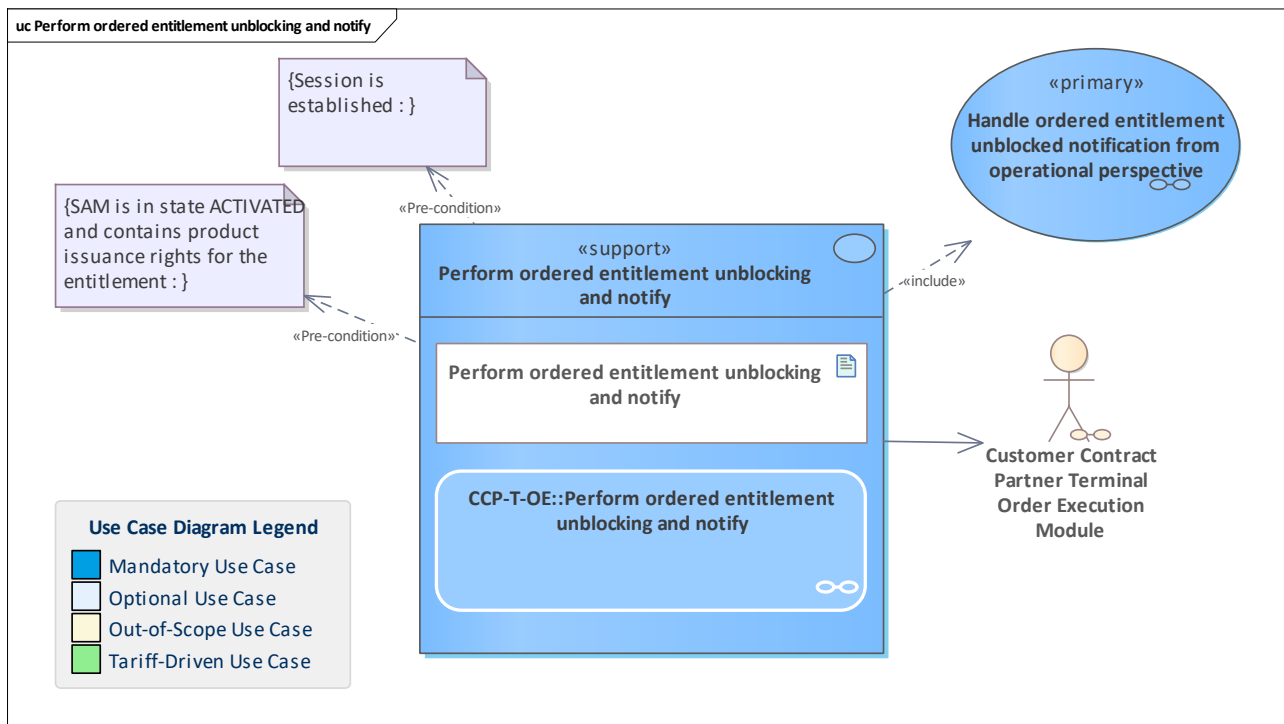


Figure 438: Perform ordered entitlement unblocking and notify

The CCP terminal executes an entitlement unblocking order and notifies the own back-office system about it.
If the transaction is aborted, the CCP back-office system is also notified for consistent monitoring data.

11.306 Perform stored-value payment method crediting and notify

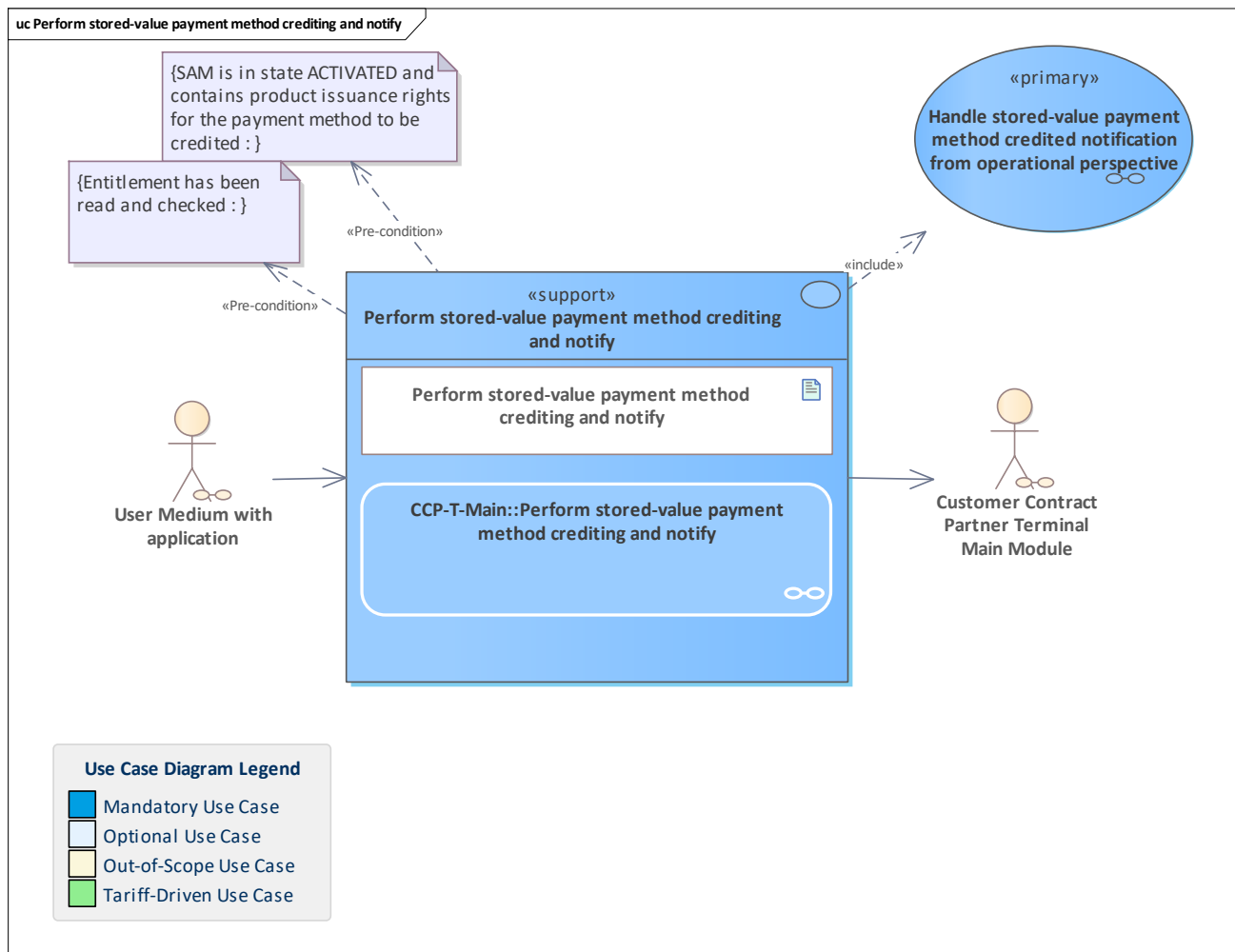


Figure 439: Perform stored-value payment method crediting and notify

The CCP terminal performs the transaction to credit a stored-value payment method and notifies the responsible CCP back-office system about it. A potential transaction abortion is also notified for consistent monitoring data.

11.307 Perform stored-value payment method debiting and notify

11.308 Perform stored-value payment method debiting and notify

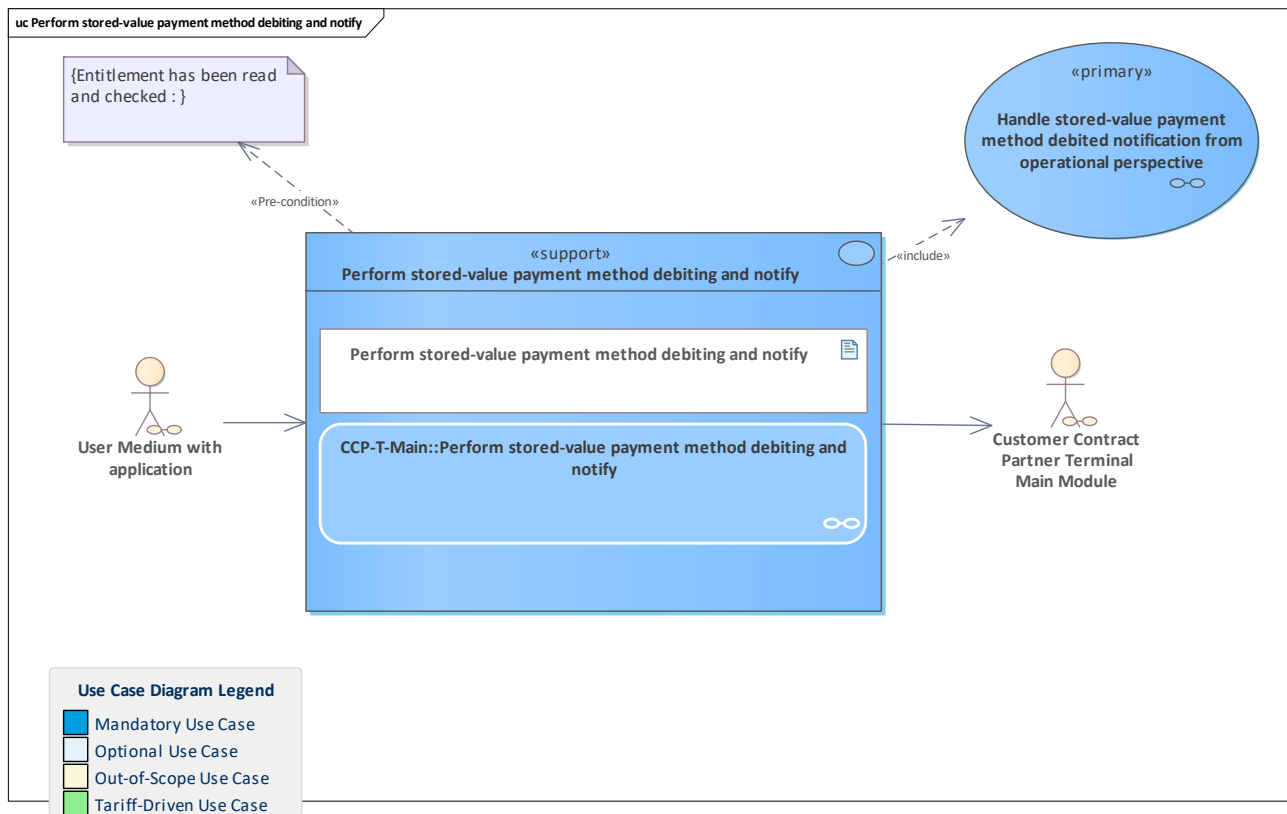


Figure 440: Perform stored-value payment method debiting and notify

The CCP terminal performs a transaction to debit a stored-value payment method on a user medium with an application and notifies the CCP back-office system about the debiting transaction.

If the transaction is aborted, the CCP back-office system is also notified.

11.309 Perform stored-value payment method recharging and notify

11.310 Perform stored-value payment method recharging and notify

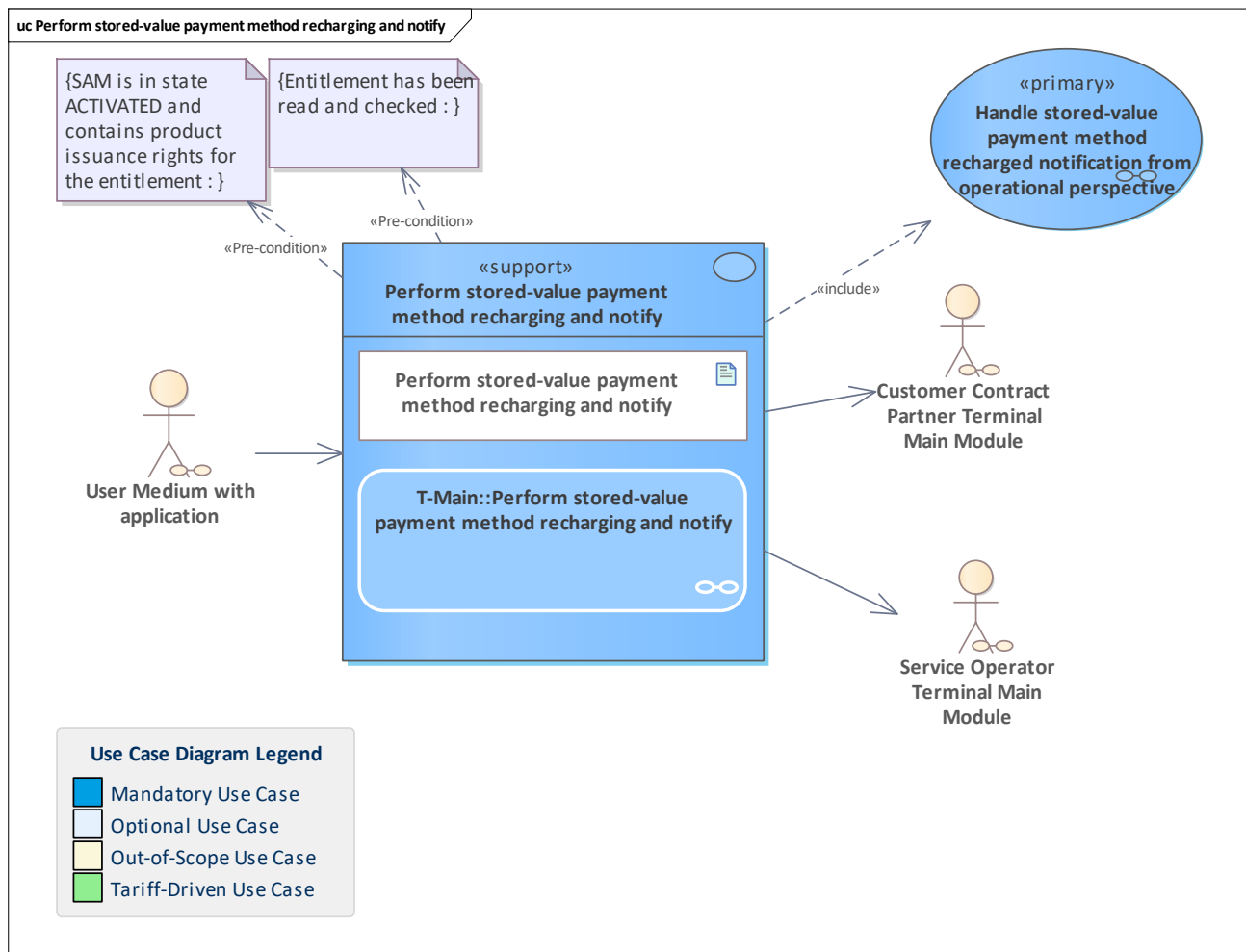


Figure 441: Perform stored-value payment method recharging and notify

The terminal performs the transaction to recharge a stored-value payment method and notifies the corresponding back-office system about it.
 In case of a performed autoload-triggered recharge in a CICO terminal, the notification is sent to the responsible SO back-office system.
 Otherwise, the notification is sent to the related CCP back-office system.
 If the transaction is aborted, the responsible back-office system is also notified.

11.311 Perform stored-value payment method reimbursing and notify

11.312 Perform stored-value payment method reimbursing and notify

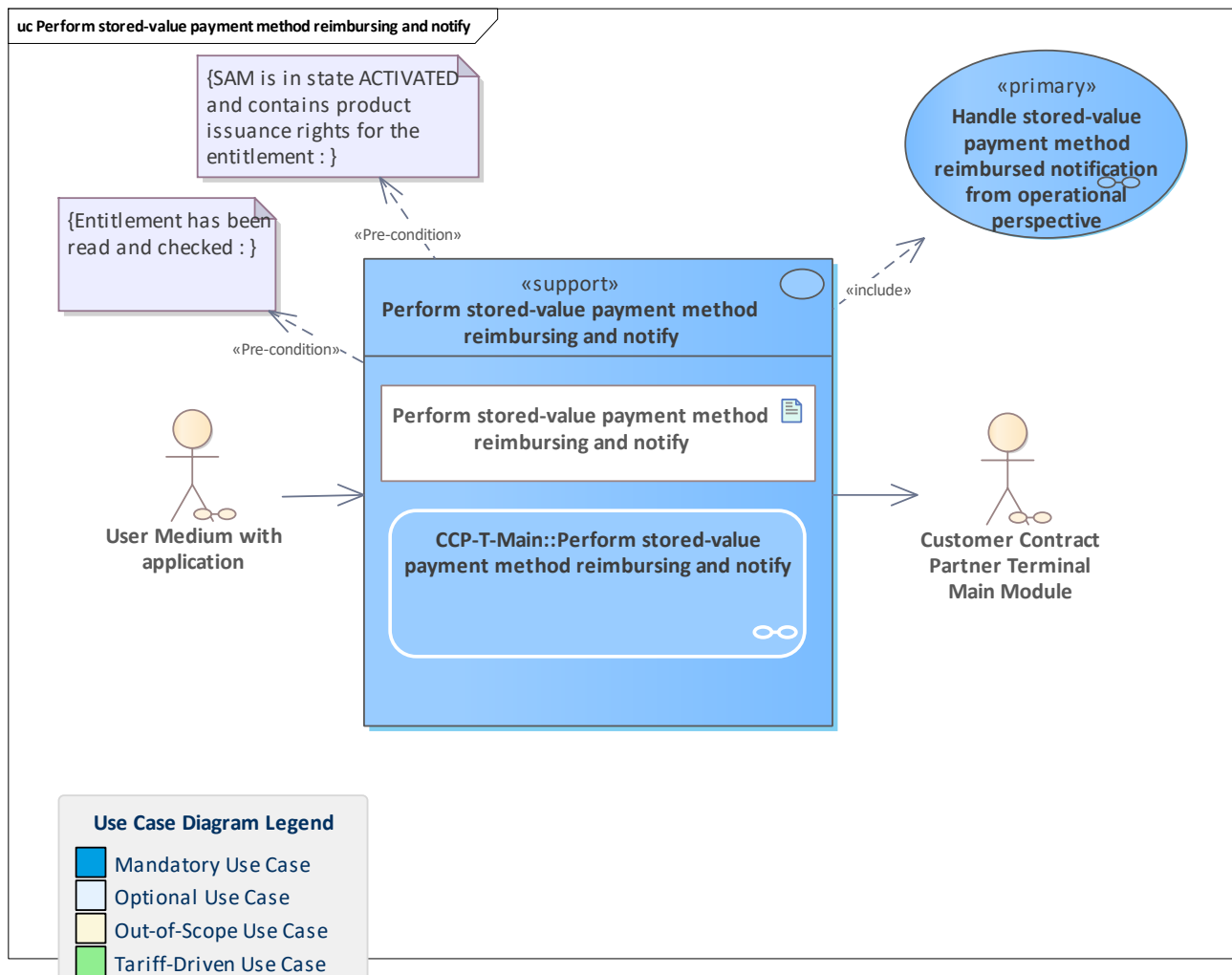


Figure 442: Perform stored-value payment method reimbursing and notify

The CCP terminal performs stored-value payment method reimbursement and notifies the responsible CCP back-office system.
If the transaction is aborted, the CCP back-office system is also notified for consistent monitoring data.

11.313 Perform user tariff parameters change and notify

11.314 Perform user tariff parameters change and notify

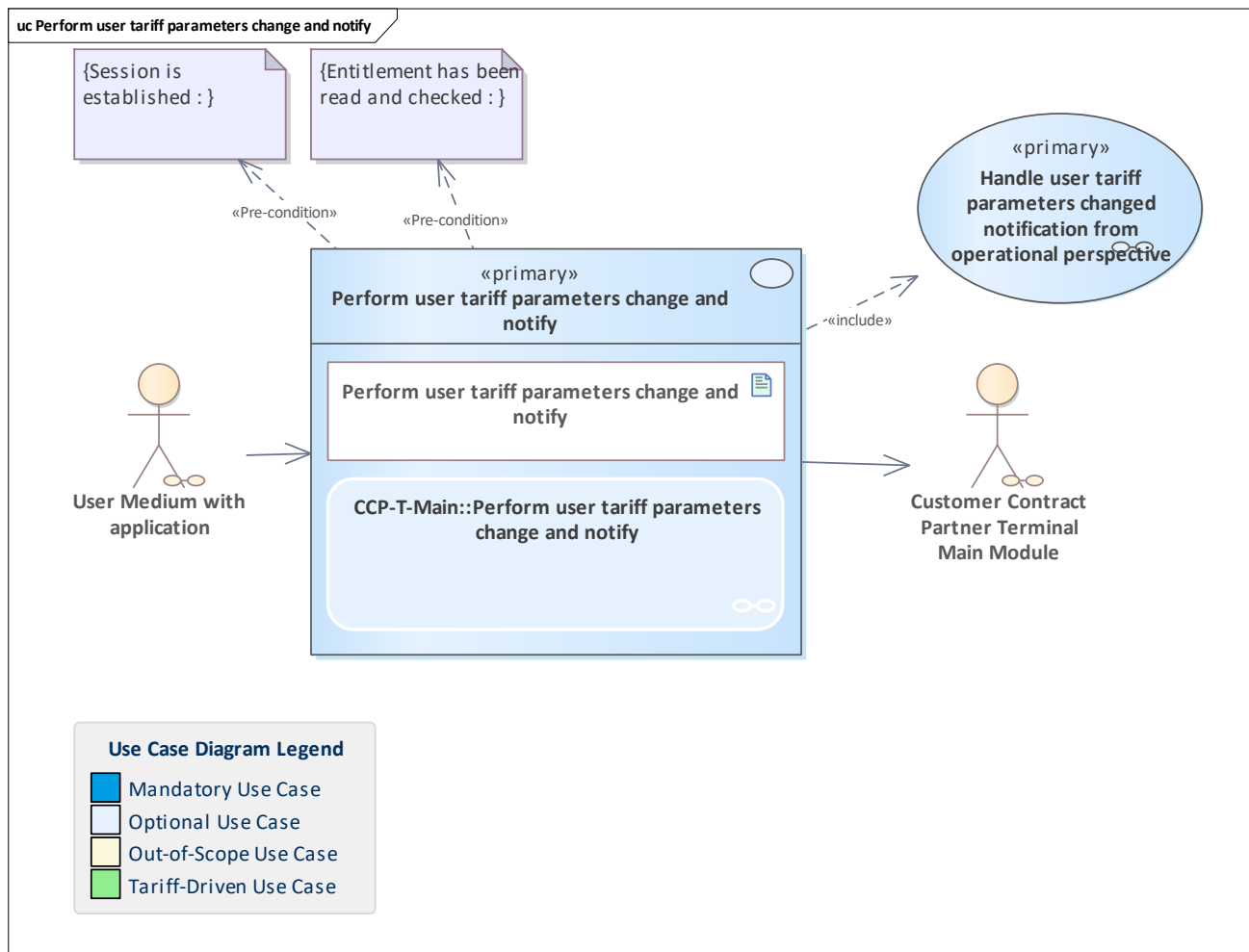


Figure 443: Perform user tariff parameters change and notify

The CCP terminal performs a transaction to change user tariff parameters and notifies the responsible CCP back-office system about the change.
A detected transaction abortion is also sent to the CCP system for consistent monitoring data.

11.315 Personalise application

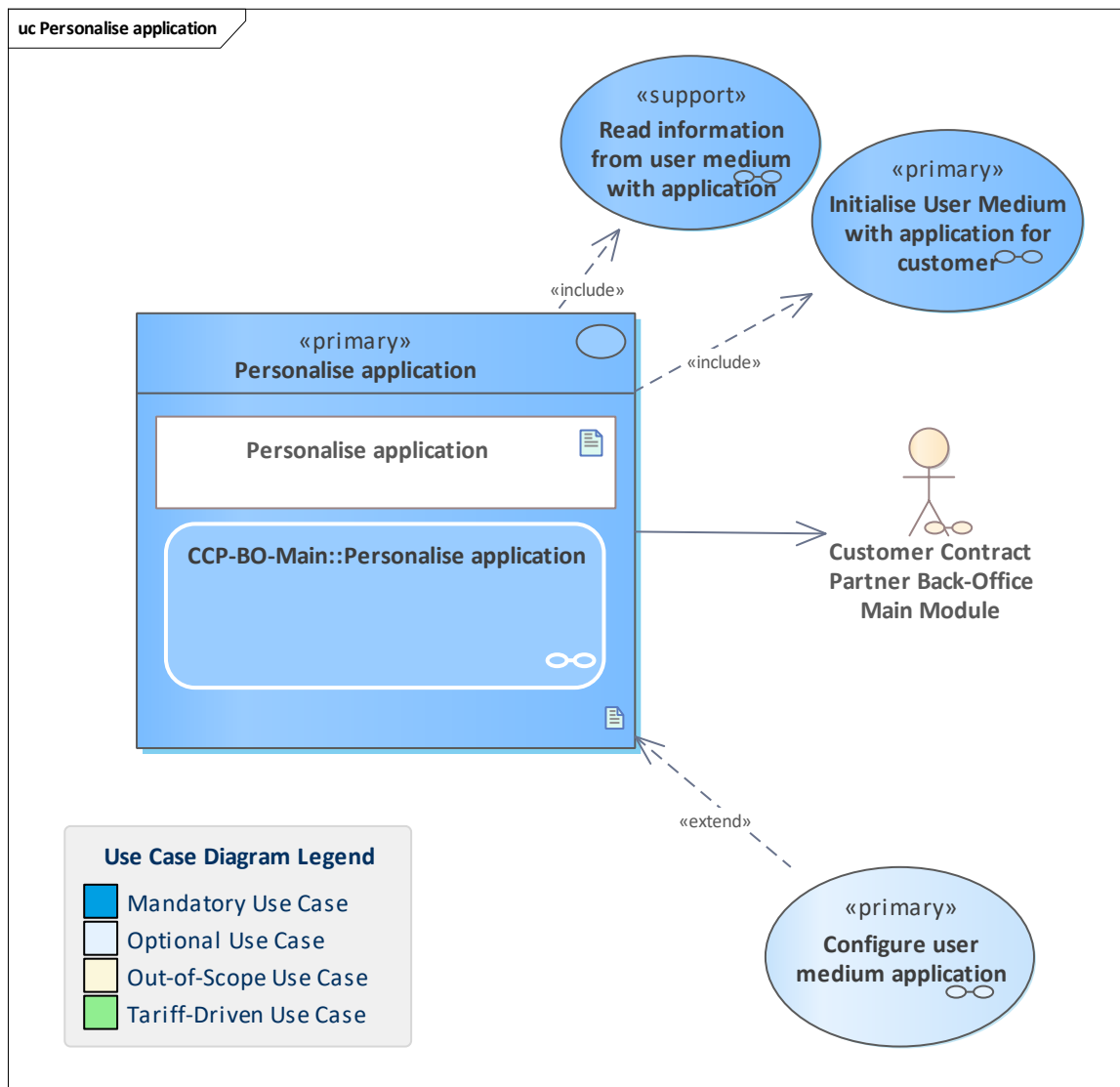


Figure 444: Personalise application

The CCP back-office system personalises a user medium application instance by first identifying it (using its app instance ID) and then writing the customer-related information to it, which is also stored in the CCP system.

11.316 Print customer receipt

11.317 Print customer receipt

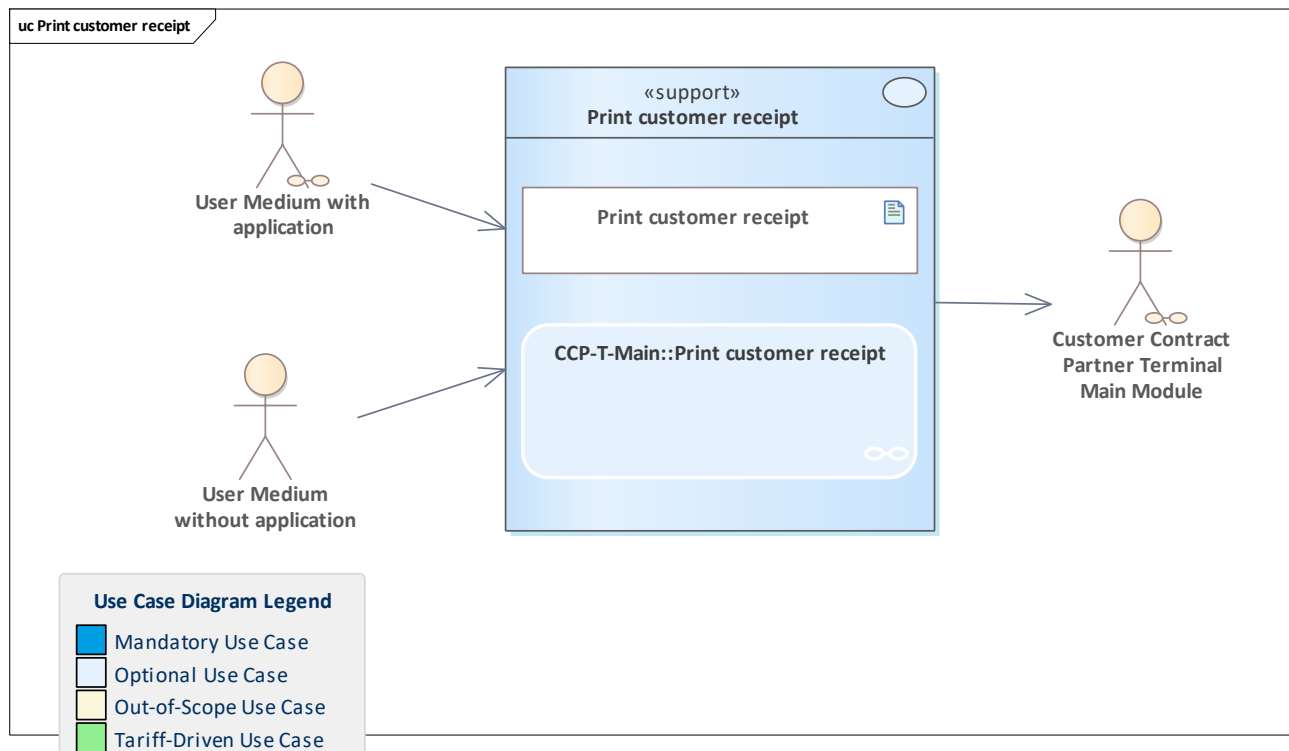


Figure 445: Print customer receipt

A customer receipt is printed e.g. after selling an entitlement or having entitlements displayed on a terminal.

11.318 Process action list retrieval configuration

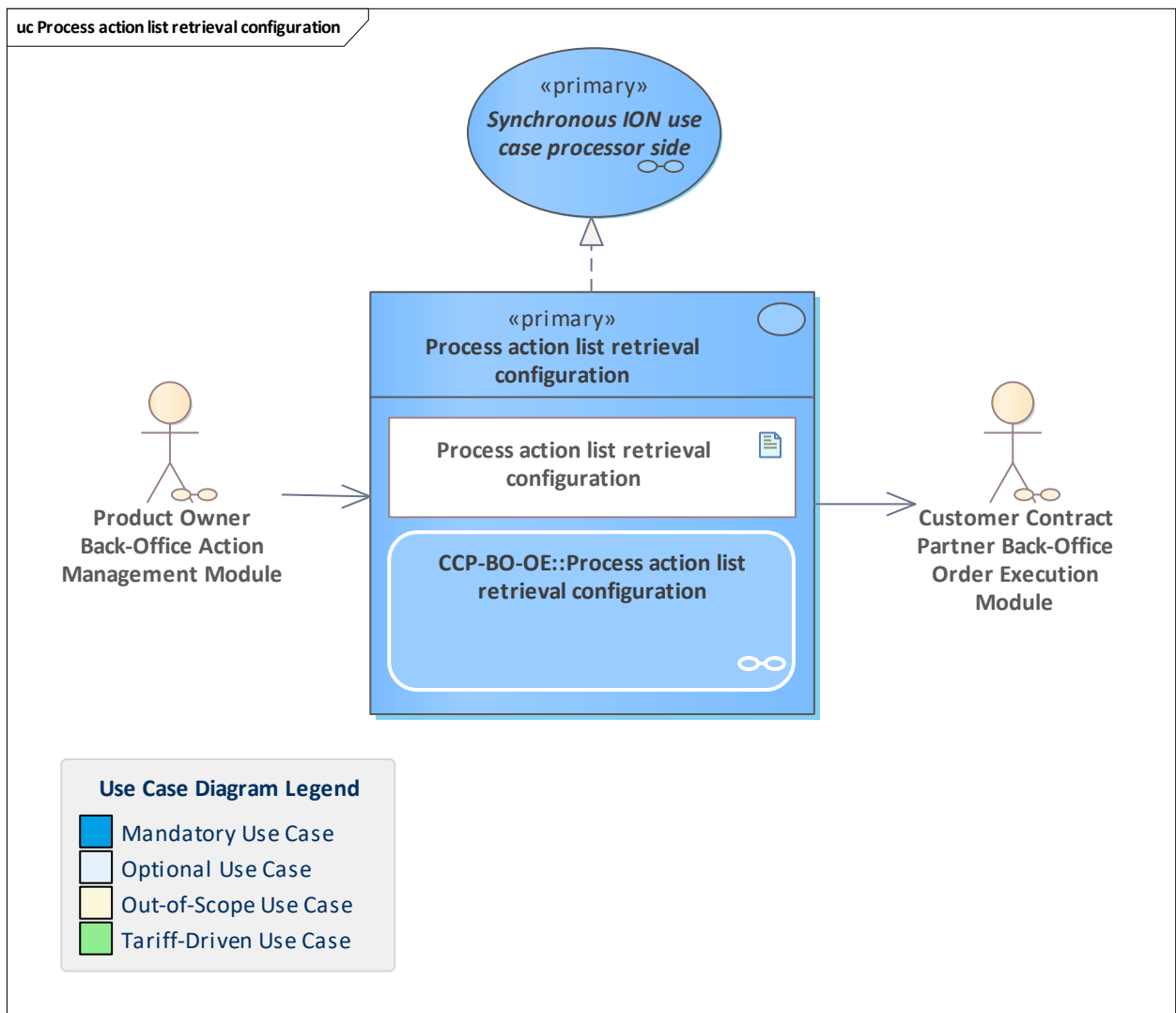


Figure 446: Process action list retrieval configuration

The action list retrieval configuration provided by the [Product Owner Back-Office Action Management Module](#) is stored in the [Customer Contract Partner Back-Office Order Execution Module](#). The configuration contains information about whether and when the action list is provided in an updated form. It overwrites any previous configurations.

11.319 Process extended logging for an application

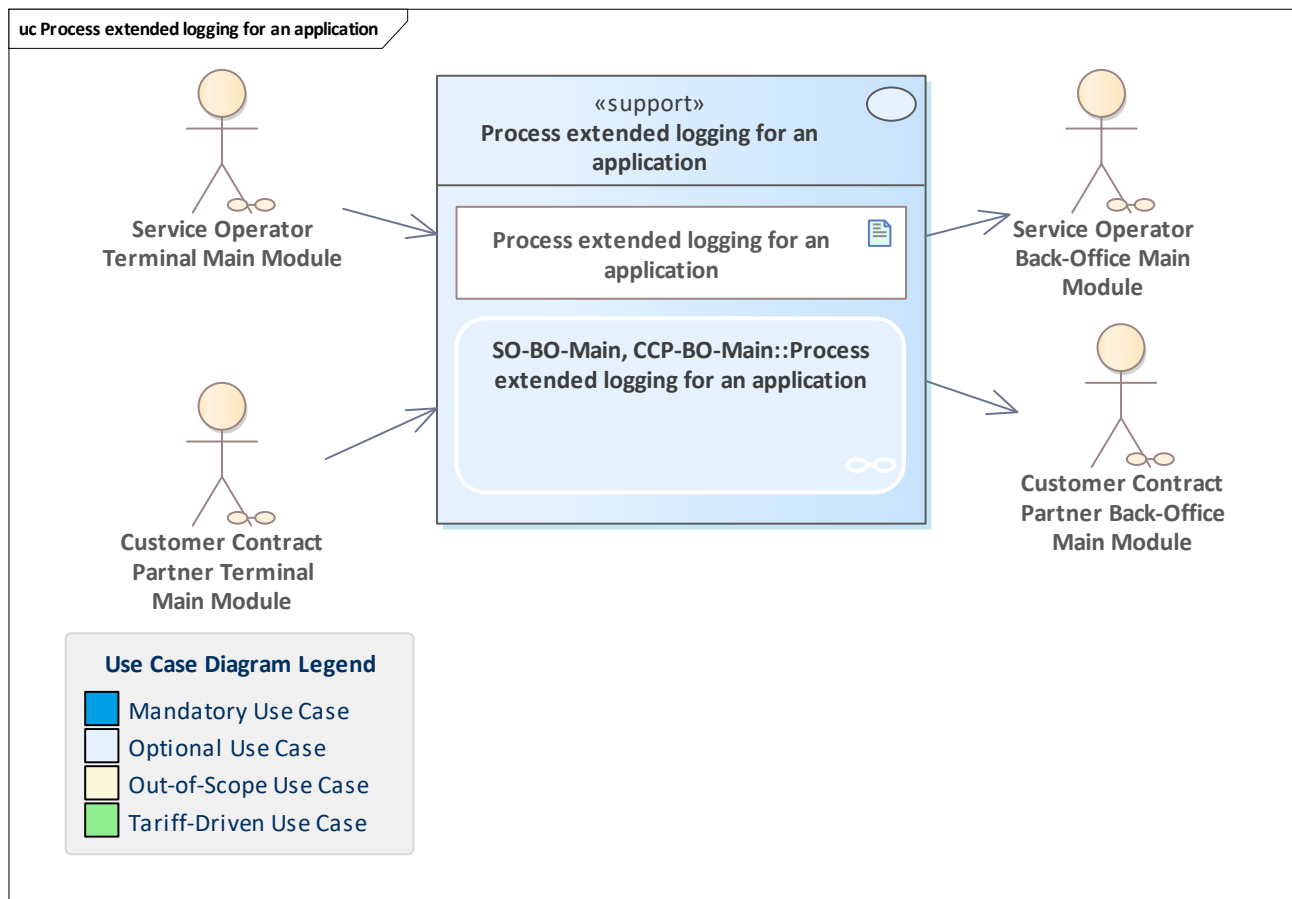


Figure 447: Process extended logging for an application

Extended logging for an application is received from a terminal by the back-office system and registered.

11.320 Process extended logging for an entitlement

11.321 Process extended logging for an entitlement

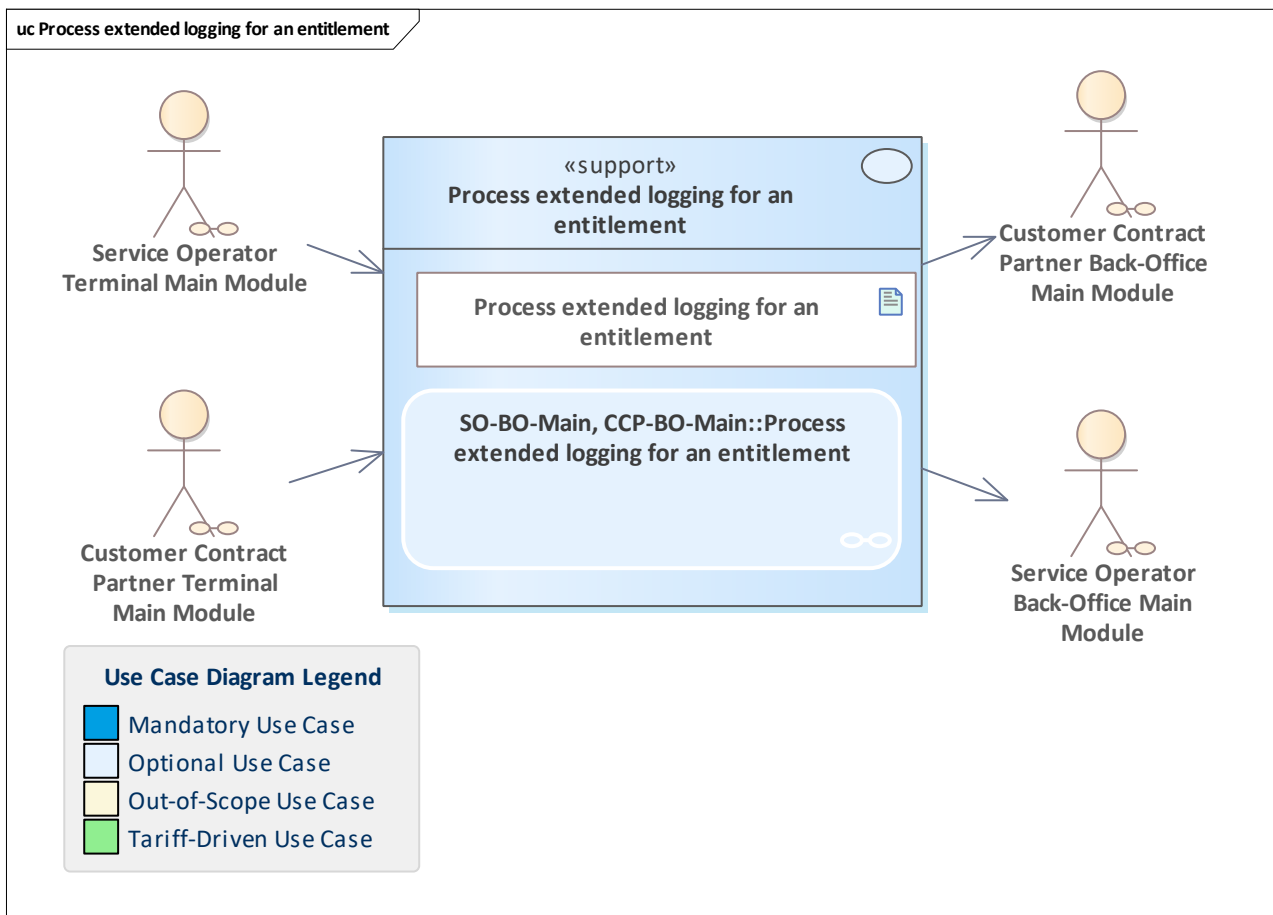


Figure 448: Process extended logging for an entitlement

Extended logging for an entitlement is received from a terminal by the back-office system and registered.

11.322 Process media shipment list

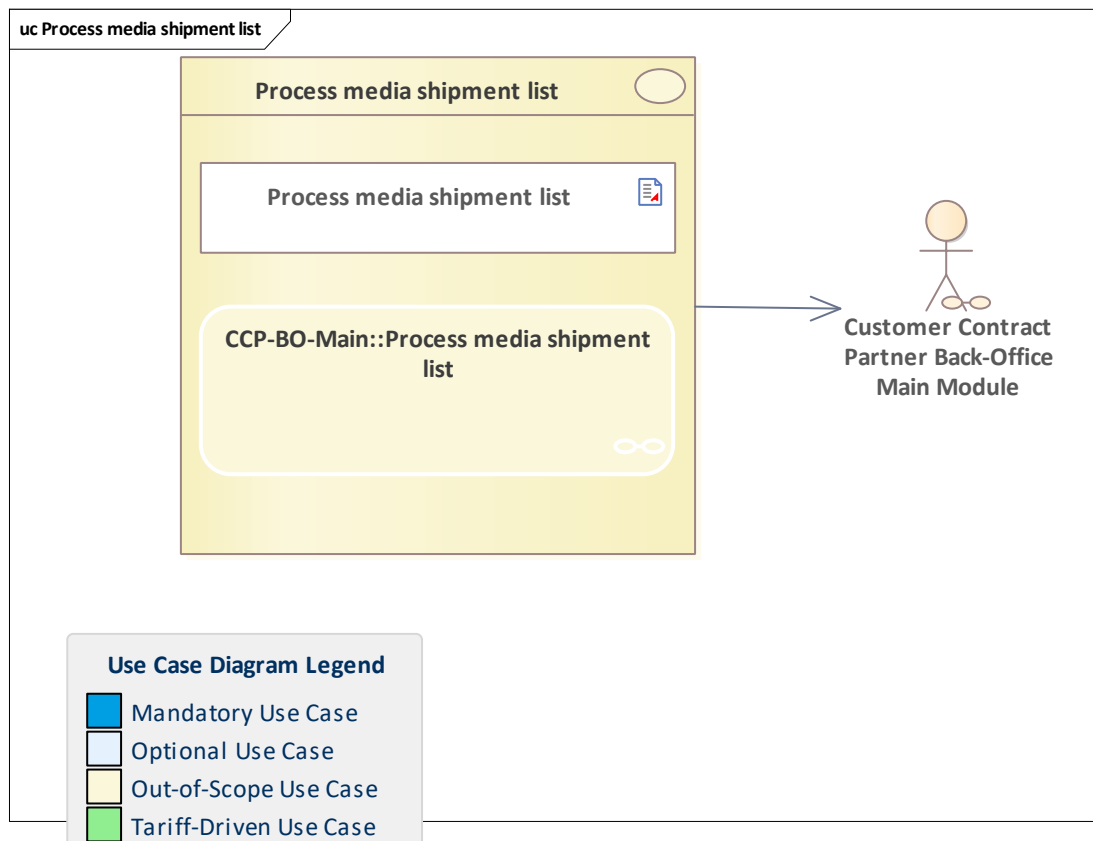


Figure 449: Process media shipment list

The CCP-BO-Main processes the media shipment list for an order of user media. The contained medium IDs and application instance IDs are registered and the order state is changed to *received*.

Apart from the interface, this use case is not specified in detail.

