

# SPEC-Hotlist

Last changed:	20.11.2025 10:25:49
Version:	3.0.0-rc.6
Creator:	VDV ETS

## Table of Contents

1	Hotlist Service	4
1.1	Hotlist Service System	4
2	Verifying Lists Updated via Increments	5
2.1	Checksum calculation for hotlist and action list verification	6
2.2	Example calculation for an action list inventory	7
2.3	Example calculation for an application hotlist inventory	8
2.4	Example calculation for an entitlement hotlist inventory	9
3	Hotlist Entry Management	10
3.1	Overview	10
3.2	Use Cases	10
3.2.1	Add application to hotlist	10
3.2.2	Retrieve CV certificate over signing key	13
3.2.3	Determine user medium owner	15
3.2.4	Remove application from hotlist	17
3.2.5	Add entitlement to hotlist	19
3.2.6	Remove entitlement from hotlist	21
3.2.7	Add SAM to hotlist	23
3.2.8	Determine SAM owner	25
3.2.9	Remove SAM from hotlist	27
3.2.10	Add authentication key to hotlist	29
3.2.11	Remove authentication key from hotlist	31
3.2.12	Add organisation to hotlist	33
3.2.13	Remove organisation from hotlist	35
4	Hotlist Retrieval	37
4.1	Overview	37
4.2	Use Cases	37
4.2.1	Handle retrieval request for application hotlist	37
4.2.2	Handle retrieval request for incremental application hotlist	40
4.2.3	Handle verification request for application hotlist updated via increments	42
4.2.4	Handle retrieval request for entitlement hotlist	44
4.2.5	Handle retrieval request for entitlement hotlist with product information	46
4.2.6	Handle retrieval request for incremental entitlement hotlist	48
4.2.7	Handle verification request for entitlement hotlist updated via increments	50
4.2.8	Handle retrieval request for SAM hotlist	52
4.2.9	Handle retrieval request for organisation hotlist	54
4.2.10	Handle retrieval request for authentication key hotlist	56
4.2.11	Generate current hotlists	58

5	Hotlist Service Configuration	60
5.1	Overview	60
5.2	Use Cases	60
5.2.1	Handle request to add product acceptance entry to hotlist service configuration	60
5.2.2	Handle request to remove product acceptance entry from hotlist service configuration	62
5.2.3	Handle request to remove product acceptance from participants	64
5.2.4	Handle request for product acceptance configuration list	66
5.2.5	Retrieve and distribute organisation list	68
6	Hotlist Reports	70
6.1	Overview	70
6.2	Use Cases	70
6.2.1	Handle retrieval request for unclaimed list information	70

# 1 Hotlist Service

This chapter contains all components and interfaces concerning the [Hotlist Service](#) as part of the [Scheme Manager](#).

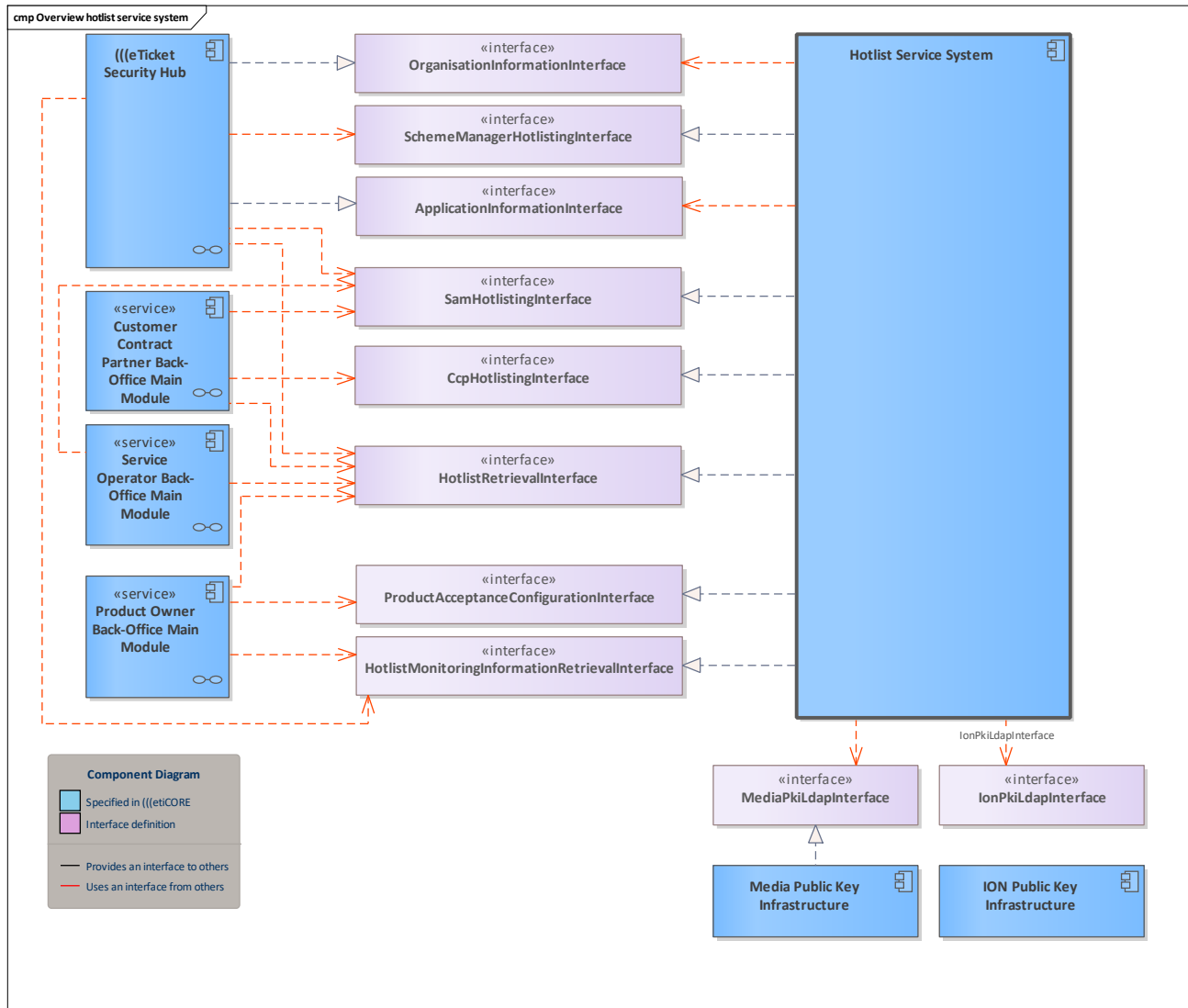


Figure 1: Overview hotlist service system

This diagram shows the interfaces which are implemented and used by the hotlist service system.

## 1.1 Hotlist Service System

System (component) which implements the necessary functionality for the [Hotlist Service](#) and implements interfaces for

- the [Application Owner Module](#),
- [Security Manager Module](#),
- [Customer Contract Partner System](#),
- [Service Operator System](#) and
- [Product Owner System](#).

## 2 Verifying Lists Updated via Increments

Incremental lists are used for large lists containing thousands of entries. These lists occur in the hotlist service for lists of hotlisted application instances and entitlements, as well as in the action list service when fetching the newest action list.

Instead of fetching a full list, an incremental list can be used. This list only contains the differences between a specified last list and the current one. These differences have to be merged into the current inventory. To verify that the updated inventory is equal to the inventory of a full list, a certain checksum algorithm is used.

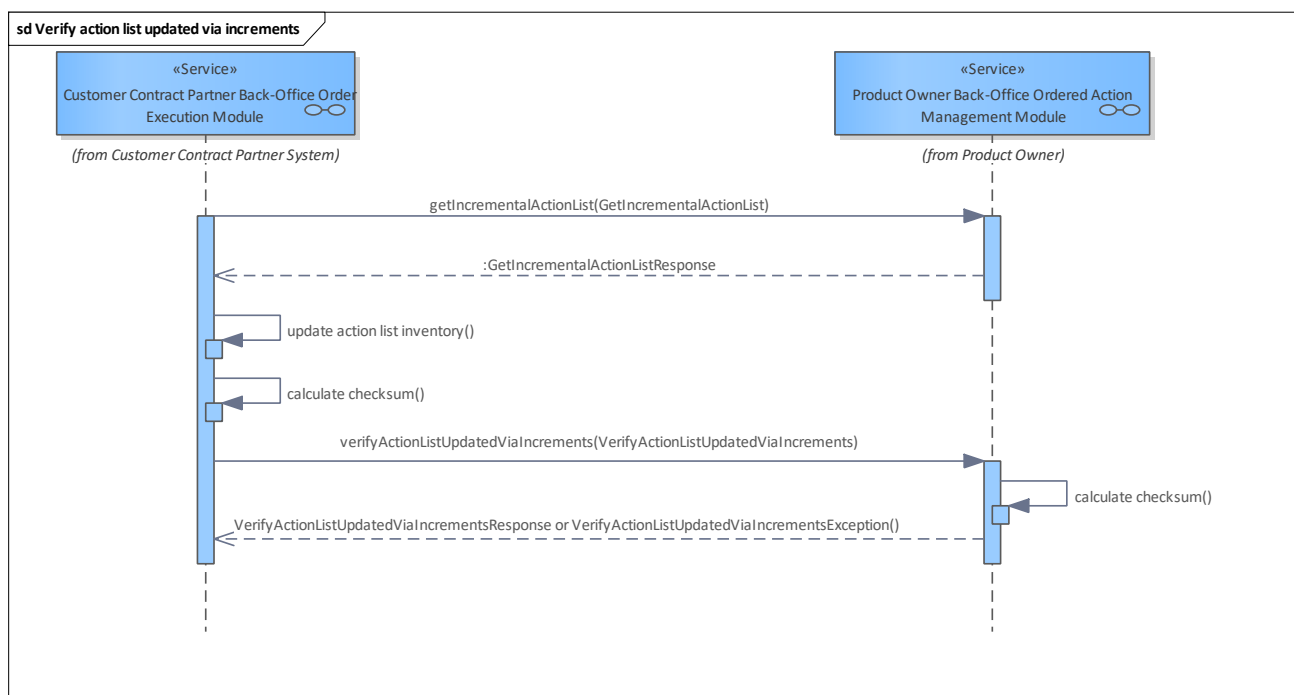


Figure 2: Verify action list updated via increments



Figure 3: Verify application hotlist updated via increments

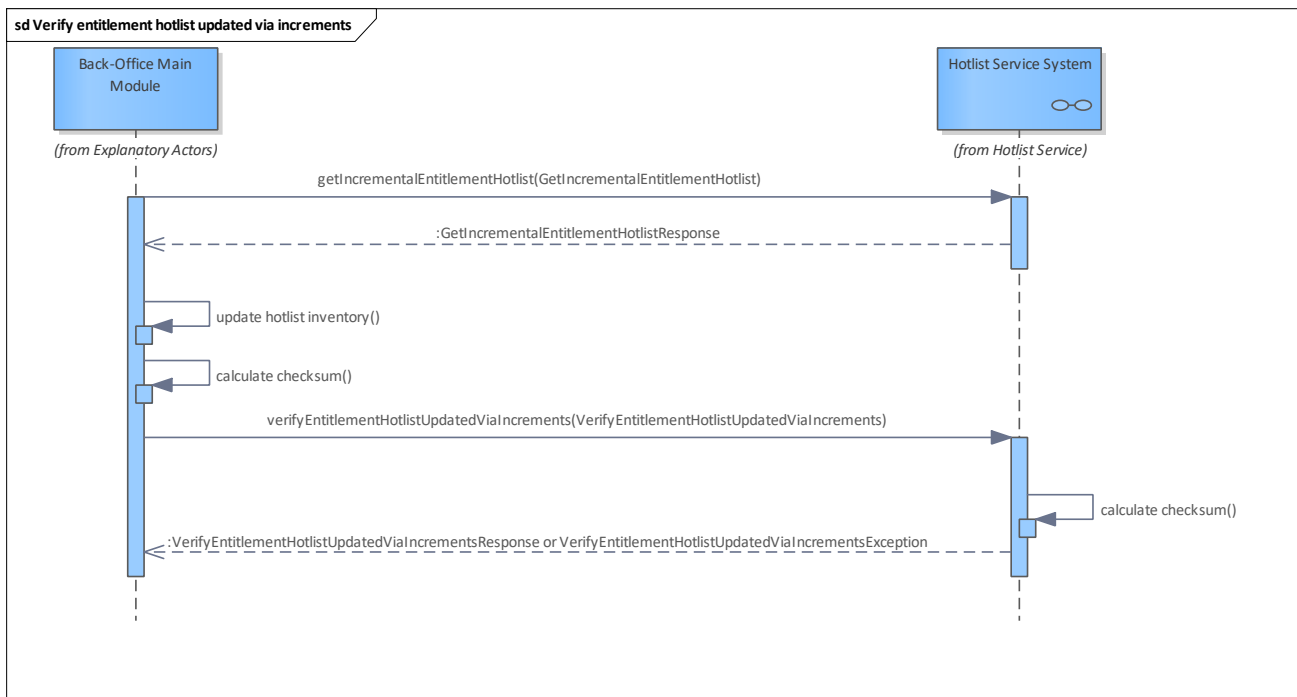


Figure 4: Verify entitlement hotlist updated via increments

## 2.1 Checksum calculation for hotlist and action list verification

The actual calculation of the checksum differs per list and is given in the corresponding use cases. Here, we describe some aspects that apply to all of them.

The data objects are encoded according to the distinguished encoding rules (DER) per the corresponding ASN.1 schemata including tag and length.

The hash algorithm to use is SHA-256.

We always sort lexicographically in ascending order.

Pseudo code to calculate the hash value:

```
hash( concatenate( sort( [ serialise(entry) for entry in entries ] ) ) )
```

## 2.2 Example calculation for an action list inventory

Action list inventory (reduced to the parts relevant to hashing) to verify:

```
[
  {
    "entitlementUnblockingActionListEntry": {
      "actionListEntryBaseInformation": {
        "orderId": {
          "orderNumber": 108,
          "orderingCcp": 123
        }
      }
    },
    {
      "entitlementIssuanceActionListEntry": {
        "actionListEntryBaseInformation": {
          "orderId": {
            "orderNumber": 109,
            "orderingCcp": 123
          }
        }
      }
    },
    {
      "entitlementTerminationActionListEntry": {
        "actionListEntryBaseInformation": {
          "orderId": {
            "orderNumber": 111,
            "orderingCcp": 123
          }
        }
      }
    },
    {
      "entitlementBlockingActionListEntry": {
        "actionListEntryBaseInformation": {
          "orderId": {
            "orderNumber": 110,
            "orderingCcp": 123
          }
        }
      }
    }
  ]
```

Intermediate result after serialisation:

```
[0x4D017B5A016C,  
0x4D017B5A016D,  
0x4D017B5A016F,  
0x4D017B5A016E]
```

Intermediate result after sorting:

```
[0x4D017B5A016C,  
0x4D017B5A016D,  
0x4D017B5A016E,  
0x4D017B5A016F]
```

Final result after applying SHA-256:

```
0x1376FF1481CB9B73DDADDC5F4588478A506EE4129AA345B2D408E37DF1FB51E8
```

## 2.3 Example calculation for an application hotlist inventory

Application hotlist inventory to verify:

```
[  
  {  
    "applicationInstanceId": {  
      "registrationAuthorityId": 10004,  
      "subjectRole": "0x1F",  
      "subjectNumber": "0x00000004"  
    },  
    "applicationTransitionCounter": 0,  
    "applicationBlockingMode": 10  
  },  
  {  
    "applicationInstanceId": {  
      "registrationAuthorityId": 10004,  
      "subjectRole": "0x1B",  
      "subjectNumber": "0x00000002"  
    },  
    "applicationTransitionCounter": 6,  
    "applicationBlockingMode": 0  
  },  
  {  
    "applicationInstanceId": {  
      "registrationAuthorityId": 10004,  
      "subjectRole": "0x1B",  
      "subjectNumber": "0x00000001"  
    },  
    "applicationTransitionCounter": 0,  
    "applicationBlockingMode": 0  
  }  
]
```

Intermediate result after serialisation:

```
[0x640D4D02271480011F82040000000458010002010A,  
0x640D4D02271480011B820400000002580106020100,  
0x640C4D02271480011B8203000001580100020100]
```



Intermediate result after sorting:

```
[0x640C4D02271480011B8203000001580100020100,  
0x640D4D02271480011B820400000002580106020100,  
0x640D4D02271480011F82040000000458010002010A]
```

Final result after applying SHA-256:

```
0x98152207B15DE600F0A98974CA510DDB6A3034E661BFE4287BF808128C61C6E2
```