11.323 Process new information about customer and discounts

11.324 Process new information about customer and discounts

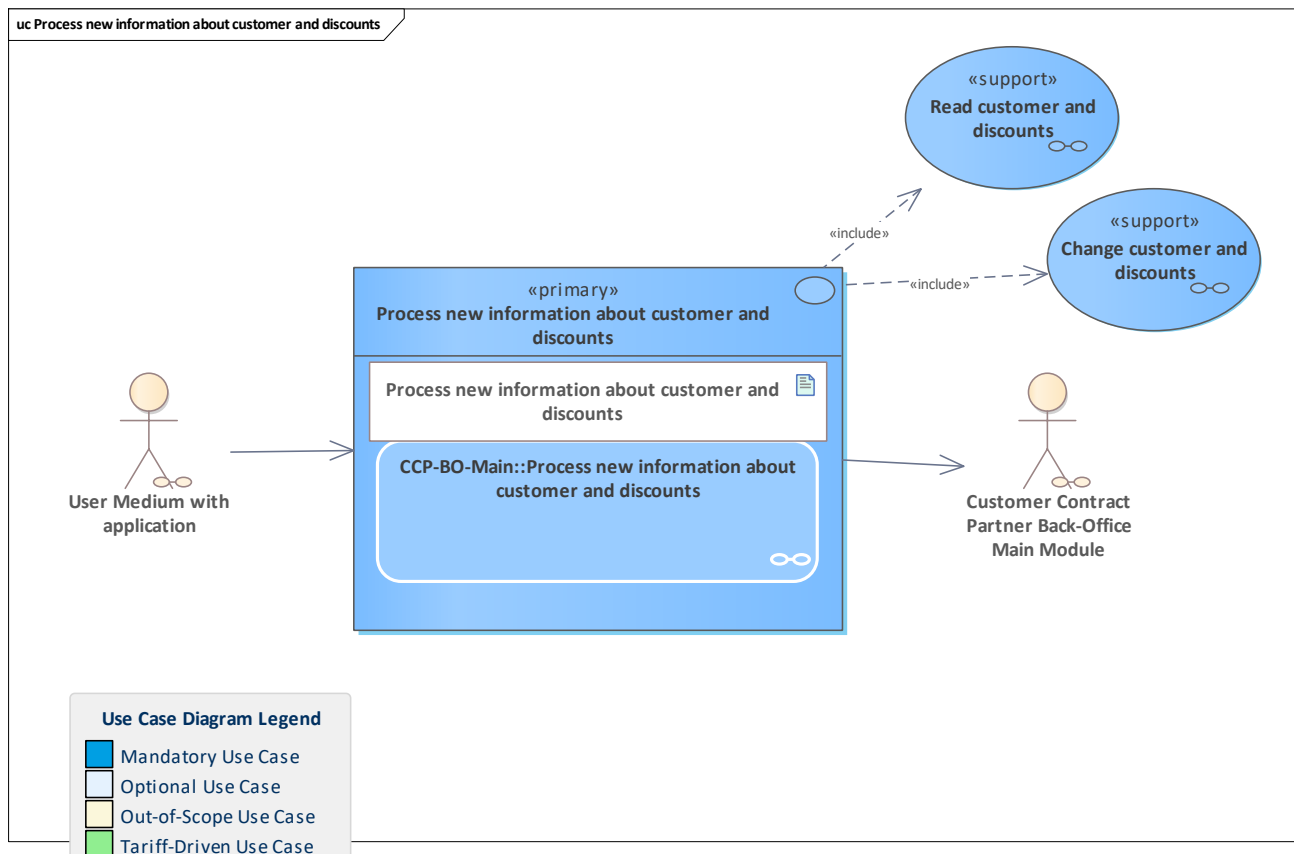


Figure 450: Process new information about customer and discounts

The CCP back-office system collects new or changed customer data and triggers the terminal to process this data on the currently involved user medium.

Information about the customer and his discounts written on the user medium with application needs to be changed. Therefore, the customer and his discounts are read, changed and written again.

Involved entitlements are changed (terminated and issued with the new data) due to the changed customer data.

11.325 Process retrieval request for organisation list

11.326 Process retrieval request for organisation list

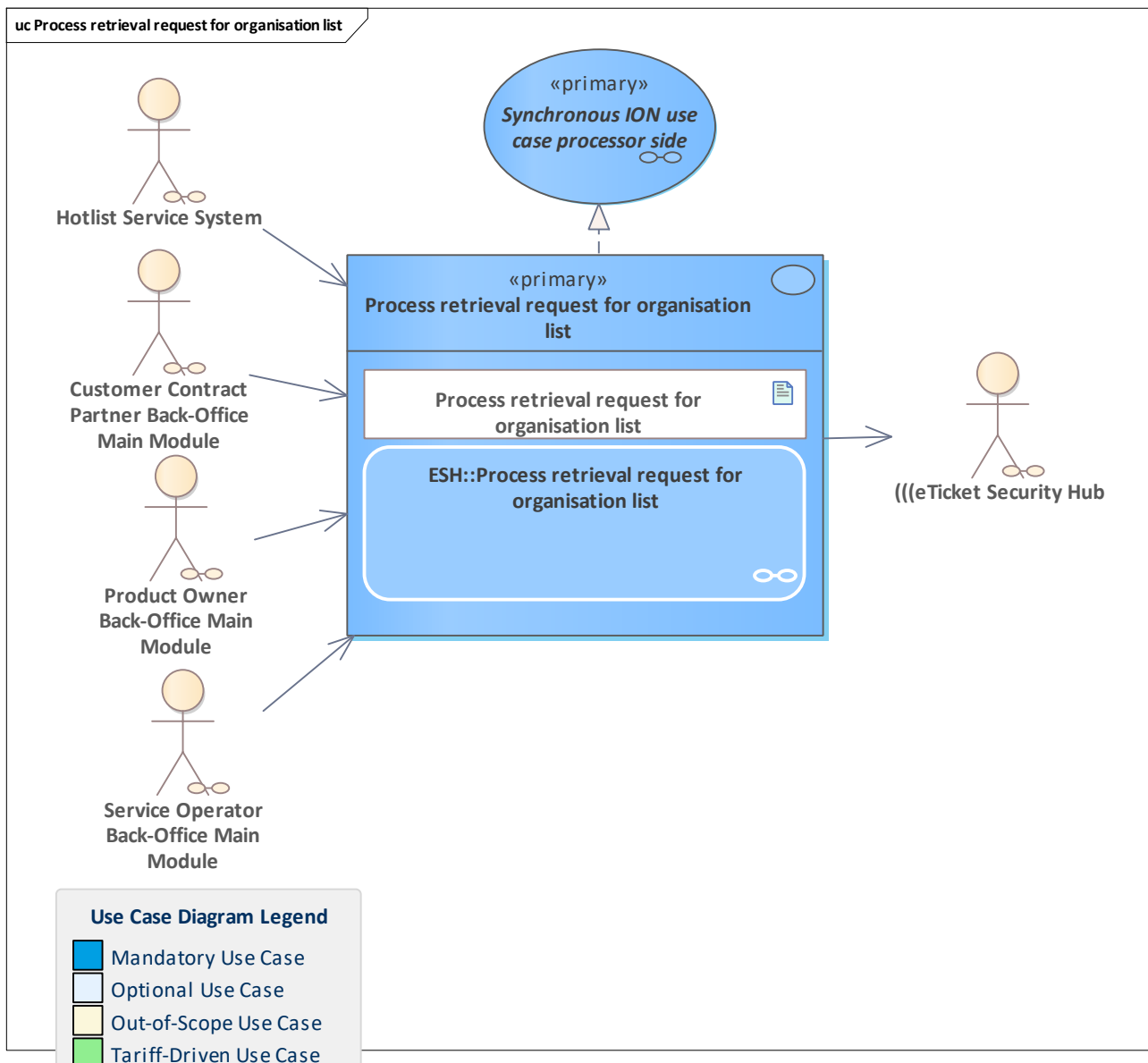


Figure 451: Process retrieval request for organisation list

The registrar creates an organisation list and sends it to the requestor.

11.327 Put action list retrieval configuration

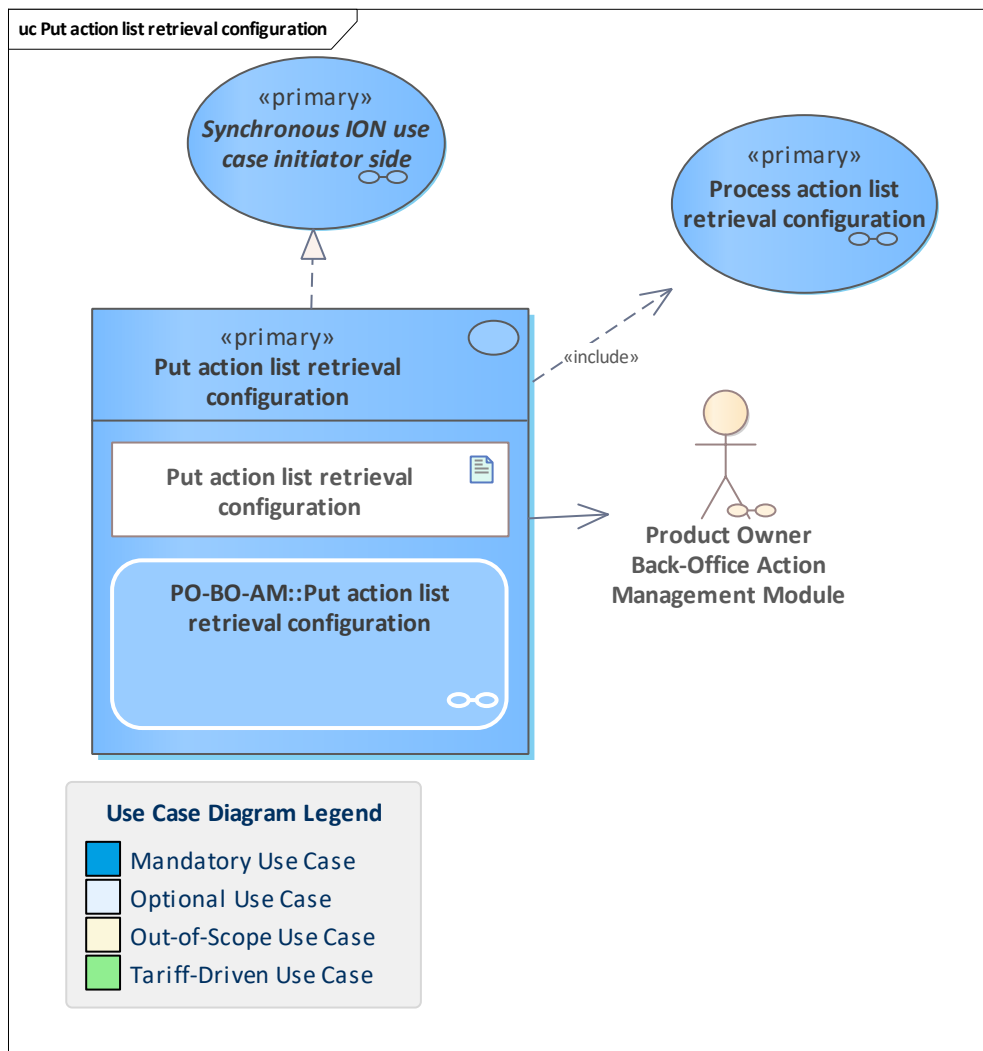


Figure 452: Put action list retrieval configuration

The action list retrieval configuration is sent to an/the [Customer Contract Partner Back-Office Order Execution Module](#)(s) by the [Product Owner Back-Office Action Management Module](#). The configuration contains information about whether and when the action list is provided in an updated form. It overwrites any previous configurations.

11.328 Read customer and discounts

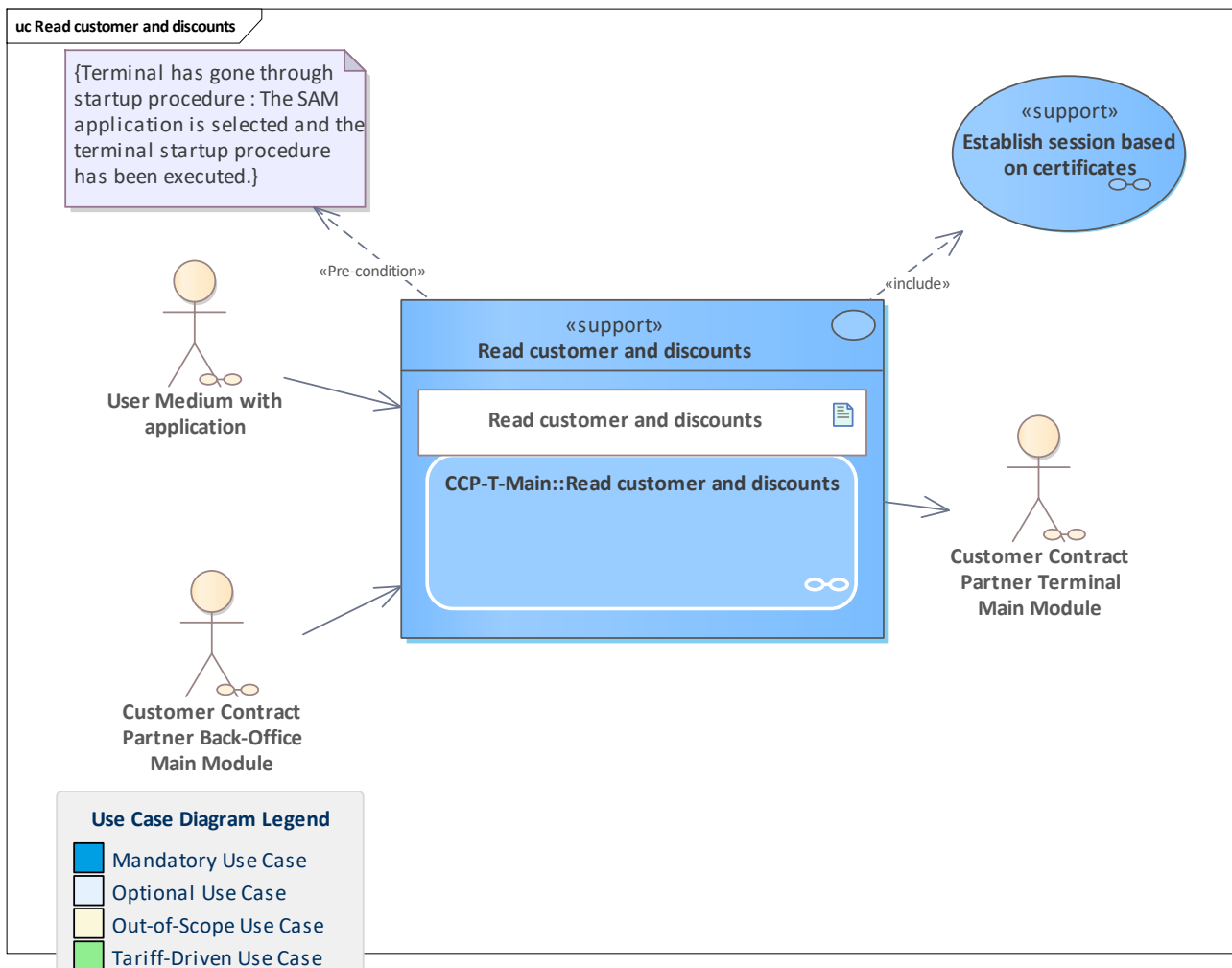


Figure 453: Read customer and discounts

The customer and his discounts are read by an authorised terminal to support primary use cases.

11.329 Read information from user medium with application

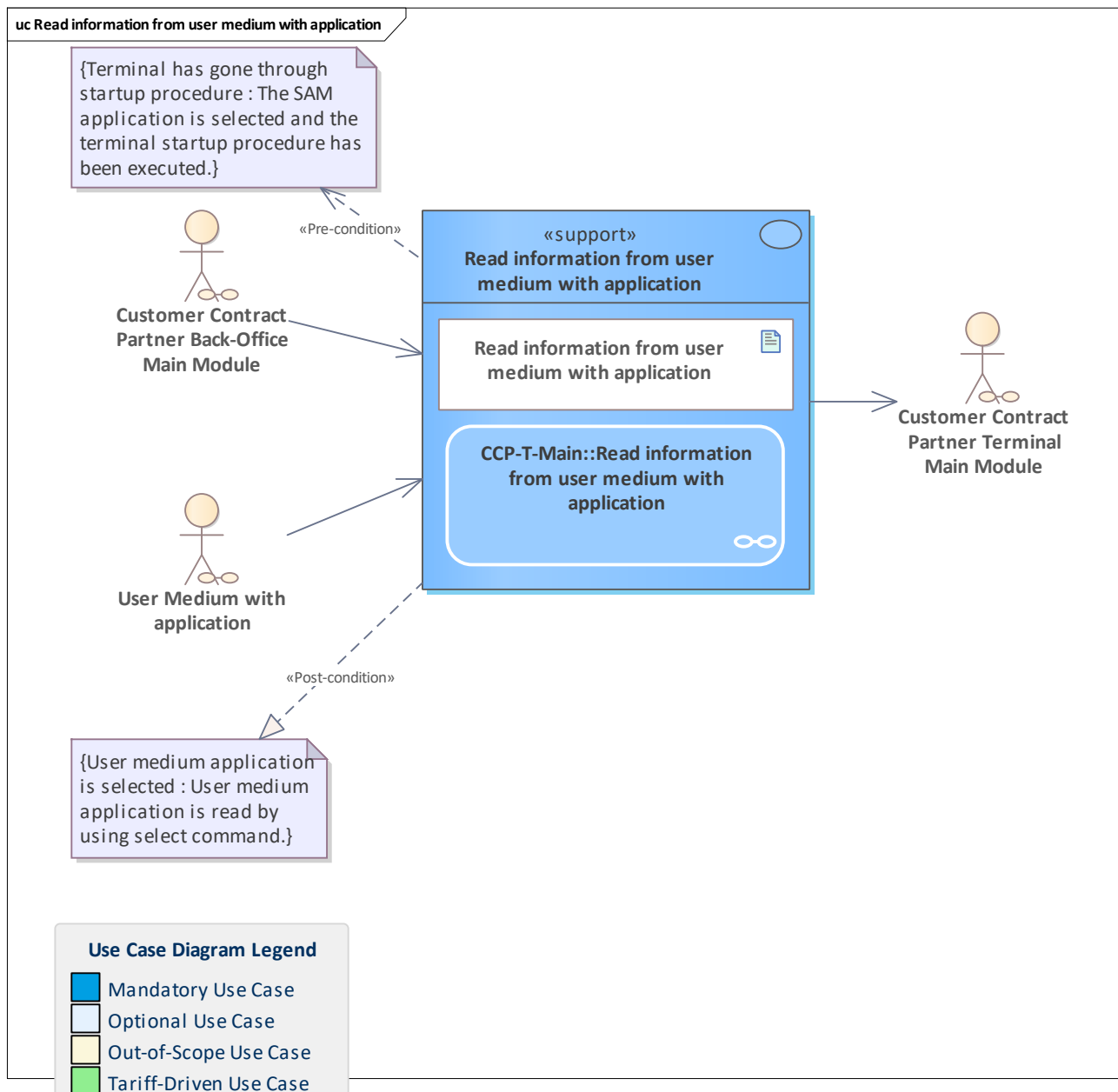


Figure 454: Read information from user medium with application

Read the application instance ID and validity period of a user medium and pass it back to the requesting CCP back-office system. Used in the context of user medium personalisation.

11.330 Recharge stored-value payment method

11.331 Recharge stored-value payment method

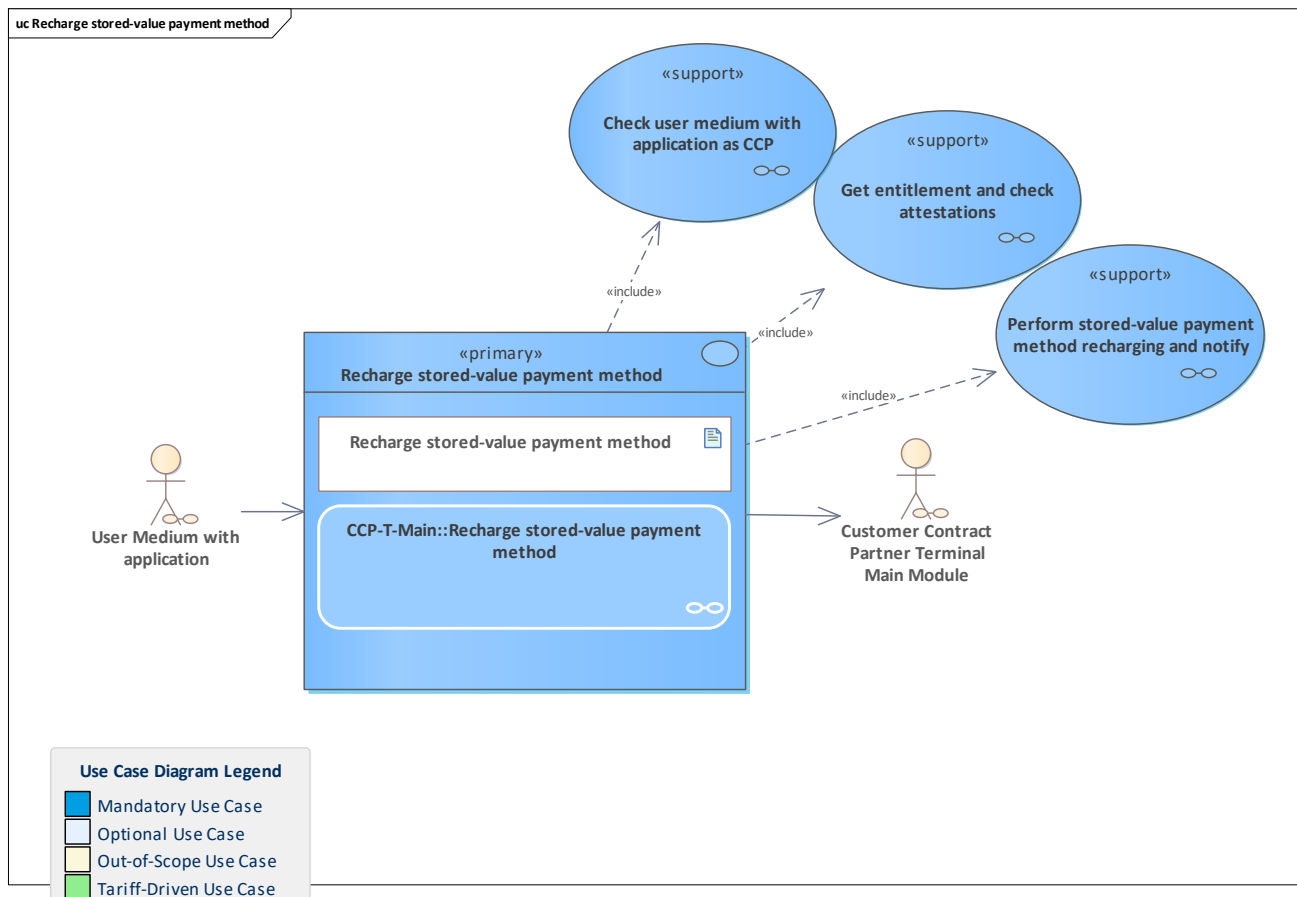


Figure 455: Recharge stored-value payment method

The CCP terminal lets the customer recharge a chosen amount onto his stored-value payment method.

The use case checks the stored-value payment method and ensures, that the maximum balance in the stored-value account is not exceeded for the intended recharge transaction.

Finally, the recharge is performed and notified to the CCP back-office system (same CCP as the terminal operator of the current terminal).

11.332 Record entitlement within CICO system

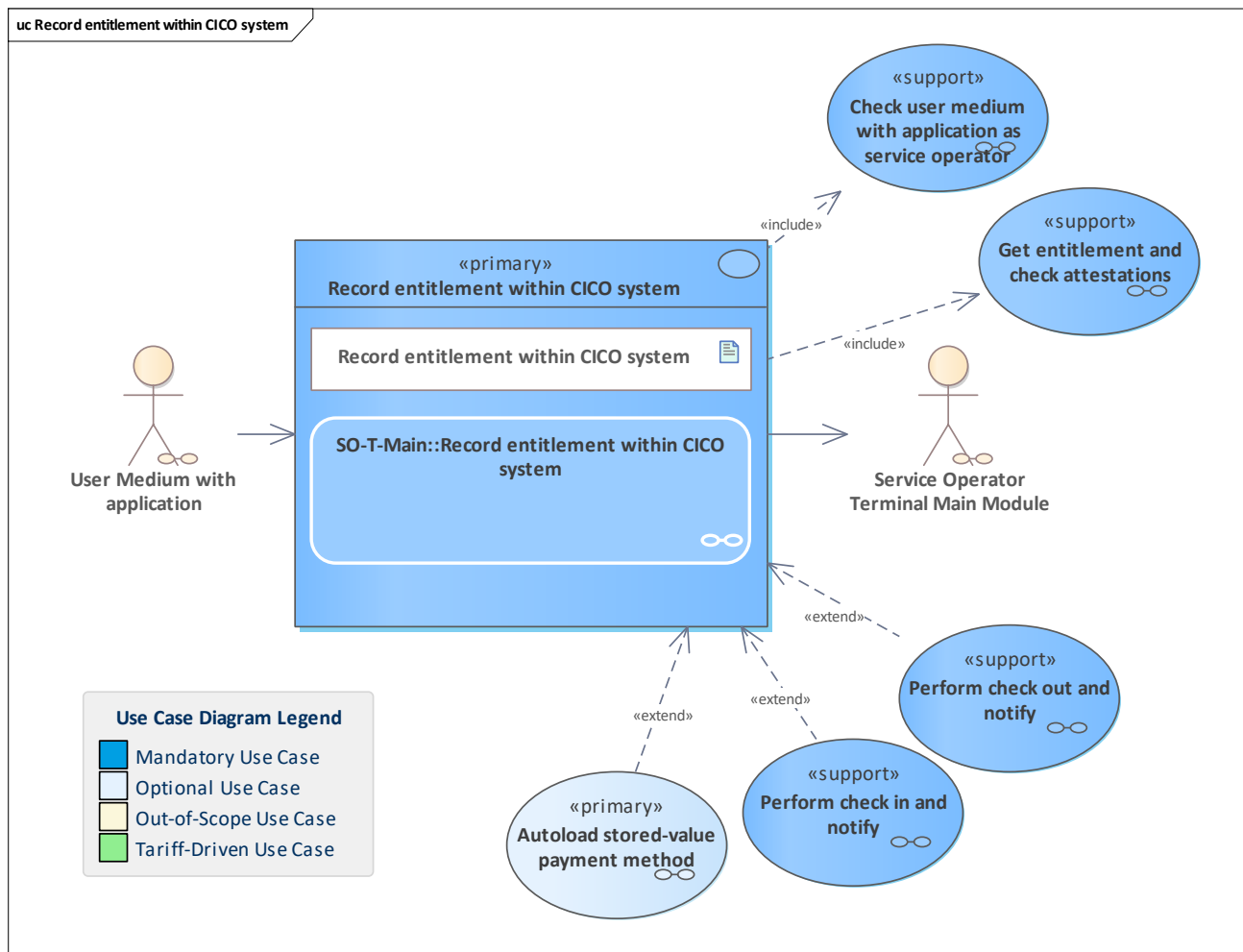


Figure 456: Record entitlement within CICO system

A customer wants to ride with her/his entitlement in the CICO system. Therefore, the entitlement needs to be recorded. The type of entitlement is always an ((etiCORE payment method. The entitlement is checked by different tariff and technical aspects. Accordingly, the check-in or check-out process is triggered and the result will be notified to the back-office system.

11.333 Reimburse and terminate account-based payment method

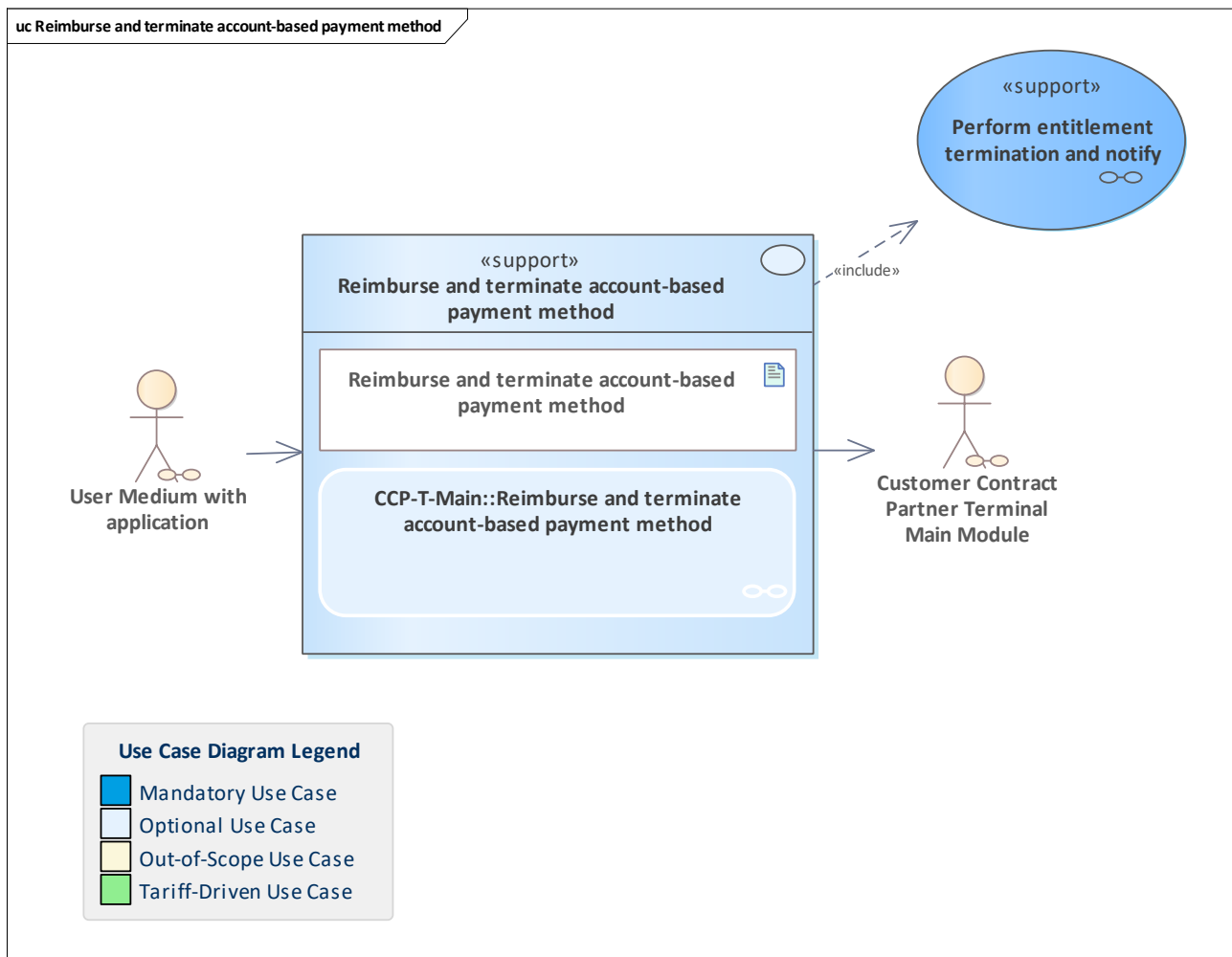


Figure 457: Reimburse and terminate account-based payment method

Reimburse and terminate an account-based payment method as a supporting use case. The primary use case is [Take back entitlement](#). Performed by the CCP terminal. The technical part is the termination of the account-based payment method which has to be done in the terminal. A reimbursement becomes necessary if the account-based payment method is based on a prepaid account with a remaining balance, or if a subscription was cancelled in the middle of the term. The termination is done and notified to the responsible CCP back-office system.

11.334 Reimburse and terminate electronic ticket

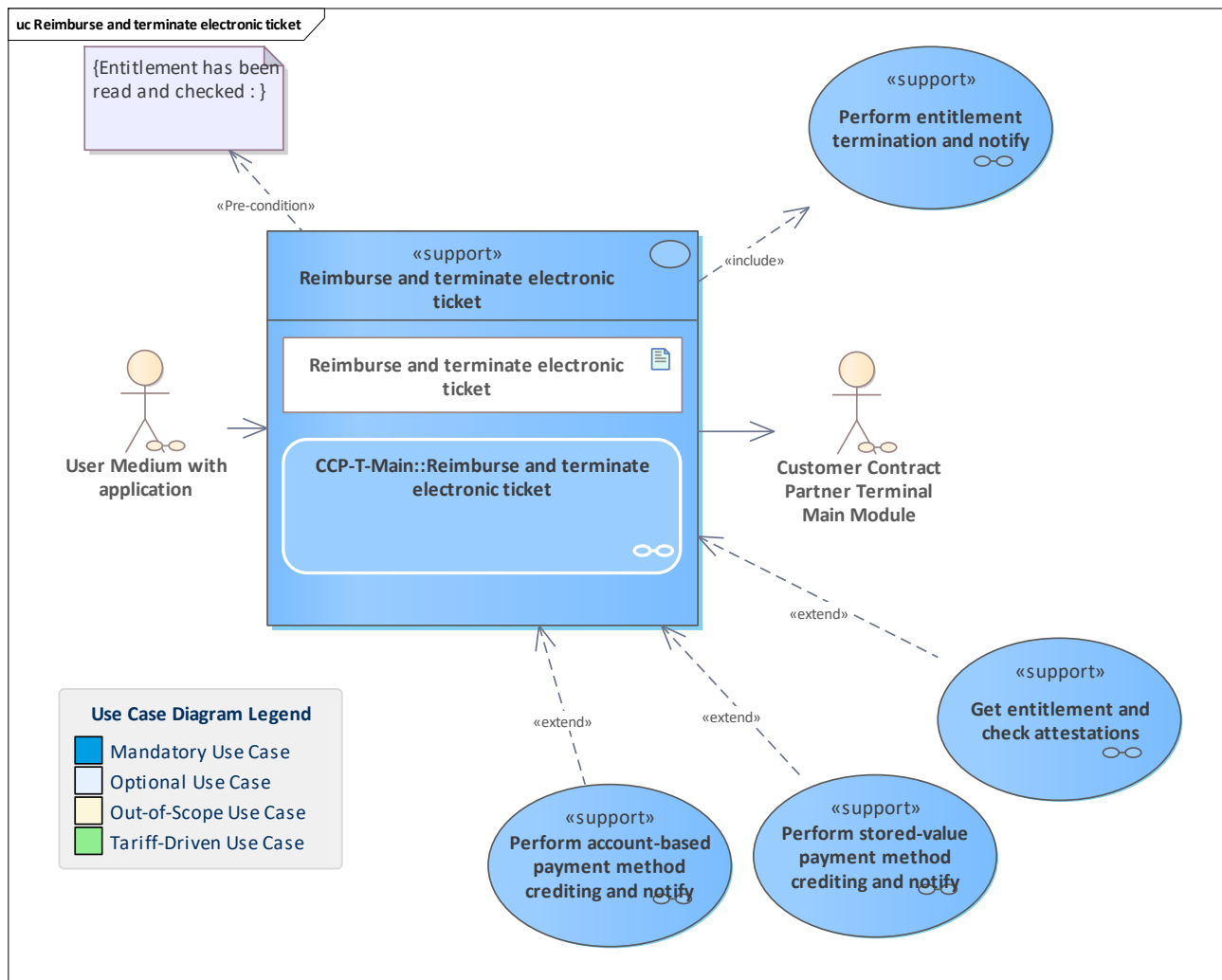


Figure 458: Reimburse and terminate electronic ticket

Reimburse and terminate an electronic ticket.

The primary use case is [Take back entitlement](#).

Performed by the CCP terminal. The technical part is the termination of the electronic ticket has to be done in the terminal.

Depending on the payment method that was used when purchasing the ticket, the reimbursement is done. If an (((etiCORE payment method was used, the refund is processed using this payment method. In this case, in addition to the termination, a credit action is done by the terminal.

The termination is done and notified to the responsible CCP back-office system, as well as the potential credit action.

11.335 Reimburse and terminate static entitlement

11.336 Reimburse and terminate static entitlement

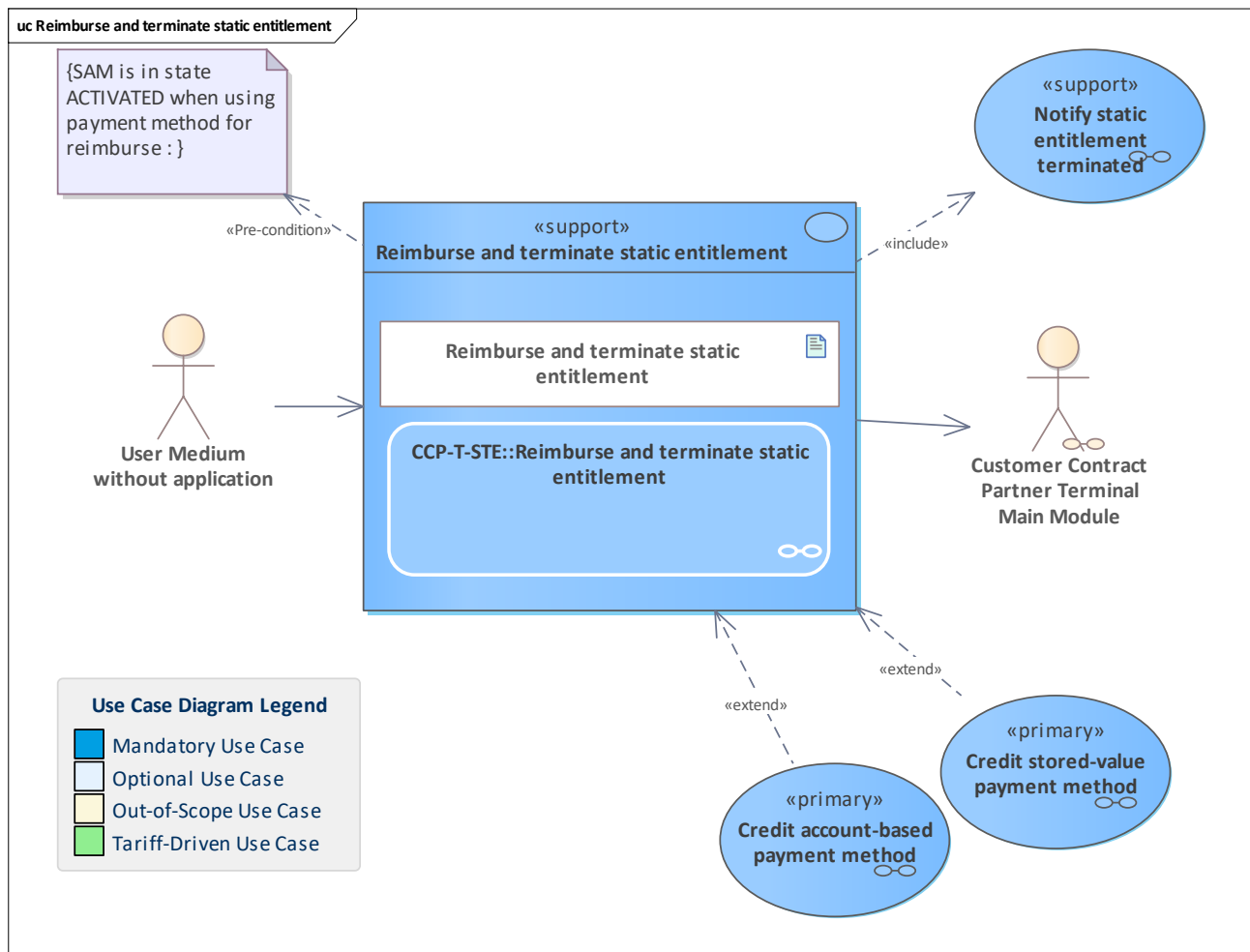


Figure 459: Reimburse and terminate static entitlement

A static entitlement is given back/terminated, and it was decided that the static entitlement can be reimbursed.

There are 3 possibilities for reimbursement:

- Credit a stored-value payment method on a user medium with an application if supported
- Credit an account-based payment method of the customer if supported
- Legal tender (must be supported)

11.337 Reimburse and terminate stored-value payment method

11.338 Reimburse and terminate stored-value payment method

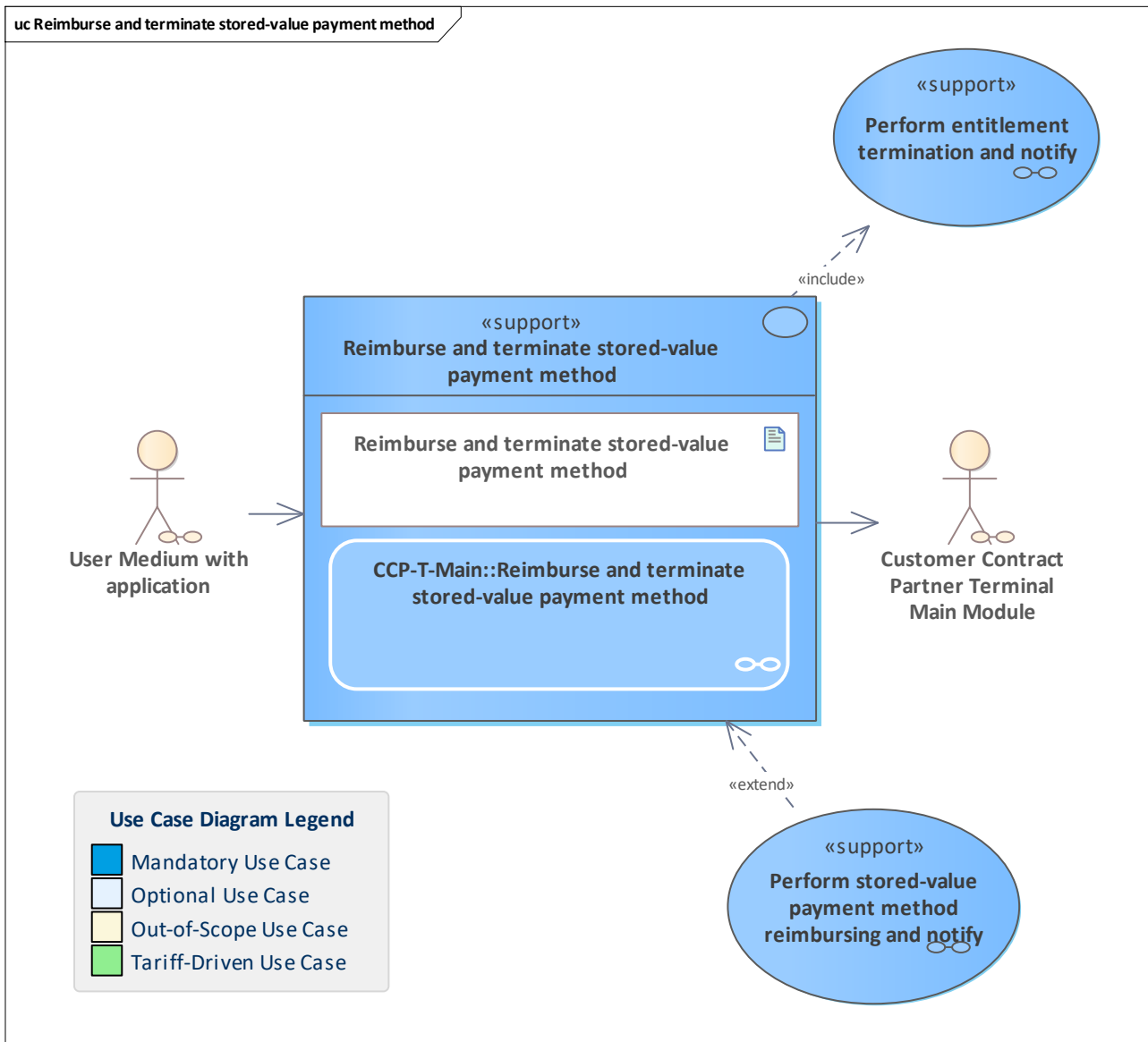


Figure 460: Reimburse and terminate stored-value payment method

Reimburse and terminate a stored-value payment method as a supporting use case. The primary use case is [Take back entitlement](#). Performed by the CCP terminal.

This process is divided into two parts:

- the potential reimbursement (if the current balance of the stored-value payment method is > 0), including a transaction and the triggering of the notification chain
- the termination transaction of the stored-value payment method, including the triggering of the notification chain

11.339 Reimburse stored-value payment method

11.340 Reimburse stored-value payment method

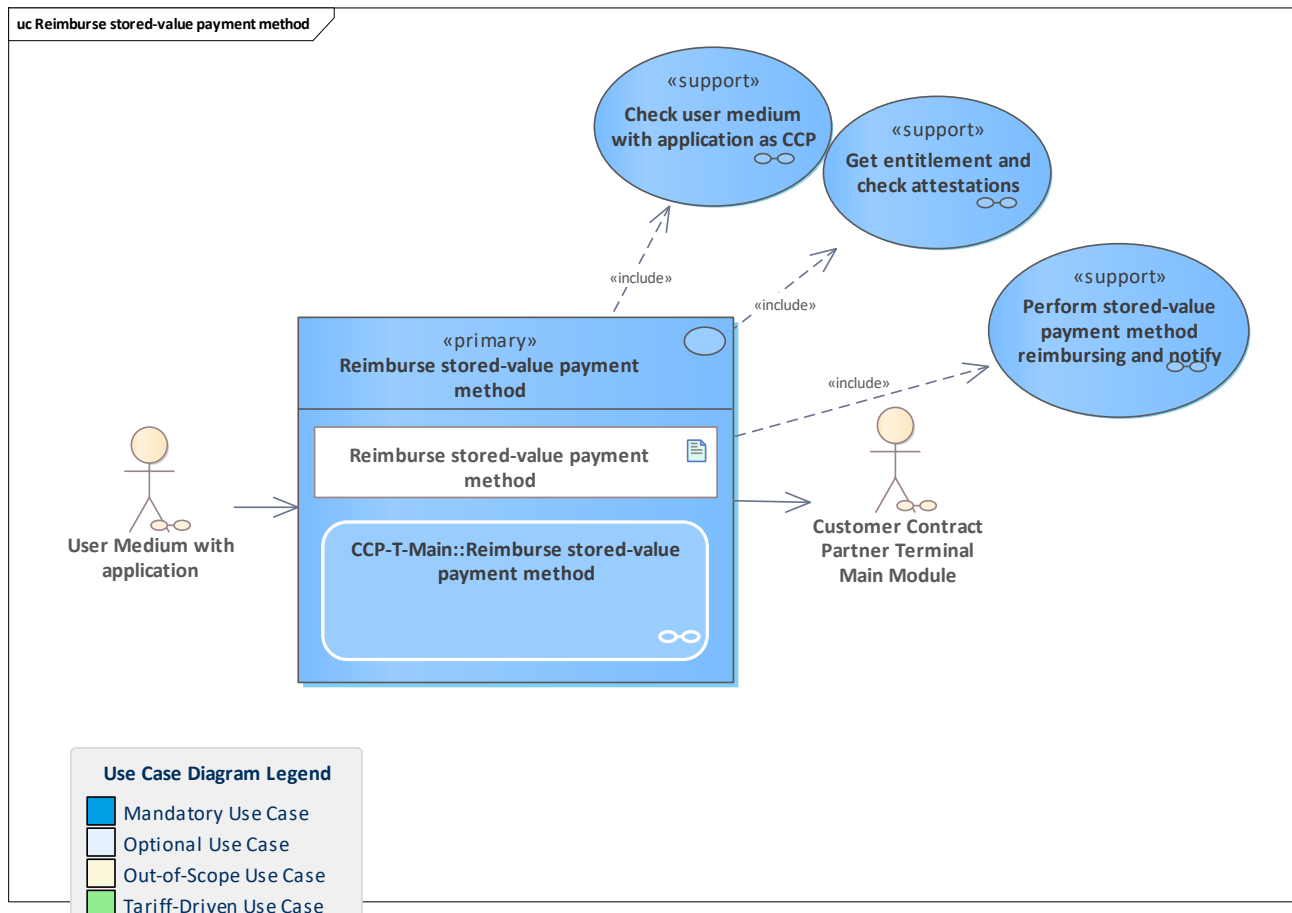


Figure 461: Reimburse stored-value payment method

Reimburse stored-values from a stored-value payment method.

Performed by the CCP terminal.

This process performs the reimbursement (if the current balance of the stored-value payment method is > 0), including a transaction and the triggering of the notification chain.

Note: the stored-value payment method is not terminated.

11.341 Reissue entitlements

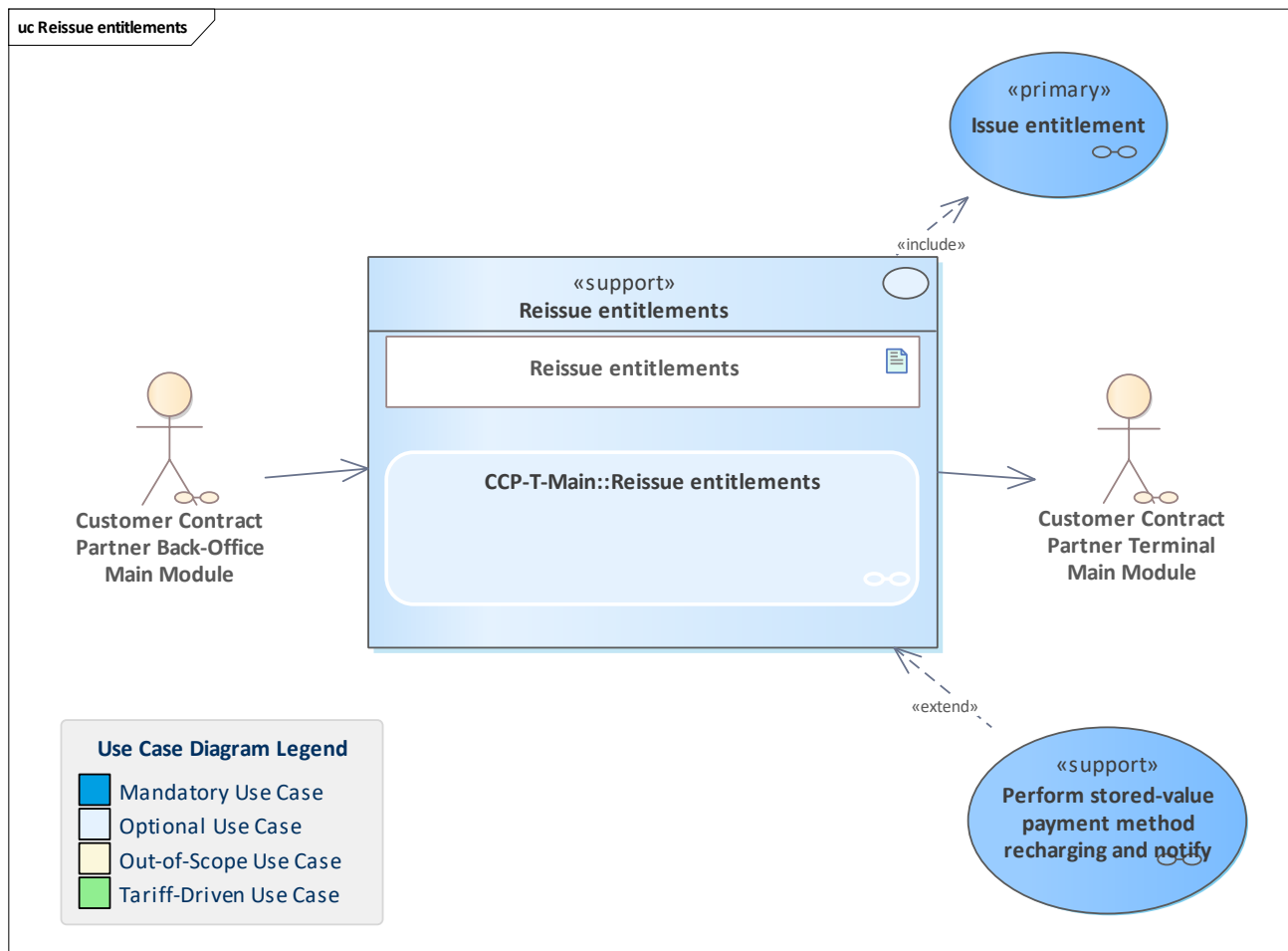


Figure 462: Reissue entitlements

Reissue entitlements based on existing entitlements. The reason for that might be lost or defective user medium.

11.342 Remove application from hotlist

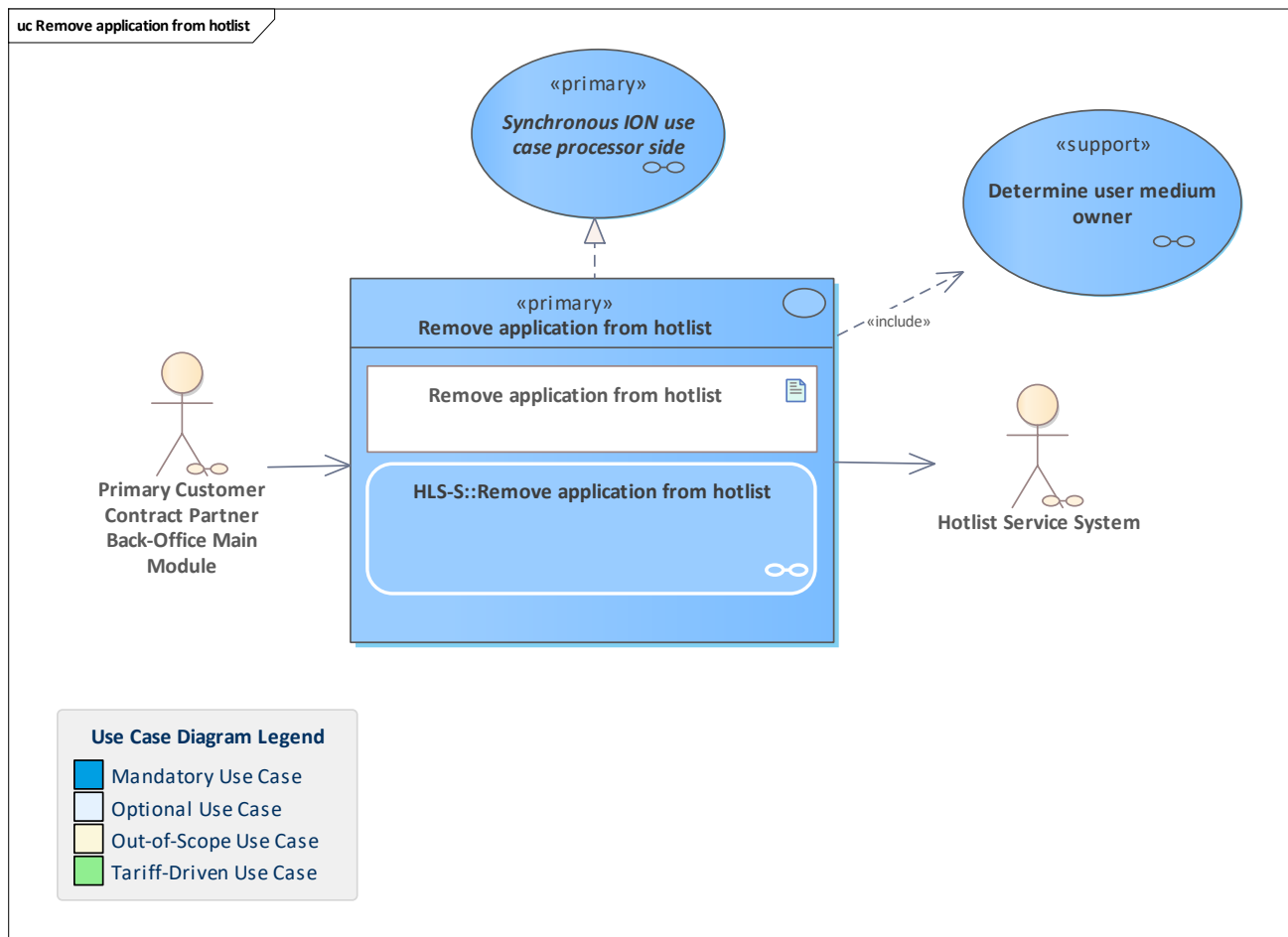


Figure 463: Remove application from hotlist

The pCCP sends a request to remove an application from the application hotlist. This can be caused either by an incoming application blocked notification or by other reasons (e.g. the customer found his "lost" user medium in his pocket). The hotlist service checks the request. The application owner is determined to ensure that the sender was authorised to remove the hotlist entry. Finally the hotlist service removes the application instance ID from the hotlist. The application hotlist in the next cycle will no longer contain the entry of this application instance. When the new hotlist is requested and distributed to the terminals, the application instance will no longer be considered during the hotlist check.

Note: in the ION context, this is a [Synchronous ION use case processor side](#) due to the synchronous call to the hotlist service system.

11.343 Remove authentication key from hotlist

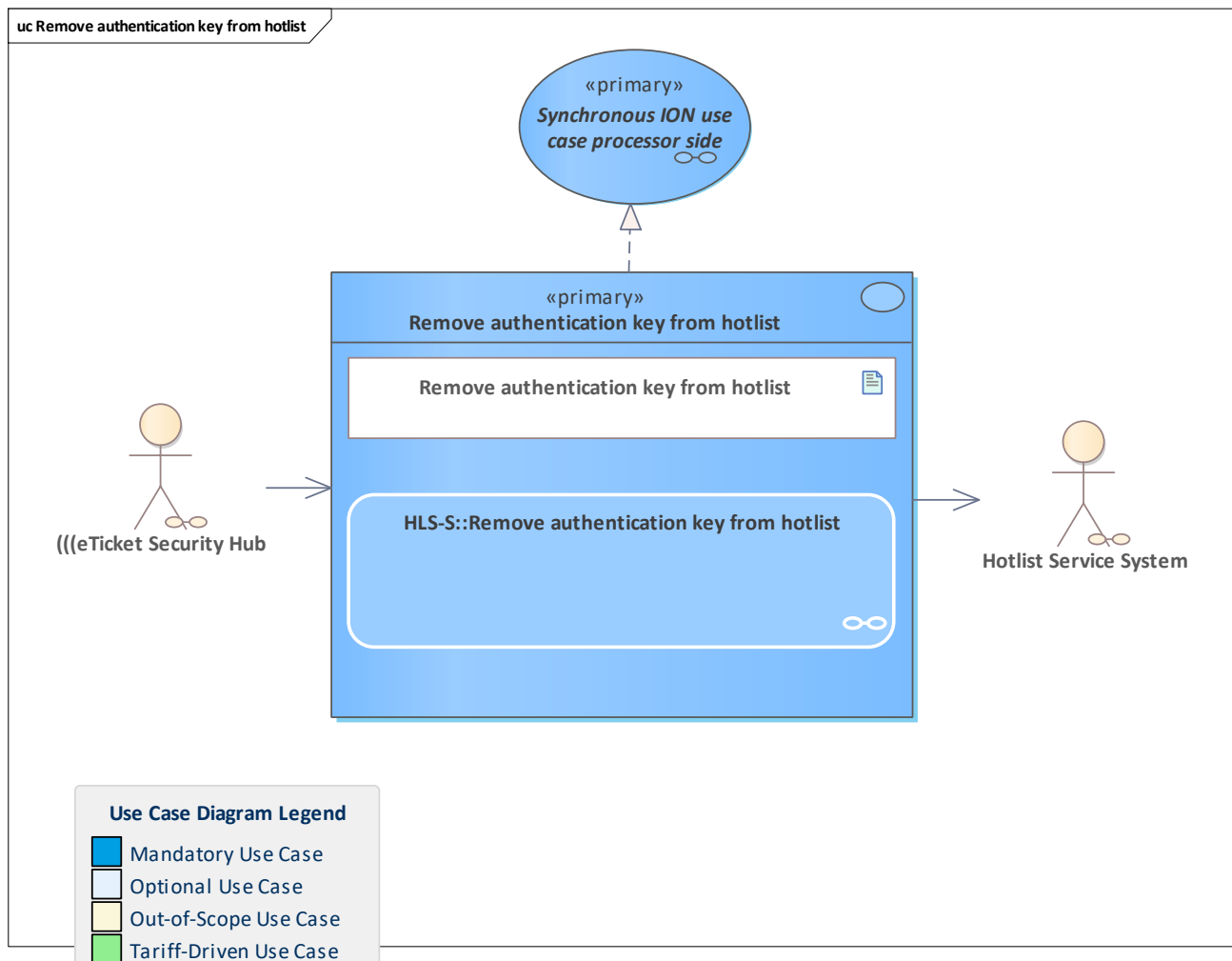


Figure 464: Remove authentication key from hotlist

Rare use case, only for the scheme manager's ESH. The authentication key is removed from the hotlist via request to the hotlist service.

Note: the authentication key hotlist in the next cycle will no longer contain the entry of this authentication key. When the new hotlist is requested and distributed to the terminals, the authentication key will no longer be considered during the hotlist check.

11.344 Remove entitlement from hotlist

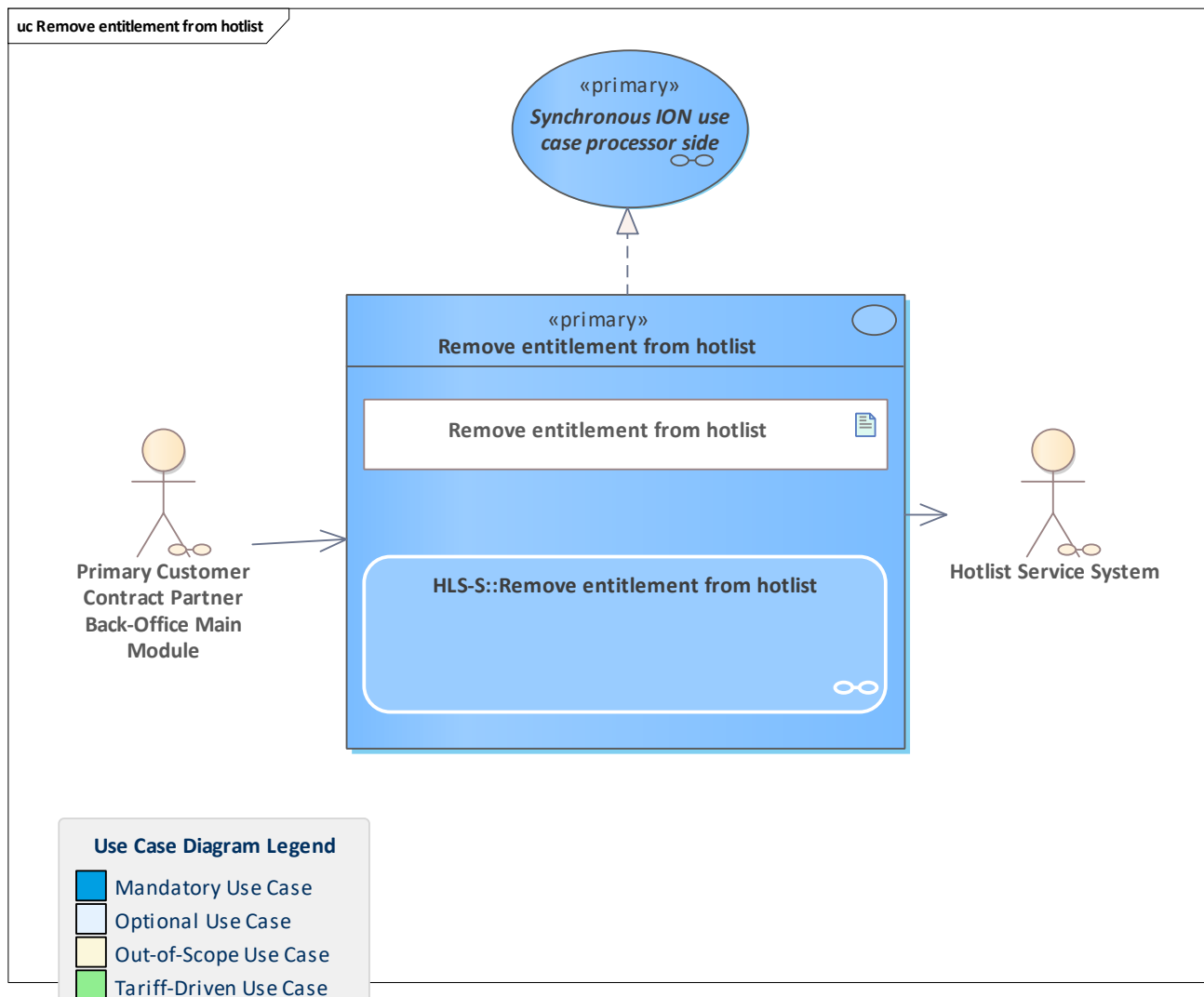


Figure 465: Remove entitlement from hotlist

The pCCP sends a request to remove an entitlement from the entitlement hotlist. This can be caused either by an incoming entitlement blocked notification or by other reasons (e.g. the customer contract allows suspension for a certain period of time. After this period, the temporary hotlist entry is removed).

The hotlist service checks the request. Finally, the hotlist service removes the entitlement from the hotlist.

The entitlement hotlist in the next cycle will no longer contain the entry of this entitlement. When the new hotlist is requested and distributed to the terminals, the entitlement ID will no longer be considered during the hotlist check.

Note: in the ION context, this is a [Synchronous ION use case processor side](#) due to the synchronous call to the hotlist service.

11.345 Remove organisation from hotlist

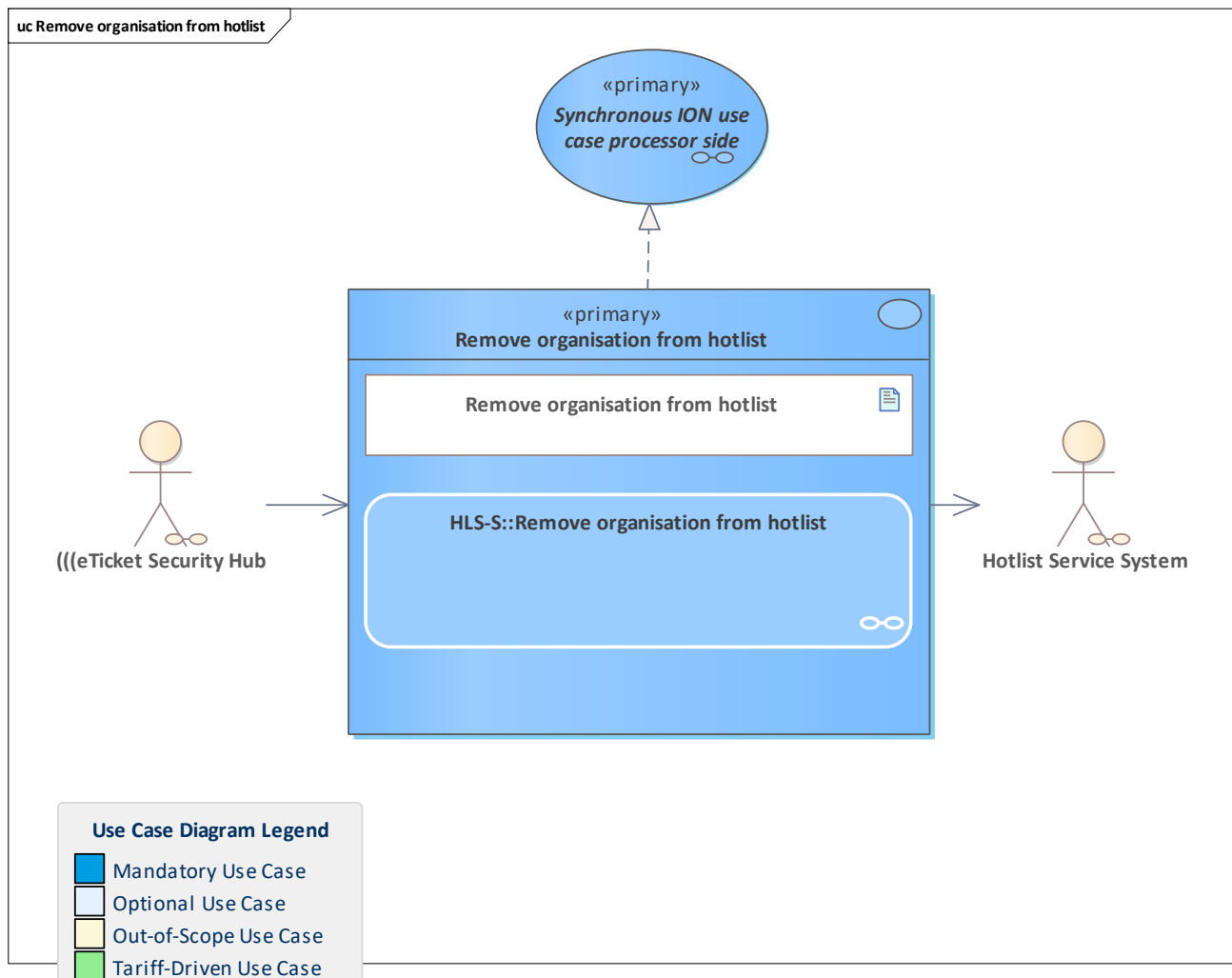


Figure 466: Remove organisation from hotlist

The requested organisation ID is removed from the organisation hotlist.
Rare use case only performed by the scheme manager's ESH as client.

Note: the organisation hotlist in the next cycle will no longer contain the entry of this organisation ID. When the new hotlist is requested and distributed to the terminals, the organisation ID will no longer be considered during the hotlist check.

11.346 Remove product acceptance entry from hotlist configuration

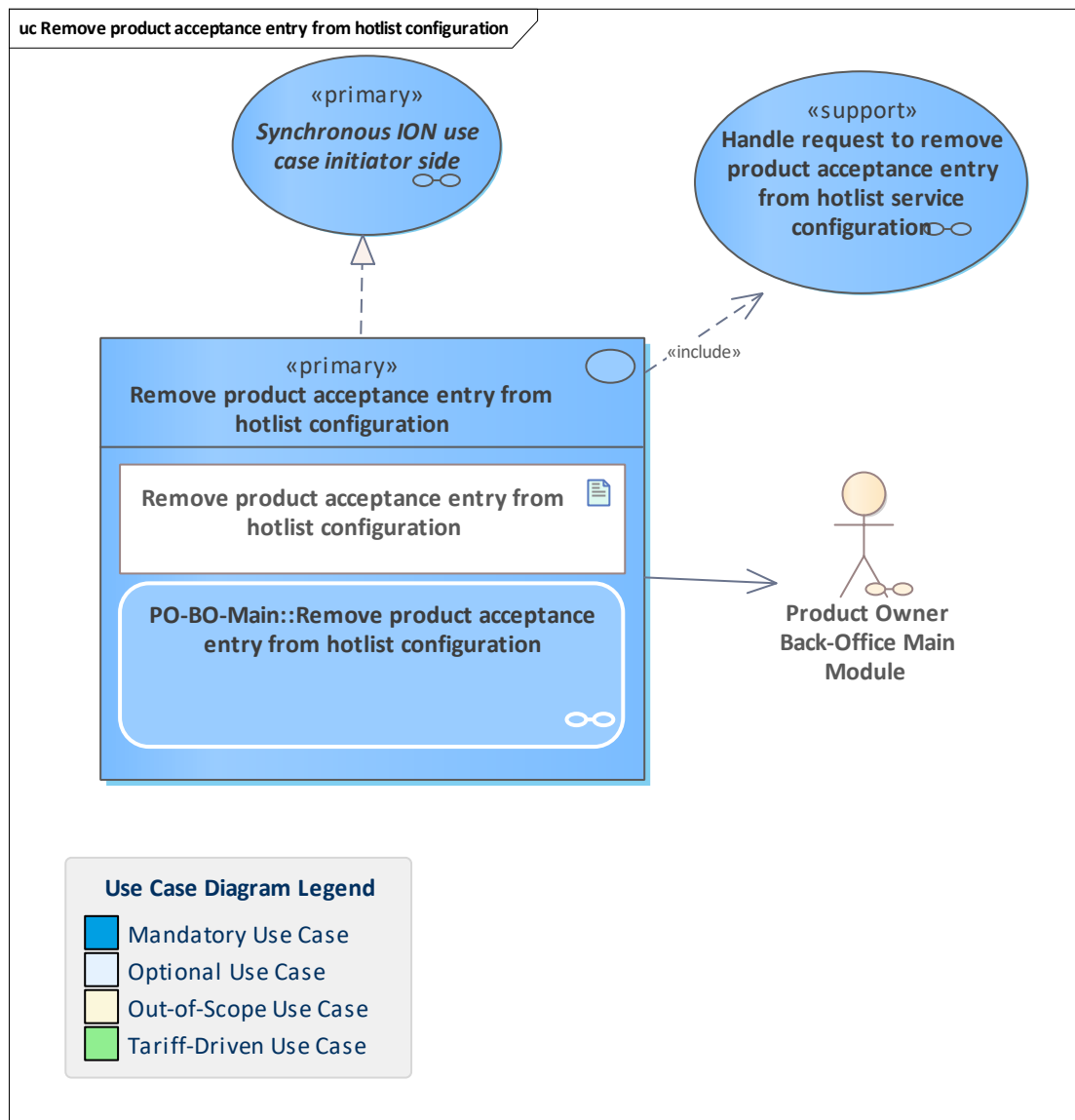


Figure 467: Remove product acceptance entry from hotlist configuration

To facilitate specific generation of hotlists, the hotlist service system uses information provided by the PO about which organisation accepts which products.

If a product is no longer accepted by an organisation, the PO sends an acceptance removal request containing the accepting organisation ID and a product number. If all products of the PO should be removed from the hotlist configuration for a certain organisation, the product number is omitted in the request.

To remove a product from all accepting organisations, for example, if the product no longer exists, there is a specific use case to [Remove product acceptance from participants](#).

Please note that, as interoperable products are not a part of the configuration list, a removal request for an interoperable product is not allowed.

11.347 Remove product acceptance from participants

11.348 Remove product acceptance from participants

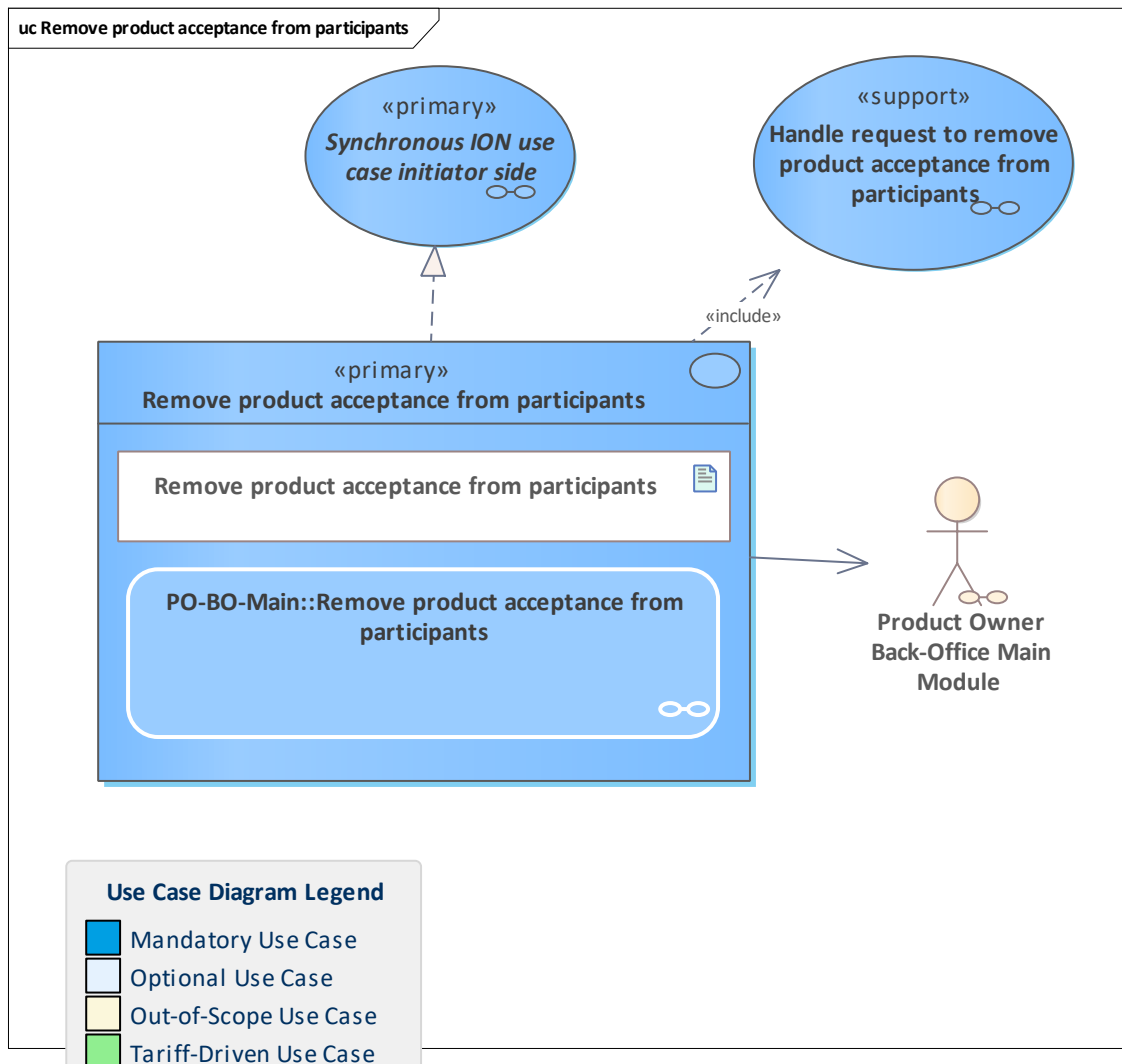


Figure 468: Remove product acceptance from participants

A product owner sends a request to remove product acceptance for one of its products from all organisations. The product number is mandatory in this request.

11.349 Remove SAM from hotlist

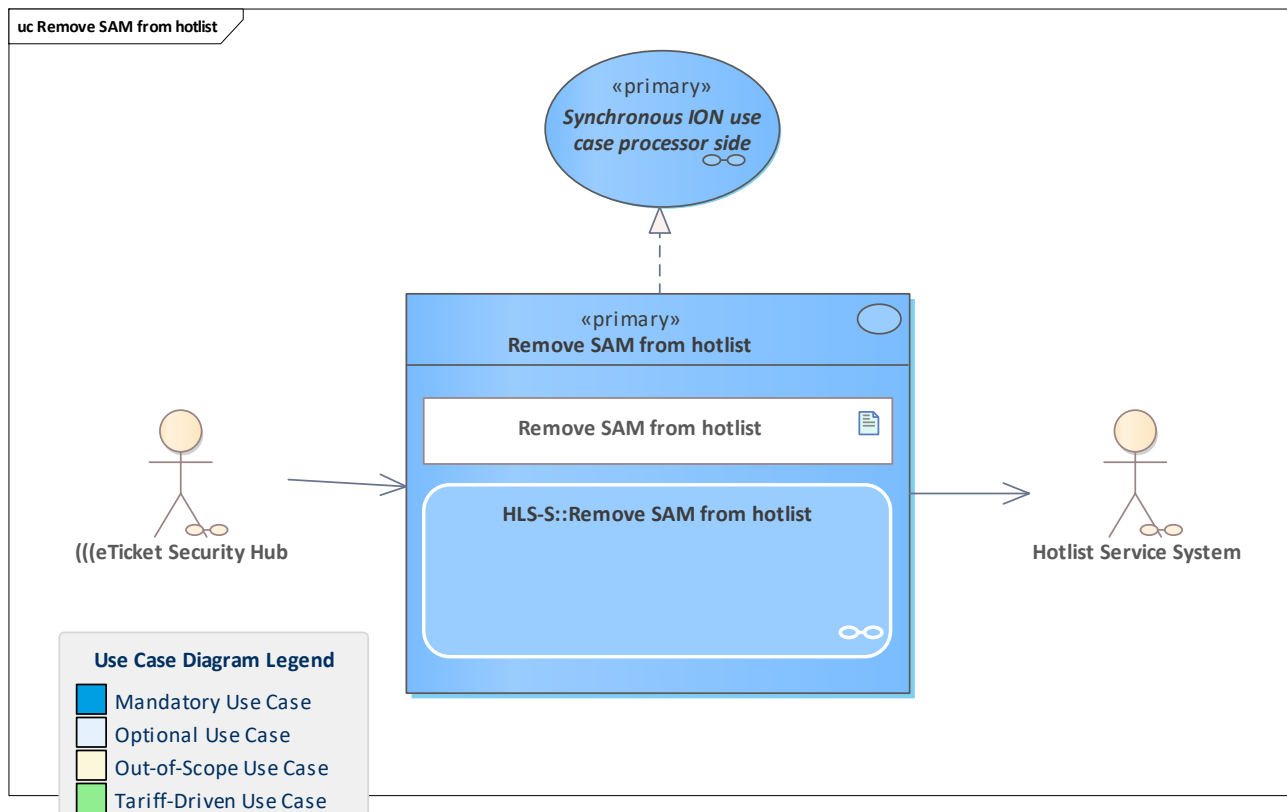


Figure 469: Remove SAM from hotlist

Rare use case for the hotlist service system to remove a SAM from the SAM hotlist. Only the scheme manager's ESH is allowed to remove a SAM from the hotlist.

Note: the SAM hotlist in the next cycle will no longer contain the entry of this SAM. When the new hotlist is requested and distributed to the terminals, the SAM will no longer be considered during the hotlist check.

11.350 Resolve notifications with timeout warnings

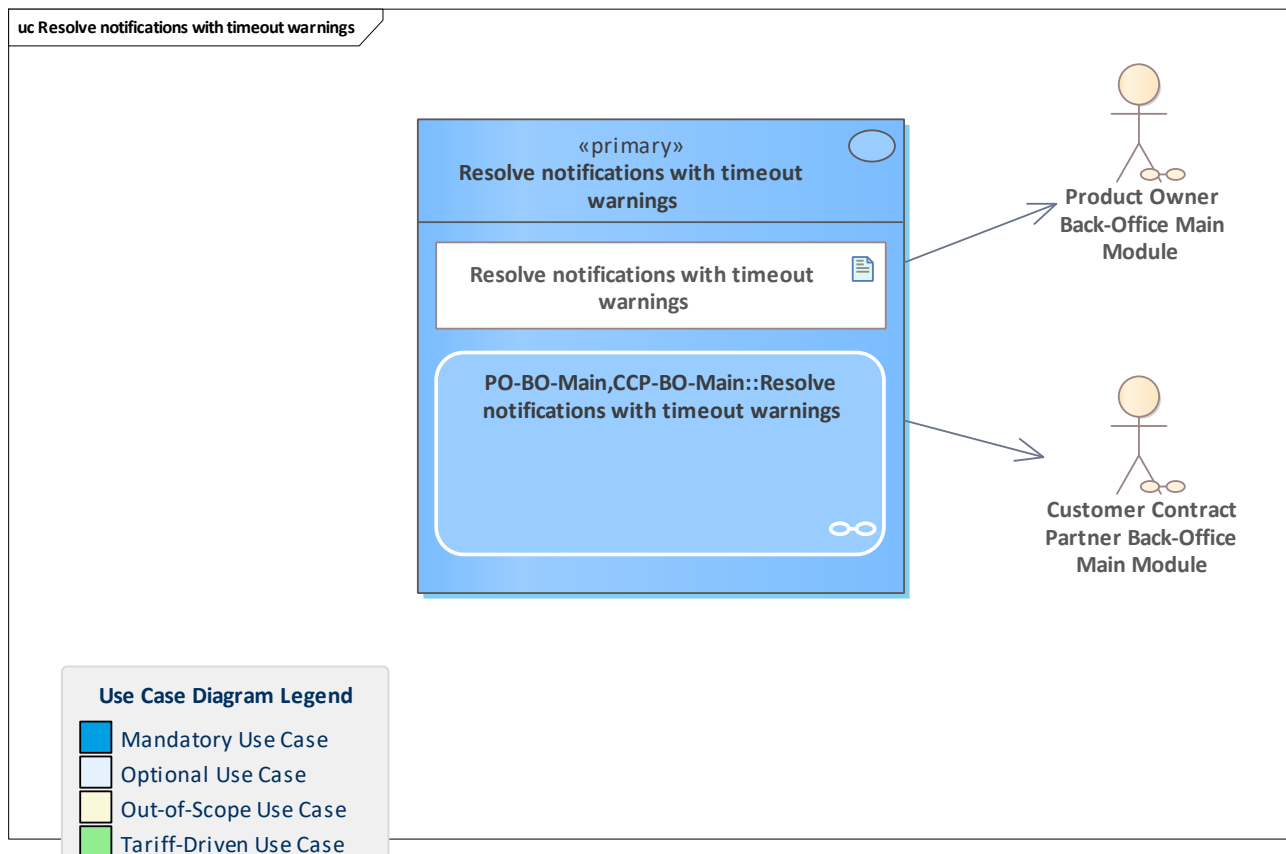


Figure 470: Resolve notifications with timeout warnings

The system periodically tries to resolve notifications with timeout warnings. A timeout warning can be resolved positively (determining that the action permanently changed the user medium) or negatively (determining the user medium performed a roll-back and was unchanged). When a notification has its timeout warning positively resolved, it can enter regular system processing.

11.351 Retrieve action list

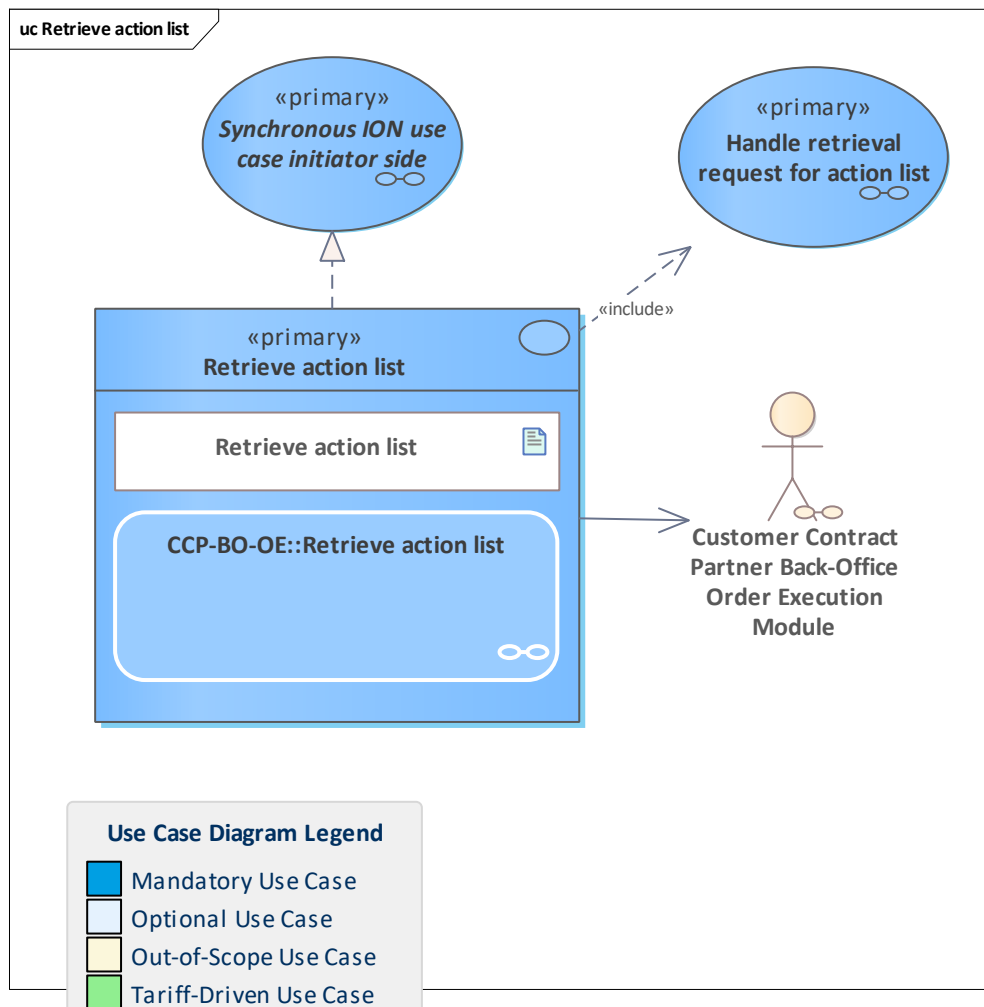


Figure 471: Retrieve action list

The [Customer Contract Partner Back-Office Order Execution Module](#) retrieves the latest action list from a [Product Owner Back-Office Action Management Module](#).

11.352 Retrieve and distribute organisation list

11.353 Retrieve and distribute organisation list

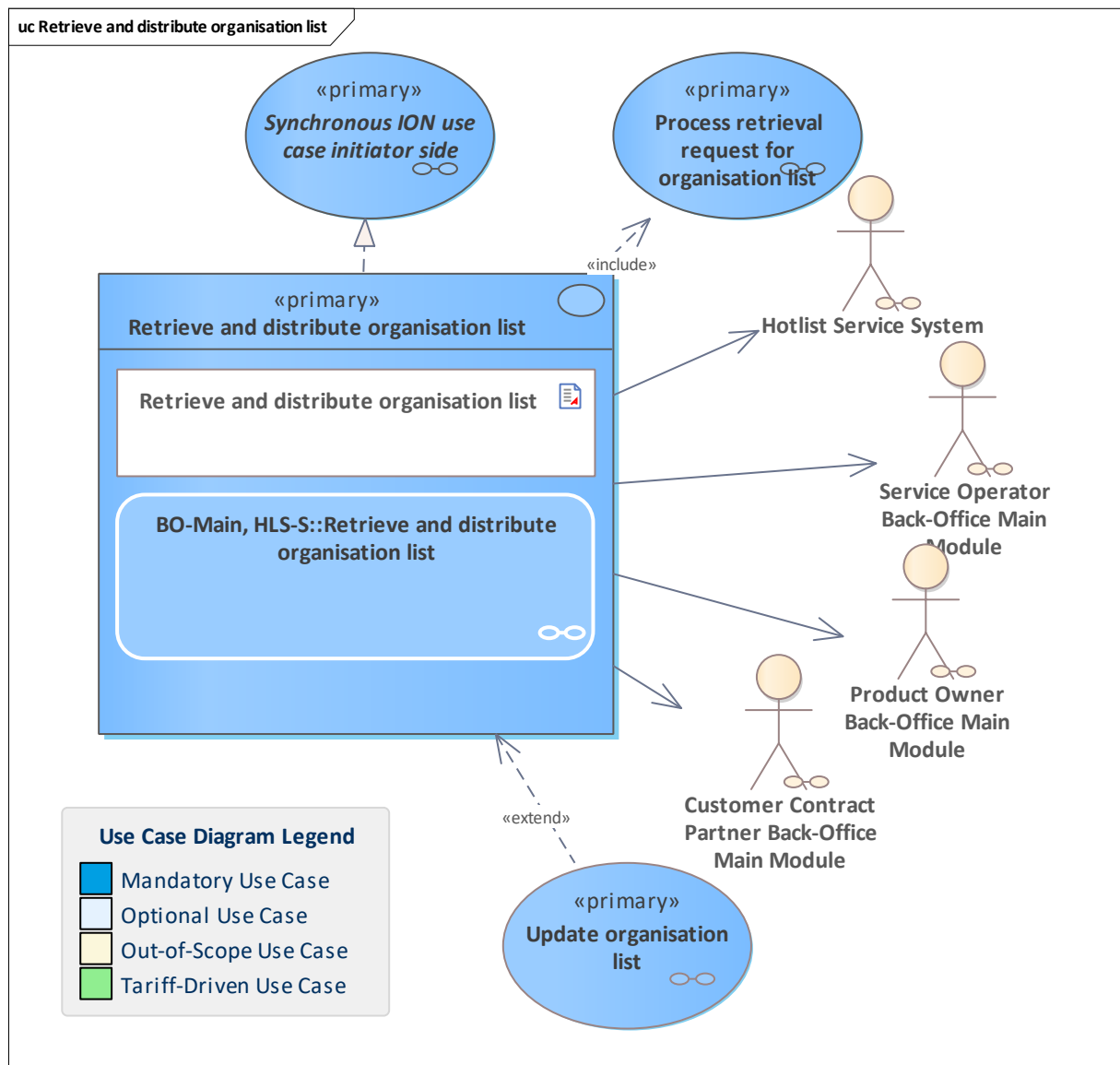


Figure 472: Retrieve and distribute organisation list

The registrar, as part of the scheme manager, provides a list of organisations. This list can be retrieved by all participants daily. An organisation list is distributed to the terminals by CCP and SO. Please note that the list does not involve any IP addresses of the organisations due to data protection.