## 2.4 Example calculation for an entitlement hotlist inventory

Entitlement hotlist inventory to verify:

```
[  
  {  
    "entitlementBlockingMode": 0,  
    "entitlementId": {  
      "ccpOrgId": 348,  
      "entitlementIssuanceCounter": 1005,  
      "samId": {  
        "registrationAuthorityId": 2713,  
        "subjectNumber": "0x00000002",  
        "subjectRole": "0x13"  
      }  
    },  
    "entitlementTransitionCounter": 0  
  },  
  {  
    "entitlementBlockingMode": 20,  
    "entitlementId": {  
      "ccpOrgId": 5730,  
      "entitlementIssuanceCounter": 120,  
      "samId": {  
        "registrationAuthorityId": 2713,  
        "subjectNumber": "0x00000002",  
        "subjectRole": "0x13"  
      }  
    },  
    "entitlementTransitionCounter": 0  
  },  
  {  
    "entitlementBlockingMode": 0,  
    "entitlementId": {  
      "ccpOrgId": 36,  
      "entitlementIssuanceCounter": 48905,  
      "samId": {  
        "registrationAuthorityId": 2713,  
        "subjectNumber": "0x00000002",  
        "subjectRole": "0x13"  
      }  
    },  
    "entitlementTransitionCounter": 6  
  }  
]
```

Intermediate result after serialisation:

```
[0x6E174D02015C640D4D020A998001138204000000024B0203ED580100020100,  
0x6E164D021662640D4D020A998001138204000000024B0178580100020114,  
0x6E174D0124640D4D020A998001138204000000024B0300BF09580106020100]
```

Intermediate result after sorting:

```
[0x6E164D021662640D4D020A998001138204000000024B0178580100020114,  
0x6E174D0124640D4D020A998001138204000000024B0300BF09580106020100,  
0x6E174D02015C640D4D020A998001138204000000024B0203ED580100020100]
```

Final result after applying SHA-256:

```
0xC58276BE0B189DEAB6350B85588F80EF016153D01002B14F192B85B1FAA38124
```

## 3 Hotlist Entry Management

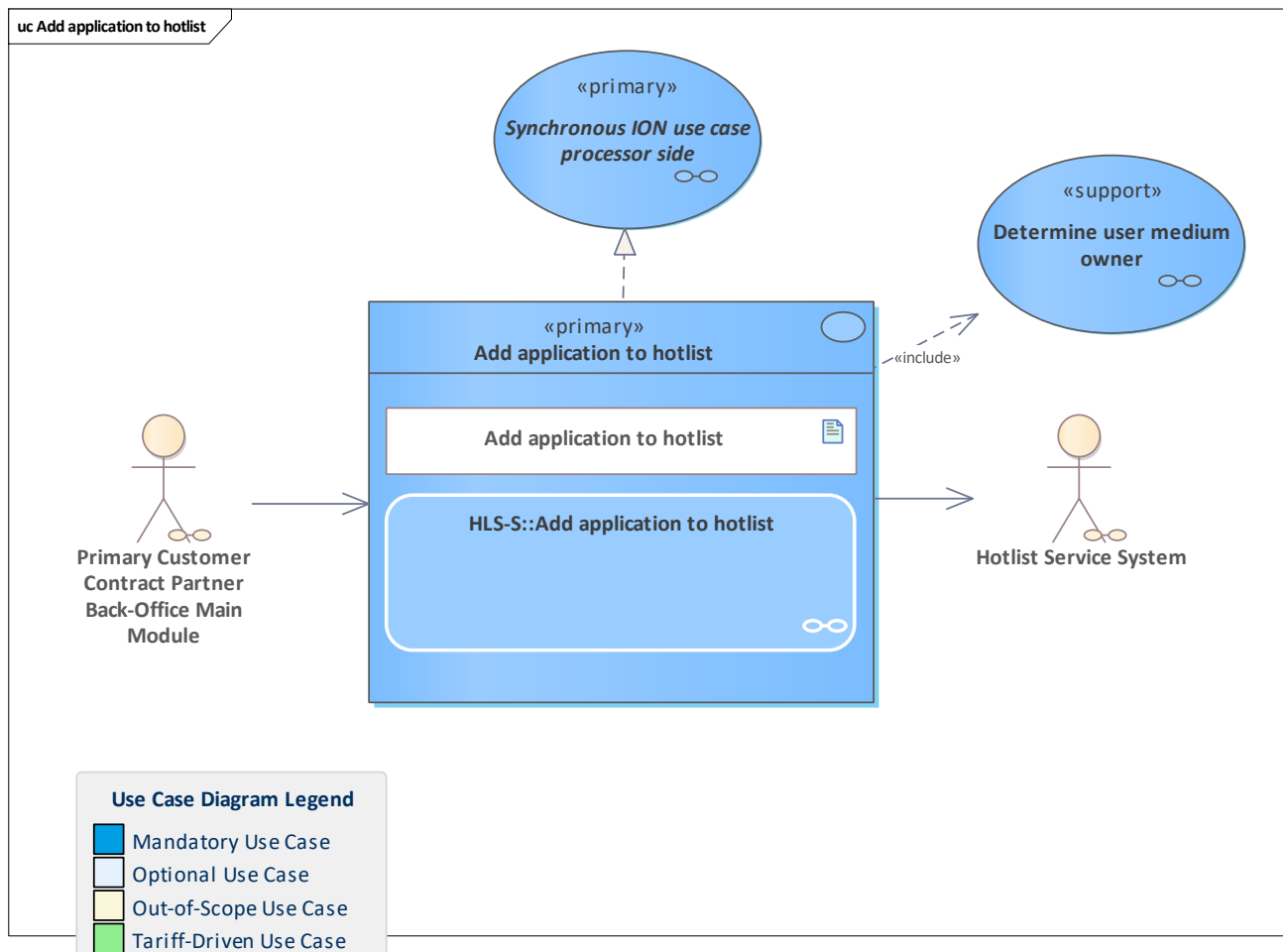
Functionality bundle that provides use cases to modify hotlists for entitlements, applications, SAMs, organisation and keys.

### 3.1 Overview

- [Add application to hotlist](#)
- [Retrieve CV certificate over signing key](#)
- [Determine user medium owner](#)
- [Remove application from hotlist](#)
- [Add entitlement to hotlist](#)
- [Remove entitlement from hotlist](#)
- [Add SAM to hotlist](#)
- [Determine SAM owner](#)
- [Remove SAM from hotlist](#)
- [Add authentication key to hotlist](#)
- [Remove authentication key from hotlist](#)
- [Add organisation to hotlist](#)
- [Remove organisation from hotlist](#)

### 3.2 Use Cases

#### 3.2.1 Add application to hotlist

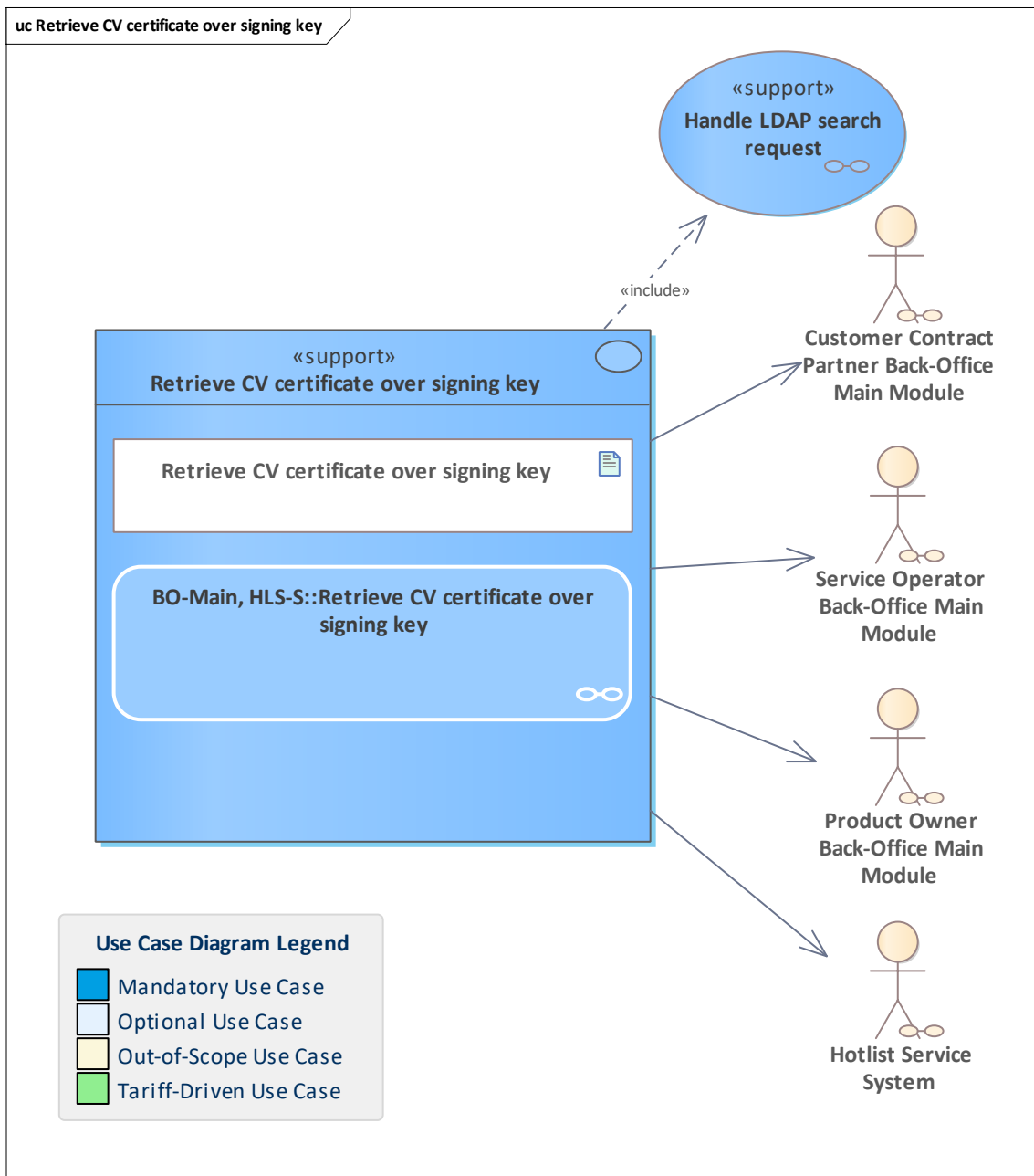


<b>Use Case</b>	<u>Add application to hotlist</u>
<b>Description</b>	<p>The pCCP sends a hotlisting request for an application instance ID coming from a chip-based user medium or an SCE ID coming from a MOTICS app.</p> <p>The Hotlist service system checks the request.</p> <p>The hotlist service system determines the application/user medium owner and compares it with the sender.</p> <p>The hotlist service system performs further checks.</p> <p>If all checks are positive, the application is added to the application hotlist.</p> <p>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.</p>
<b>Initiating Actor</b>	<u>Primary Customer Contract Partner Back-Office Main Module</u>
<b>Reacting Actor</b>	<u>Hotlist Service System</u>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	<u>Determine user medium owner</u>
<b>Linked Use Cases (Realises)</b>	<u>Synchronous ION use case processor side</u>



<b>Base Activity</b>	
<b>Inputs</b>	<u>Add application to hotlist : addApplicationToHotlist</u>
<b>Outputs</b>	<u>Add application to hotlist response : addApplicationToHotlistResponse</u>
<b>Error Cases</b>	<u>E HL HOTLIST ENTRY EXPIRATION TIME TOO FAR</u> <u>E HL UNAUTHORISED TO ADD TO OR REMOVE FROM HOTLIST</u> <u>E CO UNKNOWN BLOCKING MODE</u> <u>E HL HOTLIST ENTRY EXPIRATION TIME IN THE PAST</u> <u>E HL HOTLIST ENTRY ALREADY EXISTS</u> <u>E CO THIRD PARTY SYSTEM ERROR</u> <u>E CO APP INSTANCE ID UNKNOWN</u> <u>E HL INVALID TRANSITION COUNTER</u> <u>E HL INVALID BLOCKING MODE</u> <u>E HL INVALID BLOCKING MODE</u> <u>E HL INVALID TRANSITION COUNTER</u> <u>E CO WRONG SECURITY LEVEL</u> <u>E CO WRONG SUBJECT ROLE</u> <u>Add application to hotlist exception : addApplicationToHotlistException</u>
<b>Activity Diagram</b>	<u>HLS-S::Add application to hotlist</u>

## 3.2.2 Retrieve CV certificate over signing key

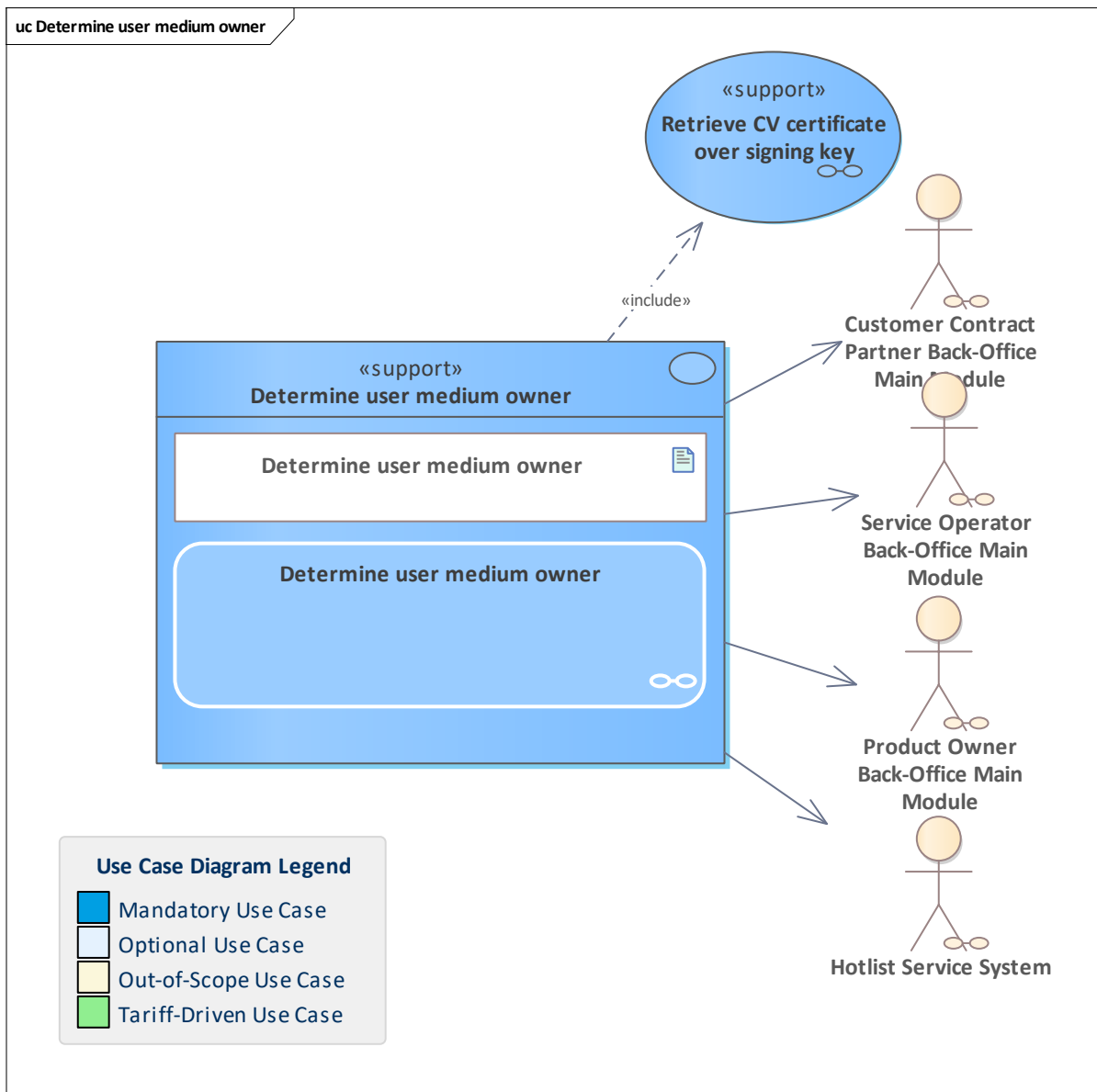


<b>Use Case</b>	<a href="#">Retrieve CV certificate over signing key</a>
<b>Description</b>	Retrieve the latest certificate over the signing key of an end entity. Note that there might be several certificates for this end entity that are not relevant here: superseded certificates and certificates over keys not used for signature purposes.
<b>Initiating Actor</b>	
<b>Reacting Actor</b>	<a href="#">Service Operator Back-Office Main Module</a> <a href="#">Customer Contract Partner Back-Office Main Module</a> <a href="#">Product Owner Back-Office Main Module</a> <a href="#">Hotlist Service System</a>
<b>Preconditions</b>	



<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	<a href="#">Handle LDAP search request</a>
<b>Linked Use Cases (Realises)</b>	
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">App instance ID : AppInstanceId</a>
<b>Outputs</b>	<a href="#">CV certificate over signing key : CvCertificate</a>
<b>Error Cases</b>	<a href="#">M-PKI not reachable</a> <a href="#">Unknown app instance ID</a>
<b>Activity Diagram</b>	<a href="#">BO-Main, HLS-S::Retrieve CV certificate over signing key</a>

### 3.2.3 Determine user medium owner



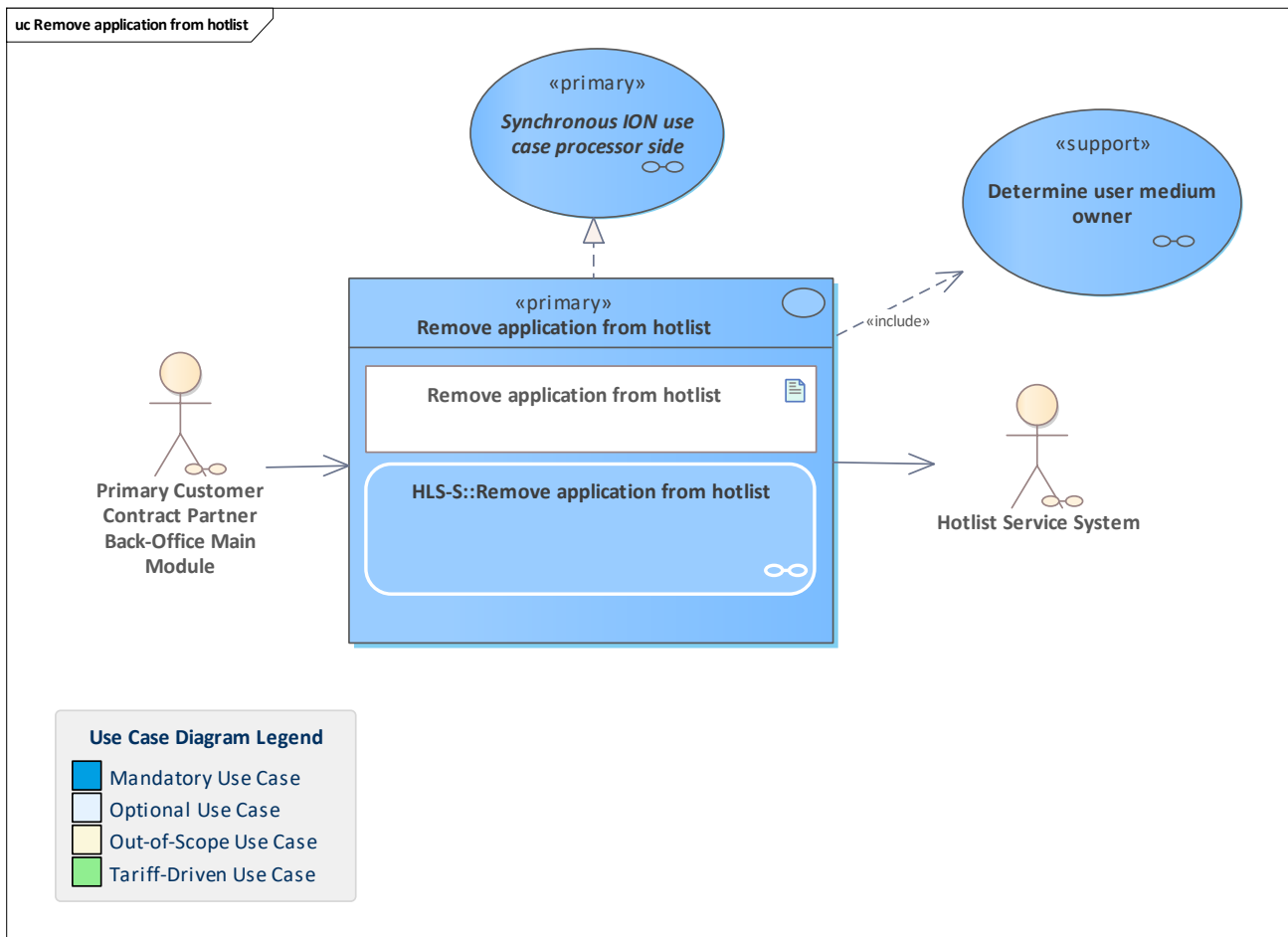
<b>Use Case</b>	<u>Determine user medium owner</u>
<b>Description</b>	<p>Determine the owner of a user medium with an application or SCE (in case of MOTICS with static entitlement) in a back-office system using the ownership information contained in the corresponding certificate.</p> <p>To determine 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.</p> <p>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.</p>
<b>Initiating Actor</b>	
<b>Reacting Actor</b>	<u>Customer Contract Partner Back-Office Main Module</u>



	<a href="#">Service Operator Back-Office Main Module</a> <a href="#">Product Owner Back-Office Main Module</a> <a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	<a href="#">Retrieve CV certificate over signing key</a>
<b>Linked Use Cases (Realises)</b>	
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">App instance ID : AppInstanceId</a>
<b>Outputs</b>	<a href="#">Org ID of user medium owner : OrganisationId</a>
<b>Error Cases</b>	<a href="#">M-PKI not reachable</a> <a href="#">Unknown app instance ID</a>
<b>Activity Diagram</b>	<a href="#">Determine user medium owner</a>



## 3.2.4 Remove application from hotlist

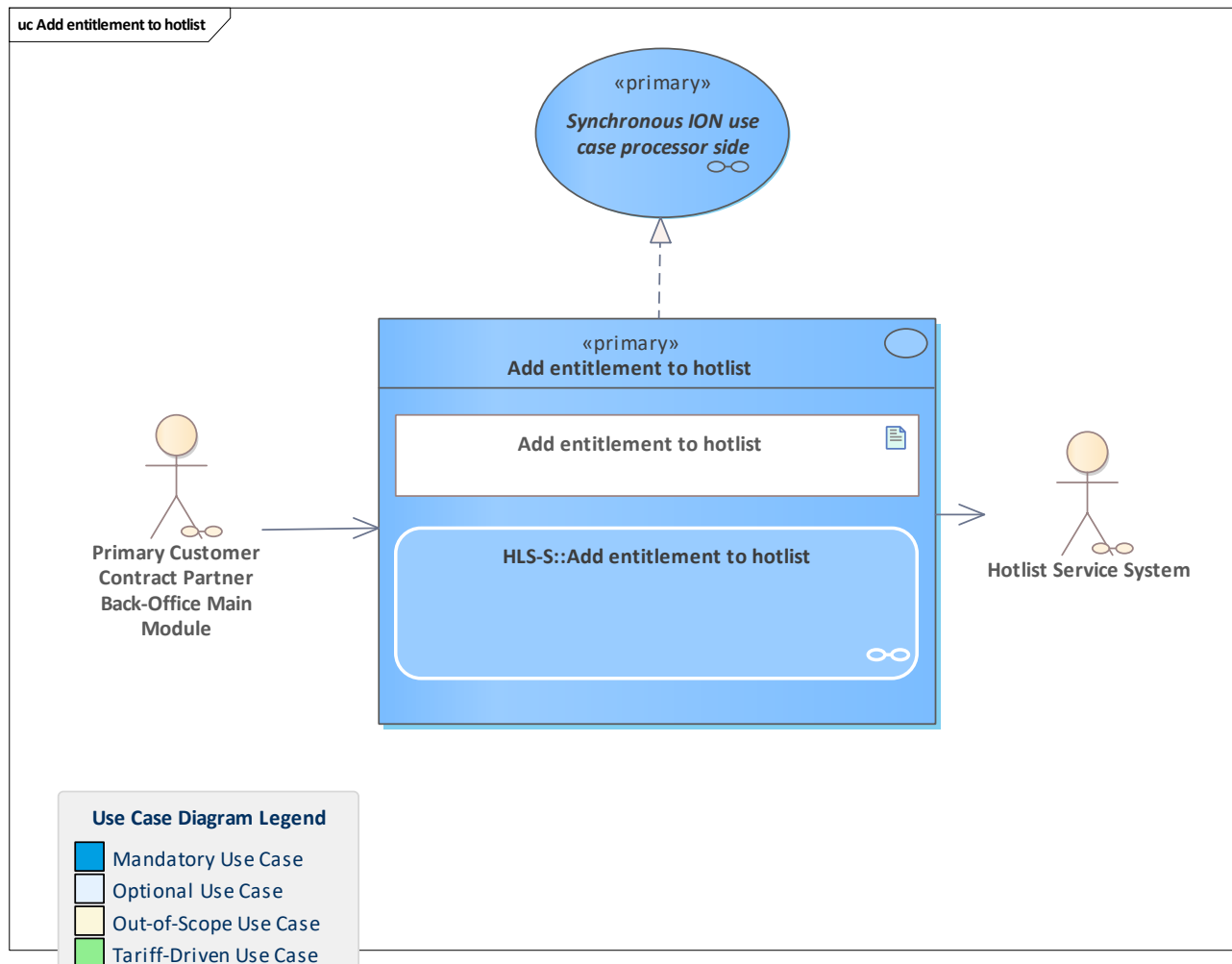


Use Case	<a href="#">Remove application from hotlist</a>
<b>Description</b>	<p>The pCCP sends a request to remove an application from the application hotlist.</p> <p>This can be caused either by an incoming application blocked notification or by other reasons (e.g. the customer found his "lost" user medium in his pocket).</p> <p>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.</p> <p>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.</p> <p><b>Note:</b> in the ION context, this is a <a href="#">Synchronous ION use case processor side</a> due to the synchronous call to the hotlist service system.</p>
<b>Initiating Actor</b>	<a href="#">Primary Customer Contract Partner Back-Office Main Module</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases</b>	



<b>(Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	<a href="#">Determine user medium owner</a>
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Remove application from hotlist : removeApplicationFromHotlist</a>
<b>Outputs</b>	<a href="#">Hotlisting removal processed : removeApplicationFromHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E HL UNAUTHORISED TO ADD TO OR REMOVE FROM HOTLIST</a> <a href="#">E HL HOTLIST ENTRY ALREADY REMOVED</a> <a href="#">E HL HOTLIST ENTRY NOT FOUND</a> <a href="#">E CO THIRD PARTY SYSTEM ERROR</a> <a href="#">E CO APP INSTANCE ID UNKNOWN</a> <a href="#">E CO WRONG SECURITY LEVEL</a> <a href="#">E CO WRONG SUBJECT ROLE</a> <a href="#">Remove application from hotlist exception : removeApplicationFromHotlistException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Remove application from hotlist</a>

## 3.2.5 Add entitlement to hotlist

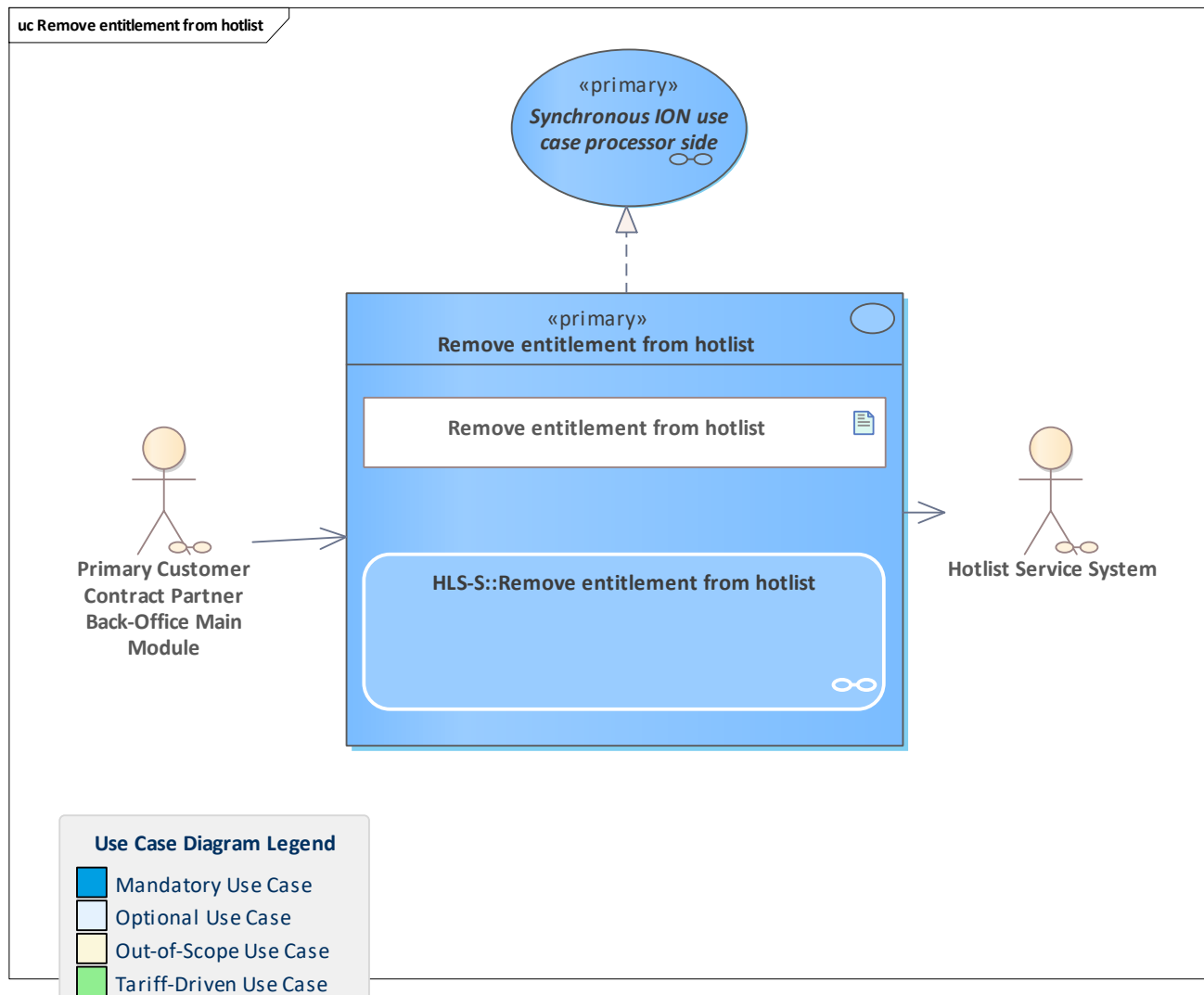


<b>Use Case</b>	<u>Add entitlement to hotlist</u>
<b>Description</b>	<p>The pCCP sends a hotlisting request for an entitlement ID coming from a chip-based user medium or an entitlement ID from a static entitlement.</p> <p>The Hotlist service system checks the request.</p> <p>If all checks are positive, the entitlement is added to the entitlement hotlist.</p> <p><b>Note:</b> 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.</p>
<b>Initiating Actor</b>	<u>Primary Customer Contract Partner Back-Office Main Module</u>
<b>Reacting Actor</b>	<u>Hotlist Service System</u>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	



<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Add entitlement to hotlist : addEntitlementToHotlist</a>
<b>Outputs</b>	<a href="#">Add entitlement to hotlist response : addEntitlementToHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E HL UNAUTHORISED TO ADD TO OR REMOVE FROM HOTLIST</a> <a href="#">E HL HOTLIST ENTRY ALREADY EXISTS</a> <a href="#">E HL HOTLIST ENTRY EFFECTIVE TIME AFTER EXPIRATION TIME</a> <a href="#">E HL HOTLIST ENTRY EXPIRATION TIME TOO FAR</a> <a href="#">E CO UNKNOWN BLOCKING MODE</a> <a href="#">E HL HOTLIST ENTRY EXPIRATION TIME IN THE PAST</a> <a href="#">E HL UNKNOWN PRODUCT OWNER</a> <a href="#">E HL CCP NOT ACCEPTING PRODUCT</a> <a href="#">E HL HOTLIST ENTRY EFFECTIVE TIME TOO FAR</a> <a href="#">E HL INVALID TRANSITION COUNTER</a> <a href="#">E HL INVALID BLOCKING MODE</a> <a href="#">E HL INVALID TRANSITION COUNTER</a> <a href="#">E HL INVALID BLOCKING MODE</a> <a href="#">E CO WRONG SECURITY LEVEL</a> <a href="#">E CO WRONG SUBJECT ROLE</a> <a href="#">Add entitlement to hotlist exception : addEntitlementToHotlistException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Add entitlement to hotlist</a>