11.354 Retrieve and distribute the CA certificate repository

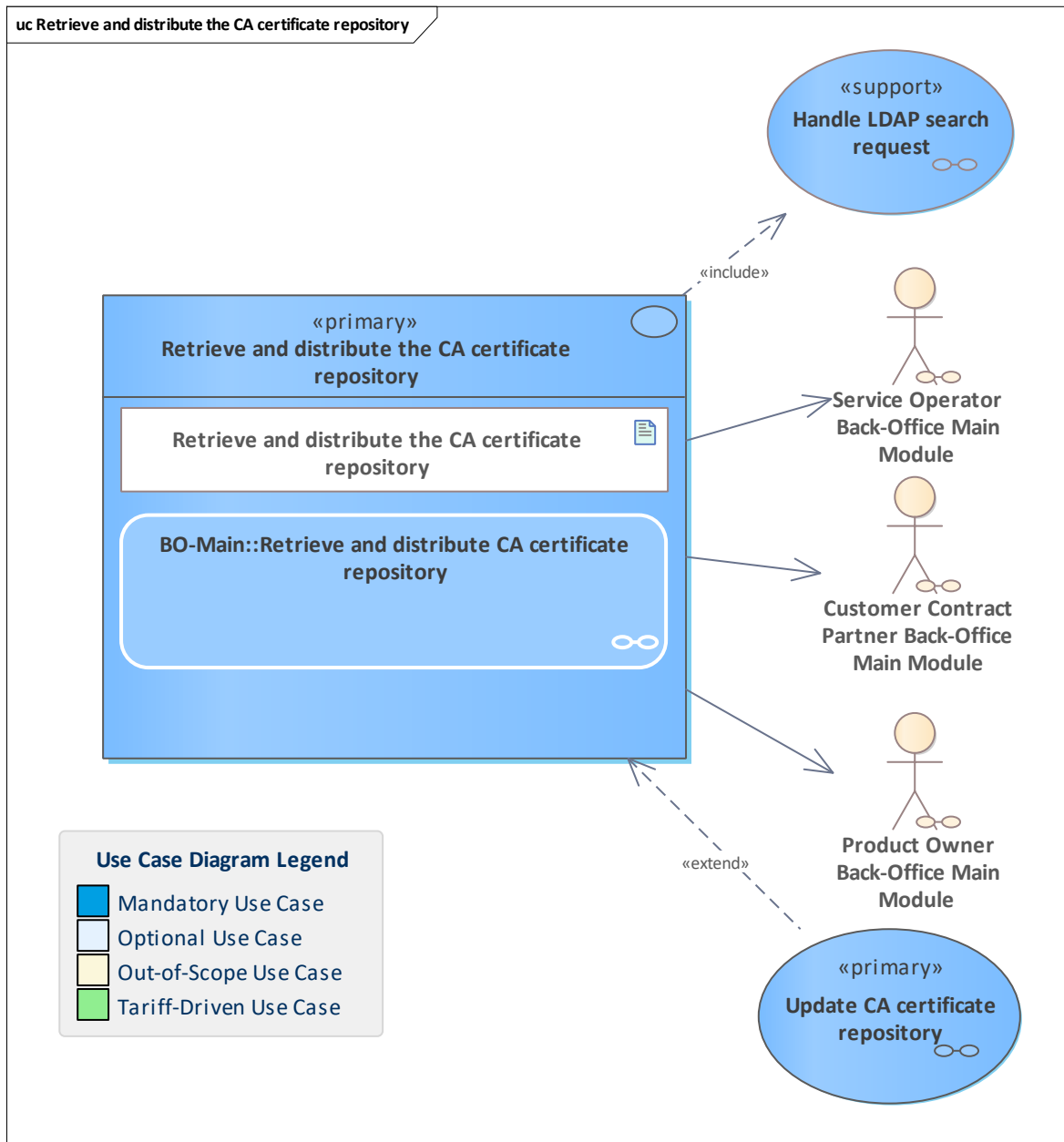


Figure 473: Retrieve and distribute the CA certificate repository

The back-office system retrieves the CA certificate repository from the media-PKI (M-PKI). If the requestor operates terminals that belong to back-office this system, the CA certificate repository is also updated in all of them.

This process needs to run periodically to keep the CA certificate repository up to date.

11.355 Retrieve and distribute the CV certificate revocation list

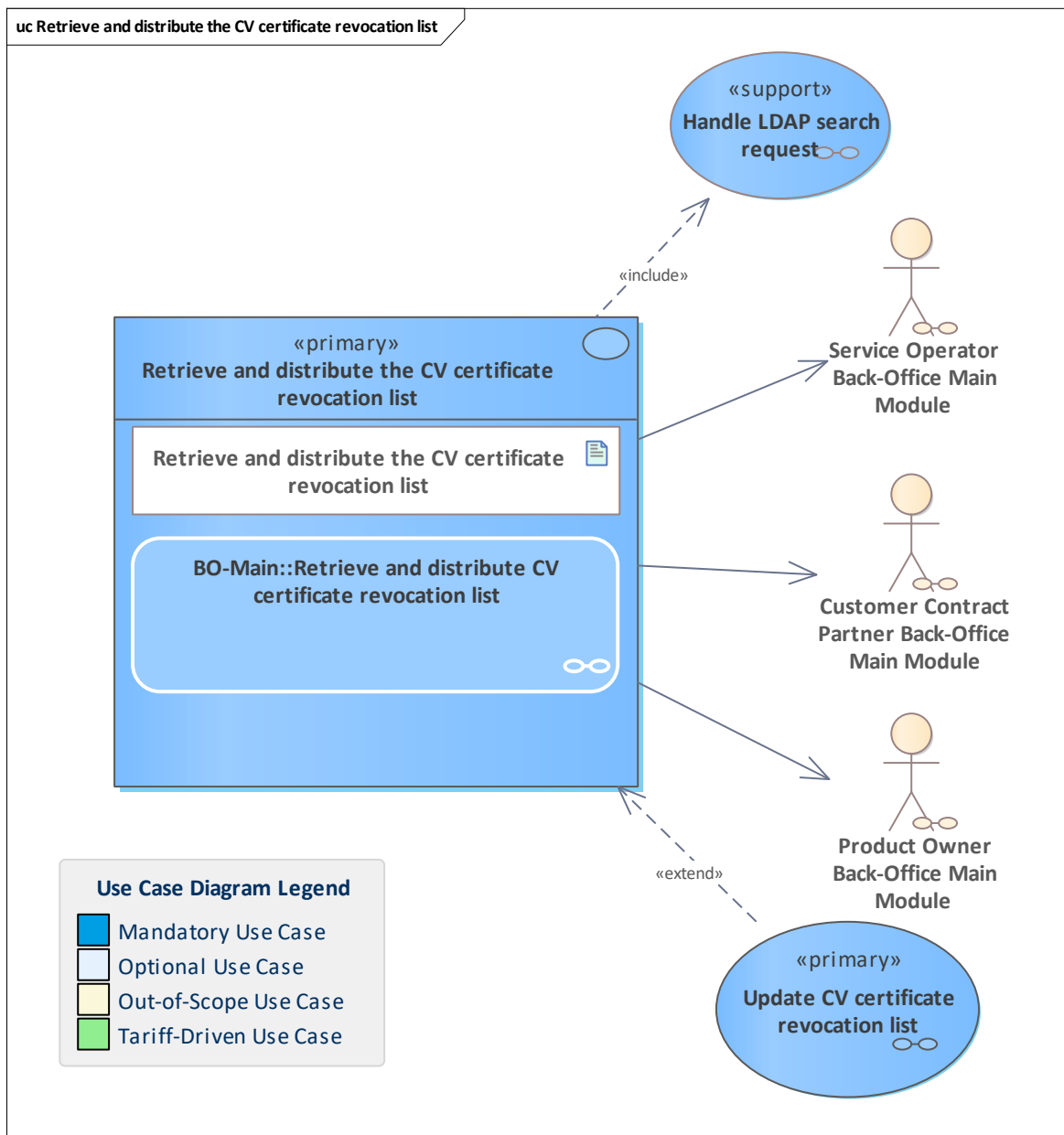


Figure 474: Retrieve and distribute the CV certificate revocation list

The back-office system retrieves the CV certificate revocation list from the media-PKI (M-PKI). If the requestor operates terminals that belong to back-office this system, the CV certificate revocation list is also updated in all of them. This process needs to run periodically to keep the CV certificate revocation list up to date.

11.356 Retrieve application hotlist

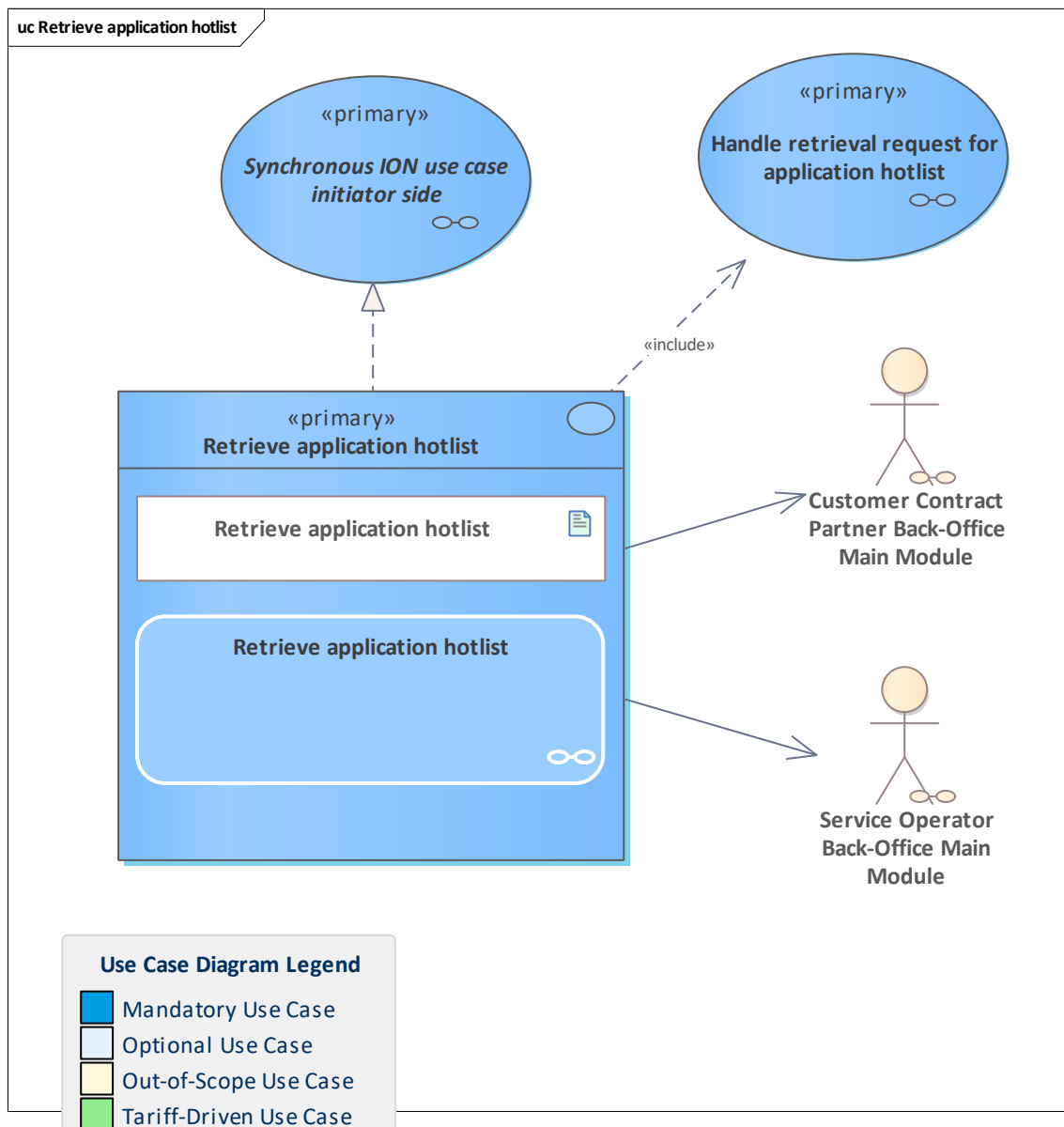


Figure 475: Retrieve application hotlist

The SO and CCP want to update their application hotlist inventory by retrieving the application hotlist from the hotlist service system.
Please note that the application hotlist can be retrieved as an incremental or a total hotlist.

11.357 Retrieve CV certificate over signing key

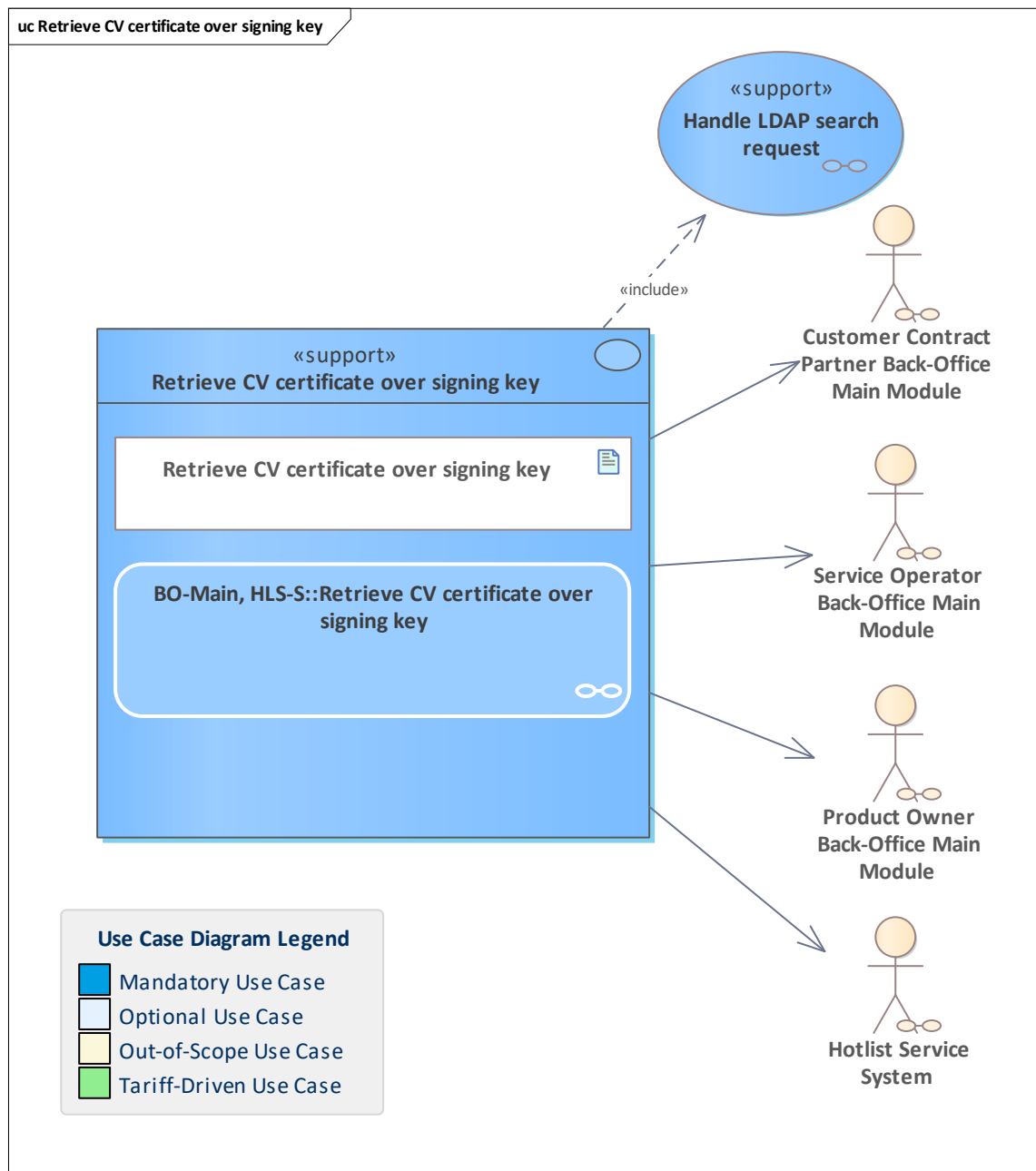


Figure 476: Retrieve CV certificate over signing key

Retrieve the latest certificate over the signing key of an end entity.
Note that there might be several certificates for this end entity that are not relevant here: superseded certificates and certificates over keys not used for signature purposes.

11.358 Retrieve entitlement hotlist

11.359 Retrieve entitlement hotlist

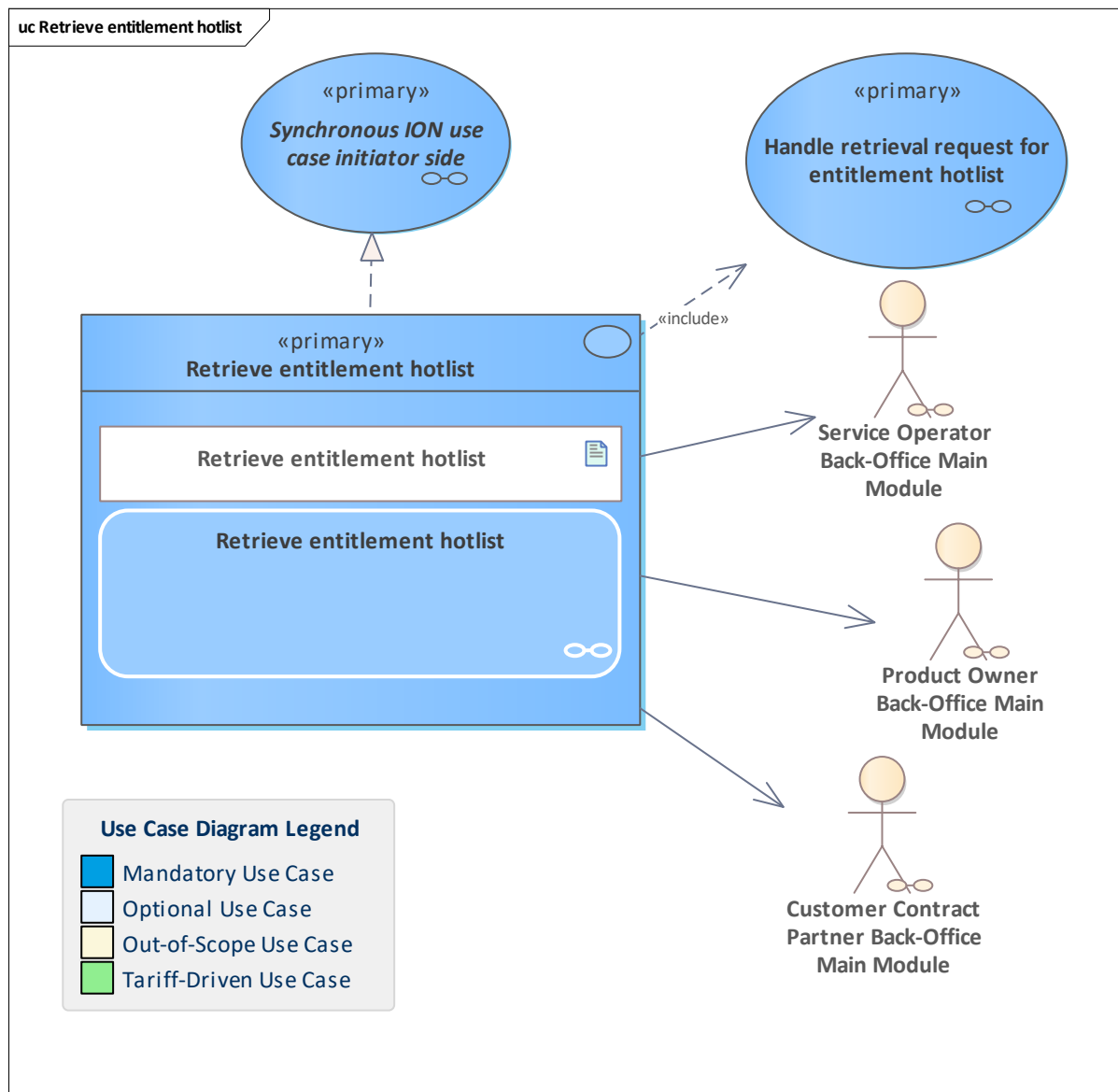


Figure 477: Retrieve entitlement hotlist

The SO, PO and CCP want to update their entitlement hotlist inventory by retrieving the current entitlement hotlist from the hotlist service system.

Please note that the entitlement hotlist can be retrieved either as an incremental or a total hotlist, as well as a total hotlist with product information.

11.360 Retrieve entitlement hotlist with product information

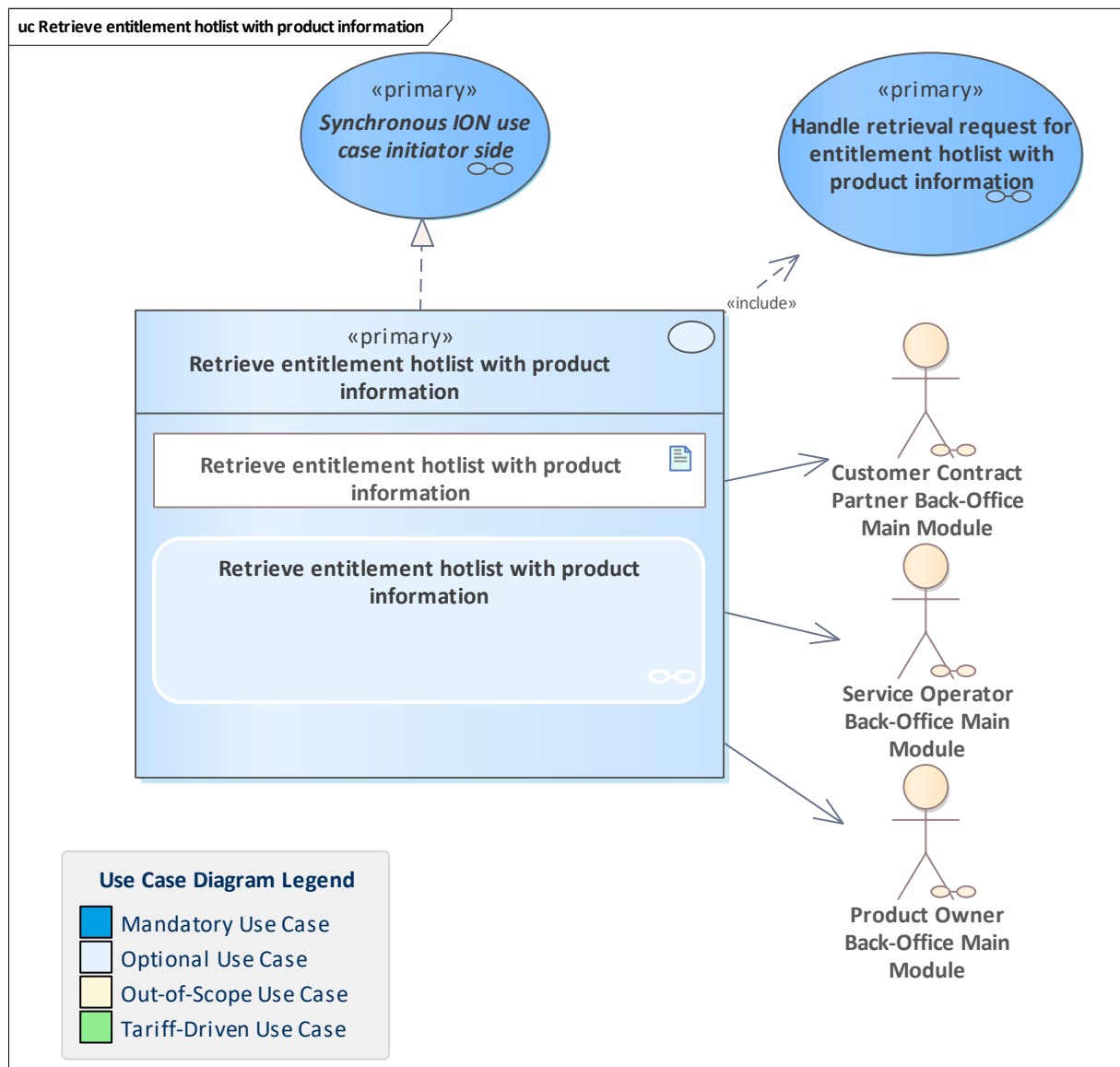


Figure 478: Retrieve entitlement hotlist with product information

The SO, PO and CCP want to update their entitlement hotlist inventory by retrieving the entitlement hotlist with additionally contained product information from the hotlist service system.

Please note that the entitlement hotlist can be retrieved either as an incremental or a total hotlist, as well as a hotlist with product information (this use case).

11.361 Retrieve incremental action list

11.362 Retrieve incremental action list

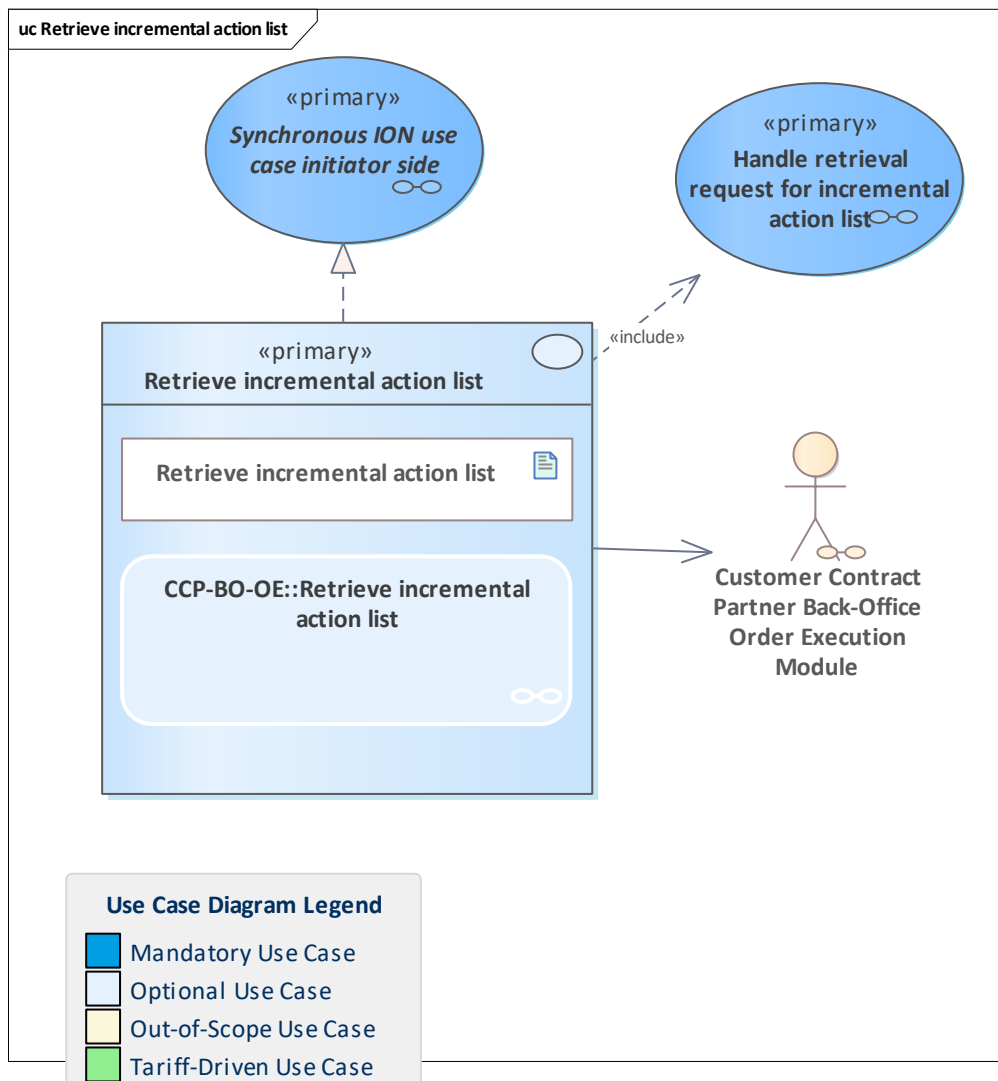


Figure 479: Retrieve incremental action list

The [Customer Contract Partner Back-Office Order Execution Module](#) retrieves the incremental action list including the increments since the cycle in its inventory from a [Product Owner Back-Office Action Management Module](#).

11.363 Retrieve incremental application hotlist

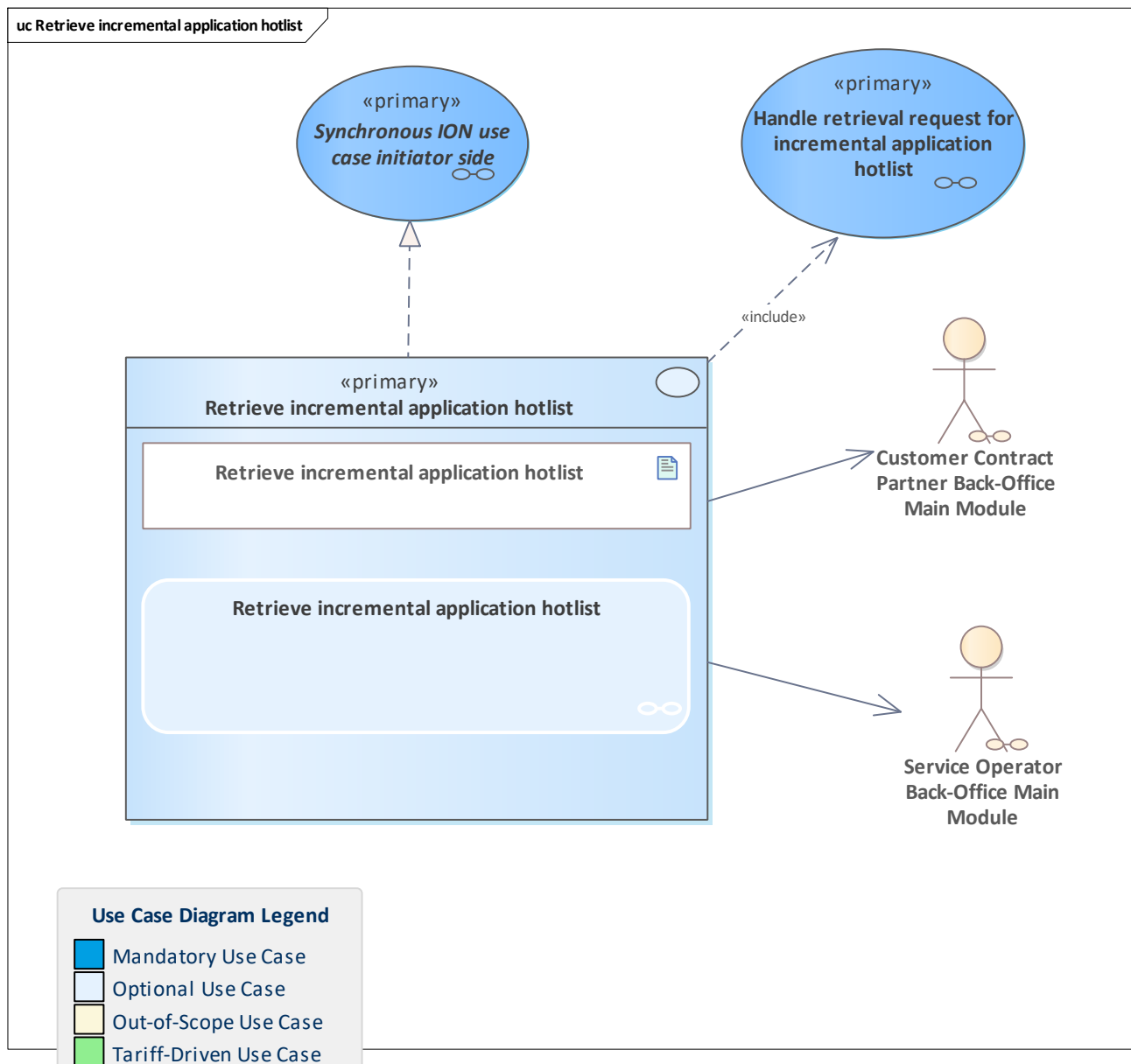


Figure 480: Retrieve incremental application hotlist

The SO and CCP want to update their application hotlist inventory by retrieving the current incremental application hotlist from the hotlist service system.

Please note that the application hotlist can be retrieved as an incremental or a total hotlist.

11.364 Retrieve incremental entitlement hotlist

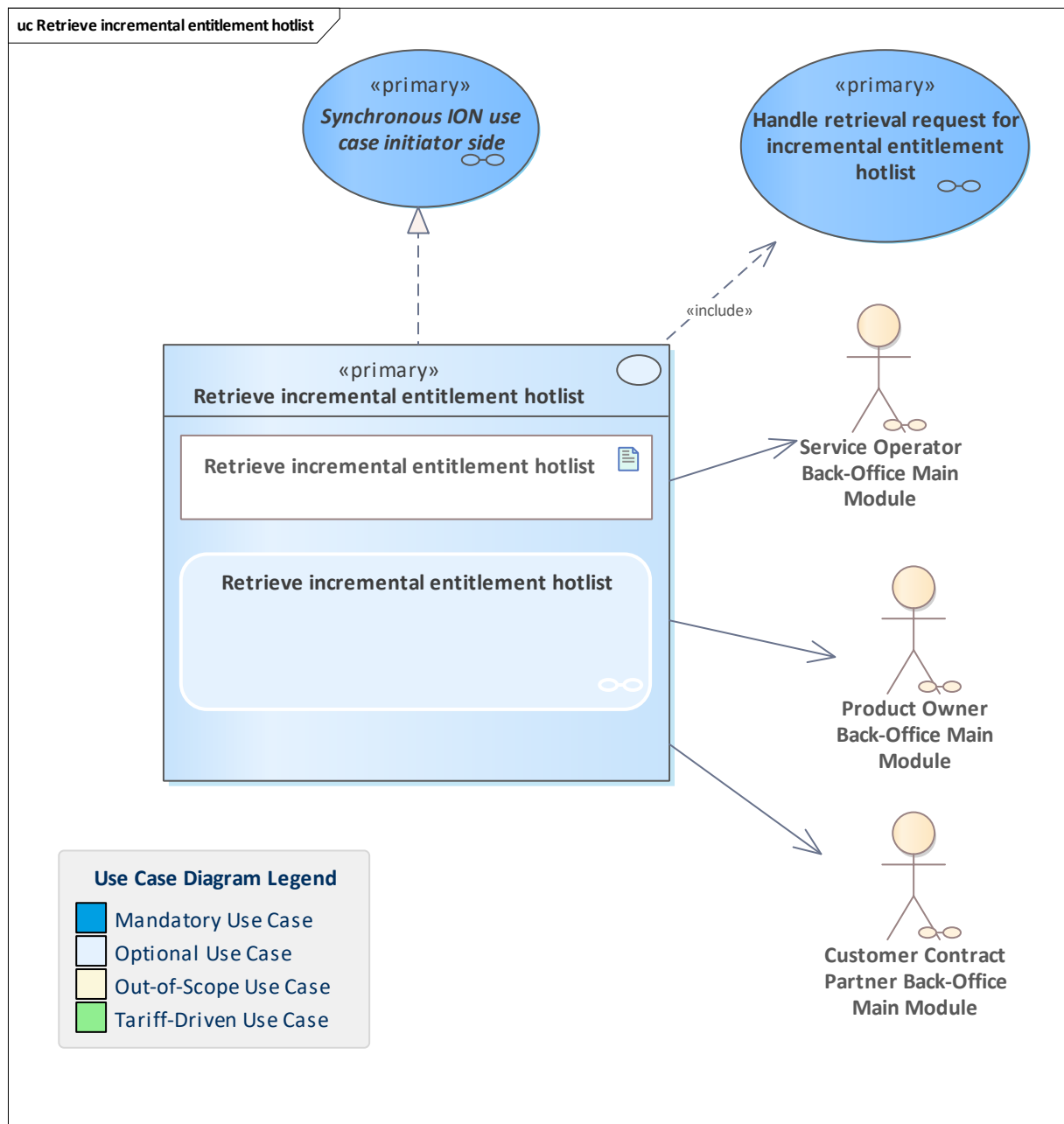


Figure 481: Retrieve incremental entitlement hotlist

The SO, PO and CCP want to update their entitlement hotlist inventory by retrieving the current incremental entitlement hotlist from the hotlist service system. Please note that the entitlement hotlist can be retrieved either as an incremental (this use case) or a total hotlist, as well as a hotlist with product information.

11.365 Retrieve valid entitlements for given app instance ID

11.366 Retrieve valid entitlements for given app instance ID

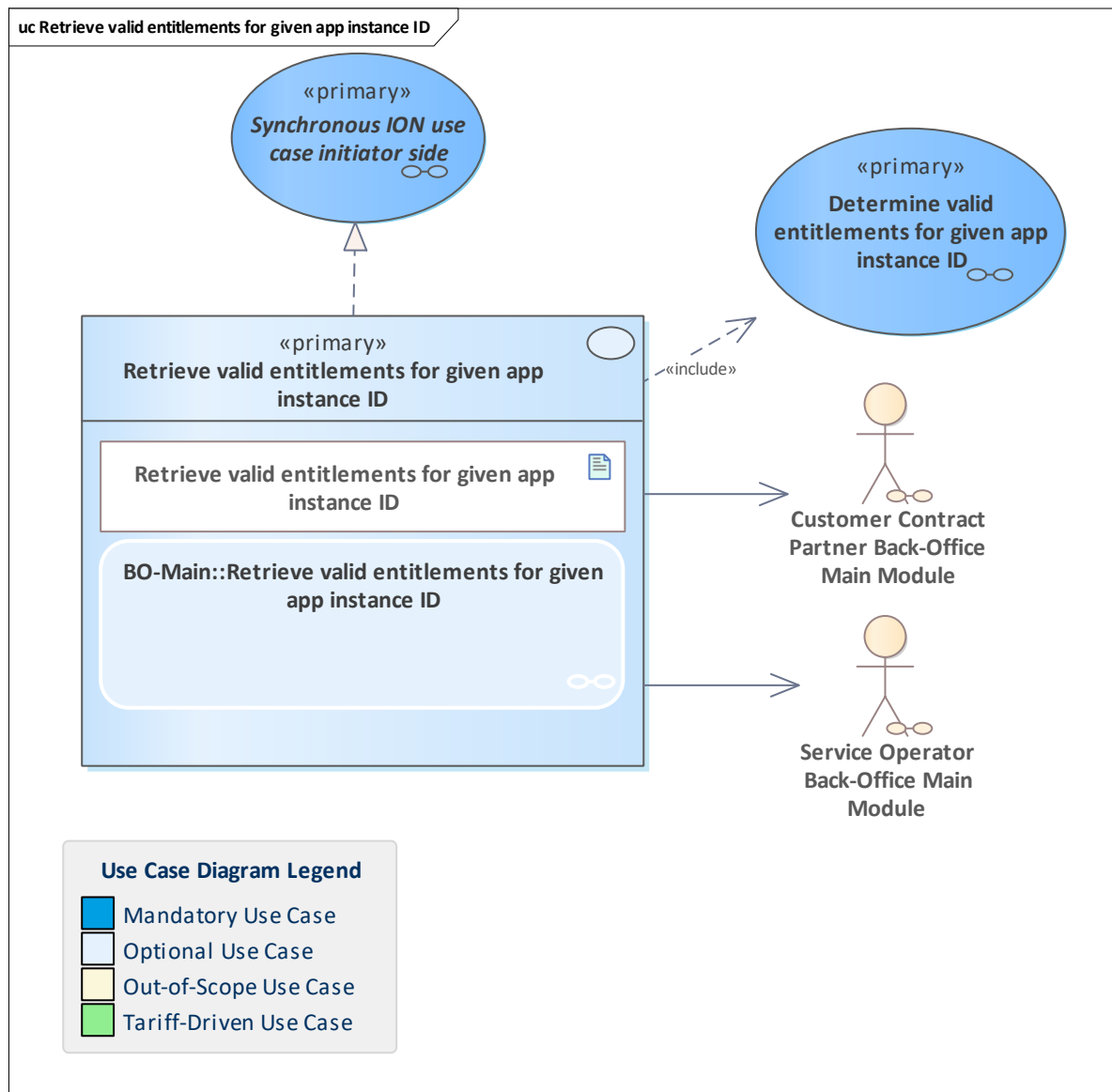


Figure 482: Retrieve valid entitlements for given app instance ID

The pCCP of the user medium with an application retrieves information about already issued and valid entitlements from each product owner that might come into question by sending the application instance ID.

Please note that this is a snapshot of the entitlements valid at a given timestamp.

Only the pCCP is informed about application hotlist entries or even an existing blocking of the application. If this use case is used as part of a user medium/entitlement replacement, the pCCP should be the only entity involved.

However, this use case can be used by a SO to access entitlement information inside a penalty fare notice process.

It should be noted that neither the requesting SO nor the PO has information about blocked applications. Likewise, the PO has no information about the hotlist of applications. The requesting SO must check the hotlist of applications.

11.367 Revoke application hotlisting demand

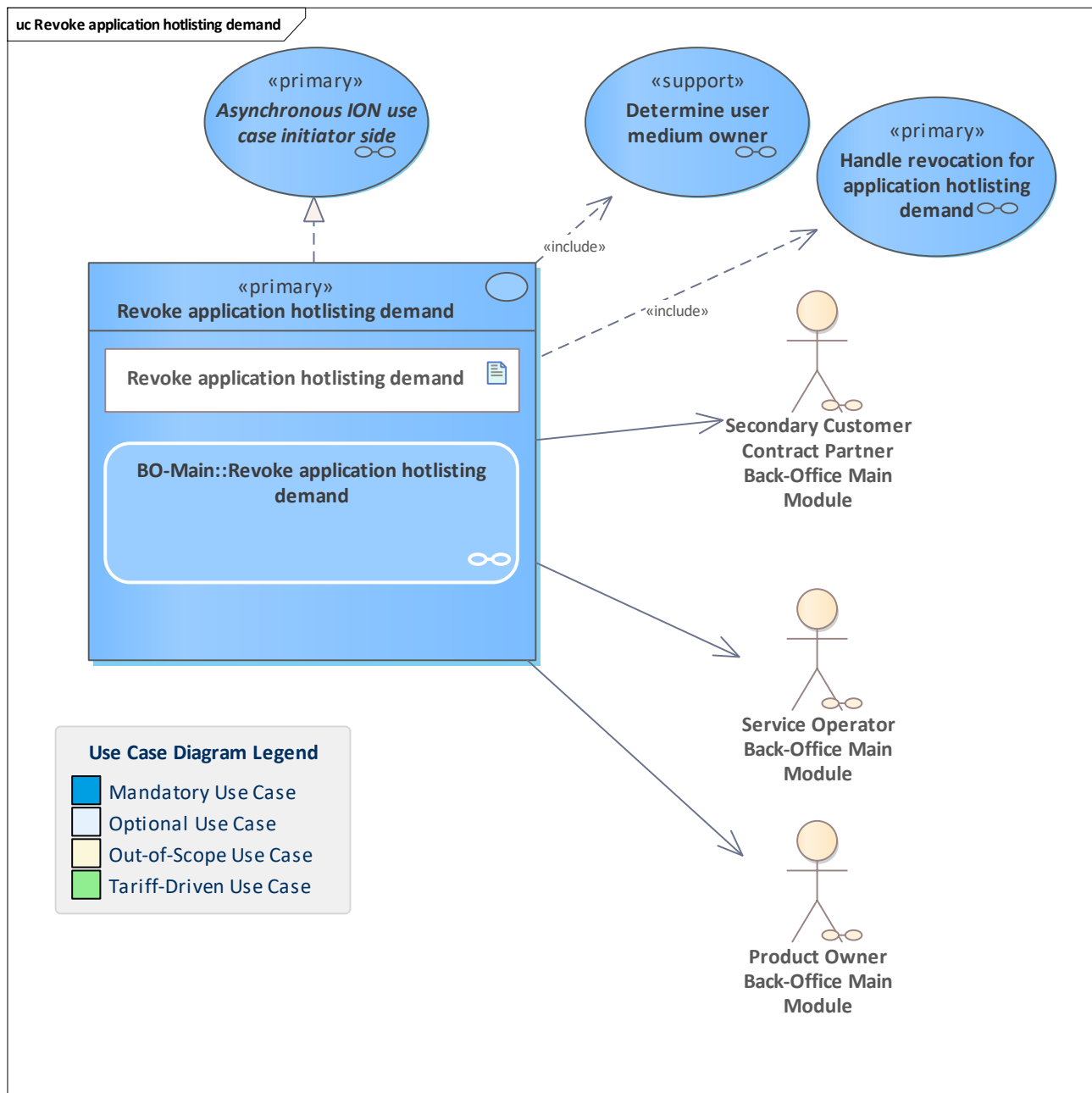


Figure 483: Revoke application hotlisting demand

An SO, PO or sCCP sends a demand for an application hotlisting revocation to the pCCP since the blocking reason no longer exists.

The pCCP will check the revocation demand and, in case of a positive decision, have the application removed from the hotlist.

The revocation references the ION message of the previous hotlisting demand.

This is a rarely used use case.

Note: if the application owner has changed, the new application will be determined by [Determine user medium owner](#).

The new application owner must have transferred the information about old hotlisting demands (if any), especially the ION message references.