## 3.2.6 Remove entitlement from hotlist

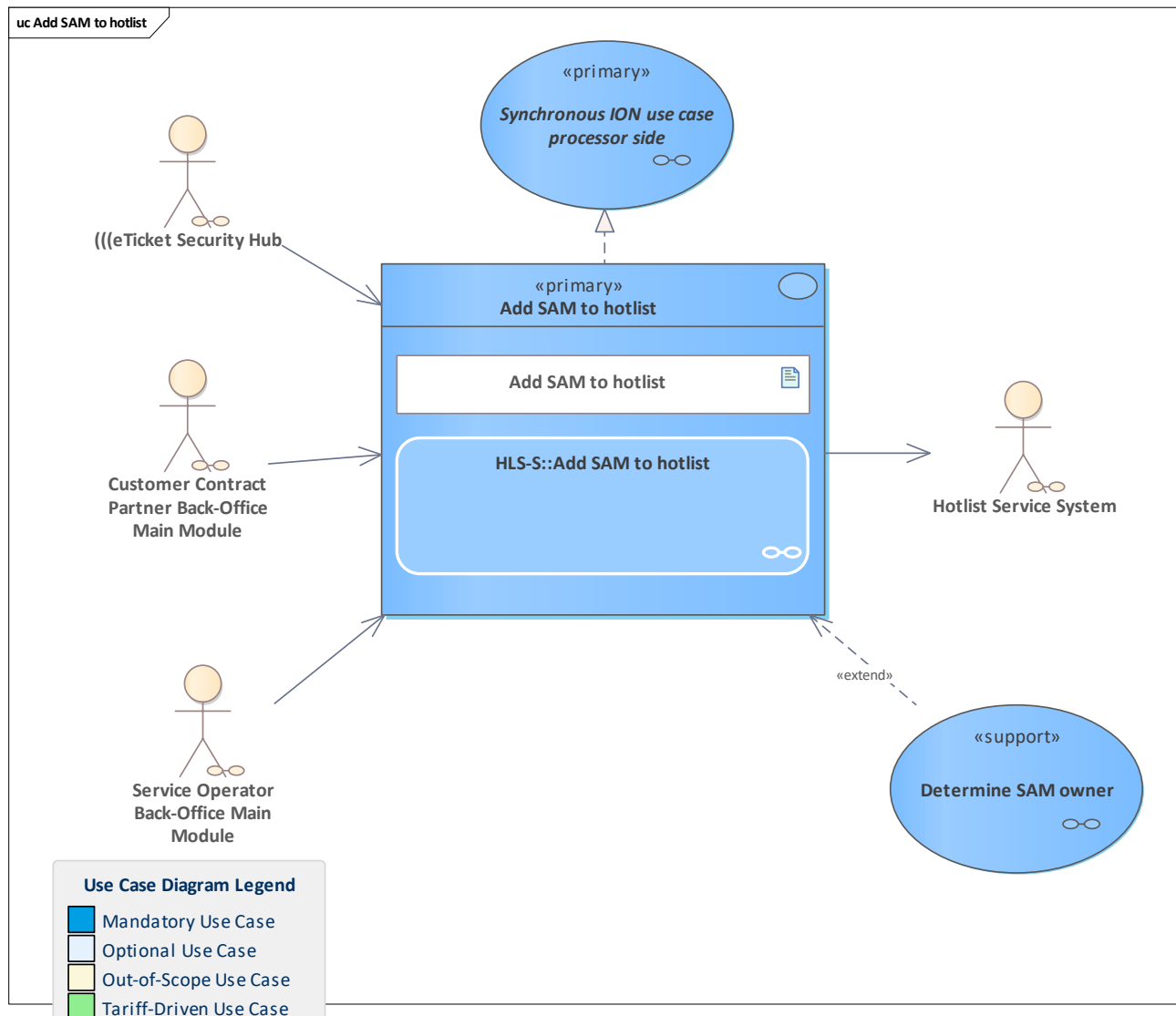


<b>Use Case</b>	<a href="#">Remove entitlement from hotlist</a>
<b>Description</b>	<p>The pCCP sends a request to remove an entitlement from the entitlement hotlist.</p> <p>This can be caused either by an incoming entitlement blocked notification or by other reasons (e.g. the customer contract allows suspension for a certain period of time. After this period, the temporary hotlist entry is removed).</p> <p>The hotlist service checks the request. Finally, the hotlist service removes the entitlement from the hotlist.</p> <p>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.</p> <p><b>Note:</b> in the ION context, this is a <a href="#">Synchronous ION use case processor side</a> due to the synchronous call to the hotlist service.</p>
<b>Initiating Actor</b>	<a href="#">Primary Customer Contract Partner Back-Office Main Module</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>



<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Remove entitlement from hotlist : removeEntitlementFromHotlist</a>
<b>Outputs</b>	<a href="#">Remove entitlement from hotlist response : removeEntitlementFromHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E HL UNAUTHORISED TO ADD TO OR REMOVE FROM HOTLIST</a> <a href="#">E HL HOTLIST ENTRY ALREADY REMOVED</a> <a href="#">E HL HOTLIST ENTRY NOT FOUND</a> <a href="#">E CO WRONG SECURITY LEVEL</a> <a href="#">E CO WRONG SUBJECT ROLE</a> <a href="#">Remove entitlement from hotlist exception : removeEntitlementFromHotlistException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Remove entitlement from hotlist</a>

## 3.2.7 Add SAM to hotlist



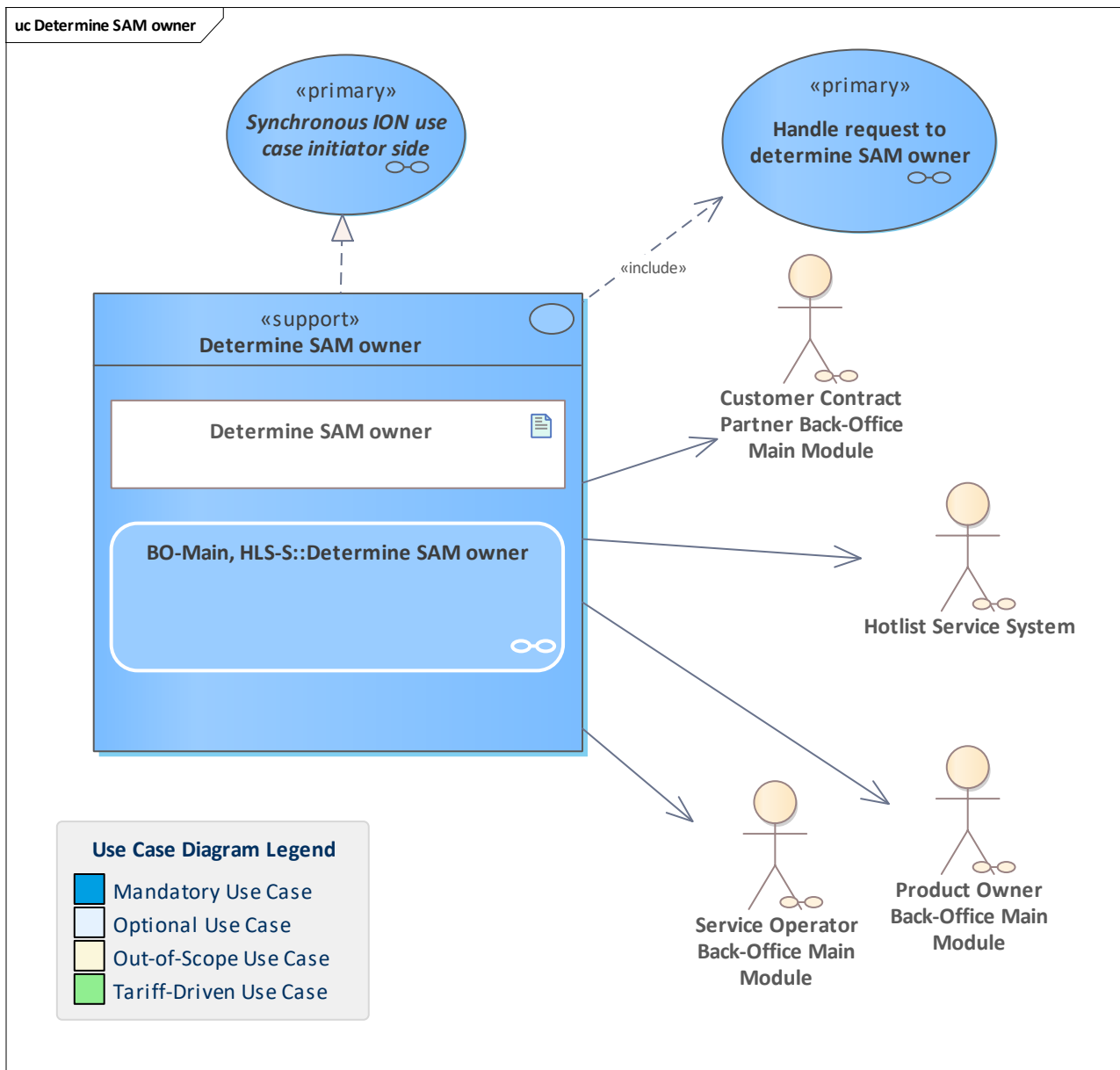
Use Case	Add SAM to hotlist
Description	<p>Use case to add a SAM to the SAM hotlist either by the SAM owner (SO or CCP) or the scheme manager (ESH module). The actor sends a request to add a SAM to the SAM hotlist. The Hotlist service system checks the request. In the case of an SO or CCP, the hotlist 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.</p> <p><b>Note:</b> 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.</p> <p><b>Note:</b> this process is irreversible since the SAM will also be terminated and its certificate revoked. This is done for security</p>



	level-3 SAMs as well as for level-2 SAMs.
<b>Initiating Actor</b>	<a href="#">Customer Contract Partner Back-Office Main Module</a> <a href="#">Service Operator Back-Office Main Module</a> <a href="#">(((eTicket Security Hub</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	<a href="#">Determine SAM owner</a>
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Add SAM to hotlist : addSamToHotlist</a>
<b>Outputs</b>	<a href="#">Add SAM to hotlist response : addSamToHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E HL HOTLIST ENTRY ALREADY EXISTS</a> <a href="#">E HL UNAUTHORISED TO ADD TO OR REMOVE FROM HOTLIST</a> <a href="#">E CO THIRD PARTY SYSTEM ERROR</a> <a href="#">E CO APP INSTANCE ID UNKNOWN</a> <a href="#">E CO WRONG SECURITY LEVEL</a> <a href="#">E CO WRONG SUBJECT ROLE</a> <a href="#">Add SAM to hotlist exception : addSamToHotlistException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Add SAM to hotlist</a>



## 3.2.8 Determine SAM owner

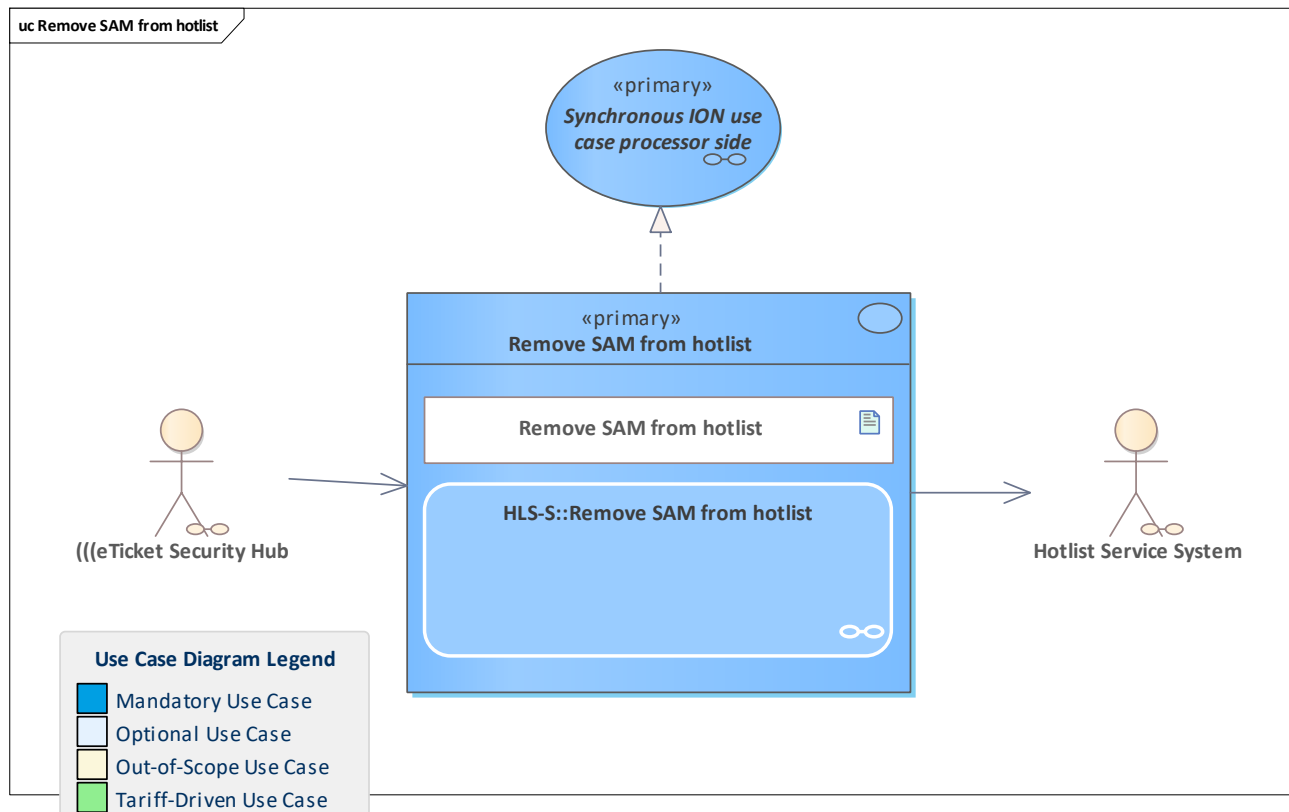


<b>Use Case</b>	<a href="#">Determine SAM owner</a>
<b>Description</b>	Determine the SAM Owner (organisation ID and role) for a given SAM ID using the service provided by the ESH.
<b>Initiating Actor</b>	
<b>Reacting Actor</b>	<a href="#">Customer Contract Partner Back-Office Main Module</a> <a href="#">Service Operator Back-Office Main Module</a> <a href="#">Hotlist Service System</a> <a href="#">Product Owner Back-Office Main Module</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases</b>	<a href="#">Handle request to determine SAM owner</a>



<b>(Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case initiator side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">SAM ID : AppInstanceId</a>
<b>Outputs</b>	<a href="#">SAM owner : OrganisationalUnit</a>
<b>Error Cases</b>	<a href="#">E_CO_APP_INSTANCE_ID_UNKNOWN</a> <a href="#">Unknown SAM : E_CO_APP_INSTANCE_ID_UNKNOWN</a> <a href="#">Timeout</a>
<b>Activity Diagram</b>	<a href="#">BO-Main, HLS-S::Determine SAM owner</a>

### 3.2.9 Remove SAM from hotlist

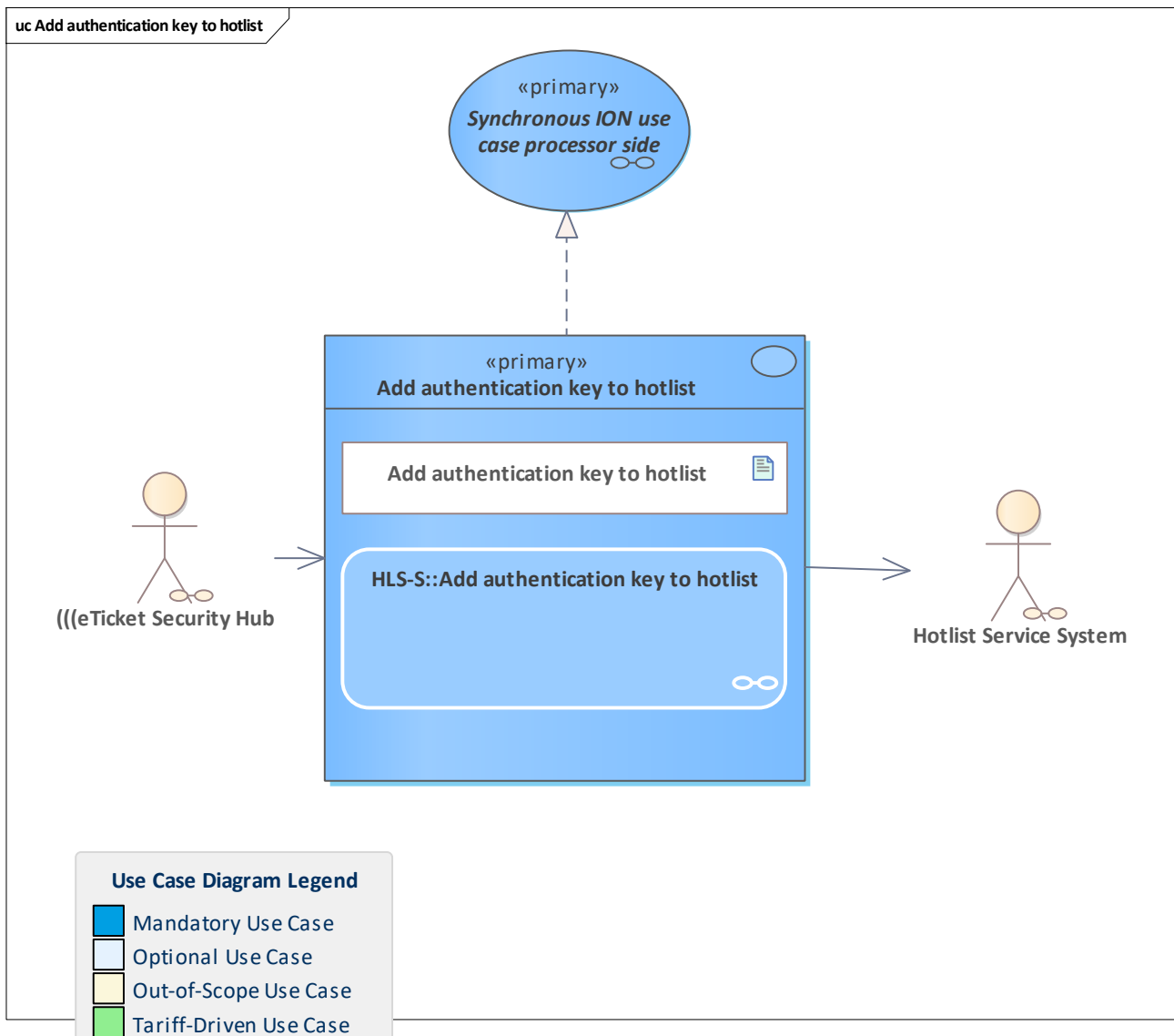


<b>Use Case</b>	<a href="#">Remove SAM from hotlist</a>
<b>Description</b>	<p>Rare use case for the hotlist service system to remove a SAM from the SAM hotlist.</p> <p>Only the scheme manager's ESH is allowed to remove a SAM from the hotlist.</p> <p><b>Note:</b> the SAM hotlist in the next cycle will 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.</p>
<b>Initiating Actor</b>	<a href="#">((eTicket Security Hub</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Remove SAM from hotlist : removeSamFromHotlist</a>
<b>Outputs</b>	<a href="#">Remove sam from hotlist response : removeSamFromHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E HL UNAUTHORISED TO ADD TO OR REMOVE FROM HOTLIST</a> <a href="#">E HL HOTLIST ENTRY NOT FOUND</a>



	<u>E HL HOTLIST ENTRY ALREADY REMOVED</u> <u>E CO WRONG SECURITY LEVEL</u> <u>E CO WRONG SUBJECT ROLE</u> <u>Remove SAM from hotlist exception :</u> <u>removeSamFromHotlistException</u>
<b>Activity Diagram</b>	<u>HLS-S::Remove SAM from hotlist</u>

### 3.2.10 Add authentication key to hotlist

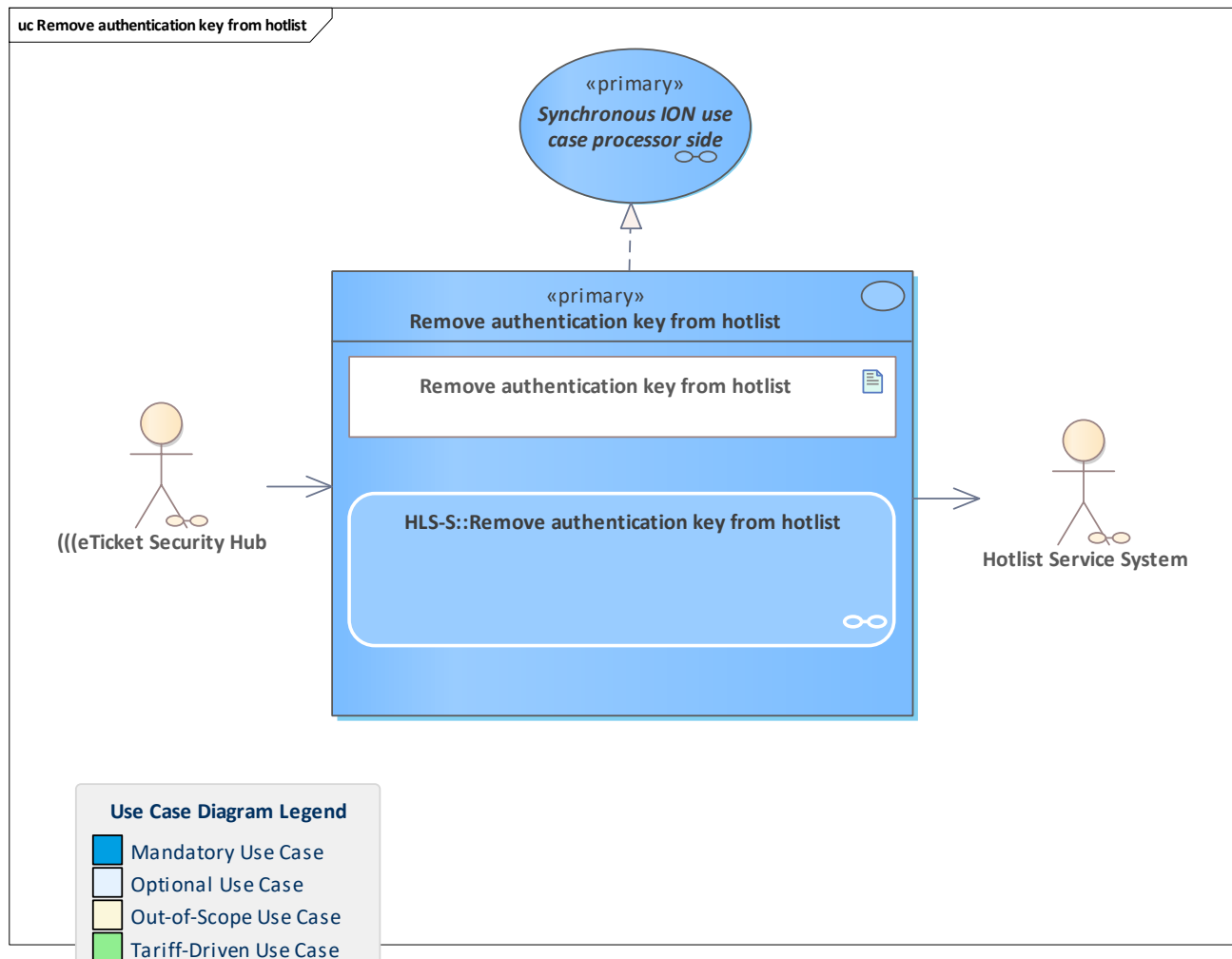


<b>Use Case</b>	<u>Add authentication key to hotlist</u>
<b>Description</b>	The authentication key is to be hotlisted. This can only be done by the scheme manager's ESH. Note: the authentication key hotlist that is compiled in the next hotlist cycle will contain the new hotlist entry. When the new hotlist is requested and distributed to the terminals, the new entry is taken into account when the check against the hotlist is performed.
<b>Initiating Actor</b>	<u>(((eTicket Security Hub</u>
<b>Reacting Actor</b>	<u>Hotlist Service System</u>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	



<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Add authentication key to hotlist : addAuthenticationKeyToHotlist</a>
<b>Outputs</b>	<a href="#">Add authentication key to hotlist response : addAuthenticationKeyToHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E HL UNAUTHORISED TO ADD TO OR REMOVE FROM HOTLIST</a> <a href="#">E HL HOTLIST ENTRY ALREADY EXISTS</a> <a href="#">Add authentication key to hotlist exception : addAuthenticationKeyToHotlistException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Add authentication key to hotlist</a>

### 3.2.11 Remove authentication key from hotlist



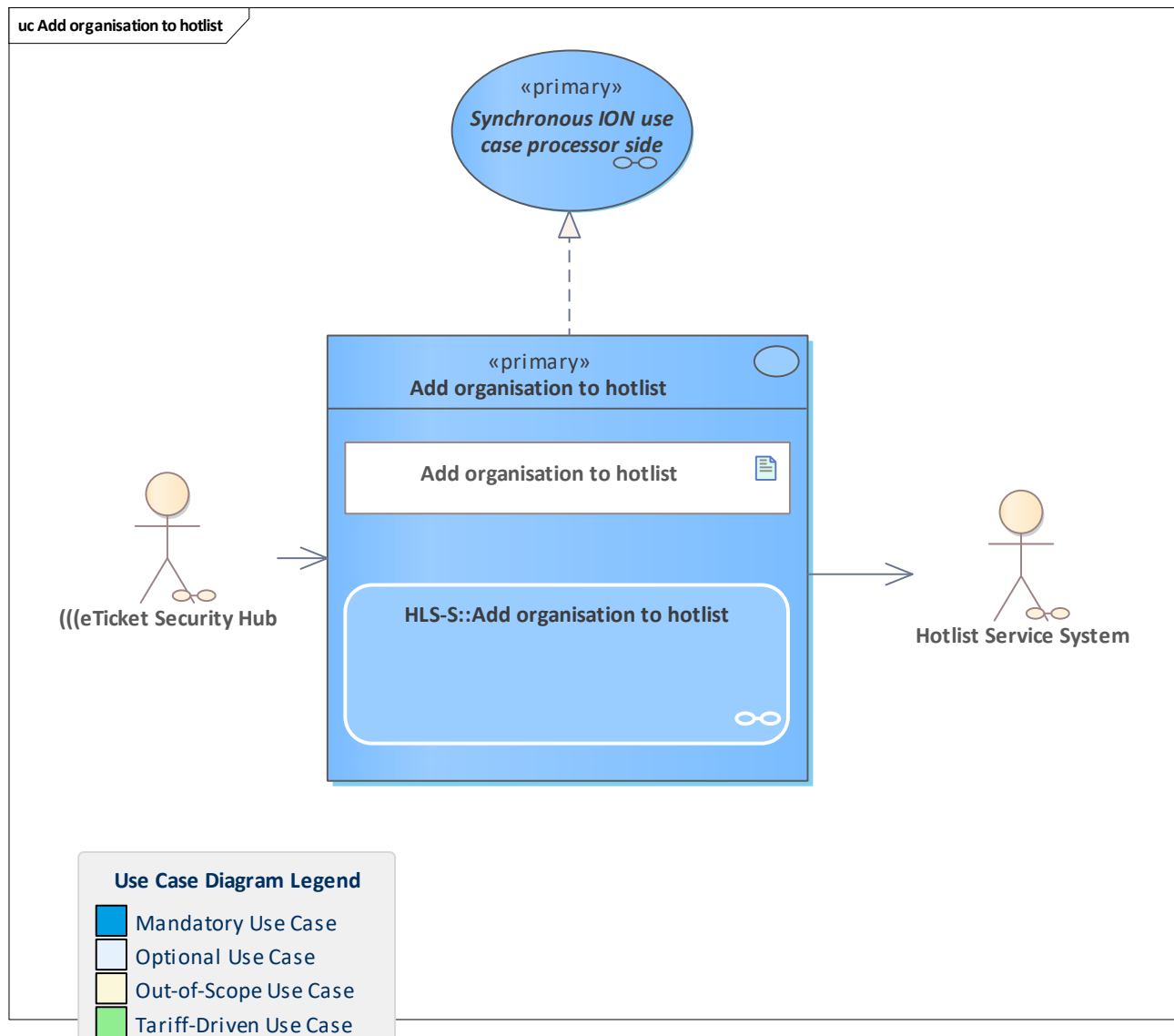
<b>Use Case</b>	<a href="#">Remove authentication key from hotlist</a>
<b>Description</b>	<p>Rare use case, only for the scheme manager's ESH. The authentication key is removed from the hotlist via request to the hotlist service.</p> <p><b>Note:</b> the authentication key hotlist in the next cycle will 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.</p>
<b>Initiating Actor</b>	<a href="#">(((eTicket Security Hub</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	



<b>Inputs</b>	<a href="#">Remove authentication key from hotlist :</a> <a href="#">removeAuthenticationKeyFromHotlist</a>
<b>Outputs</b>	<a href="#">Hotlisting removal processed :</a> <a href="#">removeAuthenticationKeyFromHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E HL UNAUTHORISED TO ADD TO OR REMOVE FROM HOTLIST</a> <a href="#">E HL HOTLIST ENTRY NOT FOUND</a> <a href="#">E HL HOTLIST ENTRY ALREADY REMOVED</a> <a href="#">Remove authentication key from hotlist exception :</a> <a href="#">removeAuthenticationKeyFromHotlistException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Remove authentication key from hotlist</a>



### 3.2.12 Add organisation to hotlist

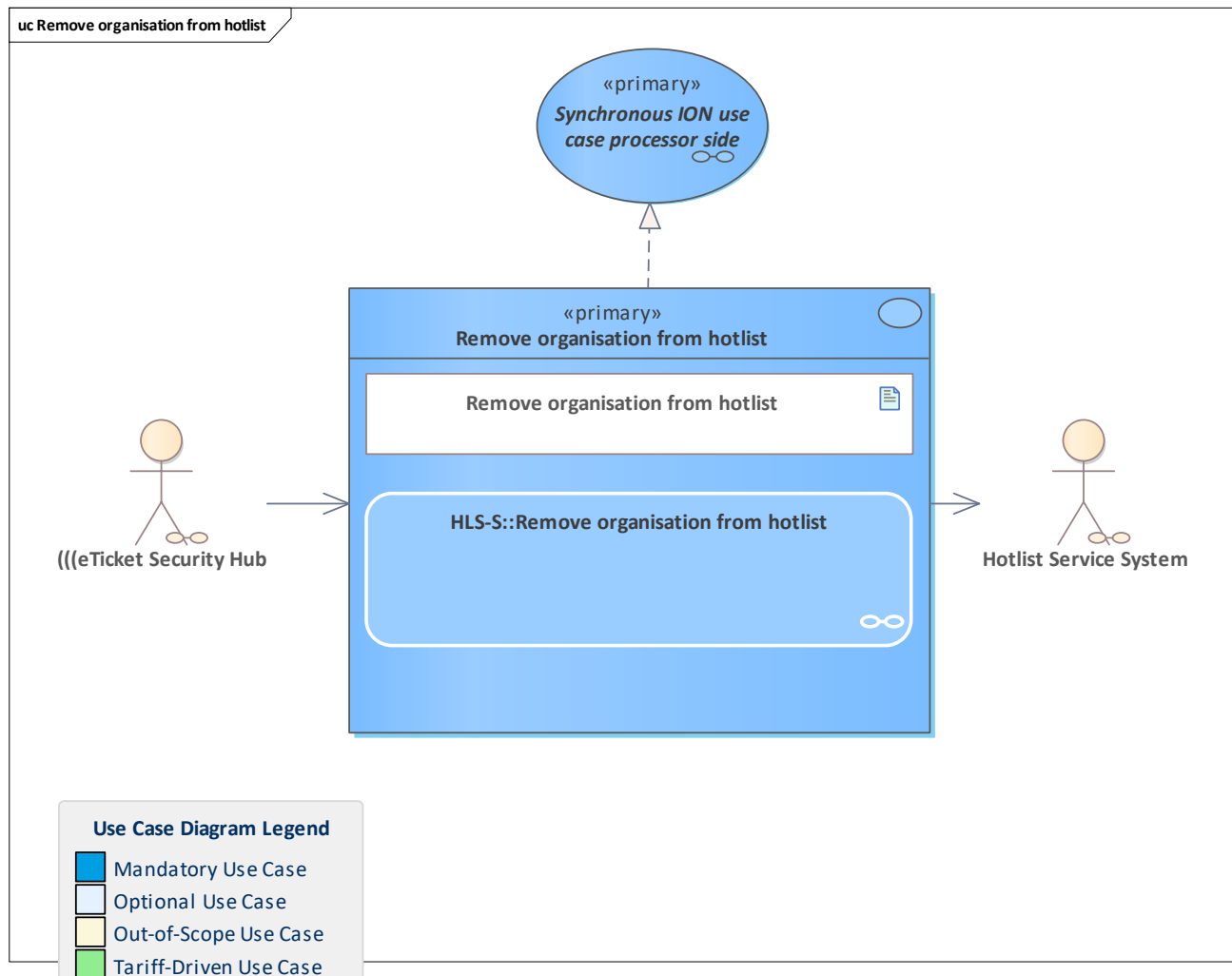


<b>Use Case</b>	<a href="#">Add organisation to hotlist</a>
<b>Description</b>	Use case to add an organisation to the organisation hotlist. This can only be done by the scheme manager's ESH. <b>Note:</b> the organisation hotlist that is compiled in the next hotlist cycle will contain the new hotlist 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.
<b>Initiating Actor</b>	<a href="#">(((eTicket Security Hub</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases</b>	<a href="#">Synchronous ION use case processor side</a>



<b>(Realises)</b>	
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Add organisation to hotlist : addOrganisationToHotlist</a>
<b>Outputs</b>	<a href="#">Add organisation to hotlist response : addOrganisationToHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E HL UNAUTHORISED TO ADD TO OR REMOVE FROM HOTLIST</a> <a href="#">E HL SCHEME MANAGER MAY NOT HOTLIST HIMSELF</a> <a href="#">E HL HOTLIST ENTRY ALREADY EXISTS</a> <a href="#">E HL UNKNOWN ORGANISATION FOR HOTLISTING</a> <a href="#">Add organisation to hotlist exception : addOrganisationToHotlistException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Add organisation to hotlist</a>

### 3.2.13 Remove organisation from hotlist



<b>Use Case</b>	<a href="#">Remove organisation from hotlist</a>
<b>Description</b>	<p>The requested organisation ID is removed from the organisation hotlist.</p> <p>Rare use case only performed by the scheme manager's ESH as client.</p> <p><b>Note:</b> 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.</p>
<b>Initiating Actor</b>	<a href="#">(((eTicket Security Hub</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>



<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Remove organisation from hotlist : removeOrganisationFromHotlist</a>
<b>Outputs</b>	<a href="#">Hotlisting removal processed : removeOrganisationFromHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E HL HOTLIST ENTRY ALREADY REMOVED</a> <a href="#">E HL HOTLIST ENTRY NOT FOUND</a> <a href="#">E HL UNAUTHORISED TO ADD TO OR REMOVE FROM HOTLIST</a> <a href="#">Remove organisation from hotlist exception : removeOrganisationFromHotlistException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Remove organisation from hotlist</a>



## 4 Hotlist Retrieval

Functionality bundle that covers the use cases for providing hotlists from hotlist service perspective.