11.368 Revoke entitlement hotlisting demand

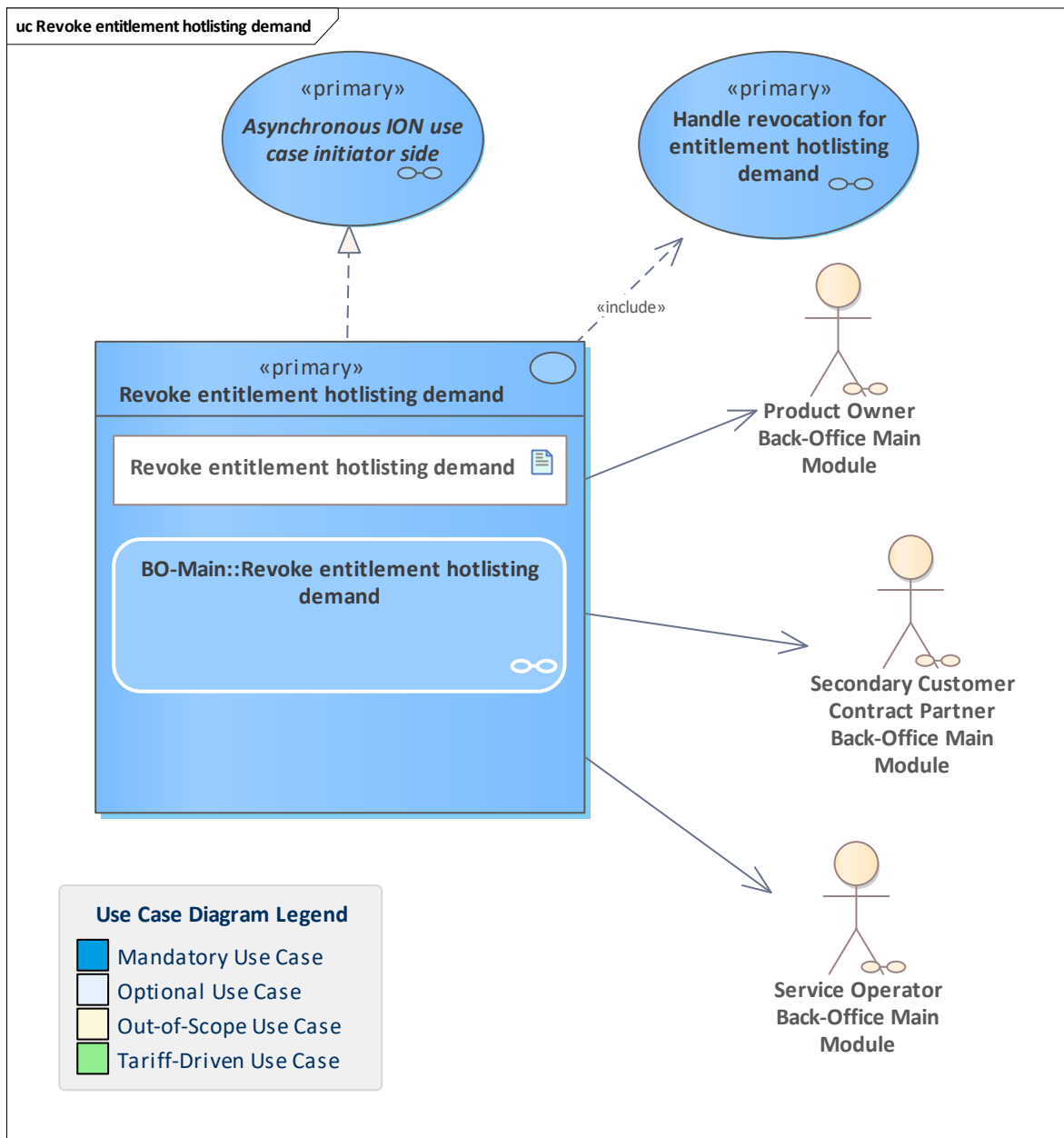


Figure 484: Revoke entitlement hotlisting demand

A PO, SO or sCCP sends a demand for hotlisting revocation to the entitlement owner (pCCP). The pCCP will check the revocation demand and, in case of a positive decision, have the entitlement removed from the hotlist. The revocation references the ION message of the previous hotlisting demand. This is a rarely used use case.

11.369 Save electronic ticket as favourite

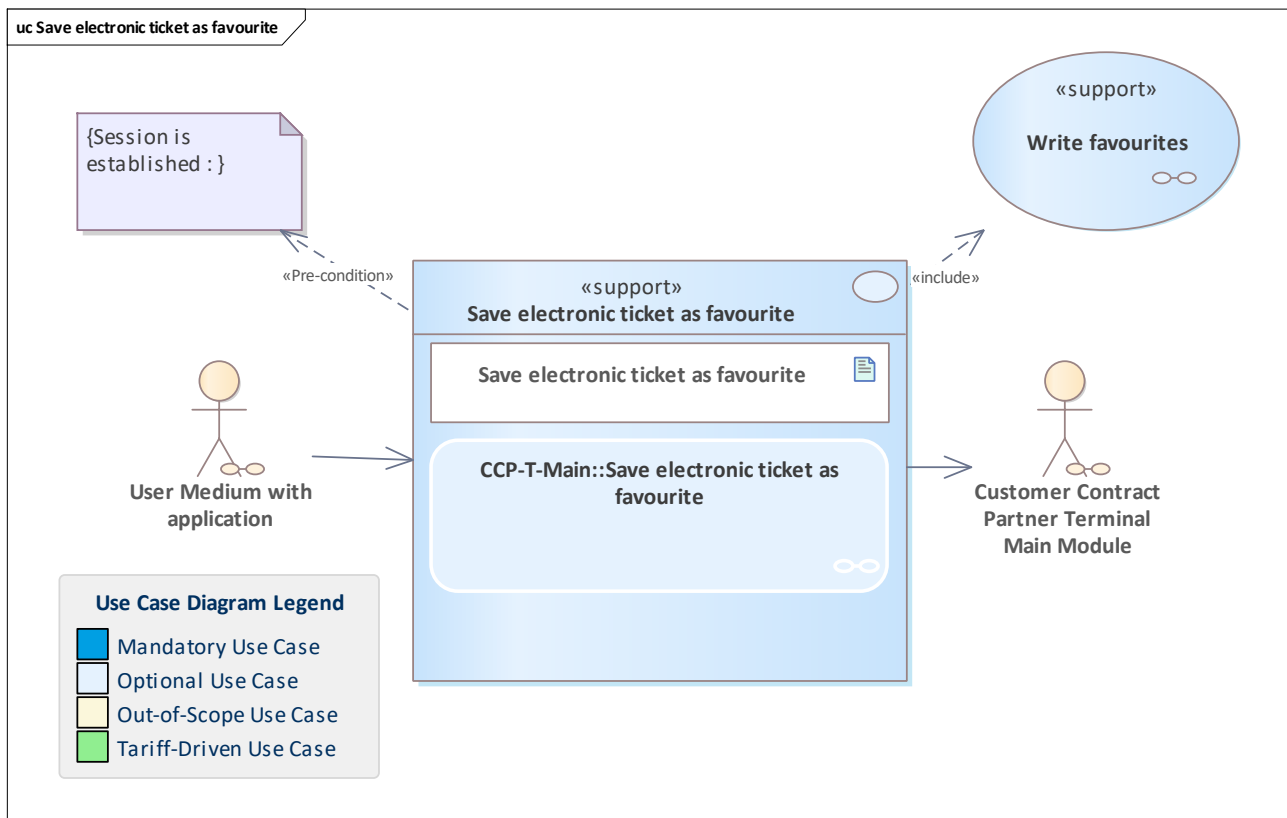


Figure 485: Save electronic ticket as favourite

Support use case that allows an authorised CCP terminal to save the information about an electronic ticket as a favourite.

11.370 Securely retrieve entitlement

11.371 Sell electronic ticket using account-based payment method

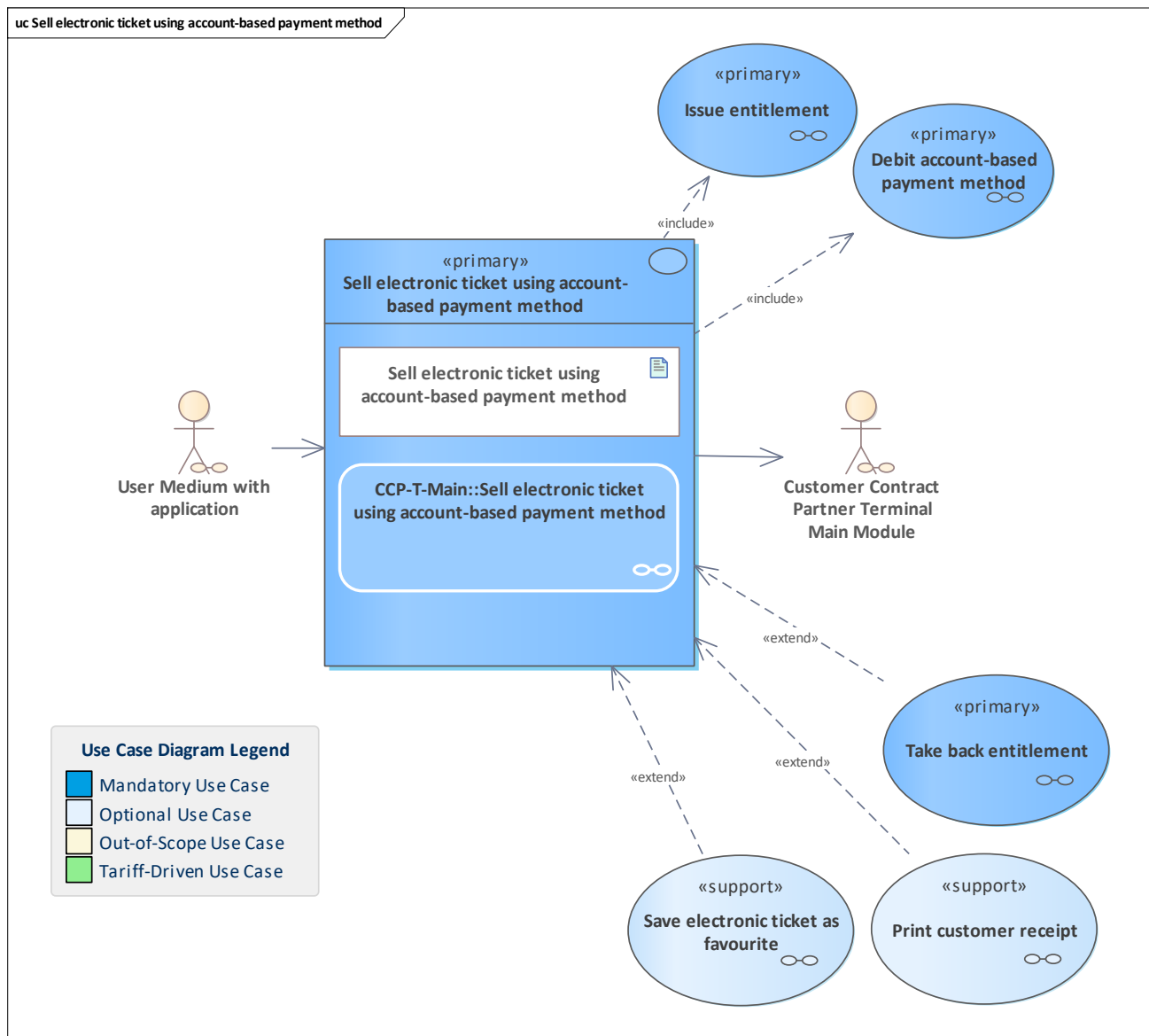


Figure 486: Sell electronic ticket using account-based payment method

An electronic ticket is sold and issued to a user medium. Payment is done using an account-based payment method already present on the same user medium. If there is not enough space on the target user medium to issue the electronic ticket, entitlements might be deleted and/or terminated after user/staff confirmation.

11.372 Sell electronic ticket using stored-value payment method

uc Sell electronic ticket using stored-value payment method

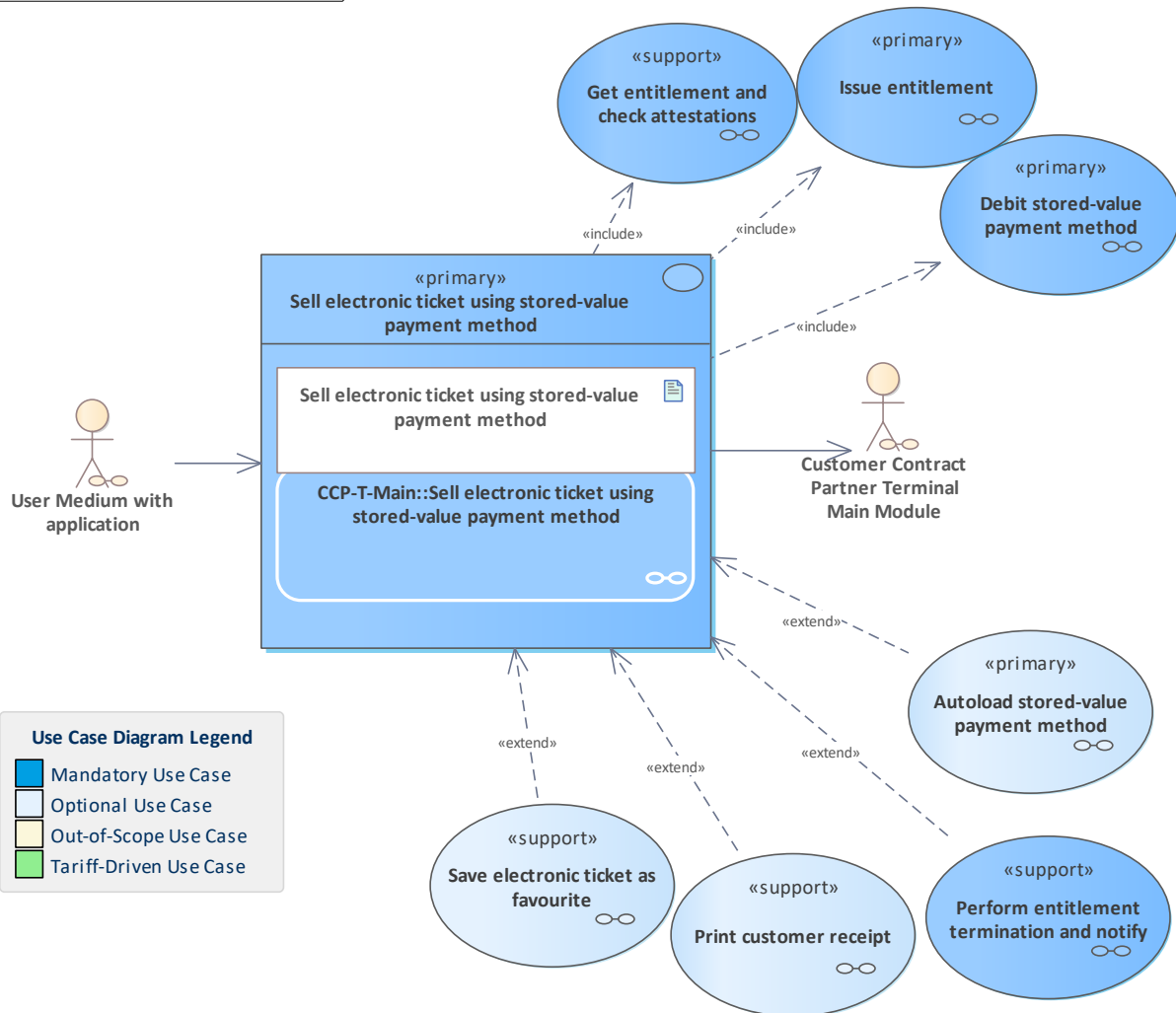


Figure 487: Sell electronic ticket using stored-value payment method

An electronic ticket is sold and issued to a user medium. Payment is done using a stored-value payment method already present on the same user medium. To do so, the payment method balance is checked.

If the balance does not suffice, the customer can recharge the payment method to be able to complete the sales process. If there is not enough space on the target user medium to issue the electronic ticket, entitlements might be deleted and/or terminated after user/staff confirmation.

11.373 Sell static entitlement using account-based payment method

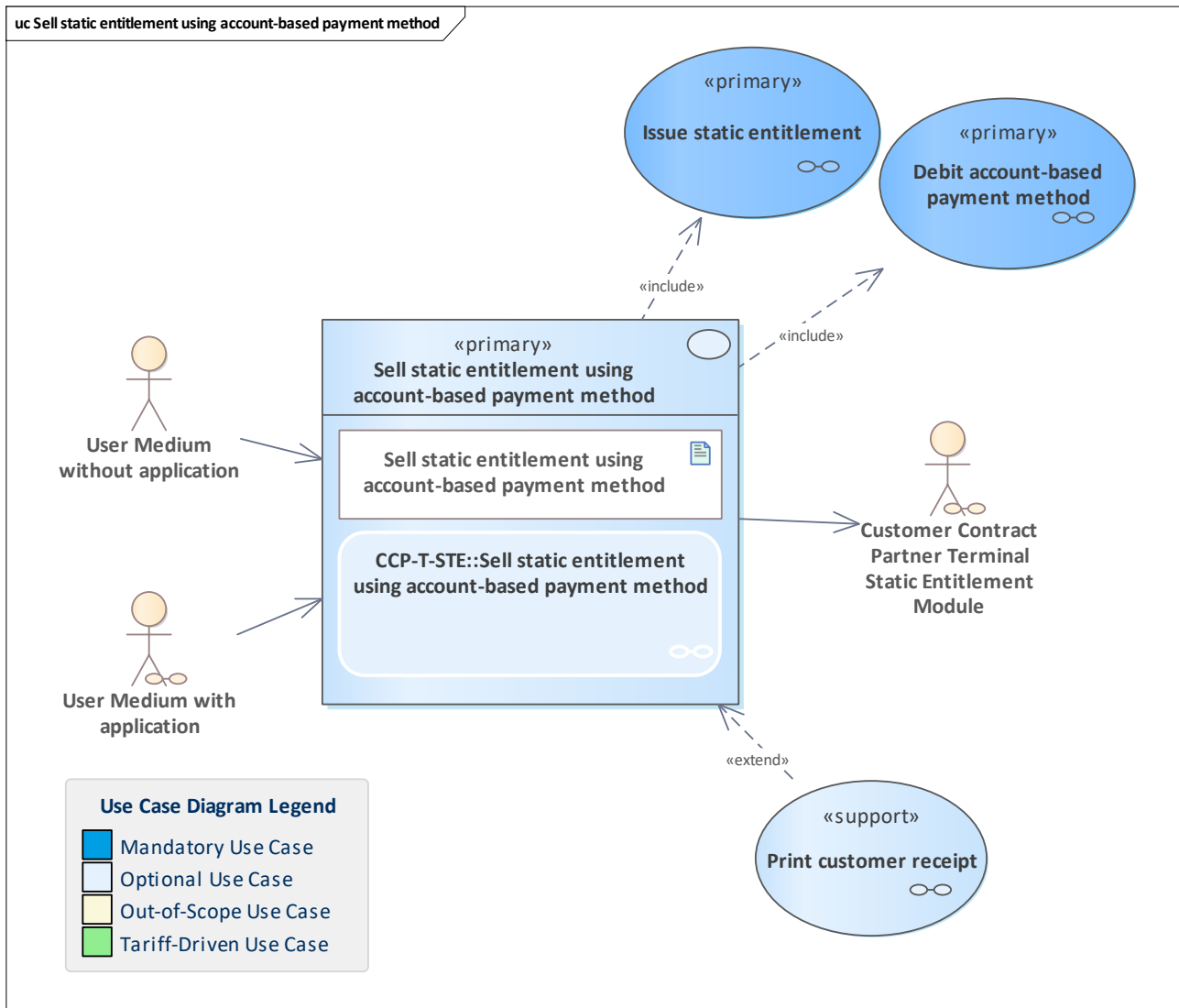


Figure 488: Sell static entitlement using account-based payment method

Sell a static entitlement using an account-based payment method.

Since this process involves both static and non-static entitlements, error scenarios and error handling mechanisms for both could apply. Thus, we provide additional steps to prevent error handling processes in the back-office systems where possible. For that, we exploit the fact that we can keep the already issued static entitlement from the customer until the payment succeeds.

Selling multiple static entitlements in a single process can be implemented in two ways, which both involve one debiting action per issued static entitlement:

1. Perform the issuance and debiting process steps (including notifying the back-office systems) multiple times and perform the delivery step only once for a [StaticEntitlements](#) structure containing the 1..* [StaticEntitlementData](#) for each issued static entitlement.
2. Perform the whole process multiple times. In the target system, e.g. the customer smartphone, the static entitlements delivered in multiple steps can then be combined into a single [StaticEntitlements](#) structure so that it can be inspected efficiently. To do so, take the [StaticEntitlementData](#) from all static entitlements and combine them with a single SAM certificate to form a single [StaticEntitlements](#) structure. Of course, the issuing SAM needs to be identical for all entitlements for this to work.

11.374 Sell static entitlement using stored-value payment method

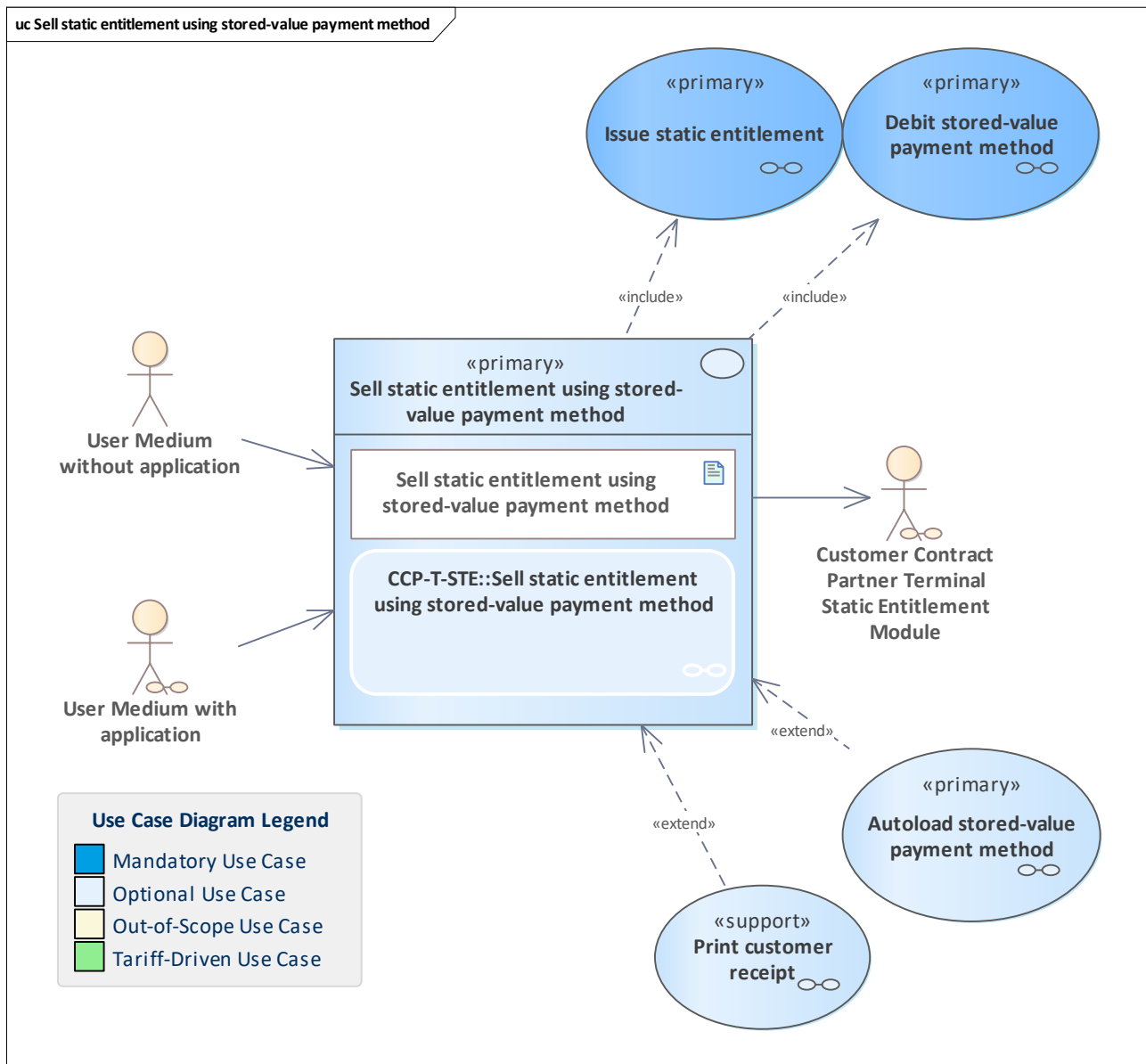


Figure 489: Sell static entitlement using stored-value payment method

Sell a static entitlement using a stored-value payment method.

Since this process involves both static and non-static entitlements, error scenarios and error handling mechanisms for both could apply. Thus, we provide additional steps to prevent error handling processes in the back-office systems where possible. For that, we exploit the fact that we can keep the already issued static entitlement from the customer until the payment succeeds.

Selling multiple static entitlements in a single process can be implemented in two ways, which both involve one debiting action per issued static entitlement:

1. Perform the issuance and debiting process steps (including notifying the back-office systems) multiple times and perform the delivery step only once for a [StaticEntitlements](#) structure containing the 1..* [StaticEntitlementData](#) for each issued static entitlement.

- Perform the whole process multiple times. In the target system, e.g. the customer smartphone, the static entitlements delivered in multiple steps can then be combined into a single [StaticEntitlements](#) structure so that it can be inspected efficiently. To do so, take the [StaticEntitlementData](#) from all static entitlements and combine them with a single SAM certificate to form a single [StaticEntitlements](#) structure. Of course, the issuing SAM needs to be identical for all entitlements for this to work.

11.375 Set service as available for a participant

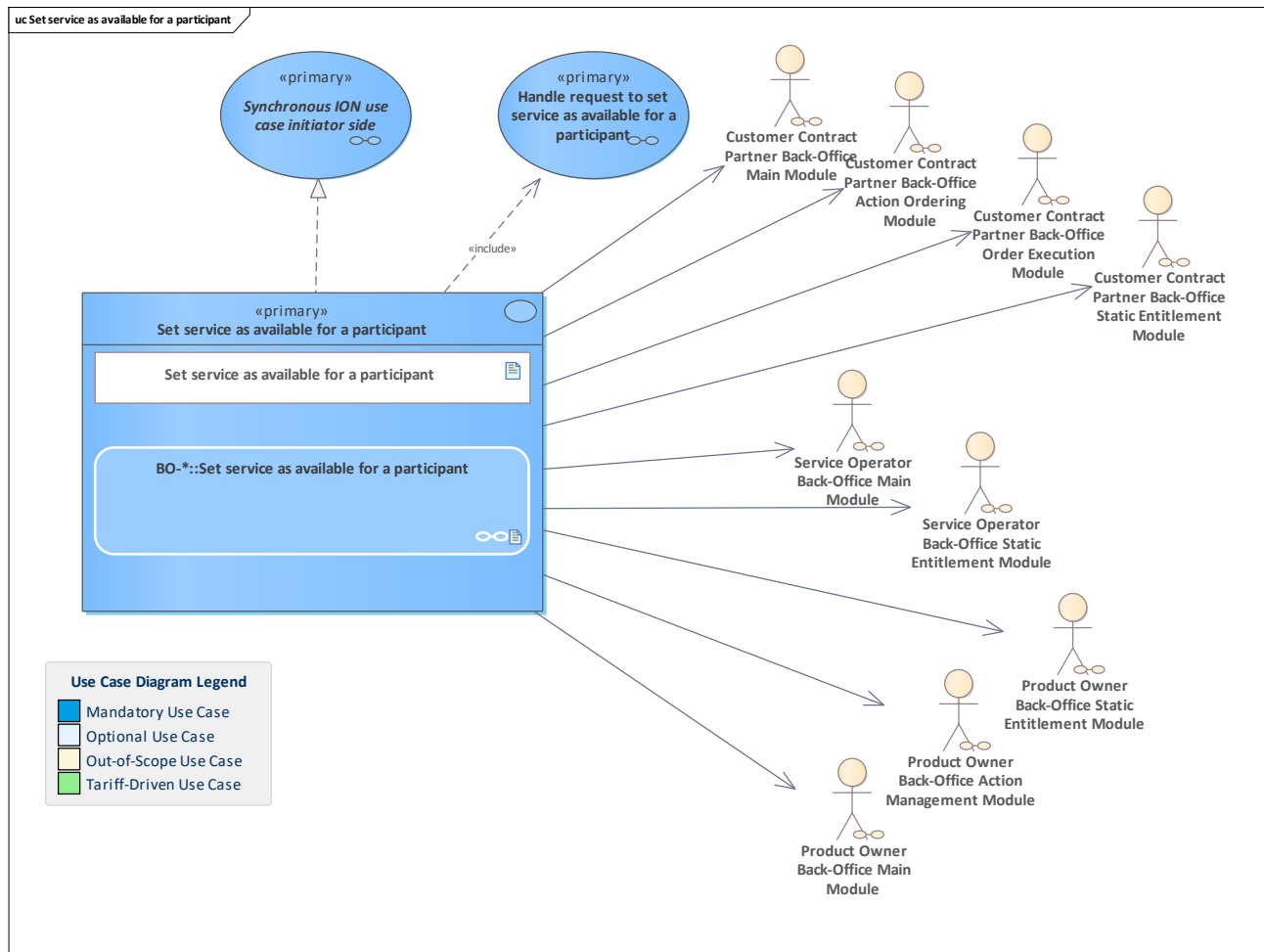


Figure 490: Set service as available for a participant

In this use case, a participant sets a certain service as being available in the central routing engine (CRE).

The purpose of this use case is to enable receiving messages for use cases in an asynchronous context. Note: if a service is announced to be available (again) the CRE will send stored messages to this service if any stored messages are still in the CRE's message queue.

11.376 Set service as unavailable for a participant

11.377 Set service as unavailable for a participant

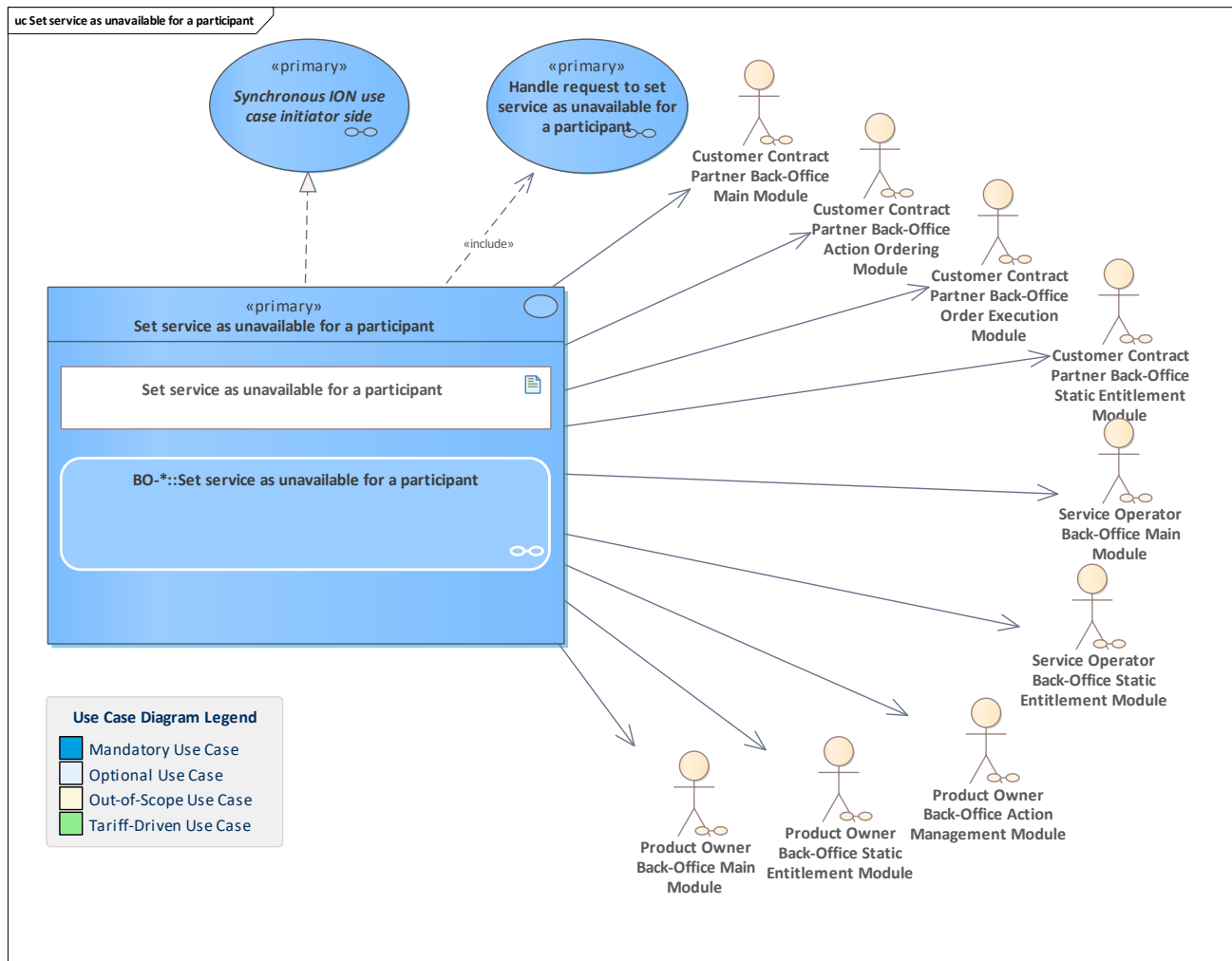


Figure 491: Set service as unavailable for a participant

In this use case, a participant sets a certain service as being unavailable in the central routing engine (CRE).

The purpose of this use case is to disable receiving messages for use cases in an asynchronous context. This means, that from now on, the CRE will store messages for a later delivery instead of routing them directly to the system that implements the service.

Note: participants must configure their services via the ESH in a preparatory step.

11.378 Take back application

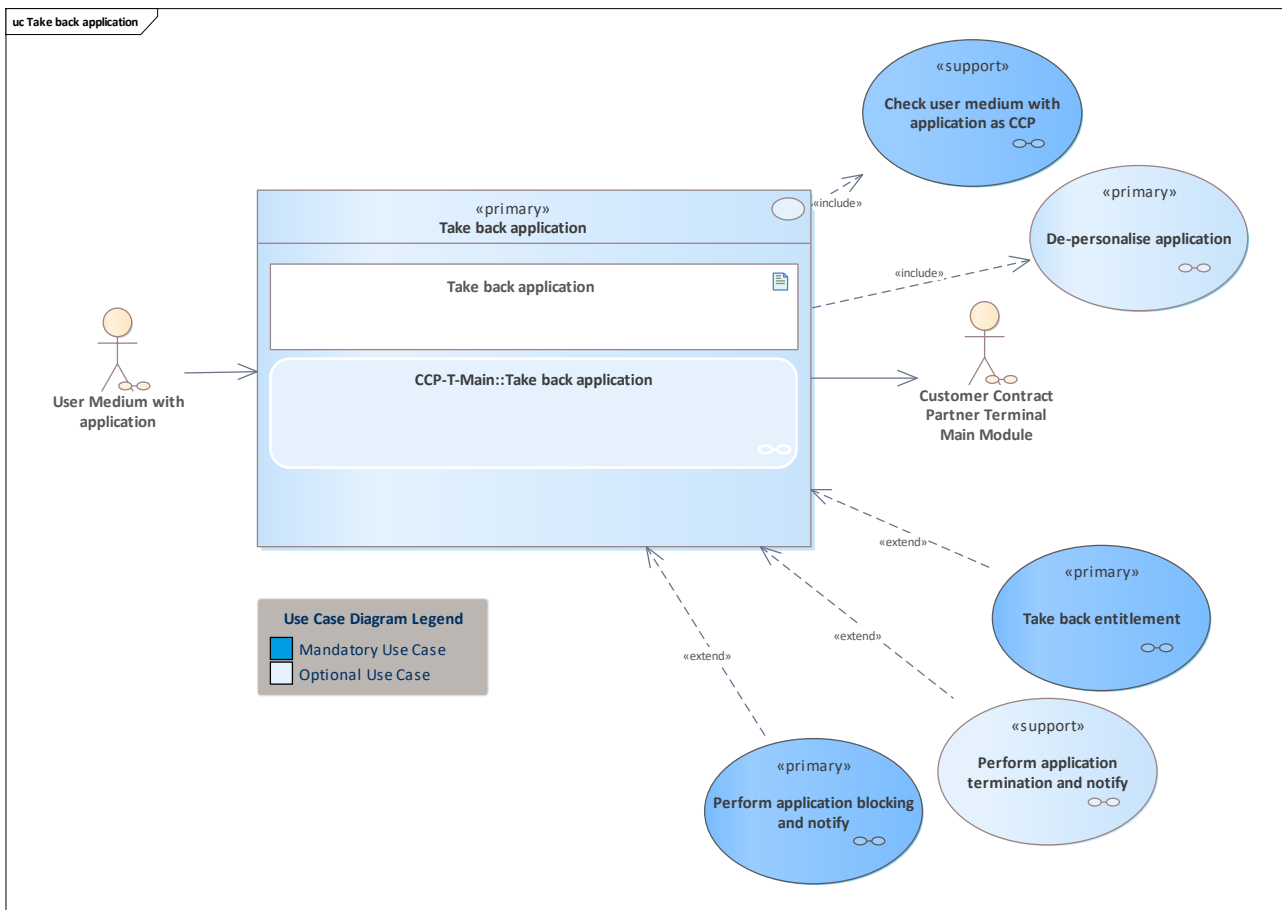


Figure 492: Take back application

Use case for a CCP terminal to take back an application.

This process may only be performed by the organisation that issued the application.

An application (of a user medium) is terminated when the underlying user medium is taken back from a customer (and has reached its end of life) or when the (((etiCORE-specific application is removed from a customer-owned user medium. This includes removing all customer-specific information from an (((etiCORE application.

The actual process of terminating an application leads to marking the application as terminated.

11.379 Take back entitlement

11.380 Take back entitlement

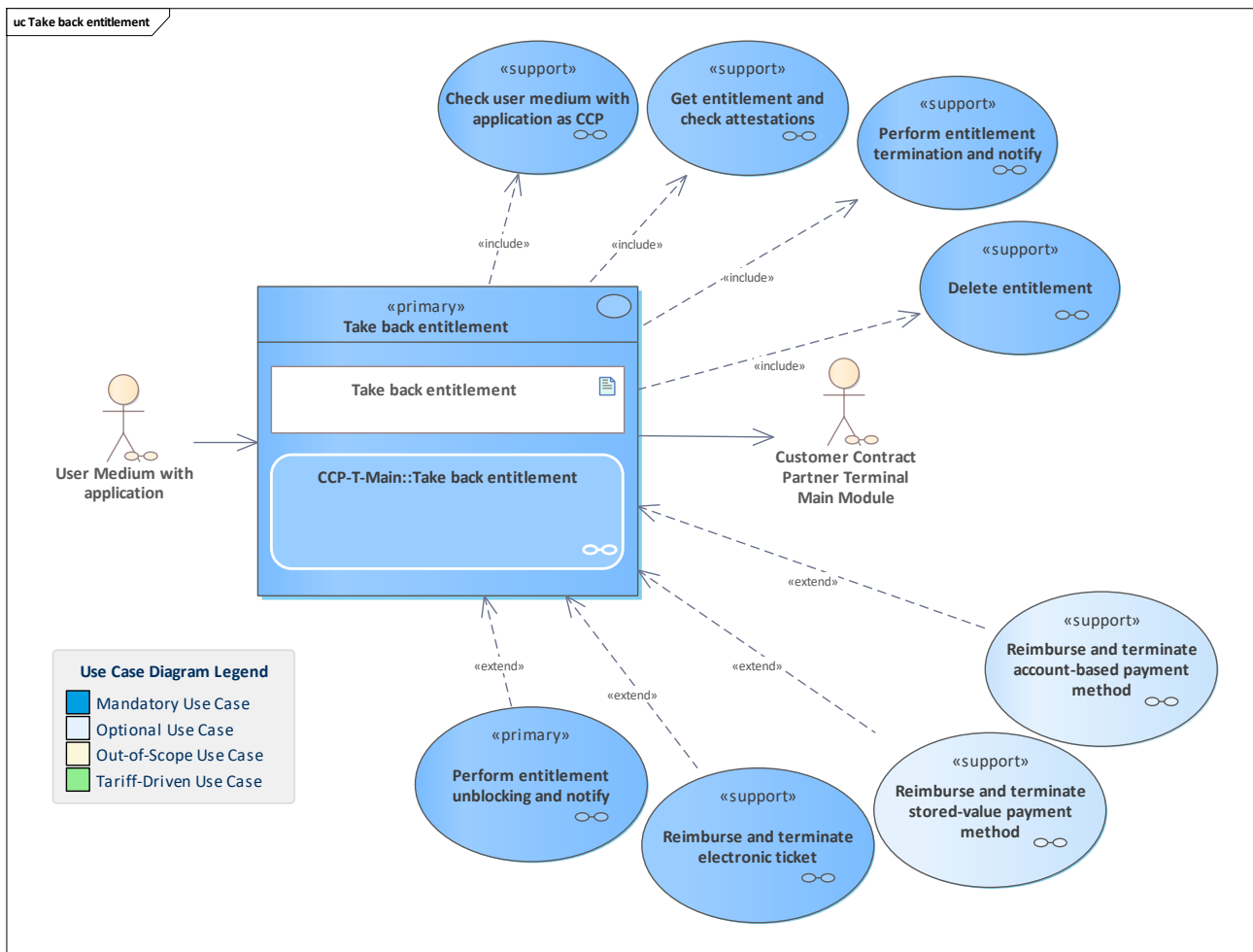


Figure 493: Take back entitlement

An entitlement is marked as terminated to prevent any further usage of the entitlement. Usually, the entitlement is deleted from the user medium application directly afterwards. This process may optionally involve reimbursement towards the customer in case it is executed at a terminal of the entitlement owner (pCCP).

11.381 Take back static entitlement

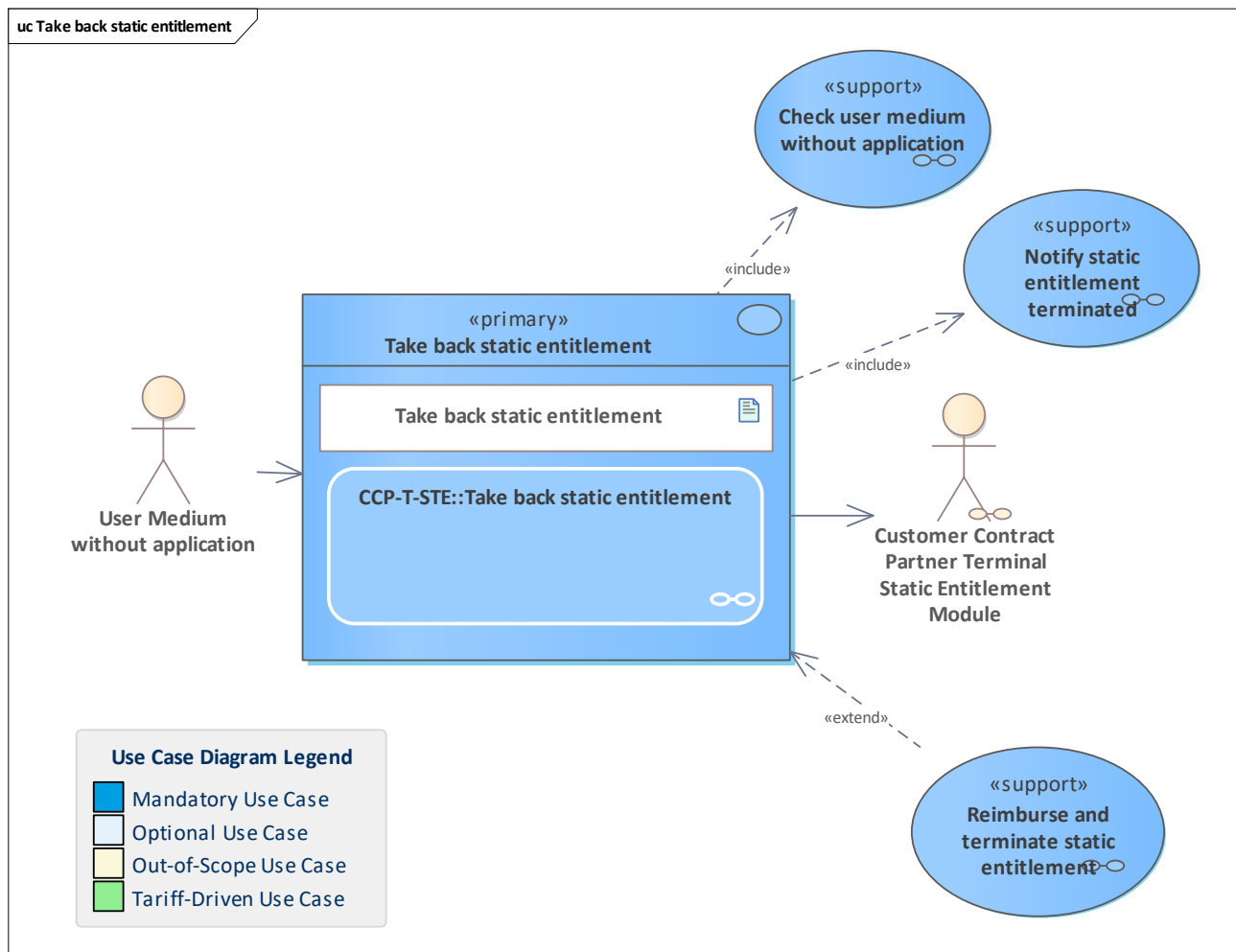


Figure 494: Take back static entitlement

A CCP terminal with static entitlement extension starts the process. Taking back of a static entitlement process (including creating and sending notifications) is always required when static entitlements that are still valid in terms of time are taken back and, in particular, when a refund is in question. Since there is no guarantee that copies might still be in circulation, this entitlement must be hotlisted as soon as possible and the option for taking back should be possible only for a pCCP.