### 4.1 Overview

[Handle retrieval request for application hotlist](#)

[Handle retrieval request for incremental application hotlist](#)

[Handle verification request for application hotlist updated via increments](#)

[Handle retrieval request for entitlement hotlist](#)

[Handle retrieval request for entitlement hotlist with product information](#)

[Handle retrieval request for incremental entitlement hotlist](#)

[Handle verification request for entitlement hotlist updated via increments](#)

[Handle retrieval request for SAM hotlist](#)

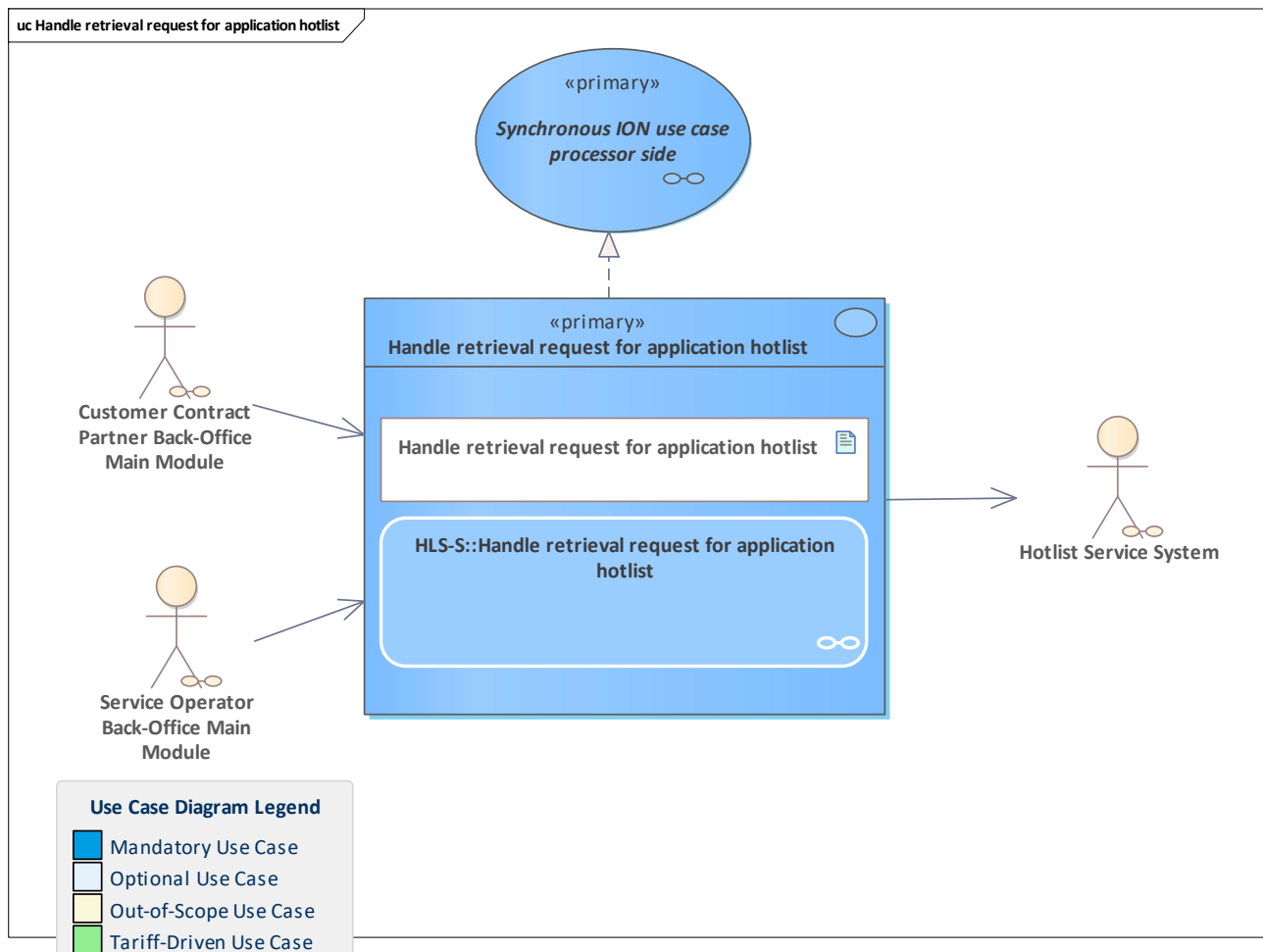
[Handle retrieval request for organisation hotlist](#)

[Handle retrieval request for authentication key hotlist](#)

[Generate current hotlists](#)

### 4.2 Use Cases

#### 4.2.1 Handle retrieval request for application hotlist

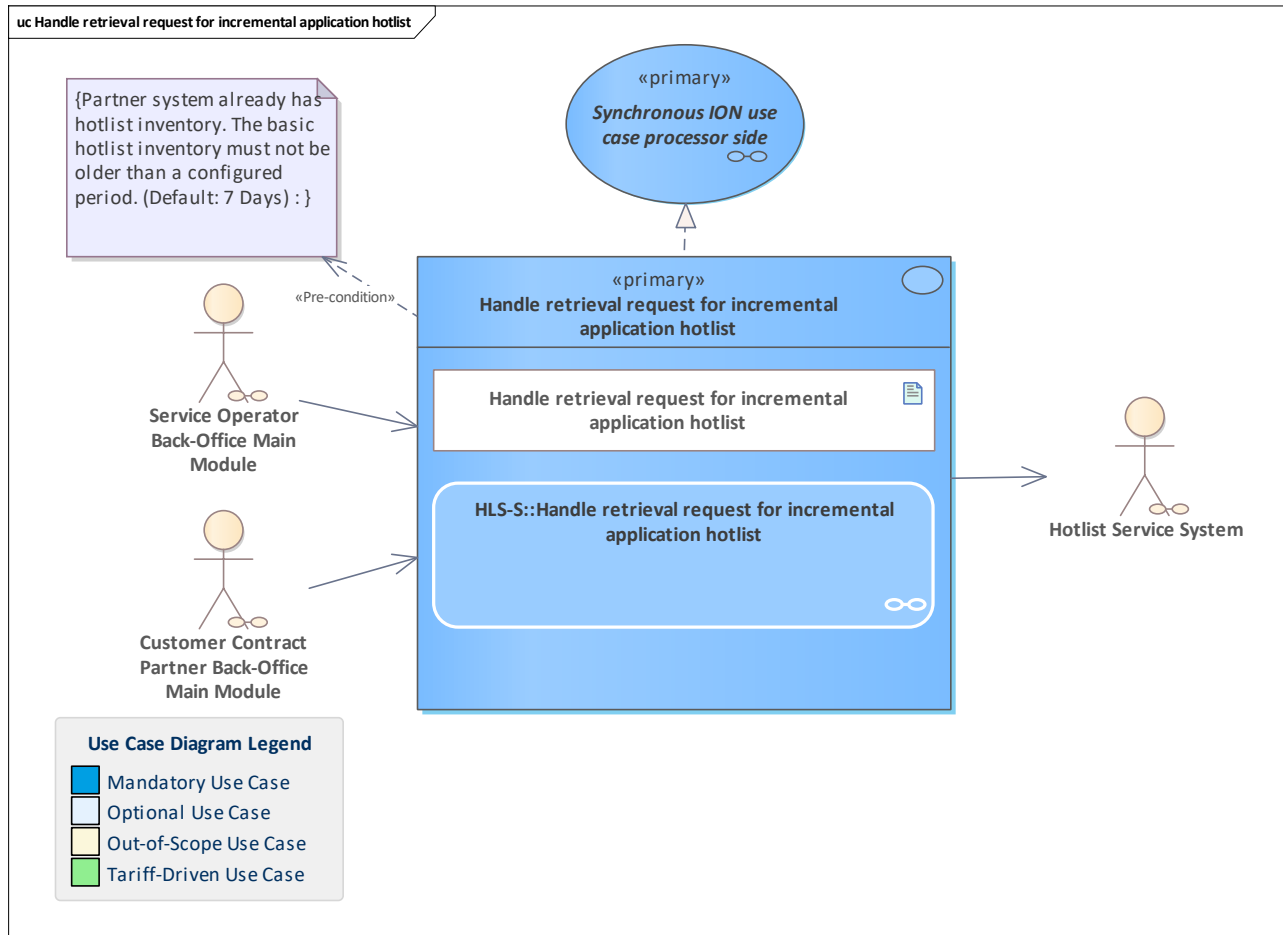


<b>Use Case</b>	<a href="#">Handle retrieval request for application hotlist</a>
<b>Description</b>	After receiving a request for an application hotlist, the hotlist service system processes the request by running checks (see activity diagram) and creating the current application hotlist as a response.
<b>Initiating Actor</b>	<a href="#">Customer Contract Partner Back-Office Main Module</a> <a href="#">Service Operator Back-Office Main Module</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Get application hotlist request : getApplicationHotlist</a>
<b>Outputs</b>	<a href="#">Get application hotlist response : getApplicationHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E HL STILL GENERATING HOTLIST</a> <a href="#">Get application hotlist exception : getApplicationHotlistException</a>



<b>Activity Diagram</b>	<a href="#">HLS-S::Handle retrieval request for application hotlist</a>
-------------------------	---

## 4.2.2 Handle retrieval request for incremental application hotlist



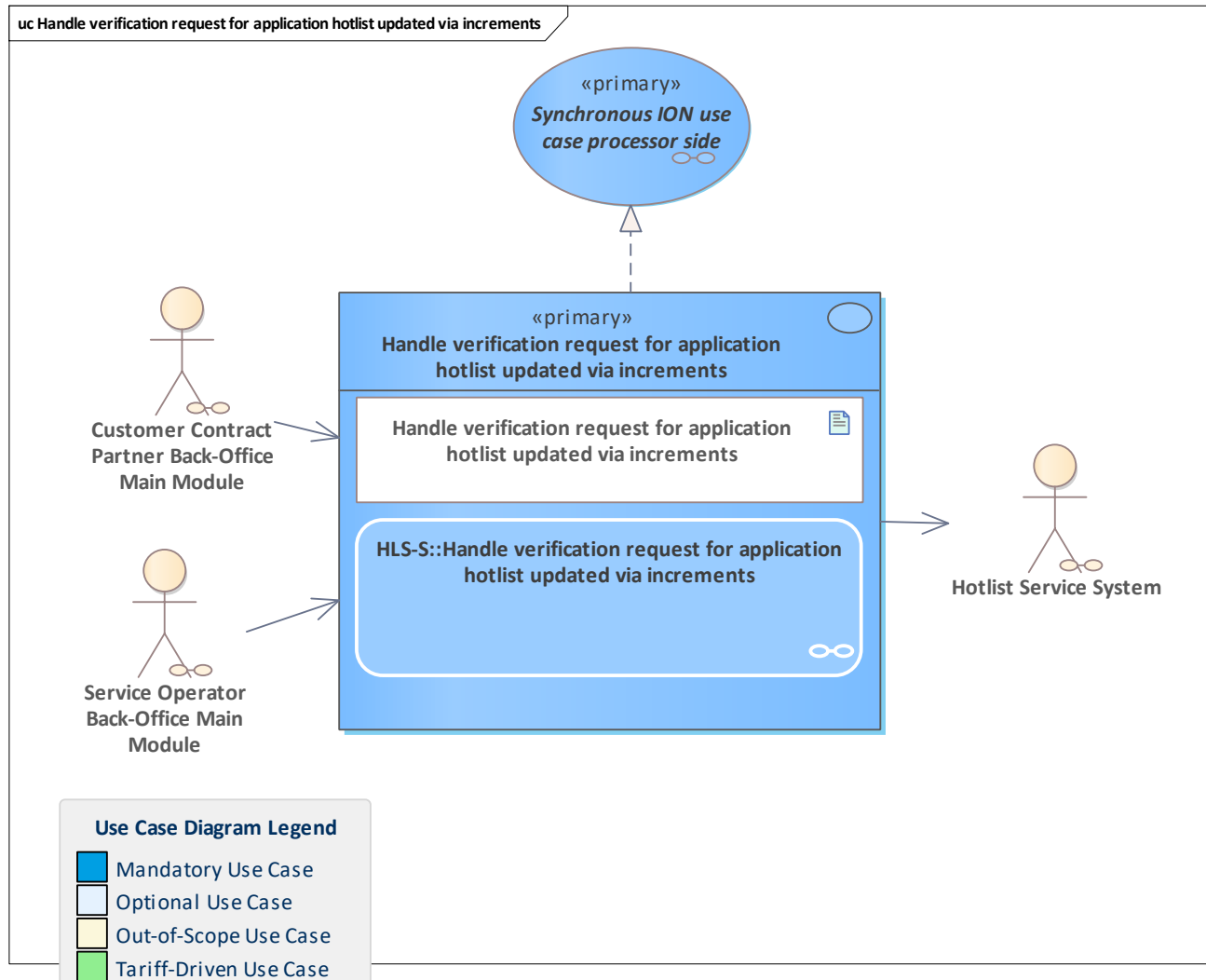
<b>Use Case</b>	<a href="#">Handle retrieval request for incremental application hotlist</a>
<b>Description</b>	After receiving a request for an application incremental hotlist, the hotlist service system processes the request by running checks (see activity diagram) and creating the application incremental 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.
<b>Initiating Actor</b>	<a href="#">Service Operator Back-Office Main Module</a> <a href="#">Customer Contract Partner Back-Office Main Module</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	<a href="#">Partner system already has hotlist inventory. The basic hotlist inventory must not be older than a configured period. (Default: 7 Days)</a>
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	





<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Get application incremental hotlist request : getIncrementalApplicationHotlist</a>
<b>Outputs</b>	<a href="#">Get application incremental hotlist response : getIncrementalApplicationHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E HL STILL GENERATING HOTLIST</a> <a href="#">E CO CYCLE NUMBER UNKNOWN OR TOO OLD</a> <a href="#">Get application incremental hotlist exception : getIncrementalApplicationHotlistException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Handle retrieval request for incremental application hotlist</a>

## 4.2.3 Handle verification request for application hotlist updated via increments

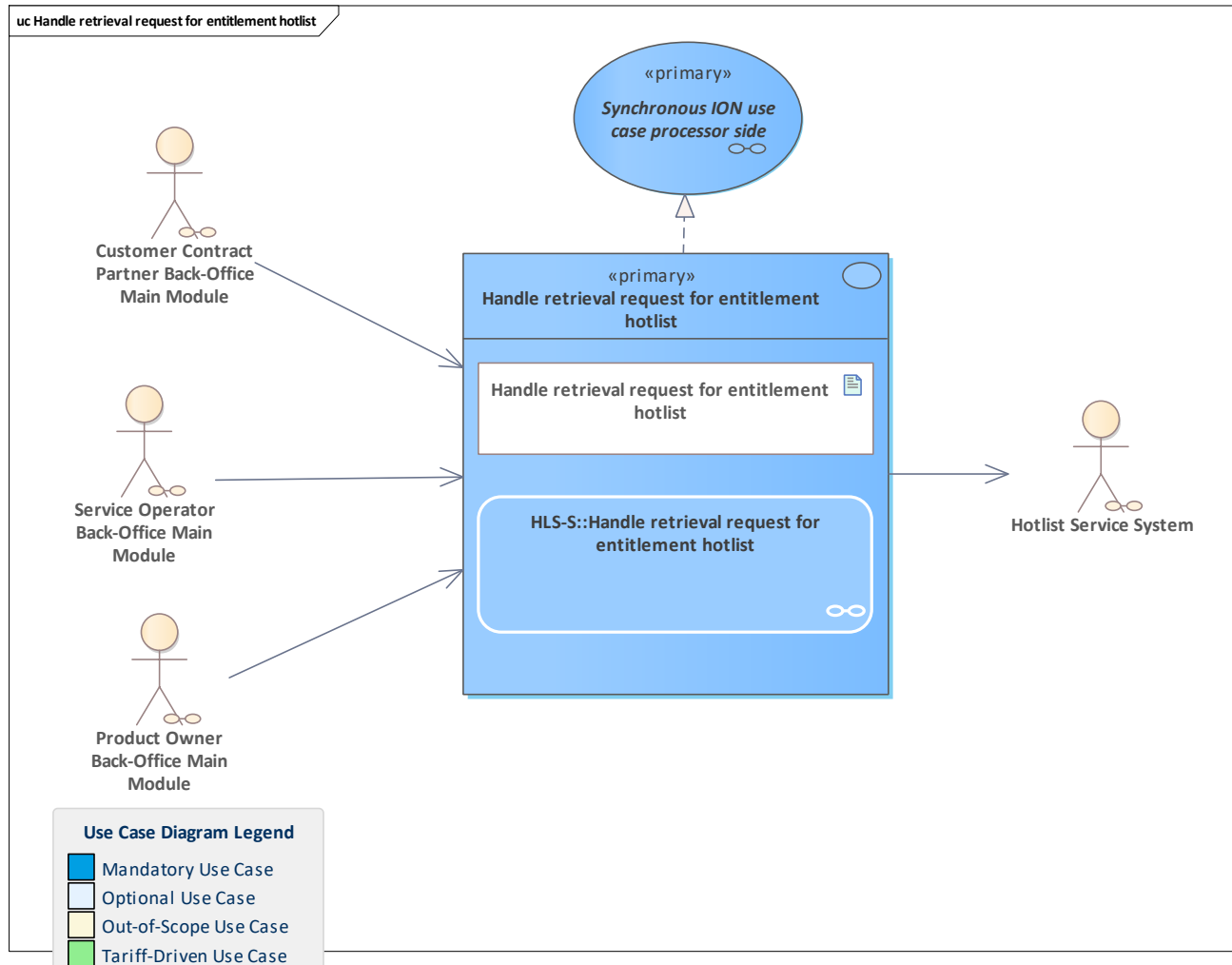


<b>Use Case</b>	<a href="#">Handle verification request for application hotlist updated via increments</a>
<b>Description</b>	The hotlist service system (HLS-S) retrieves an application hotlist list cycle reference together with the caller's computed checksum in the input and computes the check sum over 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 <a href="#">Checksum calculation for hotlist and action list verification</a> and <a href="#">Example calculation for an application hotlist inventory</a> .
<b>Initiating Actor</b>	<a href="#">Customer Contract Partner Back-Office Main Module</a> <a href="#">Service Operator Back-Office Main Module</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	



<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Verify application hotlist updated via increments : verifyApplicationHotlistUpdatedViaIncrements</a>
<b>Outputs</b>	<a href="#">Verify application hotlist updated via increments response : verifyApplicationHotlistUpdatedViaIncrementsResponse</a>
<b>Error Cases</b>	<a href="#">E_CO_INCORRECT_CHECKSUM</a> <a href="#">E_CO_CYCLE_NUMBER_UNKNOWN_OR_TOO_OLD</a> <a href="#">Verify application hotlist updated via increments exception : verifyApplicationHotlistUpdatedViaIncrementsException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Handle verification request for application hotlist updated via increments</a>

## 4.2.4 Handle retrieval request for entitlement hotlist

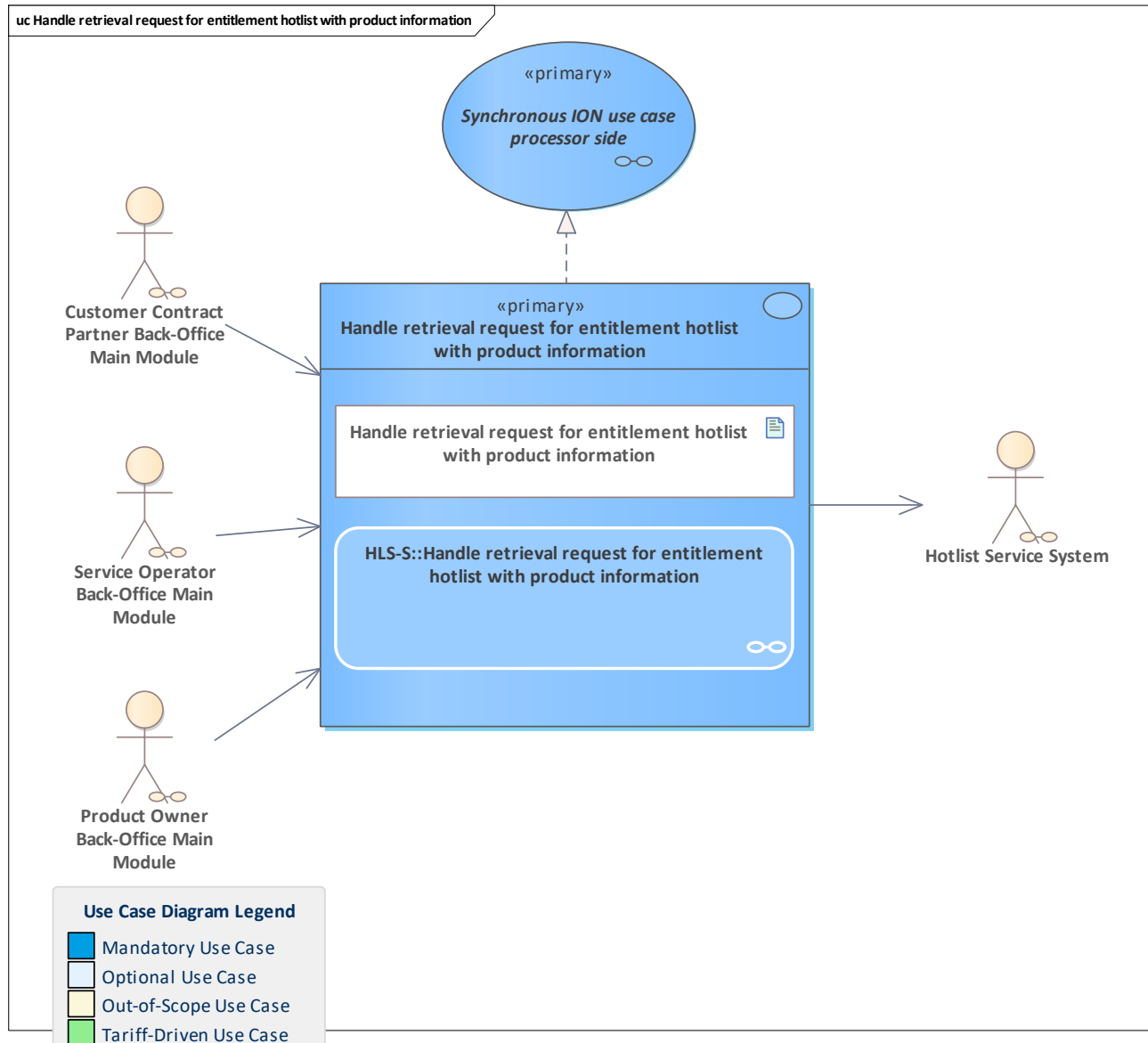


<b>Use Case</b>	<a href="#">Handle retrieval request for entitlement hotlist</a>
<b>Description</b>	After receiving a request for an entitlement hotlist, the hotlist service system processes the request by running checks (see activity diagram) and creating the current entitlement hotlist as a response.
<b>Initiating Actor</b>	<a href="#">Service Operator Back-Office Main Module</a> <a href="#">Customer Contract Partner Back-Office Main Module</a> <a href="#">Product Owner Back-Office Main Module</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	



<b>Inputs</b>	<a href="#">Get entitlement hotlist request : getEntitlementHotlist</a>
<b>Outputs</b>	<a href="#">Get entitlement hotlist response : getEntitlementHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E HL STILL GENERATING HOTLIST</a> <a href="#">Get entitlement hotlist exception : getEntitlementHotlistException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Handle retrieval request for entitlement hotlist</a>

## 4.2.5 Handle retrieval request for entitlement hotlist with product information

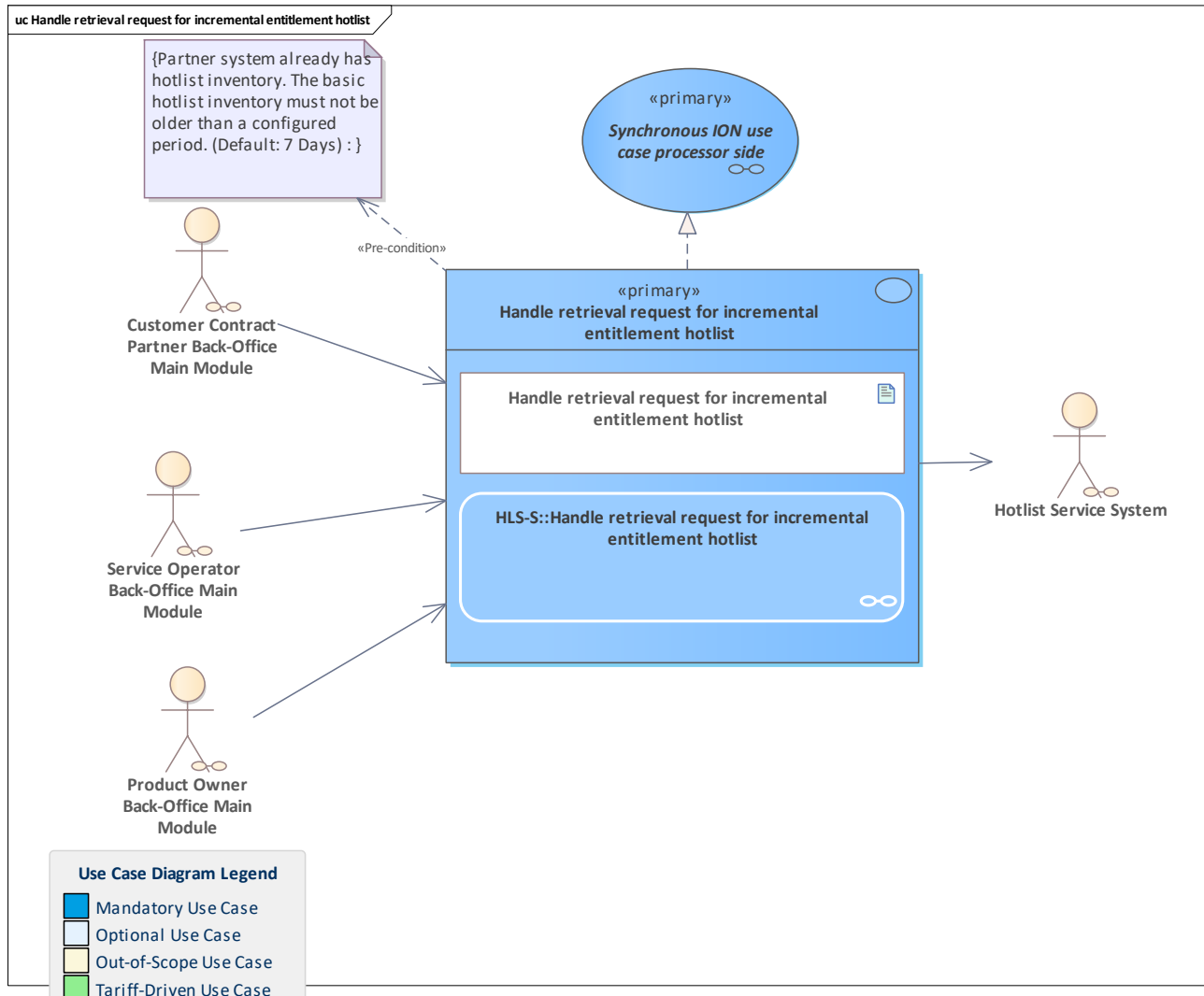


<b>Use Case</b>	<a href="#">Handle retrieval request for entitlement hotlist with product information</a>
<b>Description</b>	After receiving a request for an entitlement hotlist with product information, the hotlist service system processes the request by running checks (see activity diagram) and creating the current entitlement hotlist with product information as a response. In contrast to <a href="#">Handle retrieval request for entitlement hotlist</a> , each hotlist entry contains additional information about the product (see <a href="#">ProductId</a> ).
<b>Initiating Actor</b>	<a href="#">Service Operator Back-Office Main Module</a> <a href="#">Customer Contract Partner Back-Office Main Module</a> <a href="#">Product Owner Back-Office Main Module</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	



<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Get entitlement with product hotlist request : getEntitlementWithProductHotlist</a>
<b>Outputs</b>	<a href="#">Get entitlement with product hotlist response : getEntitlementWithProductHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E HL STILL GENERATING HOTLIST</a> <a href="#">Get entitlement with product hotlist exception : getEntitlementWithProductHotlistException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Handle retrieval request for entitlement hotlist with product information</a>

## 4.2.6 Handle retrieval request for incremental entitlement hotlist



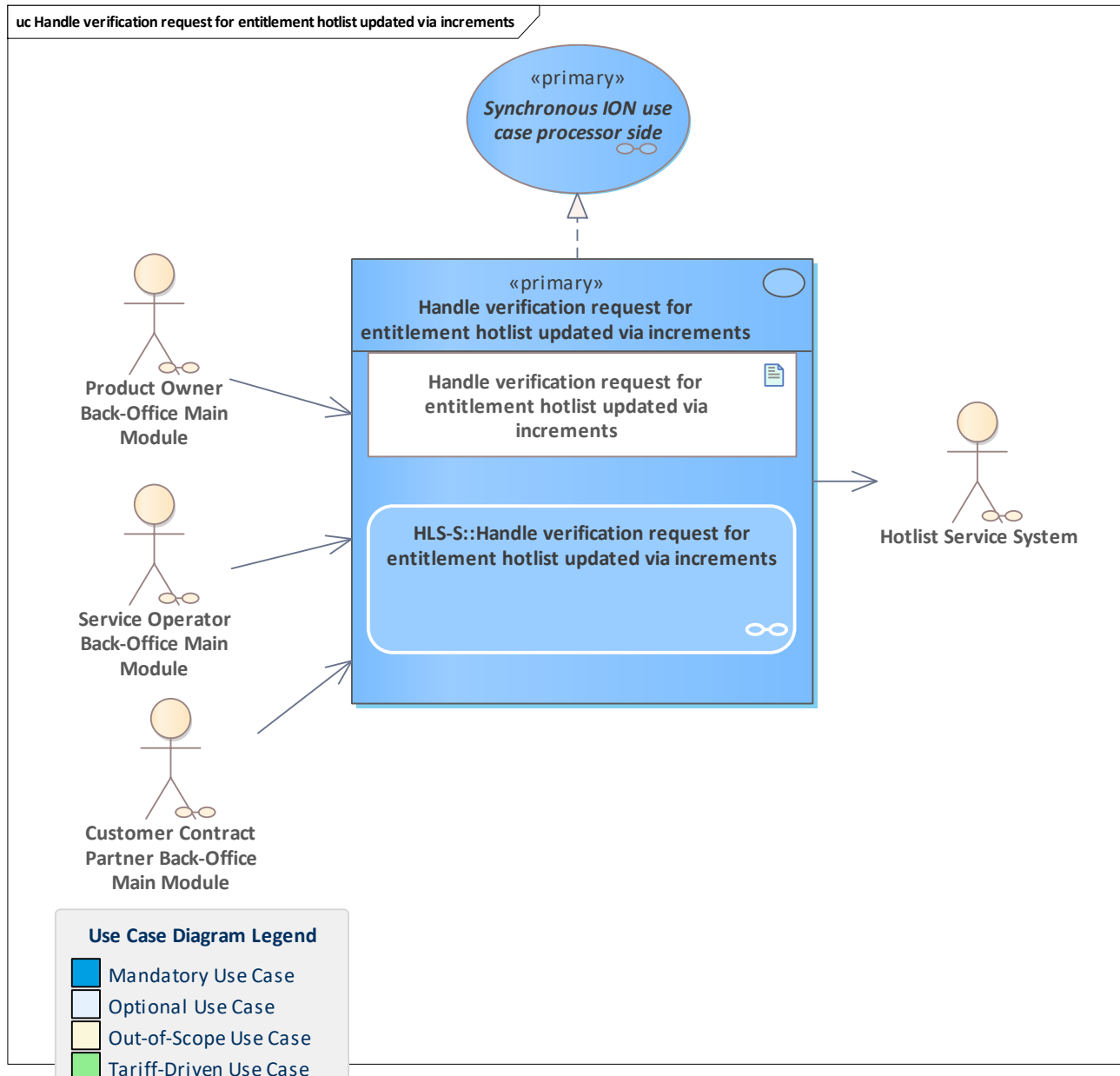
<b>Use Case</b>	<a href="#">Handle retrieval request for incremental entitlement hotlist</a>
<b>Description</b>	After receiving a request for an entitlement incremental hotlist, the hotlist service system processes the request by running checks (see activity diagram) and creating the incremental entitlement 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.
<b>Initiating Actor</b>	<a href="#">Customer Contract Partner Back-Office Main Module</a> <a href="#">Service Operator Back-Office Main Module</a> <a href="#">Product Owner Back-Office Main Module</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	<a href="#">Partner system already has hotlist inventory. The basic hotlist inventory must not be older than a configured period. (Default: 7 Days)</a>
<b>Postconditions</b>	