If a reimbursement is marked as available in the tariff regulations, this use case, in conjunction with a reimbursement, is only permissible with the CCP that issued the entitlement (primary CCP). In this case, reimbursement conditions defined within product modules for a product must be evaluated.

Irrespective of whether only the termination was performed or the additional reimbursement, the responsible CCP back-office system will be informed.

11.382 Terminal startup procedure

11.383 Terminal startup procedure

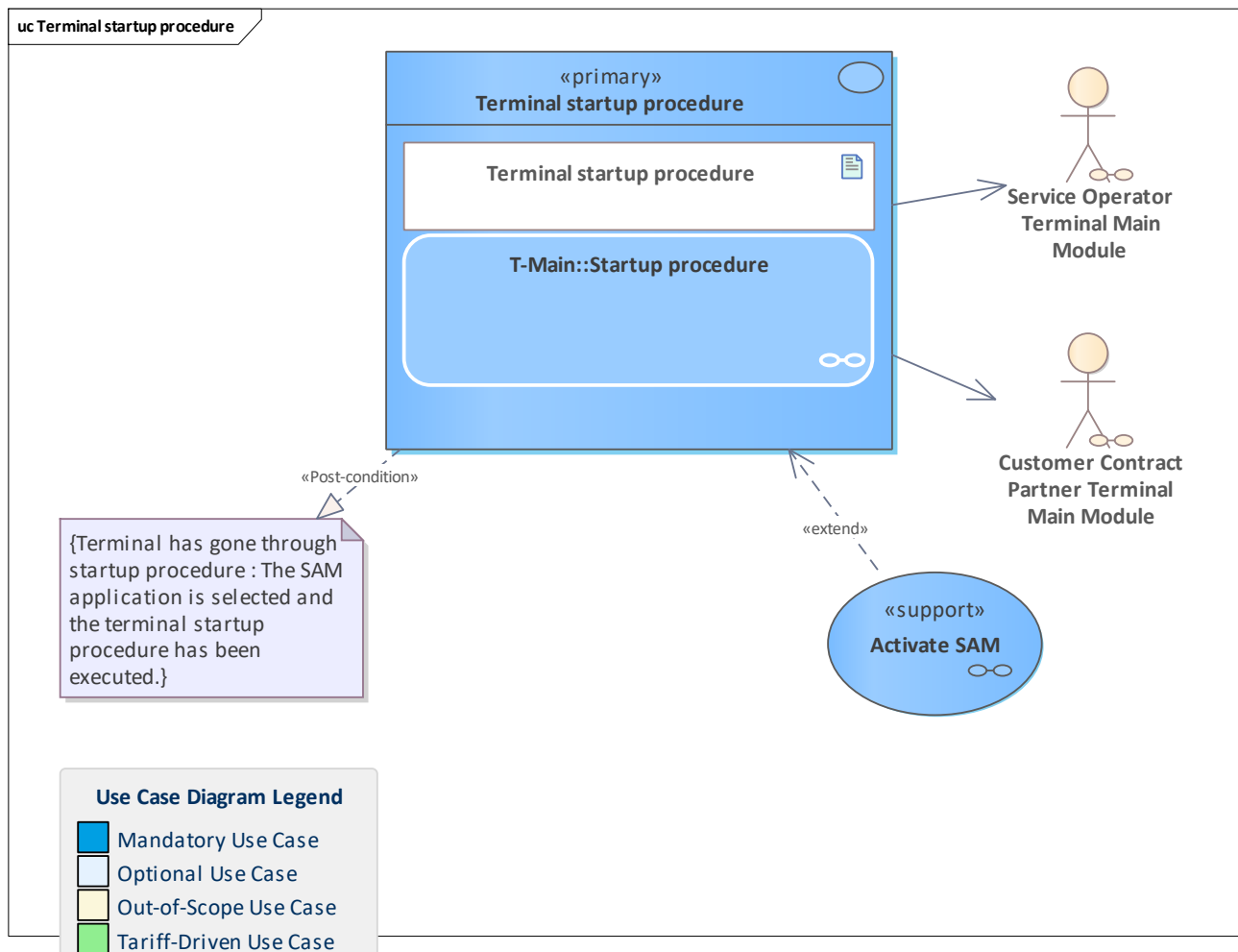


Figure 495: Terminal startup procedure

This startup procedure brings the terminal into an operational state, initialising the SAM and caching all relevant information about it for use in UM interactions.

It is used

- to initially start the terminal,
- when the SAM has been replaced,
- when the Action operator ID has been re-configured,
- when the hotlists have been updated,
- when new SAM configuration scripts are available, and
- periodically (e.g. every 24 hours) to recheck validity periods

11.384 Terminate entitlement triggered by action order

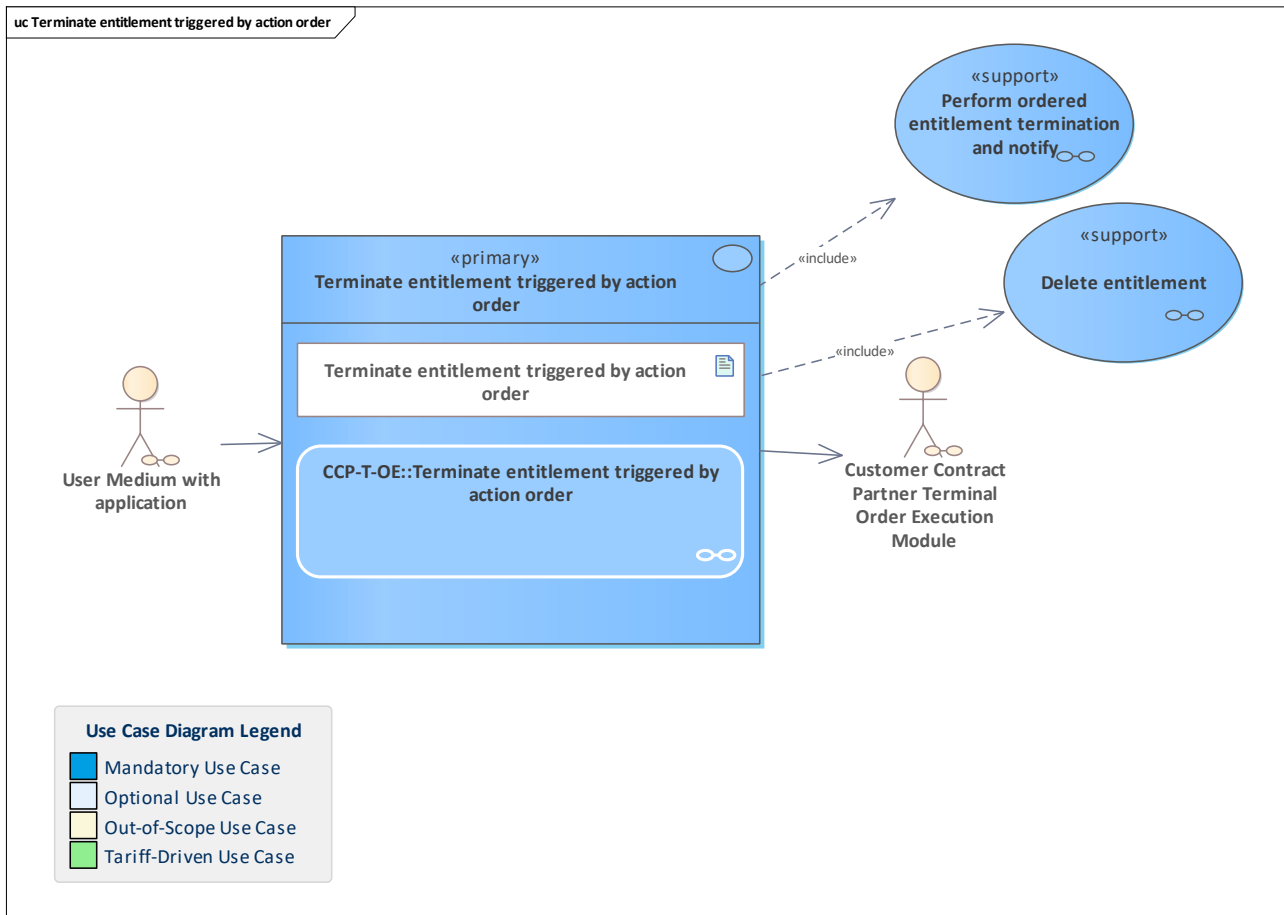


Figure 496: Terminate entitlement triggered by action order

An entitlement termination order that is potentially relevant for a given user medium is checked regarding the need to execute it and is executed if necessary. To achieve this, the terminal does a lookup in the action list for the application instance ID. To avoid a duplicate termination attempt, the user medium has to be examined to check if the entitlement in question is already in the state [Entitlement terminated](#).

11.385 Terminate UM application

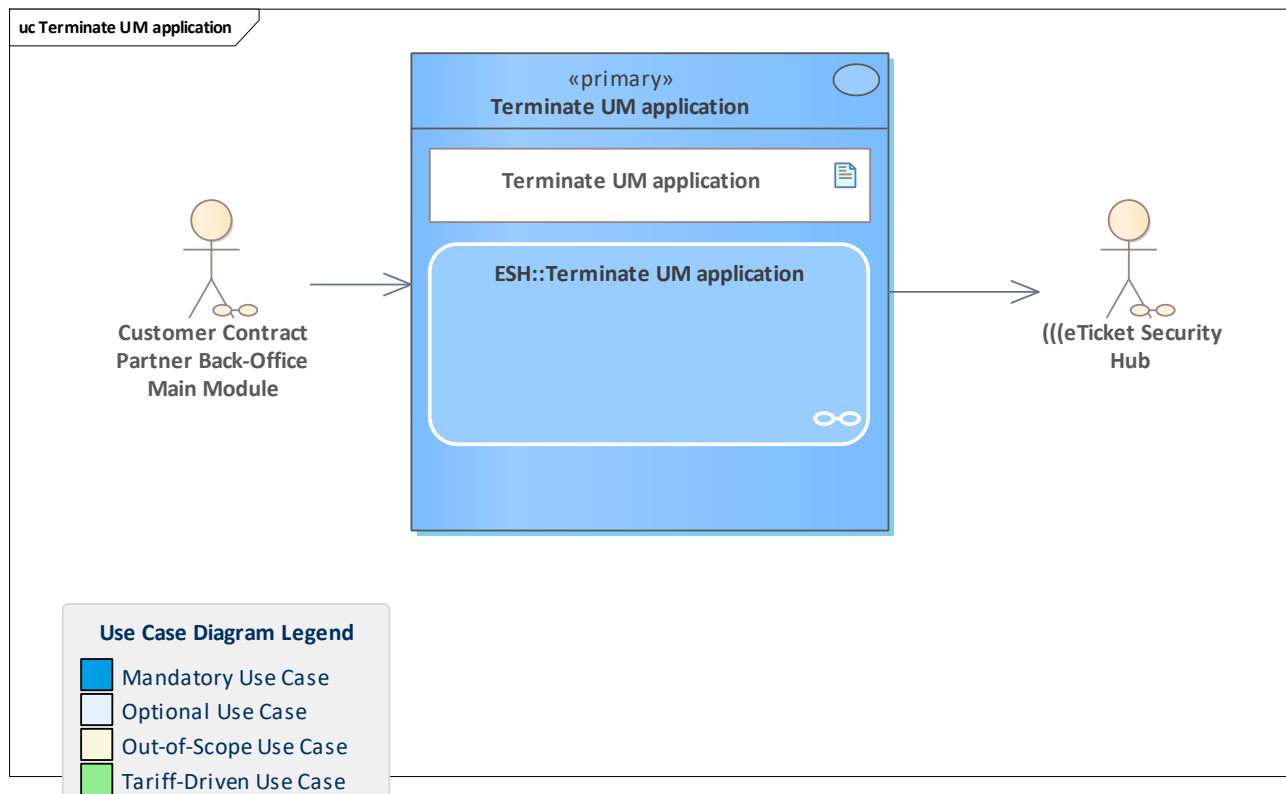


Figure 497: Terminate UM application

This use case give participants the capability to inform the Scheme Manager about UM application terminations. Subsequently, the remaining certificate fees for the current application instance will be waived.

11.386 Trigger entitlement issuance

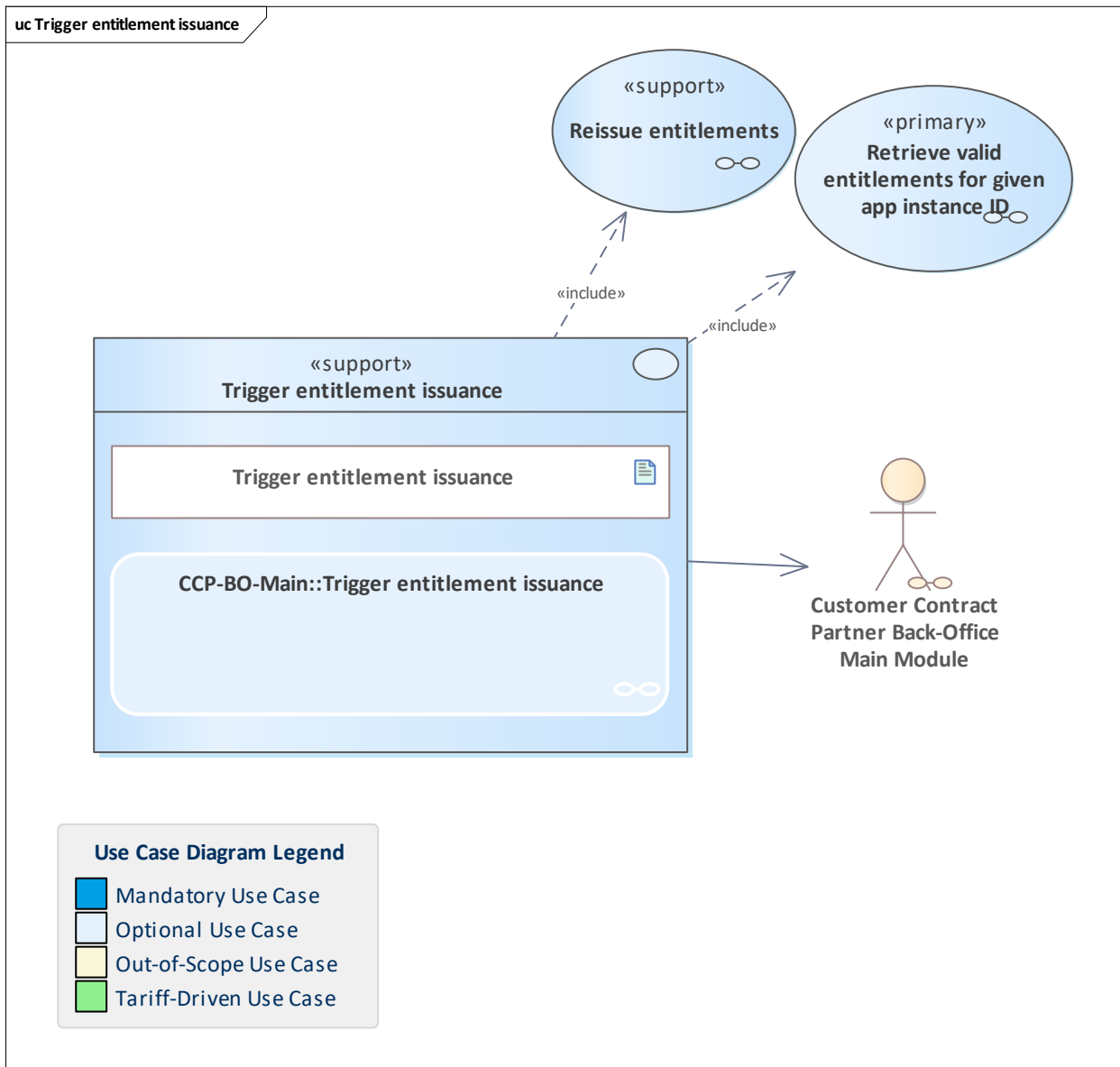


Figure 498: Trigger entitlement issuance

A customer needs a new user medium with his entitlements which were located on the old one. Information regarding these entitlements is gathered. New entitlements are issued based on that information.

11.387 Unblock application

11.388 Unblock application

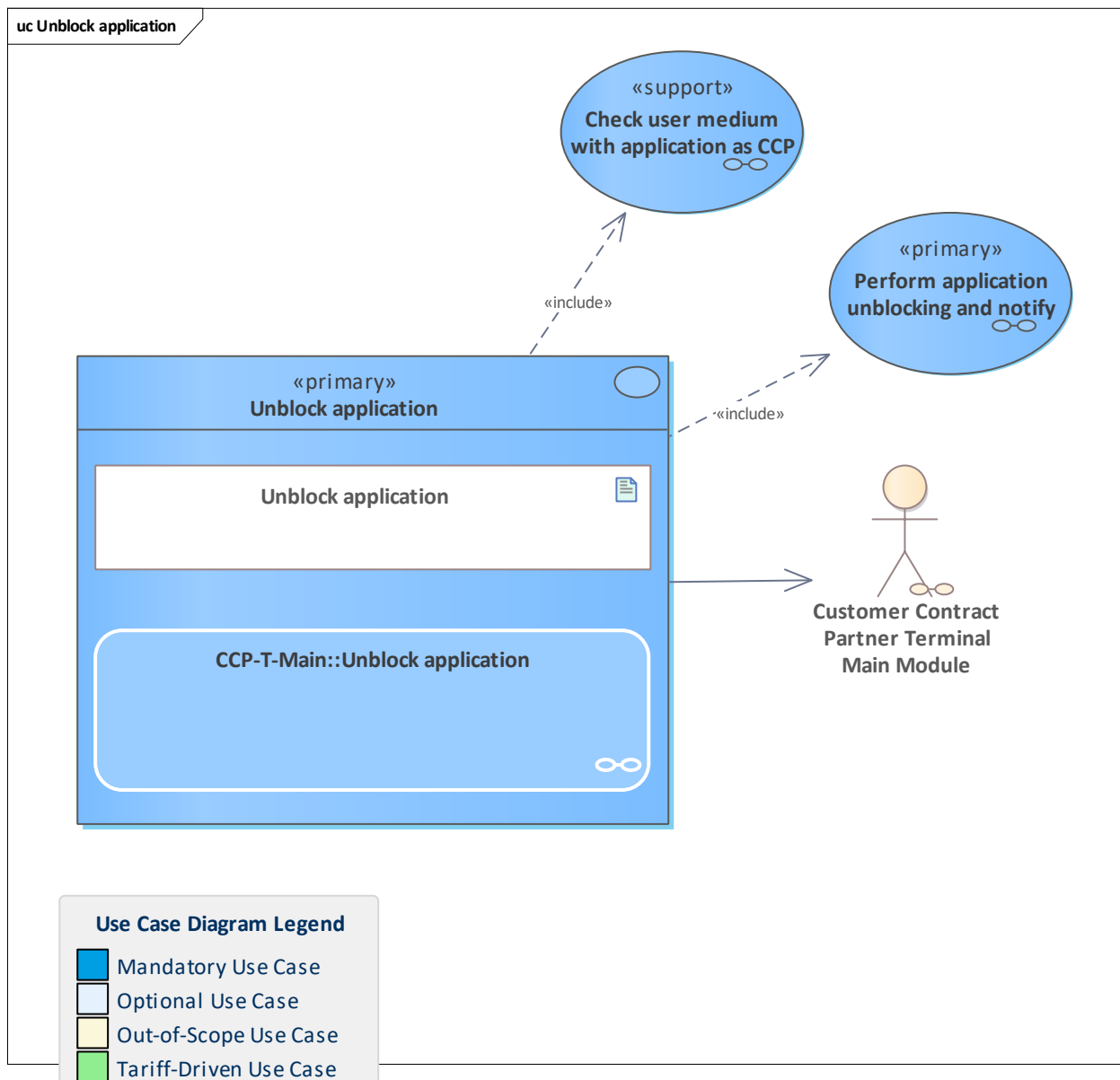


Figure 499: Unblock application

A user medium with an application will be checked in the terminal. In case of a blocked application within the validity period, the terminal checks if the application can be unblocked. If yes, the terminal performs application unblocking and notifies the back-office system. Otherwise, the unblocking process of the application is stopped.

Note: only the pCCP is allowed to unblock its owned applications.

11.389 Unblock entitlement

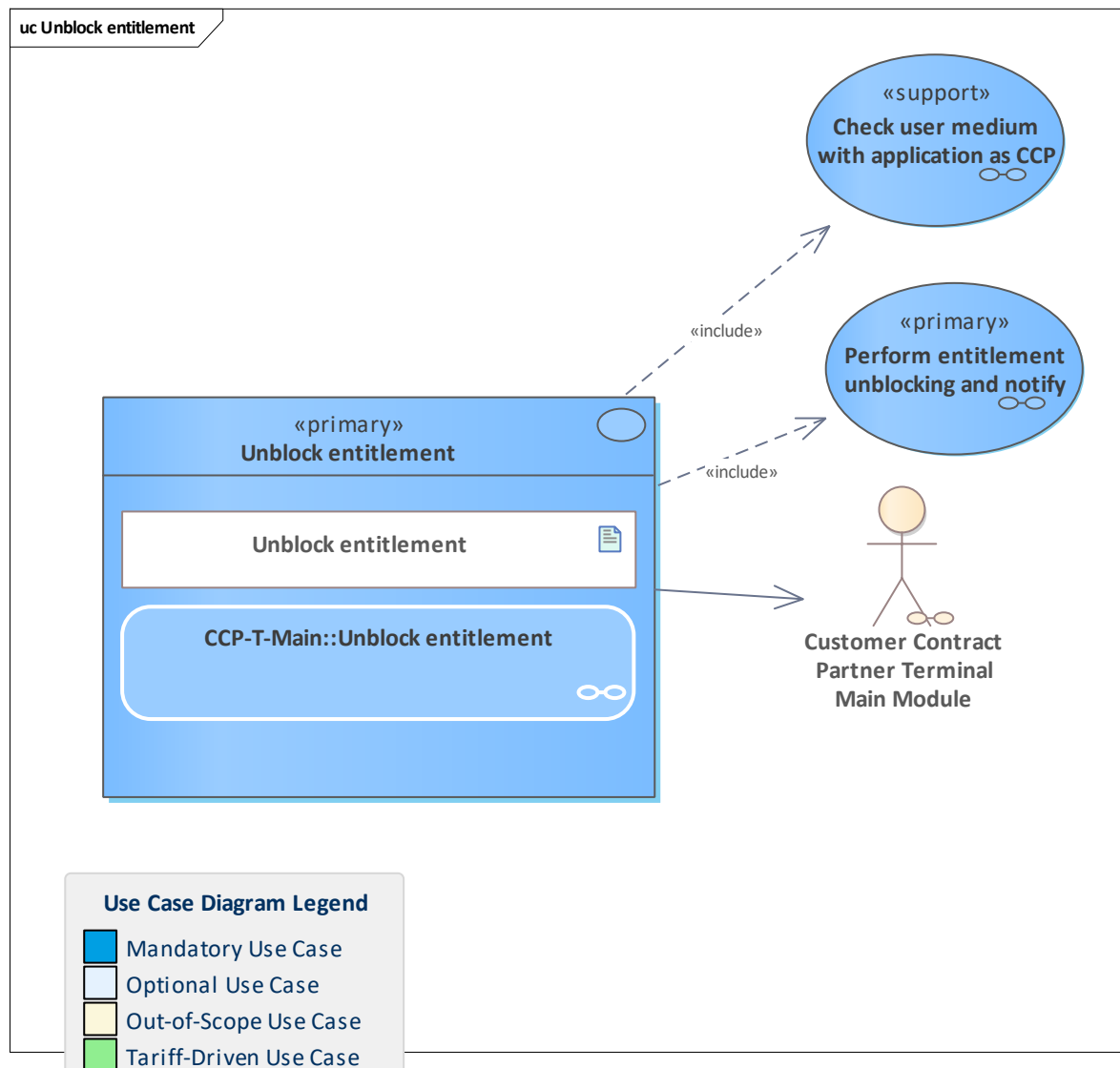


Figure 500: Unblock entitlement

A user medium with an application will be checked in the terminal. All previously blocked entitlements are determined and read.

Each entitlement in question is checked if it can be unblocked.

Otherwise, the unblocking of the entitlement in question is skipped.

If at least one entitlement exists for unblocking, the terminal performs entitlement unblocking and notifies the CCP back-office system.

Note: only the pCCP is allowed to unblock its owned entitlements.

11.390 Unblock entitlement triggered by action order

11.391 Unblock entitlement triggered by action order

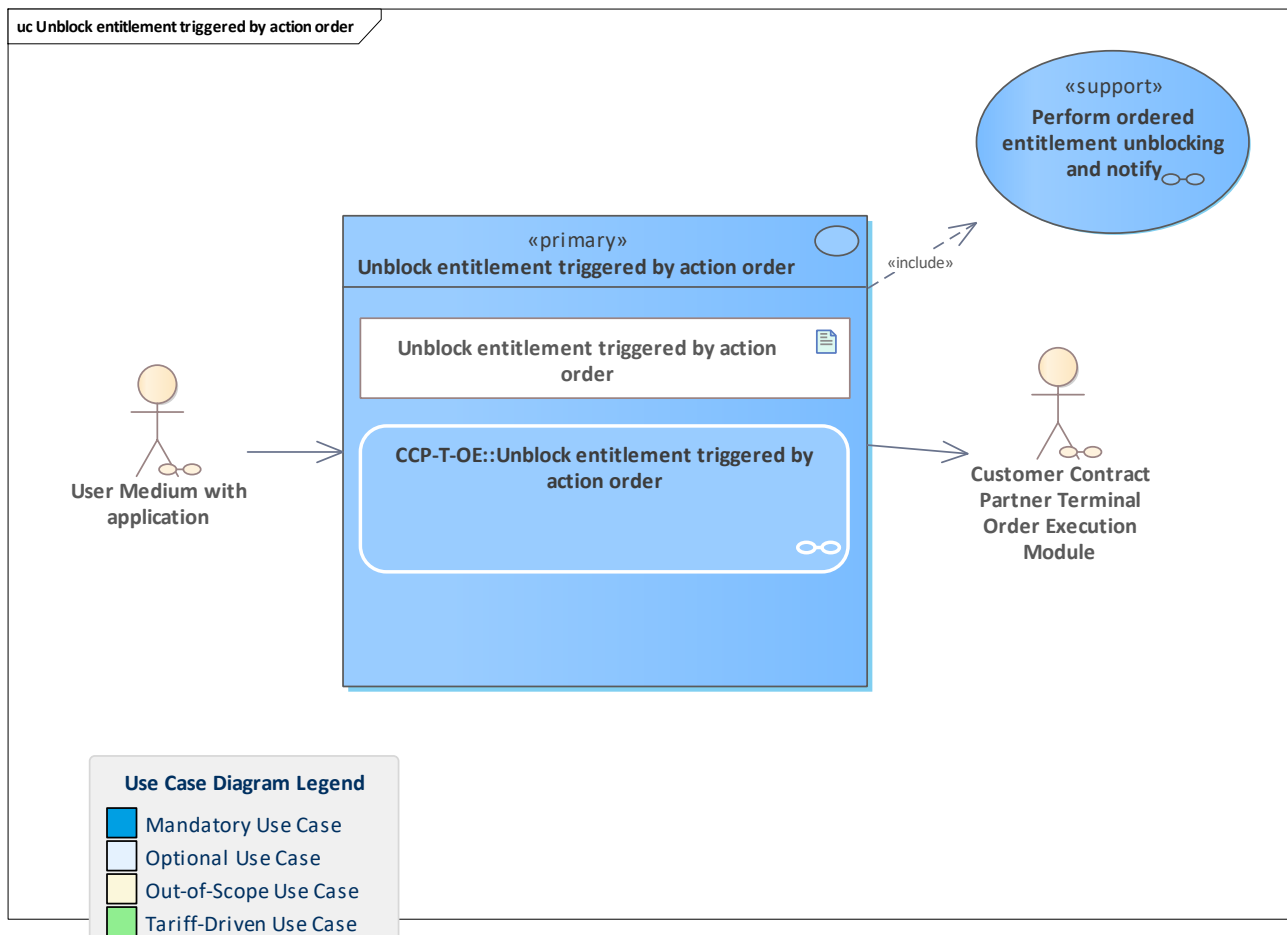


Figure 501: Unblock entitlement triggered by action order

An entitlement unblocking order that is potentially relevant for a given user medium is checked regarding the need to execute it and, if necessary, is executed.

To achieve this, the terminal does a lookup in the action list for the application instance ID. If the ID matches, the entitlements are examined. If the entitlement ID of one entitlement matches with the entry in the action list, the entitlement in question is unblocked.

To avoid a duplicate unblocking attempt, the user medium has to be examined to check if the entitlement in question is already in the regular state.

11.392 Update action list inventory from operational perspective

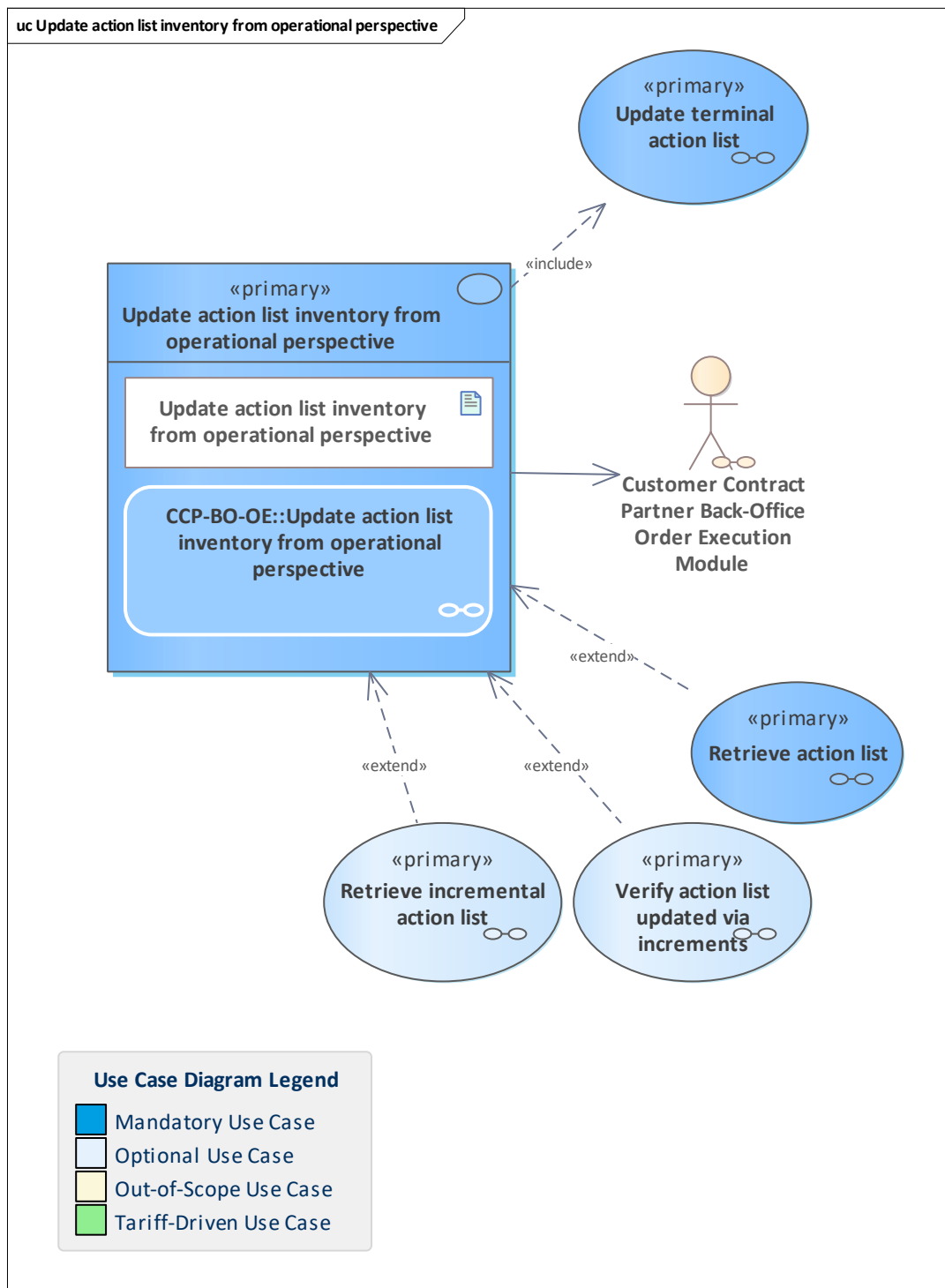


Figure 502: Update action list inventory from operational perspective

The action list is retrieved by the [Customer Contract Partner Back-Office Order Execution Module](#) from the [Product Owner Back-Office Action Management Module](#) and distributed to its relevant terminals.

When determining when to invoke this process, the retrieval configuration provided by the [Product Owner Back-Office Action Management Module](#) should be respected, see [Process action list retrieval configuration](#).

When several [Product Owner Back-Office Action Management Modules](#) are involved, the updated action lists need to be merged before passing them on to the terminals. This scenario is not shown here for simplicity.

11.393 Update authentication key hotlist inventory

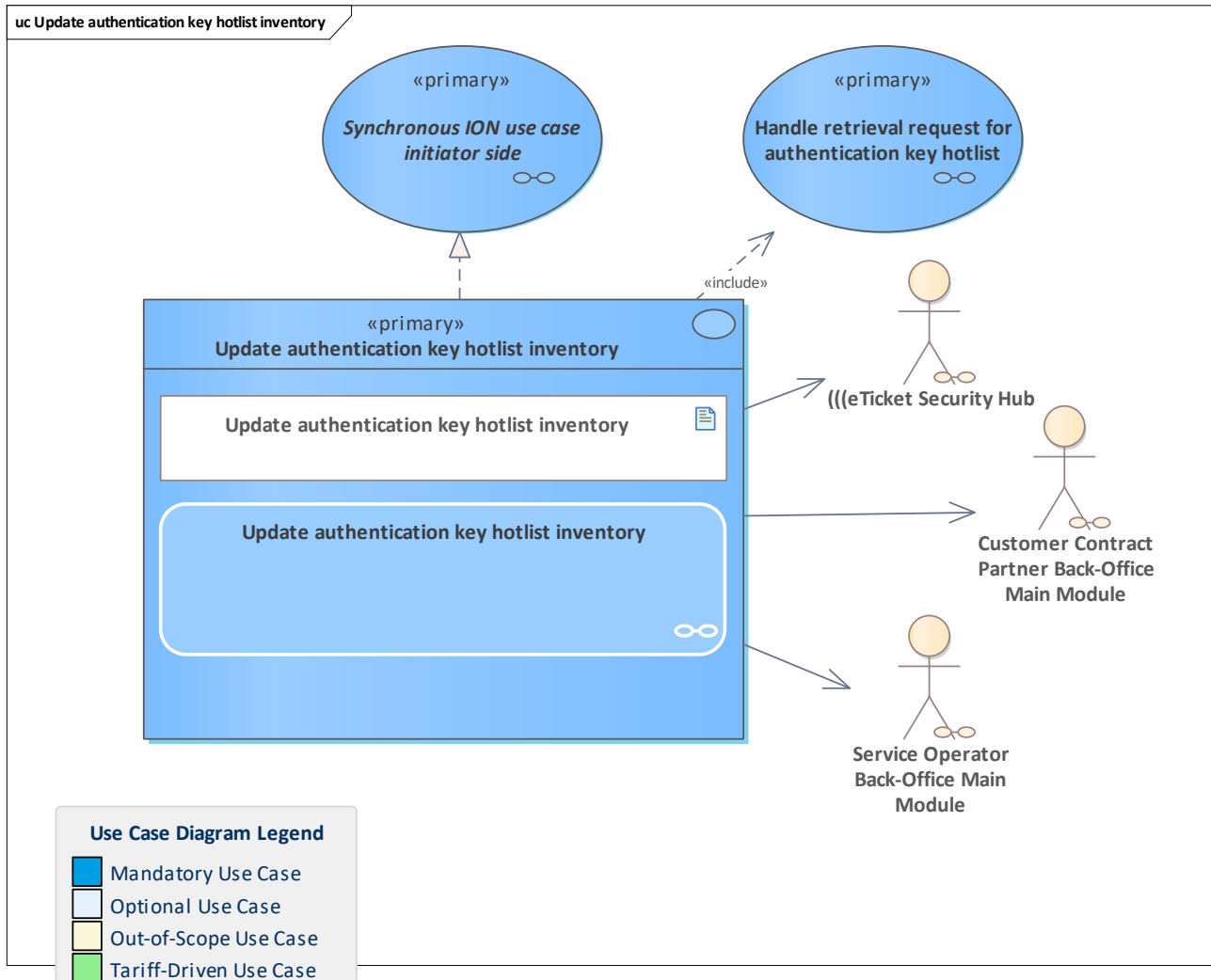


Figure 503: Update authentication key hotlist inventory

The SO, CCP and the scheme manager's ESH want to update the authentication key hotlist inventory by retrieving the current authentication key hotlist from the hotlist service system and processing it into their authentication key hotlist inventory.

11.394 Update CA certificate repository

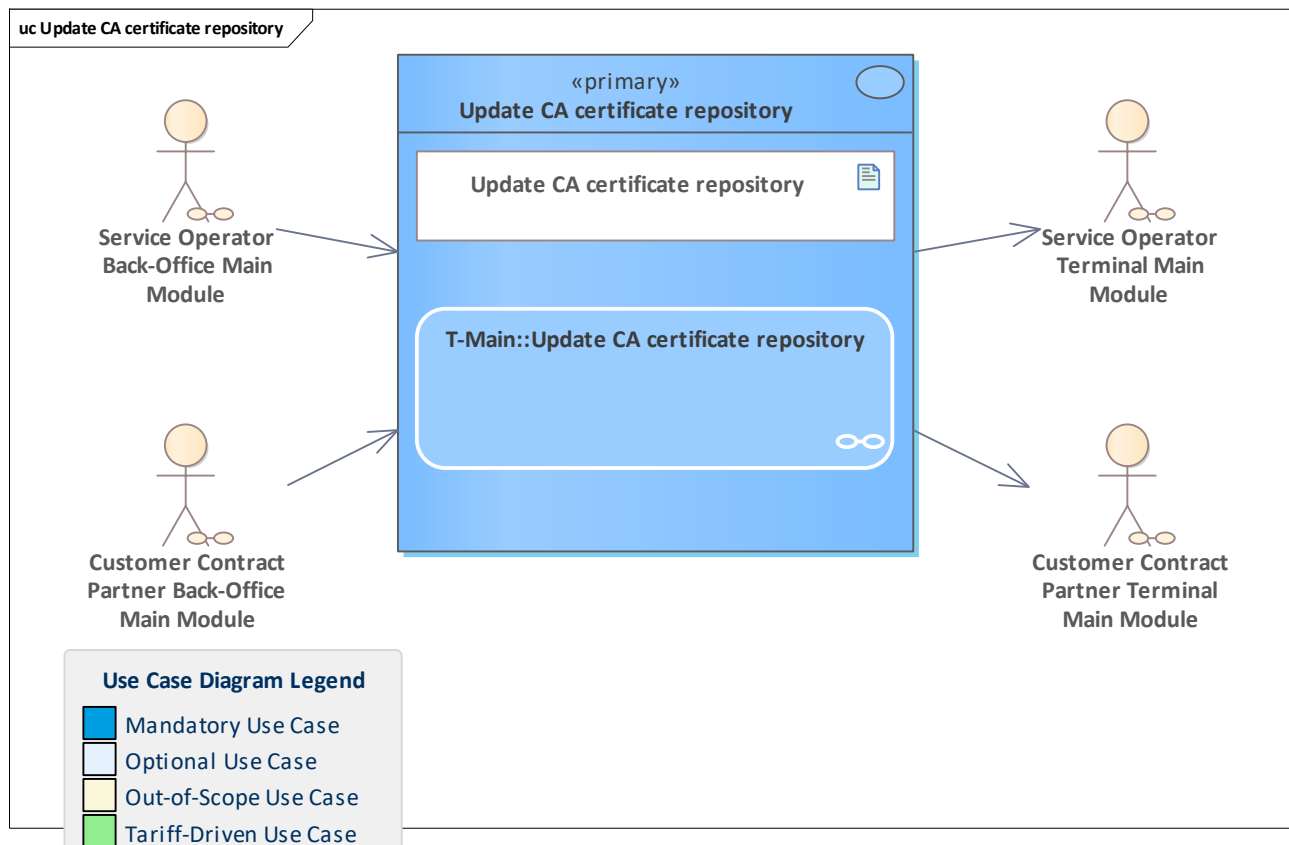


Figure 504: Update CA certificate repository

Updates the CA certificate repository in the terminal.

11.395 Update CV certificate revocation list

11.396 Update CV certificate revocation list

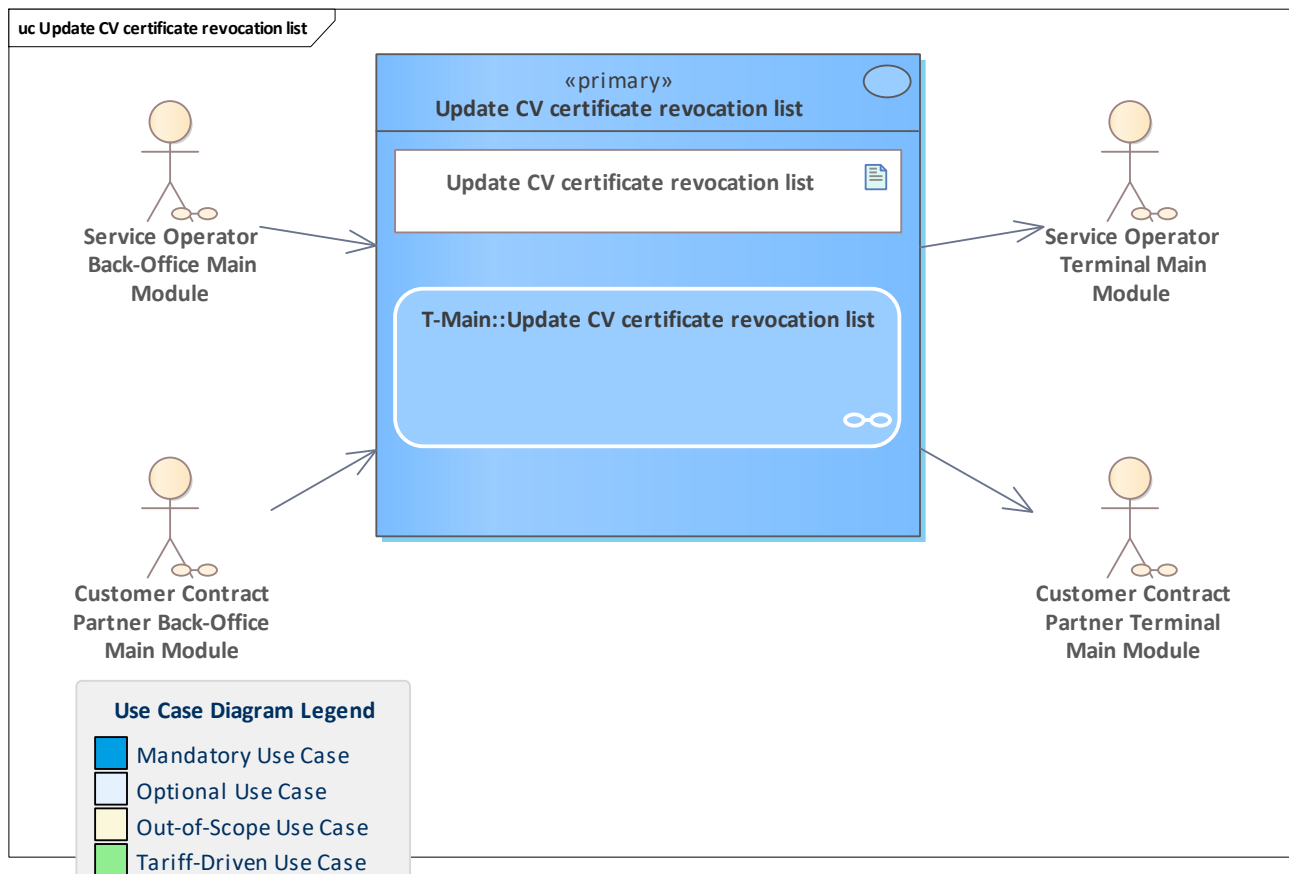


Figure 505: Update CV certificate revocation list

Updates the CV certificate revocation list in the terminal.

11.397 Update hotlist inventory from operational perspective

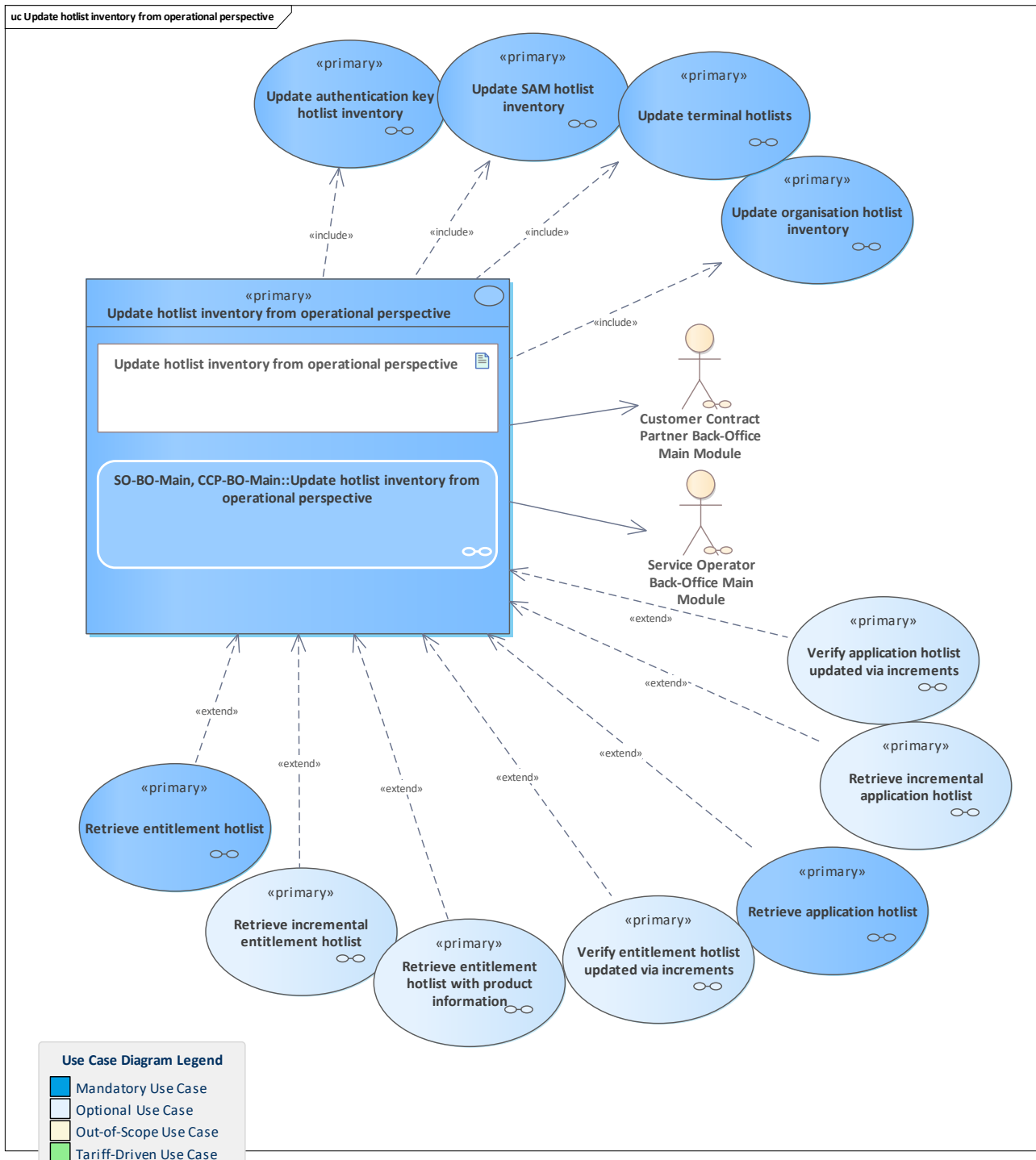


Figure 506: Update hotlist inventory from operational perspective

A CCP or SO wants to retrieve the currently available hotlists and distribute them to its terminals to inspect accessed entitlements and applications, as well as to deactivate a terminal if its SAM has been hotlisted.

The available hotlists to be updated and distributed are:

- Application hotlist:
 - either total application hotlist
 - or incremental application hotlist
- Entitlement hotlist:
 - either total entitlement hotlist



or incremental entitlement hotlist
or total entitlement hotlist with product information

- SAM hotlist
- Organisation hotlist
- Authentication key hotlist

11.398 Update hotlist inventory from product perspective

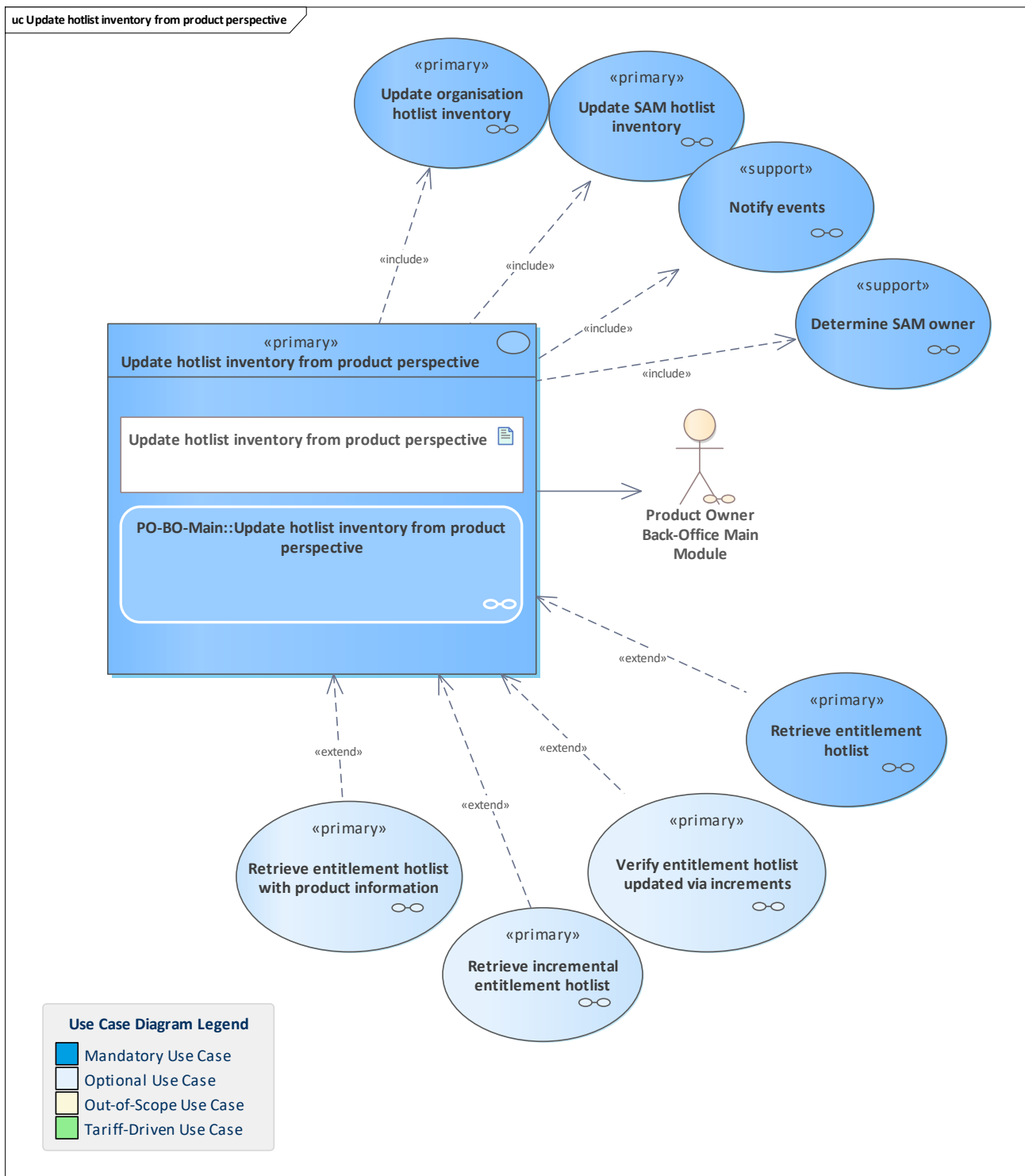


Figure 507: Update hotlist inventory from product perspective

A product owner wants to retrieve the currently available hotlists and analyse them by checking if hotlisted elements are already blocked in its inventory.

The available hotlists are:

- Entitlement hotlist:
 - either total entitlement hotlist
 - or incremental entitlement hotlist
 - or total entitlement hotlist with product information
- SAM hotlist

- Organisation hotlist

11.399 Update organisation hotlist inventory

11.400 Update organisation hotlist inventory

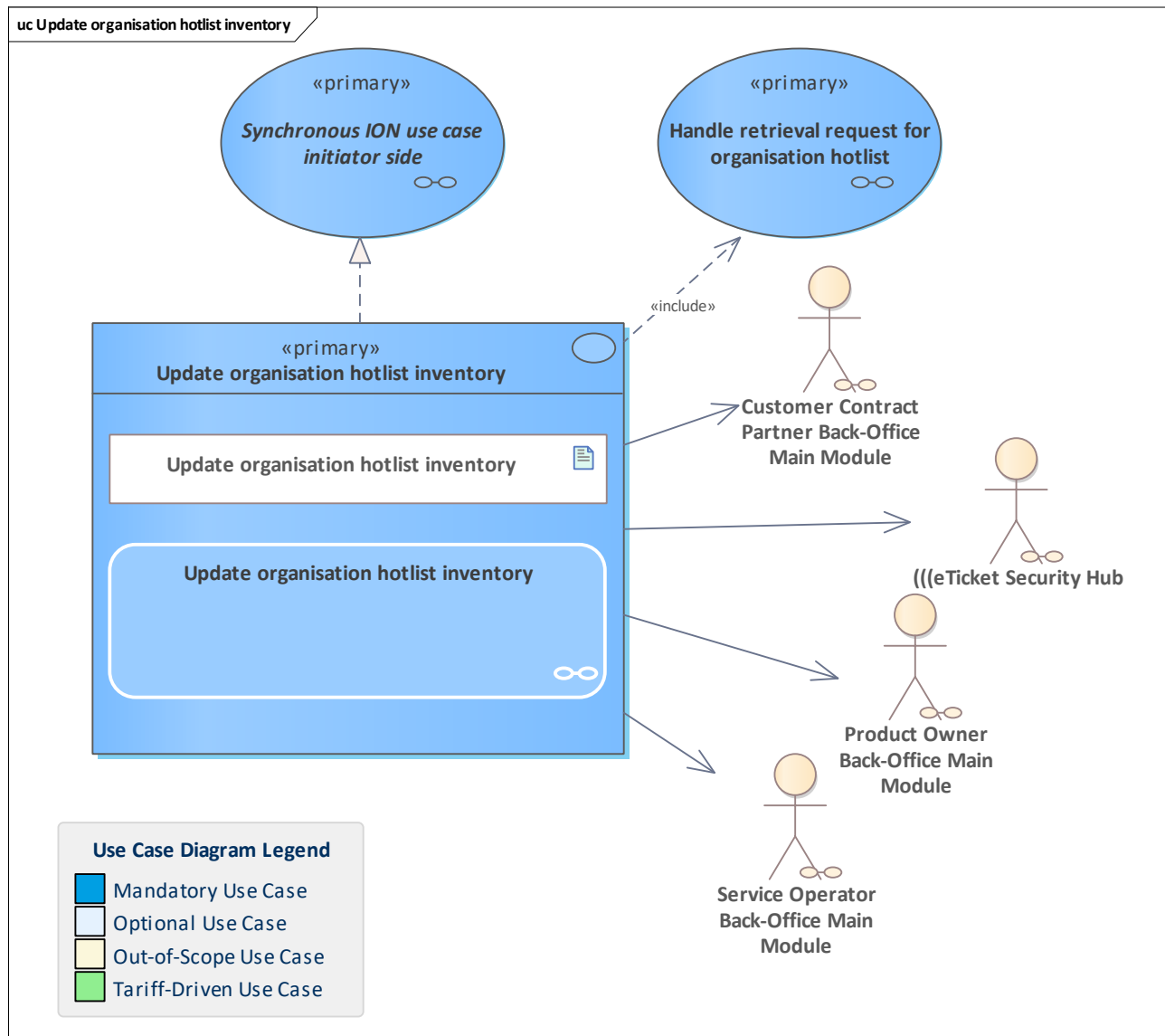


Figure 508: Update organisation hotlist inventory

The SO, CCP, PO and the scheme manager's ESH want to update their organisation hotlist inventory by retrieving the current organisation hotlist from the hotlist service system and processing it into their organisation hotlist inventory.

11.401 Update organisation list

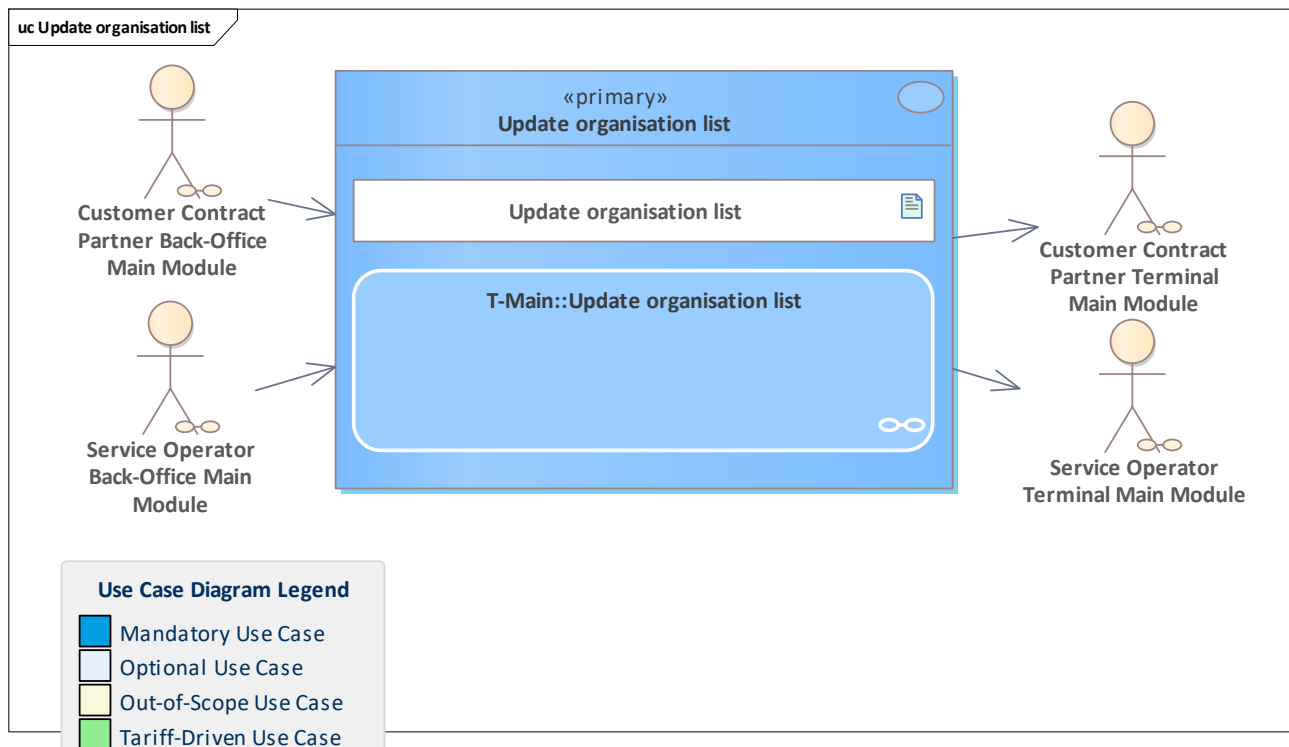


Figure 509: Update organisation list

The organisation information list is updated in the terminal by replacing the old one with new one.

11.402 Update SAM configuration

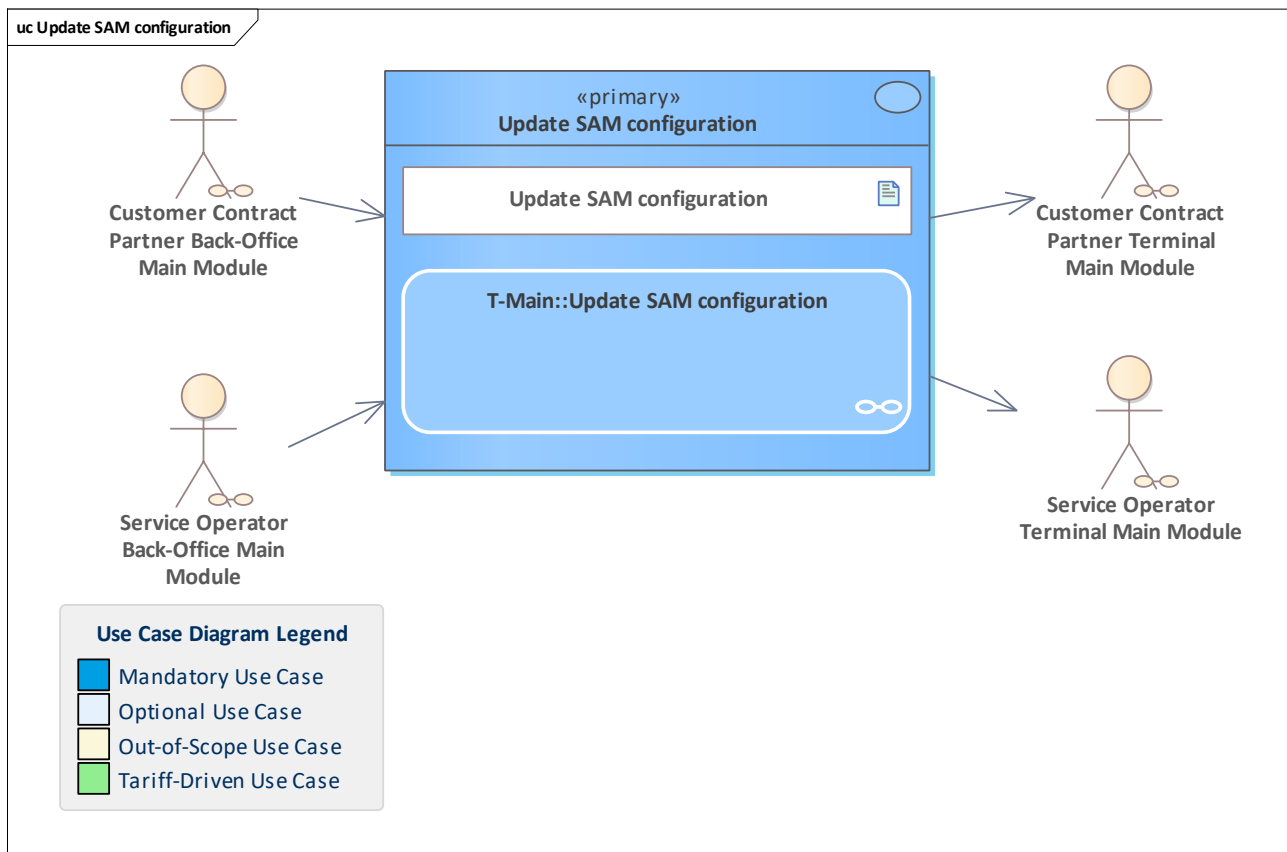


Figure 510: Update SAM configuration

The SAM configuration data is updated in the terminal.

11.403 Update SAM hotlist inventory

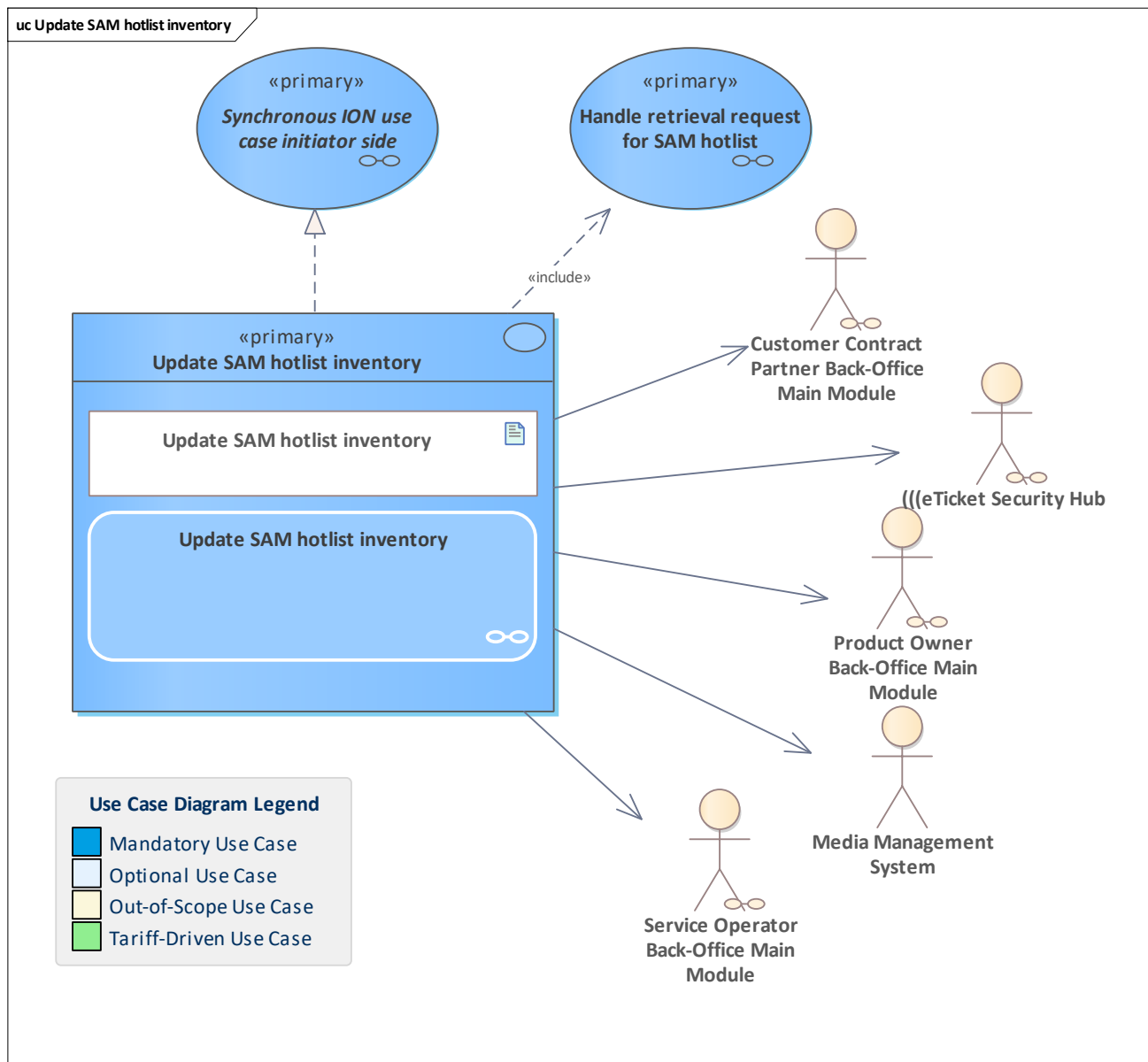


Figure 511: Update SAM hotlist inventory

The SO, CCP, PO and the scheme manager's ESH and MMS want to update their SAM hotlist inventory by retrieving the current SAM hotlist from the hotlist service system and processing it into the SAM hotlist inventory.

11.404 Update SAM reset data

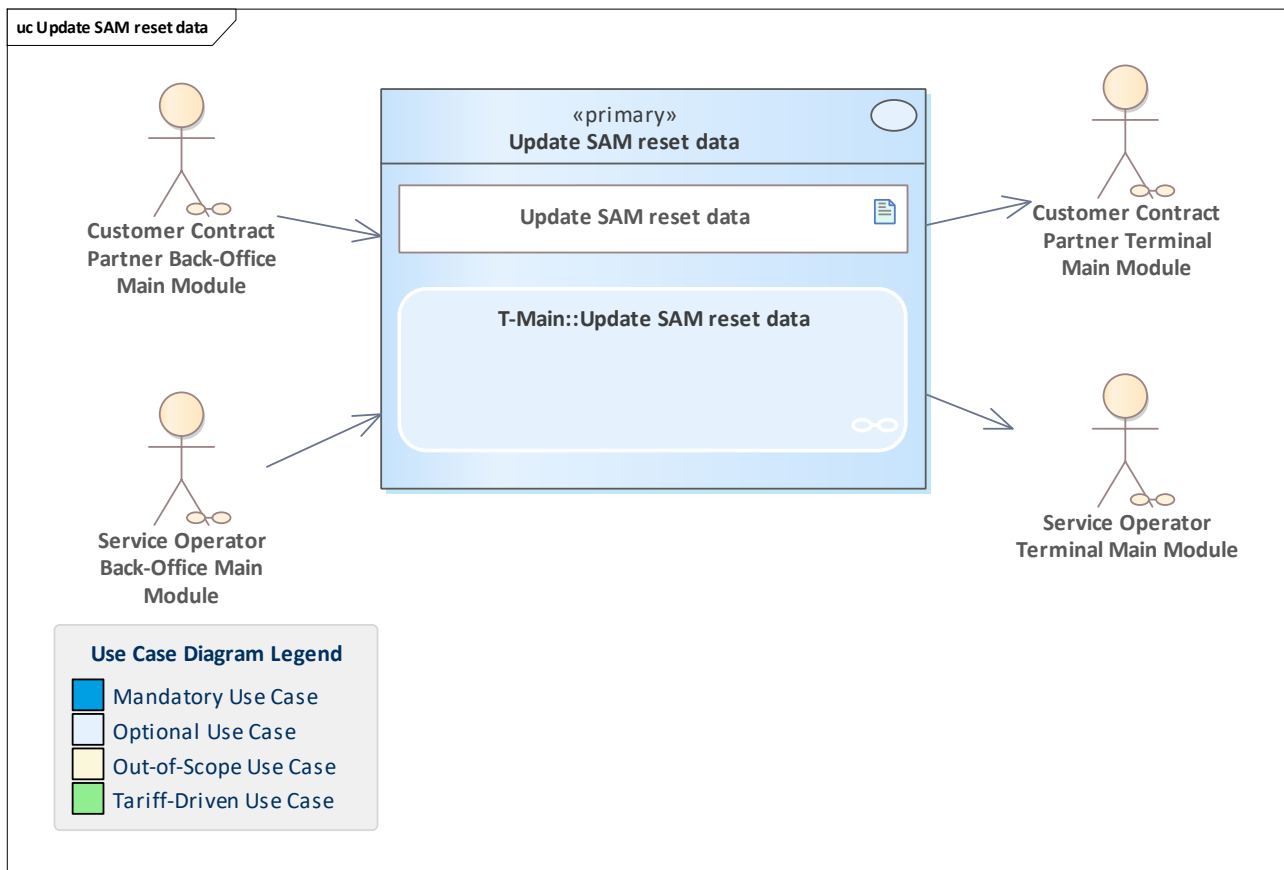


Figure 512: Update SAM reset data

The SAM reset data is updated in the terminal. Please note that after the SAM is reset, it cannot be used any more in (((etiCORE processes without being re-configured.

11.405 Update tariff module

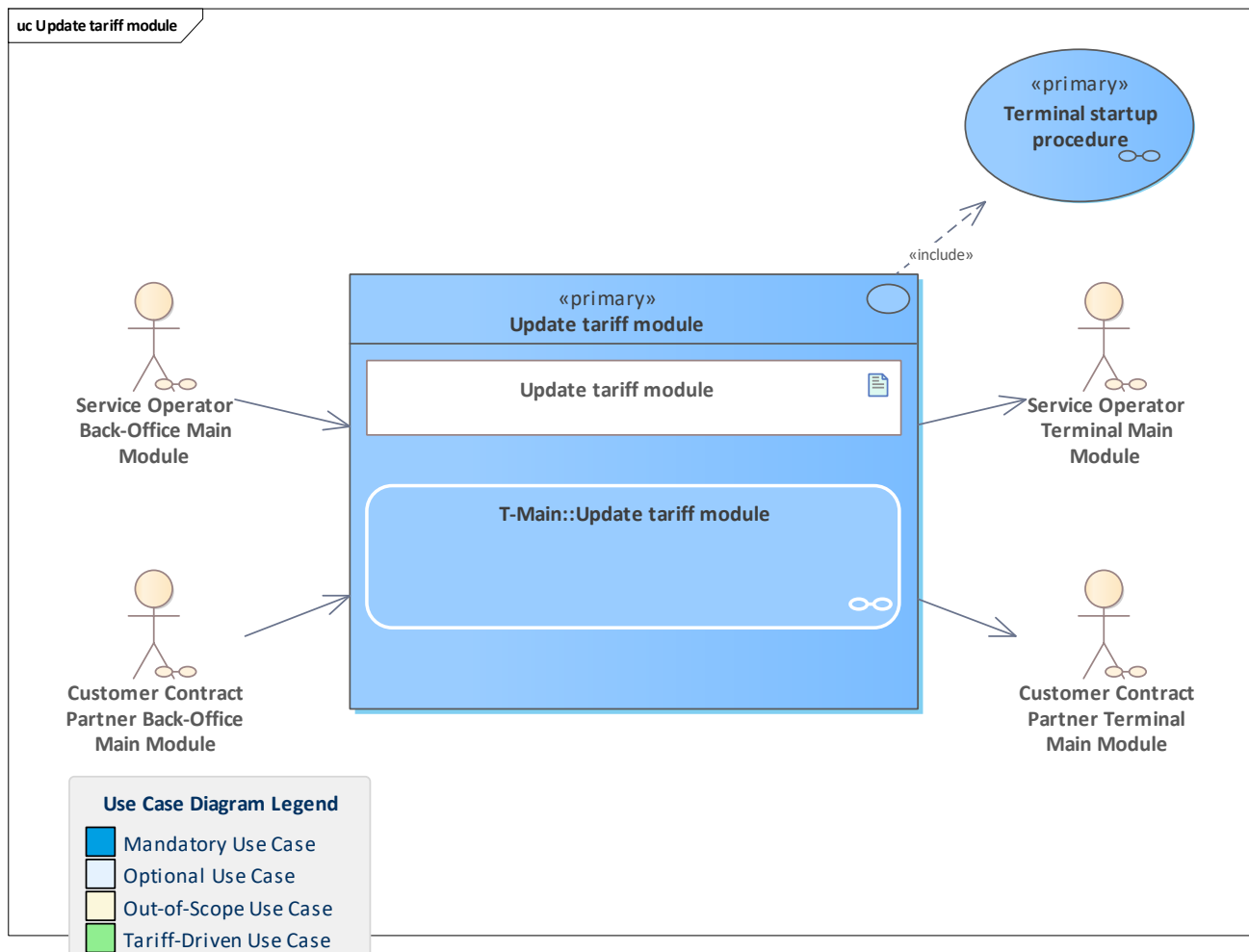


Figure 513: Update tariff module

Updates the tariff module in the terminal.

11.406 Update terminal action list

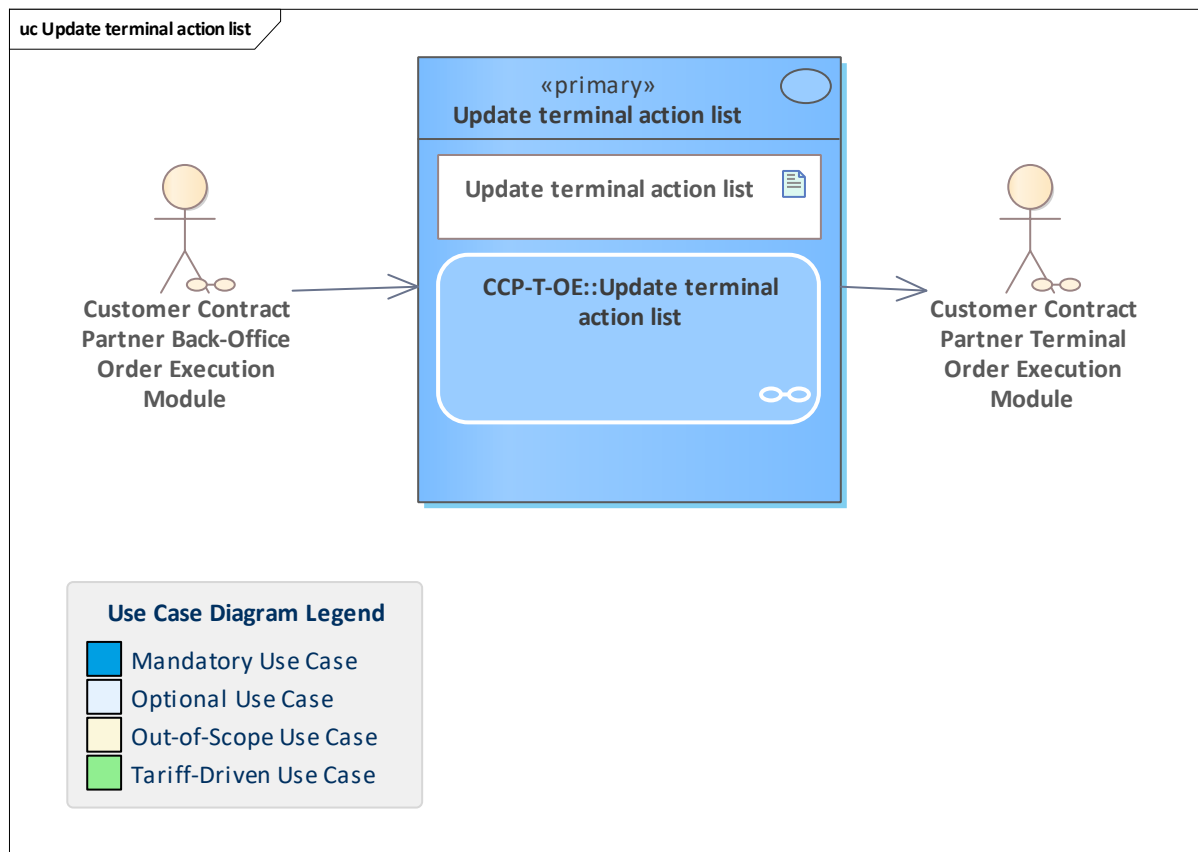


Figure 514: Update terminal action list

The executing CCP back-office system stores the latest action list in the terminal.

11.407 Update terminal hotlists

11.408 Update terminal hotlists

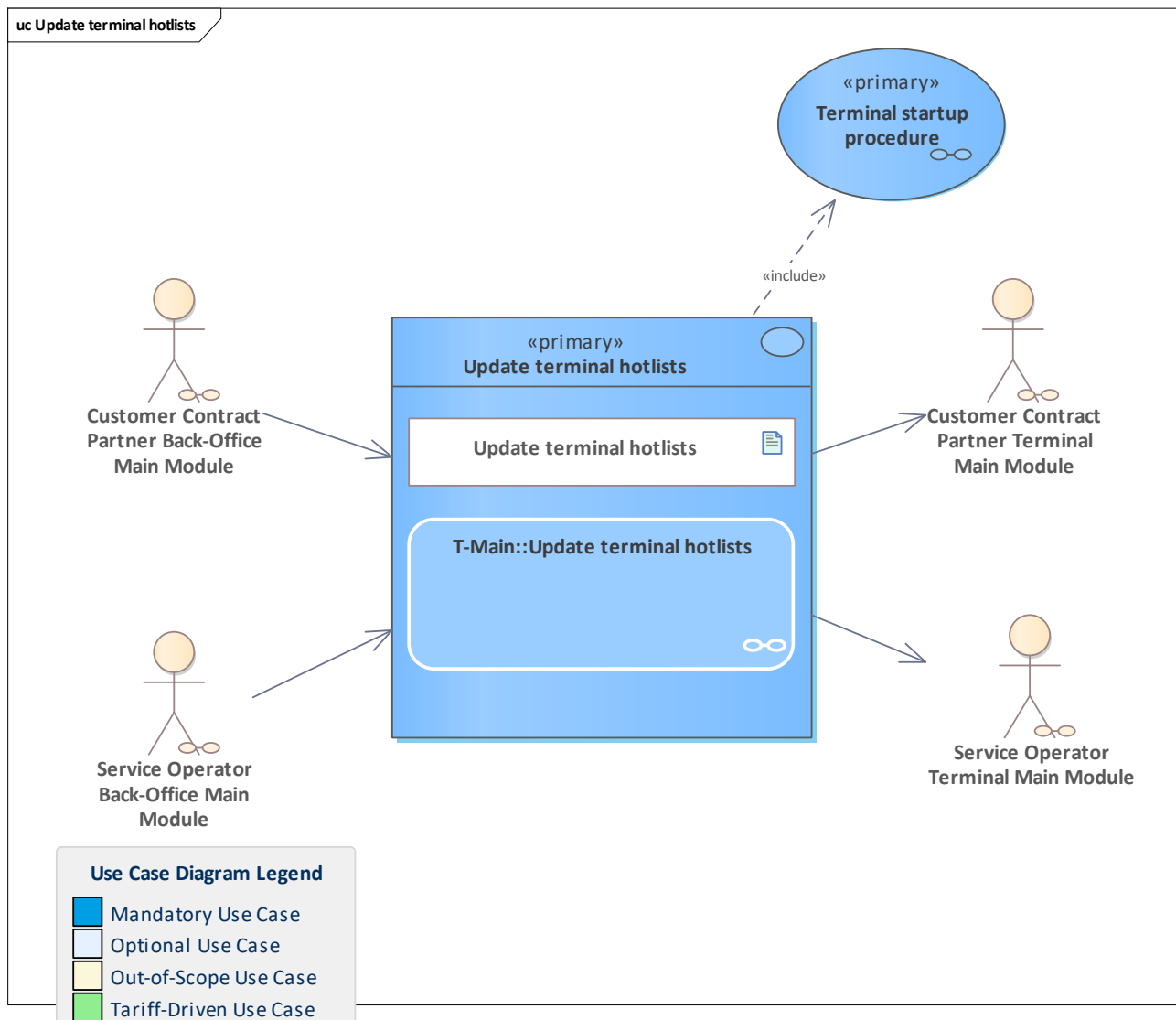


Figure 515: Update terminal hotlists

Use case for terminal operators (CCP or SO). Once per cycle, the hotlists on the terminals have to be updated or replaced with new ones.

Depending on the terminal architecture, after the update, the terminal startup process has to be performed.

Please note that it is assumed that all hotlists (entitlement, application, SAM, authentication key and organisation hotlist) are gathered together and pushed in one operation to a terminal.

Retrieval of each hotlist contains its process instance ID. Pushing all hotlists into the terminal takes a new process instance ID.

Furthermore, it is assumed that the incremental hotlists are integrated into the hotlist inventory of the back-office system and the updated status is transferred to the terminal(s). Applying the incremental hotlists would also be possible directly in the terminal. This is not described here.

11.409 Validate electronic ticket

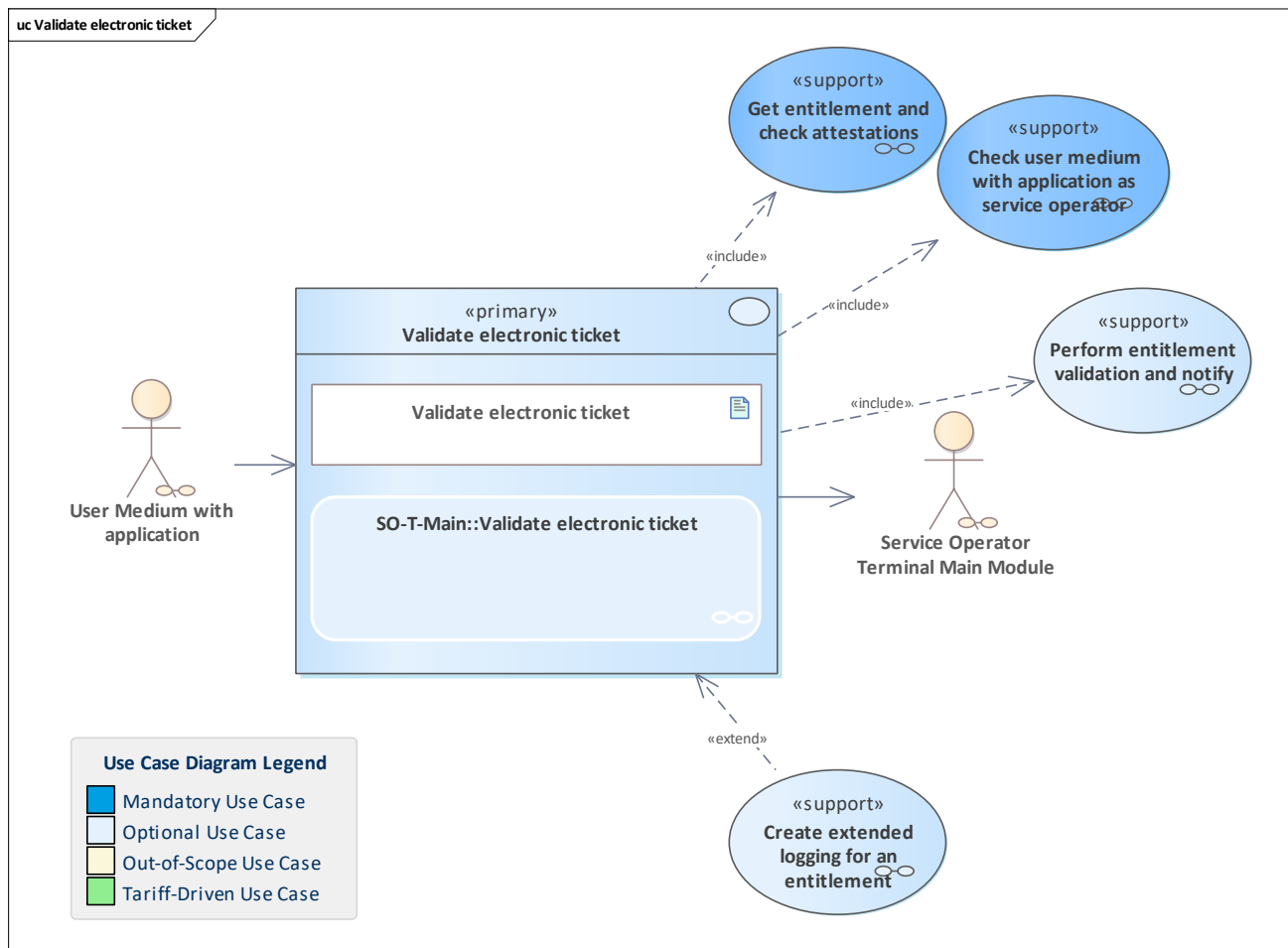


Figure 516: Validate electronic ticket

The SO terminal initiates an electronic ticket validation. These tickets belong to a special tariff so that the ticket is only valid if a validation attestation exists. This attestation is created and written to the entitlement on the user medium. The validation is executed and the responsible SO back-office system is notified.

11.410 Verify action list updated via increments

11.411 Verify action list updated via increments

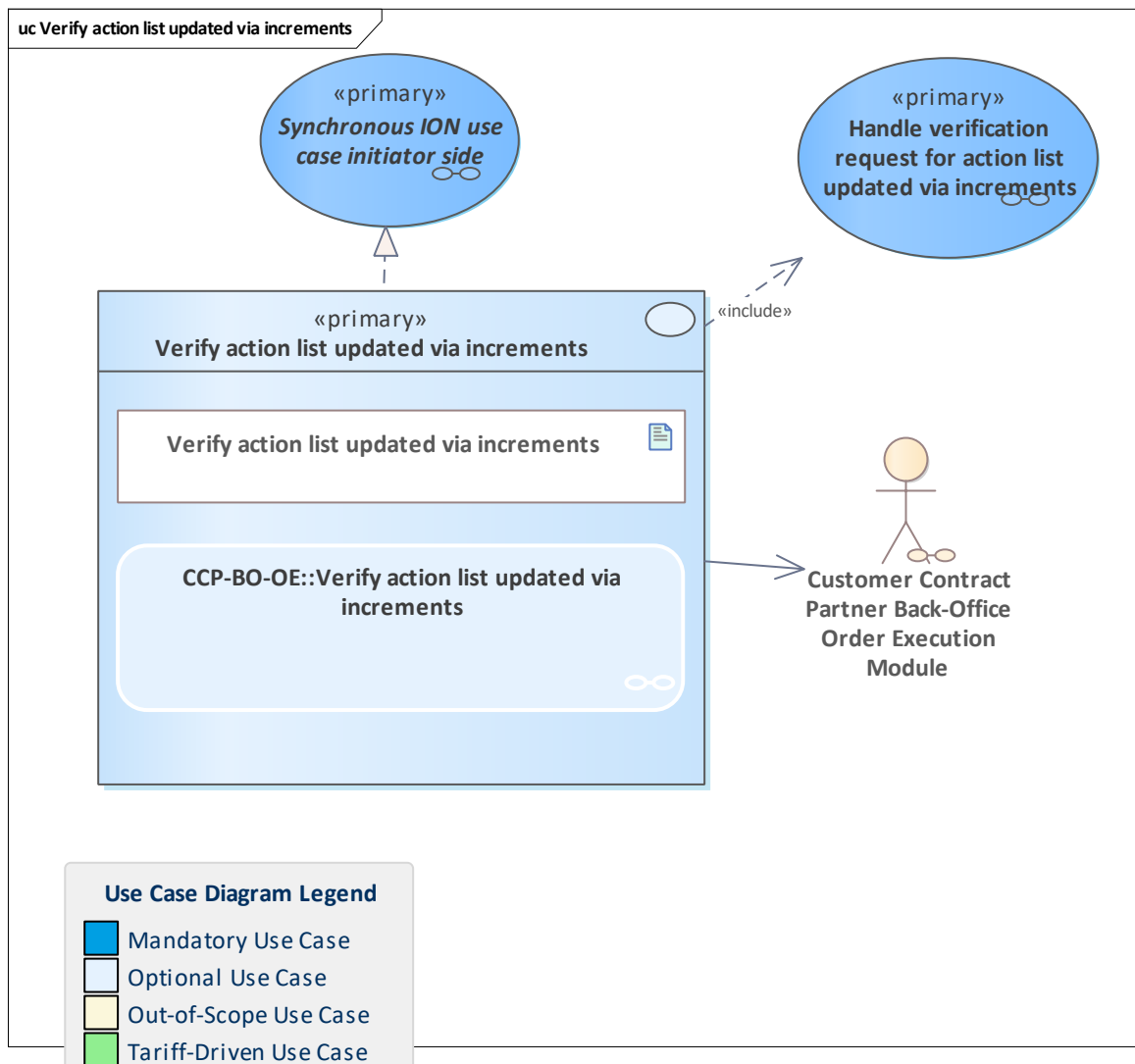


Figure 517: Verify action list updated via increments

The [Customer Contract Partner Back-Office Order Execution Module](#) verifies that its inventory is consistent with the full list by verifying the checksum after the inventory is updated via increments.

See also [Checksum calculation for hotlist and action list verification](#) and [Example calculation for an action list inventory](#).

11.412 Verify application hotlist updated via increments

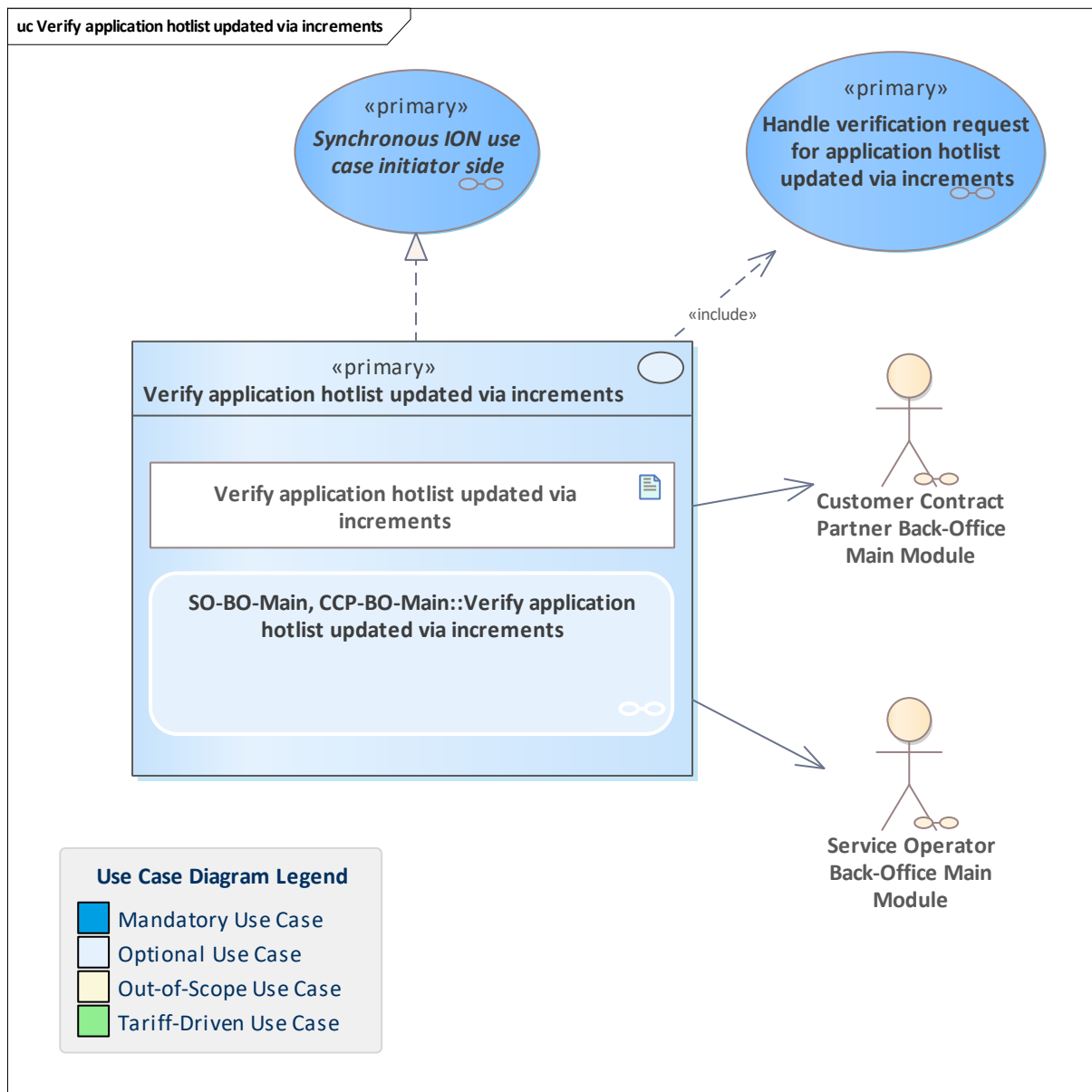


Figure 518: Verify application hotlist updated via increments

After updating the application hotlist inventory via the last incremental application hotlist, the participant must ensure that the updated hotlist inventory is the same as the hotlist inventory of the hotlist service system. For that reason, participants compute the checksum of all the application instance IDs in their inventory and send it to the hotlist service system to verify the value of the checksum. See also [Checksum calculation for hotlist and action list verification](#) and [Example calculation for an application hotlist inventory](#).

11.413 Verify entitlement hotlist updated via increments

11.414 Verify entitlement hotlist updated via increments

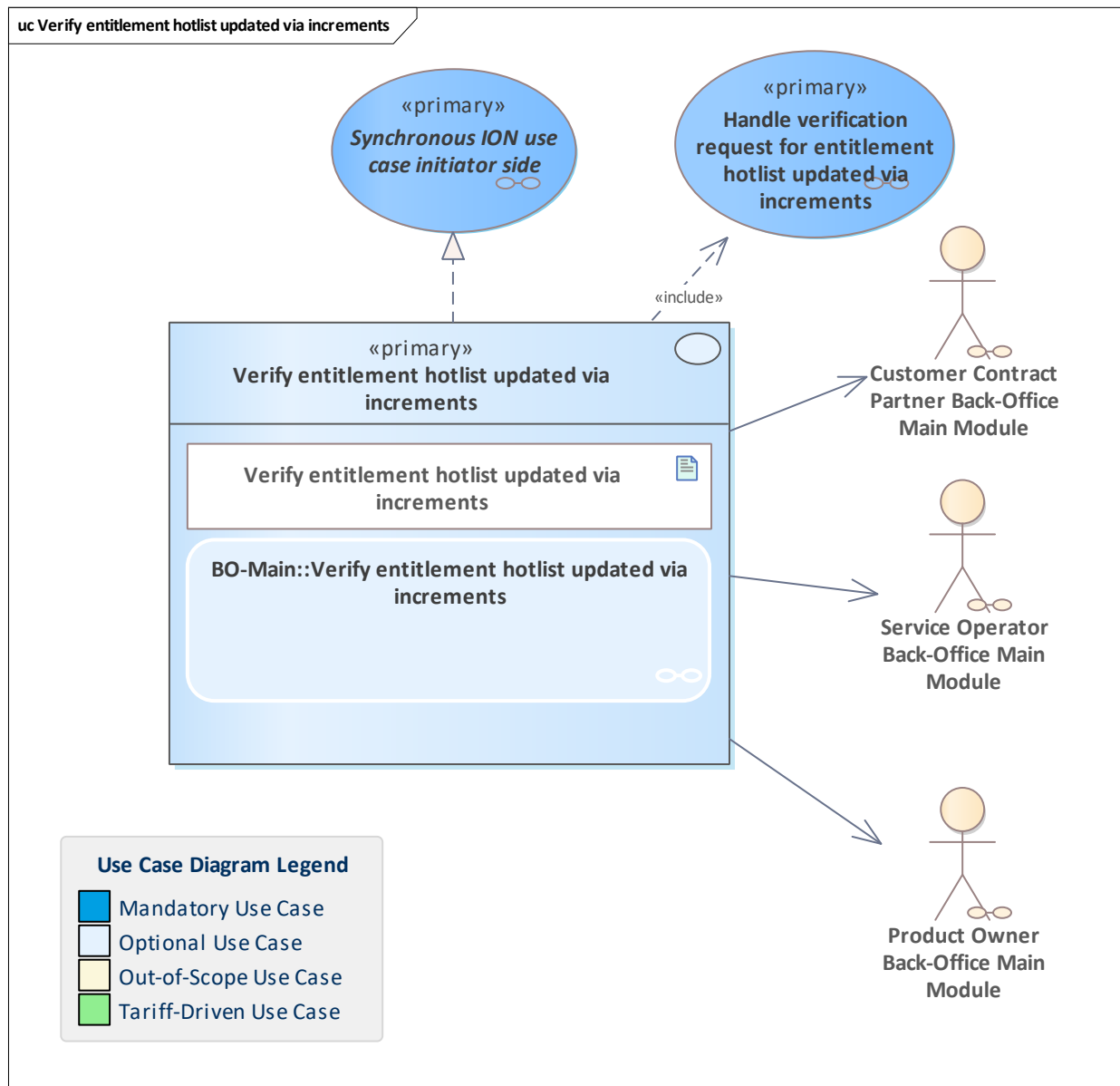


Figure 519: Verify entitlement hotlist updated via increments

After updating the entitlement hotlist inventory via the last incremental entitlement hotlist, the participant must ensure that the updated hotlist inventory is the same as the hotlist inventory of the hotlist service system (filtered to the participant's products). For that reason, participants compute the checksum of all the entitlement IDs in their inventory and send it to the hotlist service system to verify the value of the checksum. See also [Checksum calculation for hotlist and action list verification](#) and [Example calculation for an entitlement hotlist inventory](#).

11.415 Verify password

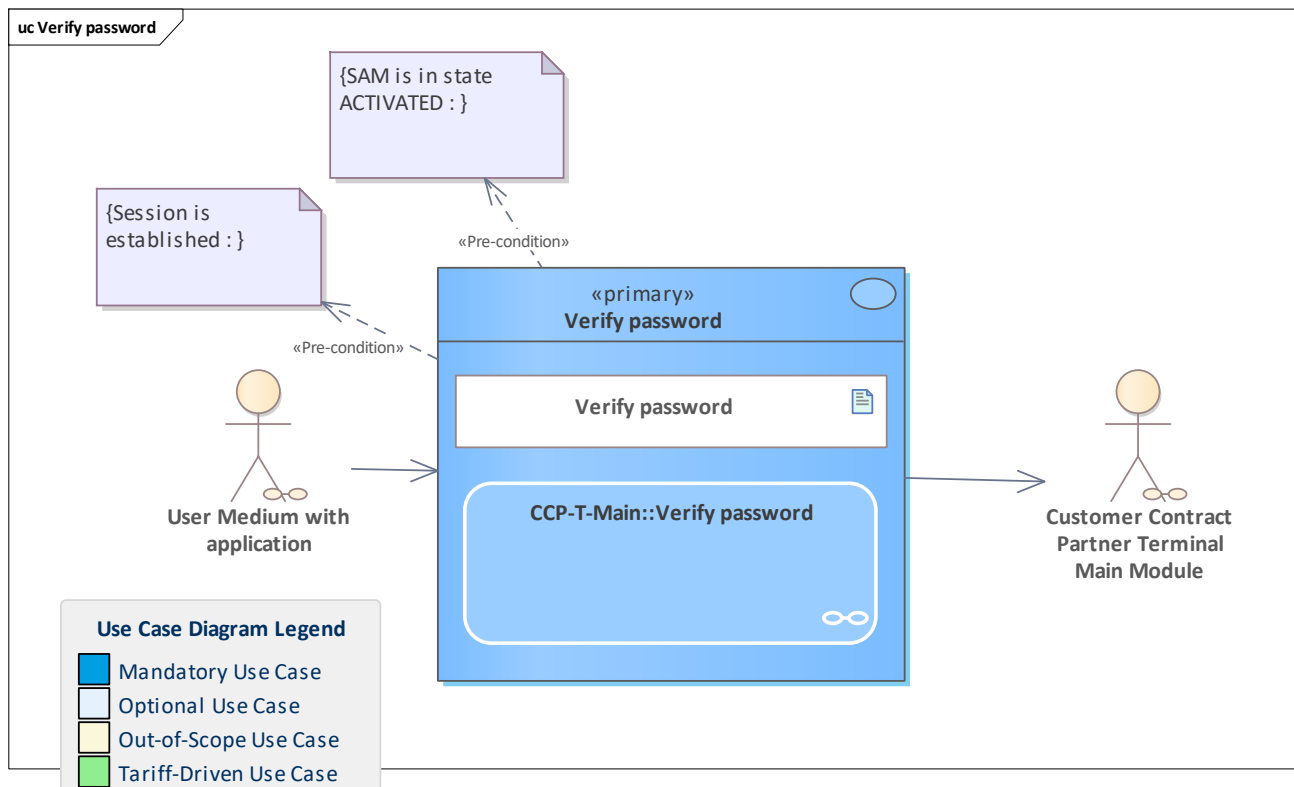


Figure 520: Verify password

A customer enters her/his password and it will be verified by the user medium with application.

11.416 Write customer

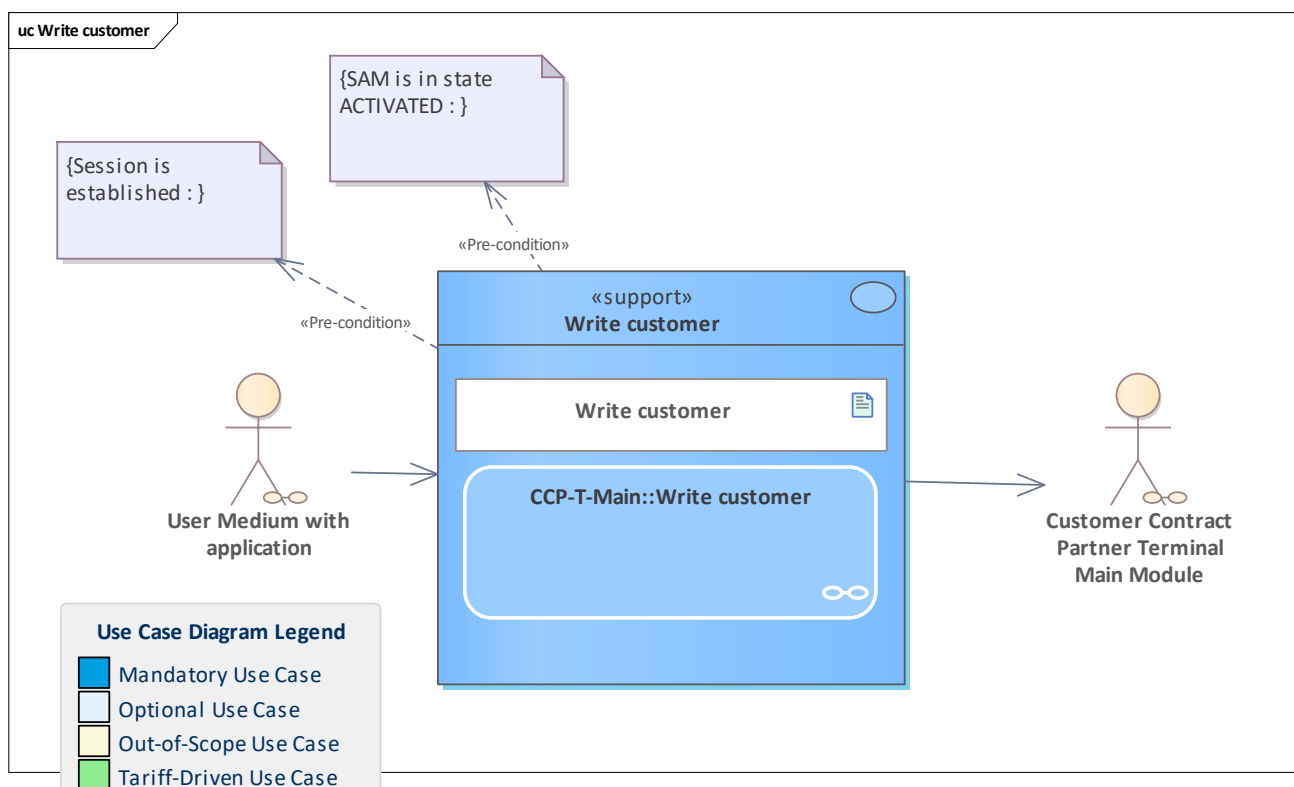


Figure 521: Write customer

Write the customer data object to the user medium with an application.

11.417 Write discounts

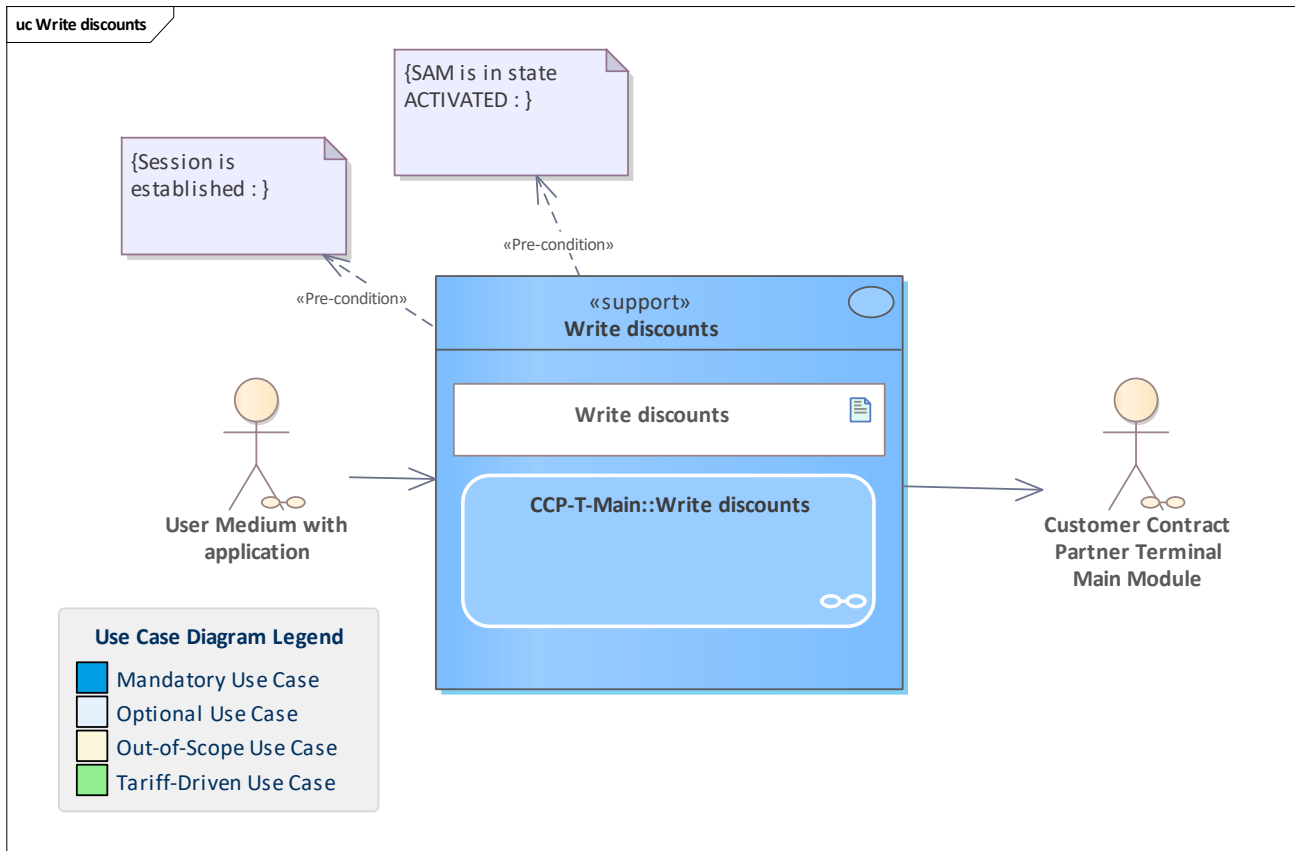


Figure 522: Write discounts

Write the discounts data object to the user medium with an application.

11.418 Write favourites

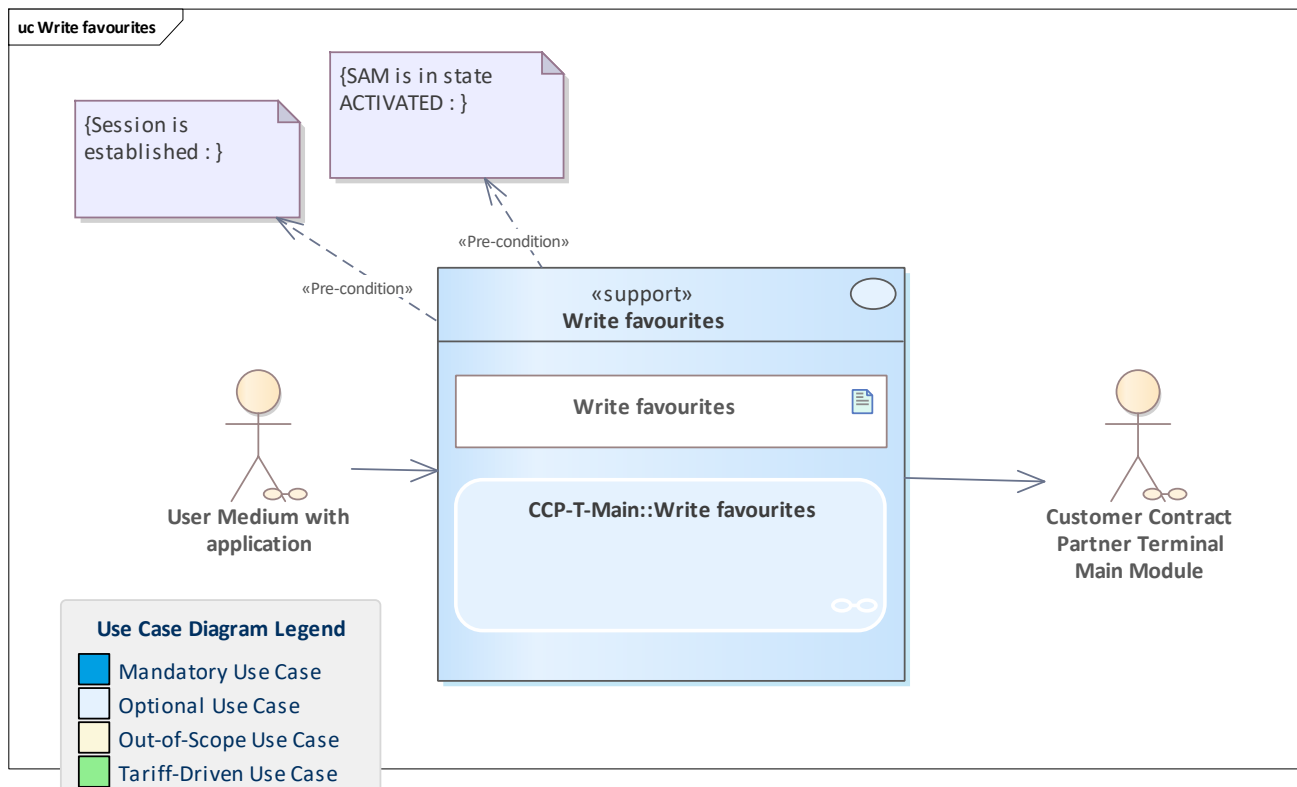


Figure 523: Write favourites

Write the favourites data object to the user medium with an application.

11.419 Write password configuration

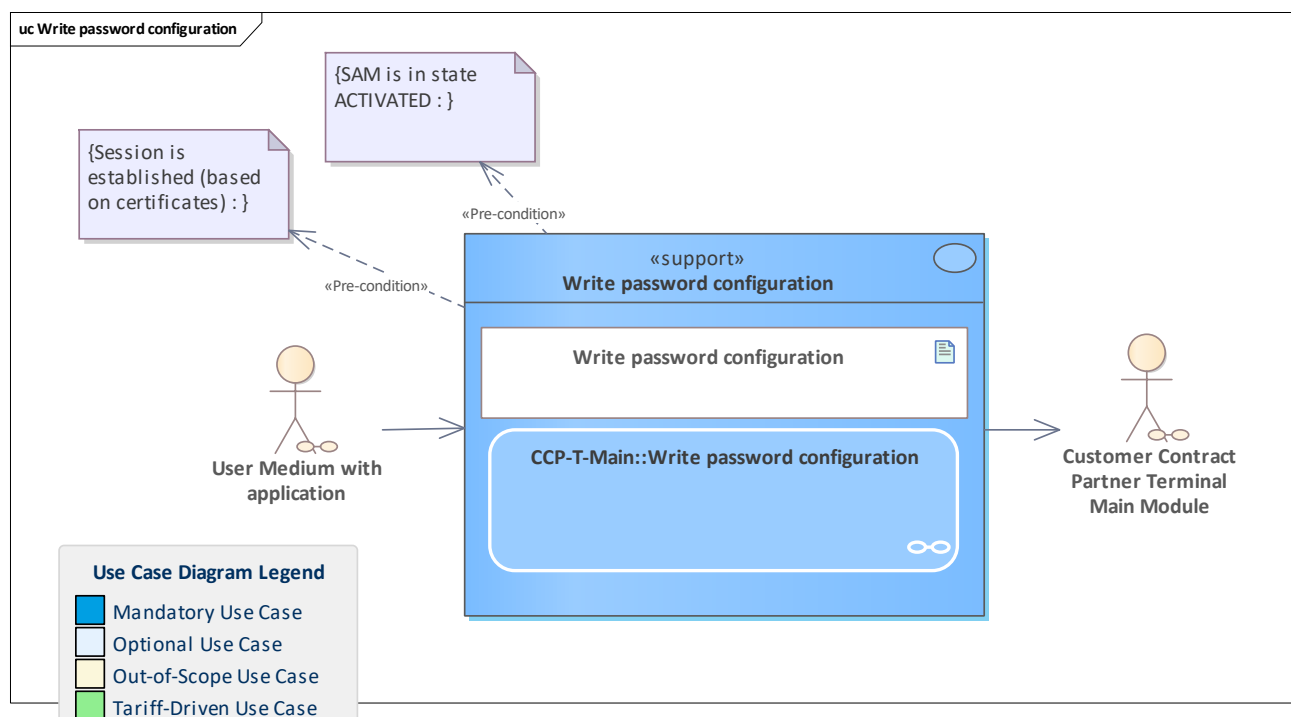


Figure 524: Write password configuration

The terminal writes the password configuration (user authentication data) to the user medium application.



12 Data Protection

Package in package 'Spec Main'

The rules described in [Data Protection Rules](#) apply to data protection.

Data Protection
Version 1.0 Phase 1.0 Proposed
Wilk Hoffmann created on 13.09.2023. Last modified 24.10.2023

13 List of References

etiCORE User Medium Specification

etiCORE User Medium Individualisation Specification

etiCORE Secure Application Module Specification

etiCORE Secure Application Module Individualisation Specification

etiCORE M2M Certificates Specification

etiCORE Cipher Suite Specification

etiCORE Media PKI Specification

This is the documentation of the LDAP service provided by the 2GSI PKI.

etiCORE Motics Specification

See <https://modell.eticket-deutschland.de/LATEST/asn.1/motics.html>

Data Protection Rules

See data protection rules due to the German DSGVO, synonymous with the GDPR in Europe.

These rules are described (only in German available) in
"Elektronisches Fahrgeldmanagement (EFM) -
Datenschutzrechtliche Grundanforderungen, Anlage Datenschutz".

14 Glossary

Contain abbreviations and terminology of terms used in the etiCORE specification model.

14.1 BPMN Terms

Terms used in BPMN

14.1.1 Abstract Process

A Process that represents the interactions between a private business process and another process or participant.

14.1.2 Activity

Work that a company or organisation performs using business processes. An activity can be atomic or non-atomic (compound). The types of activities that are a part of a Process Model are: Process, Sub-Process, and Task.

14.1.3 Artifact

A graphical object that provides supporting information about the Process or elements within the Process. However, it does not directly affect the flow of the Process.

14.1.4 Association

A connecting object that is used to link information and Artifacts with Flow Objects. An association is represented as a dotted graphical line with an arrowhead to represent the direction of flow.

14.1.5 Atomic Activity

An activity not broken down to a finer level of Process Model detail. It is a leaf in the tree-structure hierarchy of Process activities. Graphically it will appear as a Task in BPMN.

14.1.6 BPM

Business Process Management.

14.1.7 BPM System

The technology that enables BPM.

14.1.8 BPMN

Business Process Modelling Notation.

14.1.9 Business Analyst

A specialist who analyzes business needs and problems, consults with users and stakeholders to identify opportunities for improving business return through information technology, and defines, manages, and monitors the requirements into business processes.

14.1.10 Business Process

A defined set of business activities that represent the steps required to achieve a business objective. It includes the flow and use of information and resources.

14.1.11 Business Process Management

The services and tools that support process management (for example, process analysis, definition, processing, monitoring and administration), including support for human and application-level interaction. BPM tools can eliminate manual processes and automate the routing of requests between departments and applications.

14.1.12 Choreography

An ordered sequence of B2B message exchanges between two or more Participants. In a Choreography there is no central controller, responsible entity, or observer of the Process.

14.1.13 Collaboration

Collaboration is the act of sending messages between any two Participants in a BPMN model. The two Participants represent two separate BPML processes.

14.1.14 Collapsed Sub-Process

A Sub-Process that hides its flow details. The Collapsed Sub-Process object uses a marker to distinguish it as a Sub-Process, rather than a Task. The marker is a small square with a plus sign (+) inside.

14.1.15 Compensation Flow

Flow that defines the set of activities that are performed while the transaction is being rolled back to compensate for activities that were performed during the Normal Flow of the Process. A Compensation Flow can also be called from a Compensate End or Intermediate Event.

14.1.16 Compound Activity

An activity that has detail that is defined as a flow of other activities. It is a branch (or trunk) in the tree-structure hierarchy of Process activities. Graphically, it will appear as a Process or Sub-Process in BPMN.

14.1.17 Controlled Flow

Flow that proceeds from one Flow Object to another, via a Sequence Flow link, but is subject to either conditions or dependencies from other flow as defined by a Gateway. Typically, this is seen as a Sequence flow between two activities, with a conditional indicator (mini-diamond) or a Sequence Flow connected to a Gateway.

14.1.18 Decision

A gateway within a business process where the Sequence Flow can take one of several alternative paths. Also known as "Or-Split."

14.1.19 End Event

An Event that indicates where a path in the process will end. In terms of Sequence Flows, the End Event ends the flow of the Process, and thus, will not have any outgoing Sequence Flows. An End Event can have a specific Result that will appear as a marker within the center of the End Event shape. End Event Results are Message, Error, Compensation, Signal, Link, and Multiple. The End Event shares the same basic shape of the Start Event and Intermediate Event, a circle, but is drawn with a thick single line.

14.1.20 Event Context

An Event Context is the set of activities that can be interrupted by an exception (Intermediate Event). This can be one activity or a group of activities in an expanded Sub-Process.

14.1.21 Exception

An event that occurs during the performance of the Process that causes a diversion from the Normal Flow of the Process. Exceptions can be generated by Intermediate Events, such as time, error, or message.

14.1.22 Exception Flow

A Sequence Flow path that originates from an Intermediate Event attached to the boundary of an activity. The Process does not traverse this path unless the Activity is interrupted by the triggering of a boundary Intermediate Event (an Exception - see above).

14.1.23 Expanded Sub-Process

A Sub-Process that exposes its flow detail within the context of its Parent Process. An Expanded Sub-Process is displayed as a rounded rectangle that is enlarged to display the Flow Objects within.

14.1.24 Flow

A directional connector between elements in a Process, Collaboration, or Choreography. A Sequence Flows represents the sequence of Flow Objects in a Process or Choreography. A Message Flow represents the transmission of a Message between Collaboration Participants. The

term Flow is often used to represent the overall progression of how a Process or Process segment would be performed.

14.1.25 Flow Object

A graphical object that can be connected to or from a Sequence Flow. In a Process, Flow Objects are Events, Activities, and Gateways. In a Choreography, Flow Objects are Events, Choreography Activities, and Gateways.

14.1.26 Fork

A point in the Process where one Sequence Flow path is split into two or more paths that are run in parallel within the Process, allowing multiple activities to run simultaneously rather than sequentially. BPMN uses multiple outgoing Sequence Flows from Activities or Events or a Parallel Gateway to perform a Fork. Also known as "AND-Split."

14.1.27 Intermediate Event

An event that occurs after a Process has been started. An Intermediate Event affects the flow of the process by showing where messages and delays are expected, distributing the Normal Flow through exception handling, or showing the extra flow required for compensation. However, an Intermediate Event does not start or directly terminate a process. An Intermediate Event is displayed as a circle, drawn with a thin double line.

14.1.28 Join

A point in the Process where two or more parallel Sequence Flow paths are combined into one Sequence Flow path. BPMN uses a Parallel Gateway to perform a Join. Also known as "AND-Join."

14.1.29 Lane

A partition that is used to organize and categorize activities within a Pool. A Lane extends the entire length of the Pool either vertically or horizontally. Lanes are often used for such things as internal roles (e.g., Manager, Associate), systems (e.g., an enterprise application), or an internal department (e.g., shipping, finance).

14.1.30 Merge

A point in the Process where two or more alternative Sequence Flow paths are combined into one Sequence Flow path. No synchronization is required because no parallel activity runs at the join point. BPMN uses multiple incoming Sequence Flows for an Activity or an Exclusive Gateway to perform a Merge. Also know as "OR-Join."

14.1.31 Message

An Object that depicts the contents of a communication between two Participants. A message is transmitted through a Message Flow and has an identity that can be used for alternative branching of a Process through the Event-Based Exclusive Gateway.

14.1.32 Message Flow

A Connecting Object that shows the flow of messages between two Participants. A Message Flow is represented by a dashed lined.

14.1.33 Normal Flow

A flow that originates from a Start Event and continues through activities on alternative and parallel paths until reaching an End Event.

14.1.34 Parent Process

A Process that holds a Sub-Process within its boundaries.

14.1.35 Participant

A business entity (e.g., a company, company division, or a customer) or a business role (e.g., a buyer or a seller) that controls or is responsible for a business process. If Pools are used, then a Participant would be associated with one Pool. In a Collaboration, Participants are informally known as "Pools."

14.1.36 Pool

A Pool represents a Participant in a Collaboration. Graphically, a Pool is a container for partitioning a Process from other Pools/Participants. A Pool is not required to contain a Process, i.e., it can be a "black box."

14.1.37 Private Business Process

A process that is internal to a specific organisation and is the type of process that has been generally called a workflow or BPM process.

14.1.38 Process

A sequence or flow of Activities in an organisation with the objective of carrying out work. In BPMN, a Process is depicted as a graph of Flow Elements, which are a set of Activities, Events, Gateways, and Sequence Flow that adhere to a finite execution semantics.

14.1.39 Result

The consequence of reaching an End Event. Types of Results include Message, Error, Compensation, Signal, Link, and Multiple.

14.1.40 Sequence Flow

A connecting object that shows the order in which activities are performed in a Process and is represented with a solid graphical line. Each Flow has only one source and only one target. A Sequence Flow can cross the boundaries between Lanes of a Pool but cannot cross the boundaries of a Pool.

14.1.41 Start Event

An Event that indicates where a particular Process starts. The Start Event starts the flow of the Process and does not have any incoming Sequence Flow, but can have a Trigger. The Start Event is displayed as a circle, drawn with a single thin line.

14.1.42 Sub-Process

A Process that is included within another Process. The Sub-Process can be in a collapsed view that hides its details. A Sub-Process can be in an expanded view that shows its details within the view of the Process that it is contained in. A Sub-Process shares the same shape as the Task, which is a rectangle that has rounded corners.

14.1.43 Swimlane

A Swimlane is a graphical container for partitioning a set of activities from other activities. BPMN has two different types of Swimlanes. See "Pool" and "Lane."

14.1.44 Task

An atomic activity that is included within a Process. A Task is used when the work in the Process is not broken down to a finer level of Process Model detail. Generally, an end-user, an application, or both will perform the Task. A Task object shares the same shape as the Sub-Process, which is a rectangle that has rounded corners.

14.1.45 Token

A theoretical concept that is used as an aid to define the behavior of a Process that is being performed. The behavior of Process elements can be defined by describing how they interact with a token as it "traverses" the structure of the Process. For example, a token will pass through an Exclusive Gateway, but continue down only one of the Gateway's outgoing Sequence Flow.

14.1.46 Transaction

A Sub-Process that represents a set of coordinated activities carried out by independent, loosely-coupled systems in accordance with a contractually defined business relationship. This coordination leads to an agreed, consistent, and verifiable outcome across all participants.

14.1.47 Trigger

A mechanism that detects an occurrence and can cause additional processing in response, such as the start of a business Process. Triggers are associated with Start Events and Intermediate Events and can be of the type: Message, Timer, Conditional, Signal, Link, and Multiple.

14.1.48 Uncontrolled Flow

Flow that proceeds without dependencies or conditional expressions. Typically, an Uncontrolled Flow is a Sequence Flow between two Activities that do not have a conditional indicator (mini-diamond) or an intervening Gateway.





14.2 Common Terms

Commons terms used in the etiCORE specification model.

14.2.1 APDU

Application Protocol Data Unit. Smart card application protocol.

14.2.2 API

Application Programming Interface

14.2.3 CSR

Certificate Signing Request

14.2.4 EFM

Electronic Fare Management

14.2.5 IN-OUT Payment Method

The IN-OUT Payment Method allows the customer to make use of services in PT. It describes how the use of a service is settled financially. It serves also - besides of its use in IN-OUT-Systems - as an electronic payment method and can be used for payment of PT services in KA-Systems.

14.2.6 LDAP

Lightweight Directory Access Protocol

14.2.7 MTOM

SOAP Message Transmission Optimisation Mechanism <https://www.w3.org/TR/soap12-mtom>

14.2.8 NFC

Near Field Communication

14.2.9 OCSP

Online Certificate Status Protocol

14.2.10 PKI

Public Key Infrastructure

14.2.11 RSA

RSA (Rivest-Shamir-Adleman) is an asymmetric cryptographic procedure

14.2.12 SAM

Secure Application Module

14.2.13 SAMs

Secure Application Modules

14.2.14 SLA

Service Level Agreement

14.2.15 SM

Secure Messaging

14.2.16 TLS

Transport Layer Security

14.2.17 TLV

Tag, Length, Value

14.2.18 UM

User Medium

14.2.19 UML

Unified Modelling Language.

14.2.20 Use case

Term used in the UML to define a certain functionality performed by a dedicated system.

14.2.21 WS

Web Service



14.2.22 WS-I

Web service interoperability. Rules to ease the collaboration of different web service frameworks.

14.2.23 WSDL

Web Service Definition Language.

14.2.24 WSS

Webservice Security

14.3 EN 24014 Terms

Terms from EN/ISO 24014 used in the etiCORE specification model.

14.3.1 AO

Application Owner

14.3.2 Application Template

Executable technical pattern of the application specification

14.3.3 Commercial Rules

Rules defining the settlement and commission within the [Interoperable Fare Management System](#)

14.3.4 Customer Medium

Medium initialised with an application through an application contract

14.3.5 IFM

Shortcut for Interoperable Fare Management. It enables seamless travel with contactless user media (smartcard, smartphone, etc.).

14.3.6 MAD

Medium Acceptance Device. Each device which reads or evaluates a User Medium.

14.3.7 Pricing Rules

Rules defining the price and payment/billing relationships to the customer

14.3.8 Product Rules

Set of usage, pricing and commercial rules defined by the product owner

14.3.9 Product Template

Technical pattern of the product specification

14.3.10 Usage Rules

Rules defining the usage time, the usage area, the personal status and the type of service





14.4 EN1545 Terms

Terms from EN1545 used in the etiCORE specification model.

14.4.1 Hotlist

List for any elements in the IFM which are corrupted, stolen or compromised in any way. Elements can be application, entitlement, SAM, company, key

14.4.2 Hotlist Service

The service which provides operations directed to the hotlist such as ordering a new entry or getting the current hotlist for distribution to MADs.



14.5 etiCORE Terms

Terms introduced in the etiCORE specification model.

14.5.1 ABPM

Account-based payment method: the customer has an account in the related CCP-S that is used for payments.

14.5.2 ALISE

Abbreviation for Action List Service

14.5.3 Basic process

A basic process in (((etiCORE consists of one or more use cases that run on distributed systems. Example: issuing an electronic ticket starts in the terminal and has to be announced to the product owner and customer contract partner.

14.5.4 BO-*

One of the back-office modules.

14.5.5 BO-Main

One of the back-office main modules: [Customer Contract Partner Back-Office Main Module](#), [Product Owner Back-Office Main Module](#) or [Service Operator Back-Office Main Module](#).

14.5.6 BO-S

Abbreviation for Back-Office System.

14.5.7 CCP

Customer Contract Partner

14.5.8 CCP-BO-AO

[Customer Contract Partner Back-Office Action Ordering Module](#)

14.5.9 CCP-BO-Main

[Customer Contract Partner Back-Office Main Module](#)



14.5.10 CCP-BO-OE

[Customer Contract Partner Back-Office Order Execution Module](#)

14.5.11 CCP-BO-STE

[Customer Contract Partner Back-Office Static Entitlement Module](#)

14.5.12 CCP-T-Main

[Customer Contract Partner Terminal Main Module](#)

14.5.13 CCP-T-OE

[Customer Contract Partner Terminal Order Execution Module](#)

14.5.14 CCP-T-STE

[Customer Contract Partner Terminal Static Entitlement Module](#)

14.5.15 Central Routing Engine

As a part of the [Interoperability Network](#) (ION), the central routing engine transmits messages of one ION participant to another, based on the recipient's [Organisation-ID](#) given in the message.

The ION participants can be the [Application Owner](#), the [Hotlist Service](#), a [Customer Contract Partner](#), a [Service Operator](#) or a [Product Owner](#). They can be connected to the central routing engine either directly or via an ION adaptor.

14.5.16 CICO

Check-in, check-out.

Since a registration has to be done when entering and leaving (de-registration) a vehicle, in etiCORE we use the abbreviation CICO. The variants be-in be-out or check-in, be-out are only technical variants and not treated separately.

14.5.17 Component

In the context of a specified (((etiCORE component. Sometimes also "system component". Any logical unit that has specific functionality and interacts with other (((etiCORE components.

These are

- User medium
- SAM
- Terminals
- Central routing engine
- Back-office systems
- Central systems such as ESH or Hotlist service



14.5.18 CRE

[Central Routing Engine](#)

14.5.19 EO

Entitlement owner. His organisation ID is part of the Entitlement ID as CCP Org ID.

14.5.20 ESH

(((eTicket Security Hub

14.5.21 ESI

External System Interface

14.5.22 etiCORE

Name of the new version of the German e-ticket standard. Composed of the prefix *eti* for e-ticket and the English word *CORE*.

14.5.23 EUI

Execution Unit Interface

14.5.24 HLS-S

Abbreviation for Hotlist Service System.

14.5.25 IN-OUT

System for IN-OUT which can be check-in & check-out, be-in & be-out or check-in & be-out. A registration has to be done when entering (IN) and leaving (OUT, de-registration) a vehicle. The kinds of (de-)registration are only technical variants and not treated separately. An IN-OUT system records trip events automatically.

14.5.26 ION

Interoperable Network

14.5.27 JSB

Joint Service Broker

14.5.28 KA

Abbreviation for Kernapplikation

14.5.29 Layer

Layer in the specification model. It defines how deep the specification elements dive into technical aspects.

- Layer 1 is a kind of overview for the basic processes.
- Layer 2 shows the workflows inside one basic process.
- Layer 3 shows the use cases and activities.
- Layer 4 shows the components and interfaces.
- Layer 5 shows certain implementation details.

14.5.30 Level-1

Test environment for certain (((etiCORE components such as user medium and SAM for development test purposes. Uses fixed test keys and certificates.

14.5.31 Level-2

Staging environment for test purposes for all (((etiCORE components: user medium, SAM, terminal, back-office systems and central systems as well as the PKI. Already fitted with the complete security features, but using the staging PKI.

14.5.32 Level-3

Production environment for all (((etiCORE components: user medium, SAM, terminal, back-office systems and central systems. Fitted with the complete security features, using the production PKI.

14.5.33 MMS

Media Management System

14.5.34 MOTICS

Mobile Ticketing Crypto Service.

Central system for certificate administration and copy protection container for electronic tickets for mobile devices at the same time.

14.5.35 OA

Abbreviation for ordered action

14.5.36 Ordered action

An action to be performed on a user medium. The action can be ordered in a product owner system with the action order service. This system distributes action lists to executing customer contract partners that store these actions in the dedicated terminals.



14.5.37 Organisational unit

Combination of organisation and role

14.5.38 pCCP

primary Customer Contract Partner

14.5.39 pCCP-BO-Main

[Primary Customer Contract Partner Back-Office Main Module](#)

14.5.40 pCCP-BO-STE

[Primary Customer Contract Partner Back-Office Static Entitlement Module](#)

14.5.41 PO

Product Owner

14.5.42 PO-BO-AM

[Product Owner Back-Office Action Management Module](#)

14.5.43 PO-BO-Main

[Product Owner Back-Office Main Module](#)

14.5.44 PO-BO-STE

[Product Owner Back-Office Static Entitlement Module](#)

14.5.45 sCCP

Secondary Customer Contract Partner

14.5.46 sCCP-BO-Main

[Secondary Customer Contract Partner Back-Office Main Module](#)

14.5.47 SCE

Secure Crypto Element.

Unique ID in the scope of a MOTICS app that identifies a mobile device. Has the same data structure as an [AppInstanceId](#).



14.5.48 SO

Service Operator

14.5.49 SO-BO-Main

[Service Operator Back-Office Main Module](#)

14.5.50 SO-BO-STE

[Service Operator Back-Office Static Entitlement Module](#)

14.5.51 SO-T-Main

[Service Operator Terminal Main Module](#)

14.5.52 SO-T-STE

[Service Operator Terminal Static Entitlement Module](#)

14.5.53 Static Entitlement

Entitlement to use public transport, normally issued as an electronic ticket displayed as a 2D barcode. Static in this context means, that the ticket content cannot be updated or changed.

14.5.54 STE

Static entitlement: signed entitlement fixed at the time of issuing. Cannot be changed.

14.5.55 SVPM

Stored-value payment method: a customer has a payment method directly located on the user medium, where the current value of the balance is stored.

14.5.56 T-Main

One of the terminal main modules: [Service Operator Terminal Main Module](#) or [Customer Contract Partner Terminal Main Module](#)

14.5.57 TO

Terminal Operator. An abstraction of the organisation performing actions with user media. Can be a Service Operator or a Customer Contract Partner.



14.5.58 UMO

User medium (application instance) owner. The owner of the user medium application instance. The information about the owner of a user medium is stored on the user medium and in the corresponding certificate. Normally the primary customer contract partner, pCCP.

14.5.59 User Medium

Also Customer Medium. Medium initialised with an Application through an Application Contract

14.5.60 User Medium Owner

See UMO.

14.5.61 Workflow

A workflow consists of one or more basic process instances that interact with each other.