<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Get entitlement incremental hotlist request : <code>getIncrementalEntitlementHotlist</code></a>
<b>Outputs</b>	<a href="#">Get incremental entitlement hotlist response : <code>getIncrementalEntitlementHotlistResponse</code></a>
<b>Error Cases</b>	<a href="#">E HL STILL GENERATING HOTLIST</a> <a href="#">E CO CYCLE NUMBER UNKNOWN OR TOO OLD</a> <a href="#">Get entitlement incremental hotlist exception : <code>getIncrementalEntitlementHotlistException</code></a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Handle retrieval request for incremental entitlement hotlist</a>

## 4.2.7 Handle verification request for entitlement hotlist updated via increments

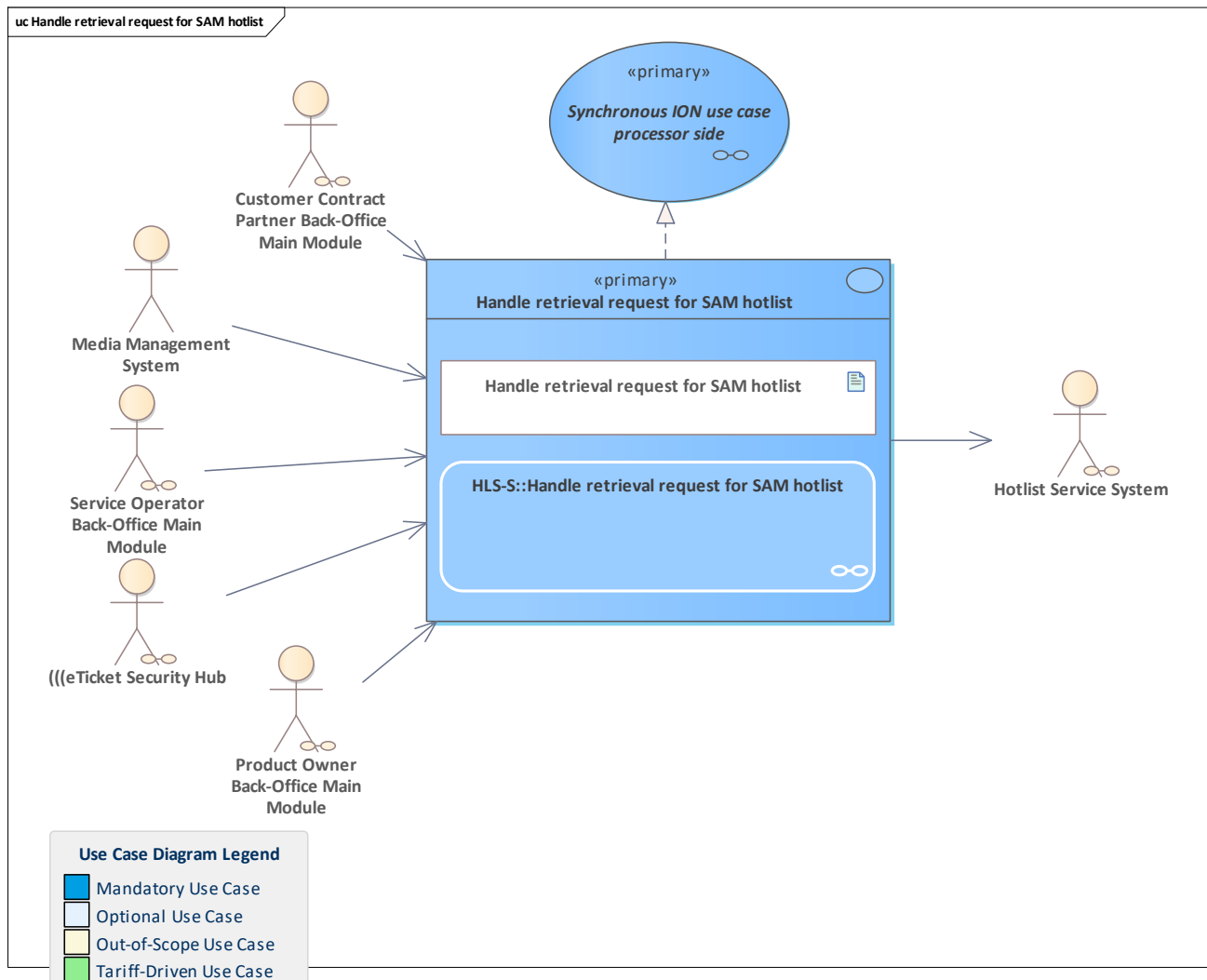


<b>Use Case</b>	<a href="#">Handle verification request for entitlement hotlist updated via increments</a>
<b>Description</b>	<p>The HLS-S retrieves an entitlement hotlist reference together with the caller's computed checksum in the input and computes the check sum over the full entitlement hotlist that belongs to the reference of the request.</p> <p>The HLS-S asserts the equality of the internally and externally computed checksum and reports the result back to the requestor. See also <a href="#">Checksum calculation for hotlist and action list verification</a> and <a href="#">Example calculation for an entitlement hotlist inventory</a>.</p>



<b>Initiating Actor</b>	<a href="#">Customer Contract Partner Back-Office Main Module</a> <a href="#">Service Operator Back-Office Main Module</a> <a href="#">Product Owner Back-Office Main Module</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Verify entitlement hotlist updated via increments : verifyEntitlementHotlistUpdatedViaIncrements</a>
<b>Outputs</b>	<a href="#">Verify entitlement hotlist updated via increments response : verifyEntitlementHotlistUpdatedViaIncrementsResponse</a>
<b>Error Cases</b>	<a href="#">E_CO_CYCLE_NUMBER_UNKNOWN_OR_TOO_OLD</a> <a href="#">E_CO_INCORRECT_CHECKSUM</a> <a href="#">Verify entitlement hotlist updated via increments exception : verifyEntitlementHotlistUpdatedViaIncrementsException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Handle verification request for entitlement hotlist updated via increments</a>

## 4.2.8 Handle retrieval request for SAM hotlist

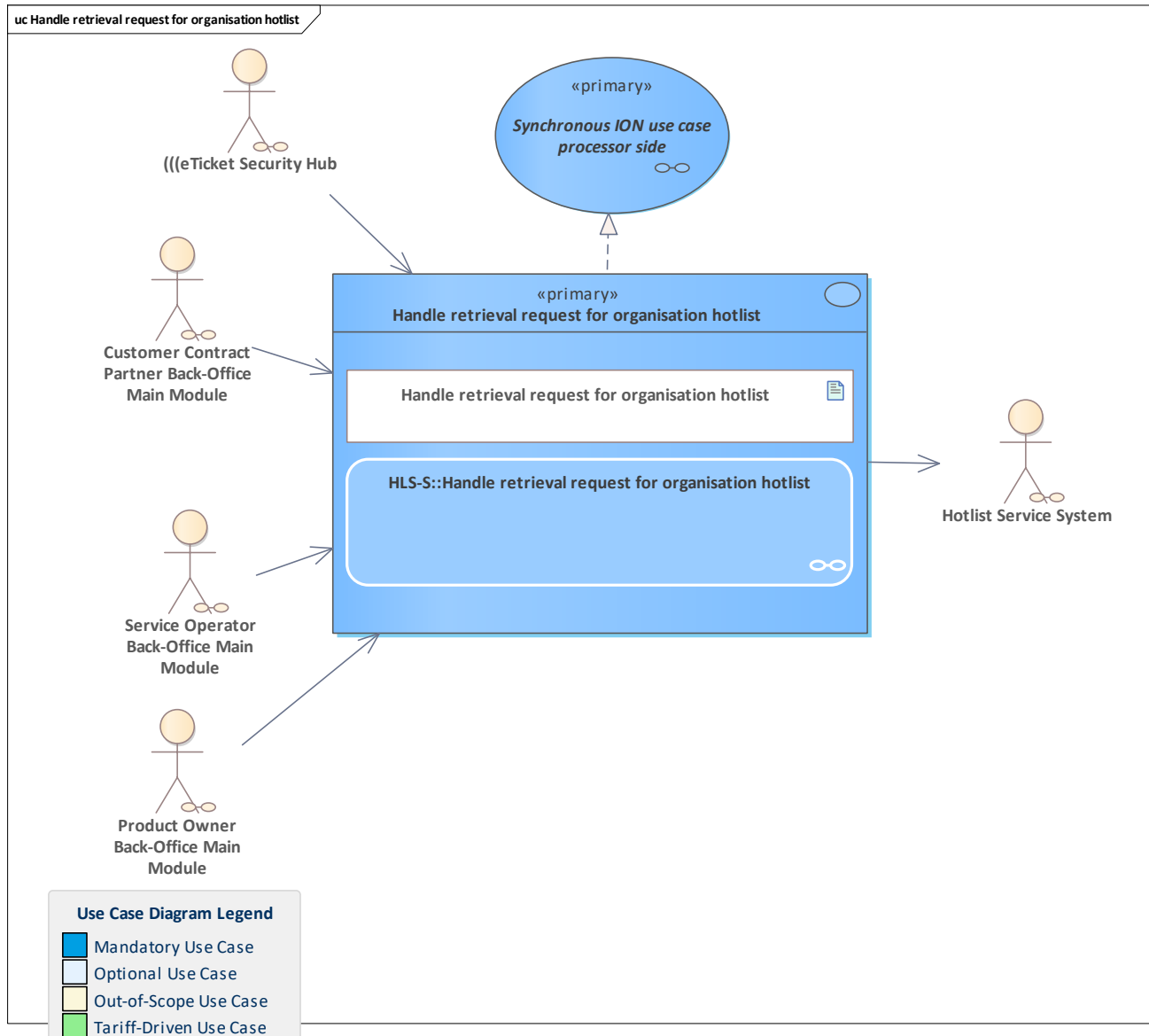


<b>Use Case</b>	<a href="#">Handle retrieval request for SAM hotlist</a>
<b>Description</b>	After receiving a request for a SAM hotlist, the hotlist service system processes the request by running checks (see activity diagram) and creating the current SAM hotlist as a response.
<b>Initiating Actor</b>	<a href="#">Customer Contract Partner Back-Office Main Module</a> <a href="#">Service Operator Back-Office Main Module</a> <a href="#">Product Owner Back-Office Main Module</a> <a href="#">Media Management System</a> <a href="#">(((eTicket Security Hub</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	



<b>Inputs</b>	<a href="#">Get SAM hotlist request : getSamHotlist</a>
<b>Outputs</b>	<a href="#">Get SAM hotlist response : getSamHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E HL STILL GENERATING HOTLIST</a> <a href="#">Get SAM hotlist exception : getSamHotlistException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Handle retrieval request for SAM hotlist</a>

## 4.2.9 Handle retrieval request for organisation hotlist

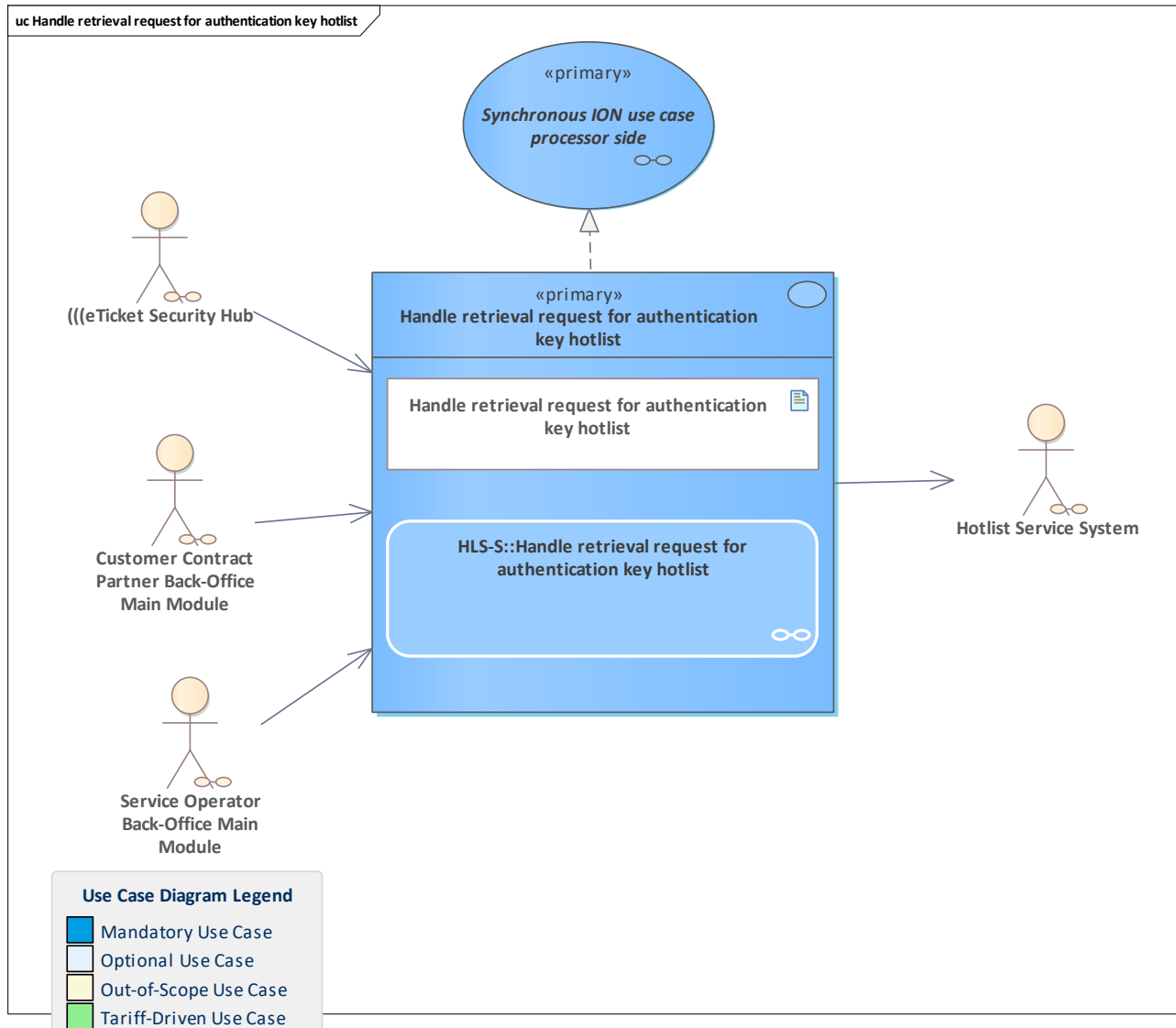


<b>Use Case</b>	<a href="#">Handle retrieval request for organisation hotlist</a>
<b>Description</b>	After receiving a request for an organisation hotlist, the hotlist service system processes the request by running checks (see activity diagram) and creating the current organisation hotlist as a response.
<b>Initiating Actor</b>	<a href="#">Customer Contract Partner Back-Office Main Module</a> <a href="#">Service Operator Back-Office Main Module</a> <a href="#">Product Owner Back-Office Main Module</a> <a href="#">(((eTicket Security Hub</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	



<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Get organisation hotlist request : getOrganisationHotlist</a>
<b>Outputs</b>	<a href="#">Get organisation hotlist response : getOrganisationHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E_HL_STILL_GENERATING_HOTLIST</a> <a href="#">Get organisation hotlist exception : getOrganisationHotlistException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Handle retrieval request for organisation hotlist</a>

## 4.2.10 Handle retrieval request for authentication key hotlist



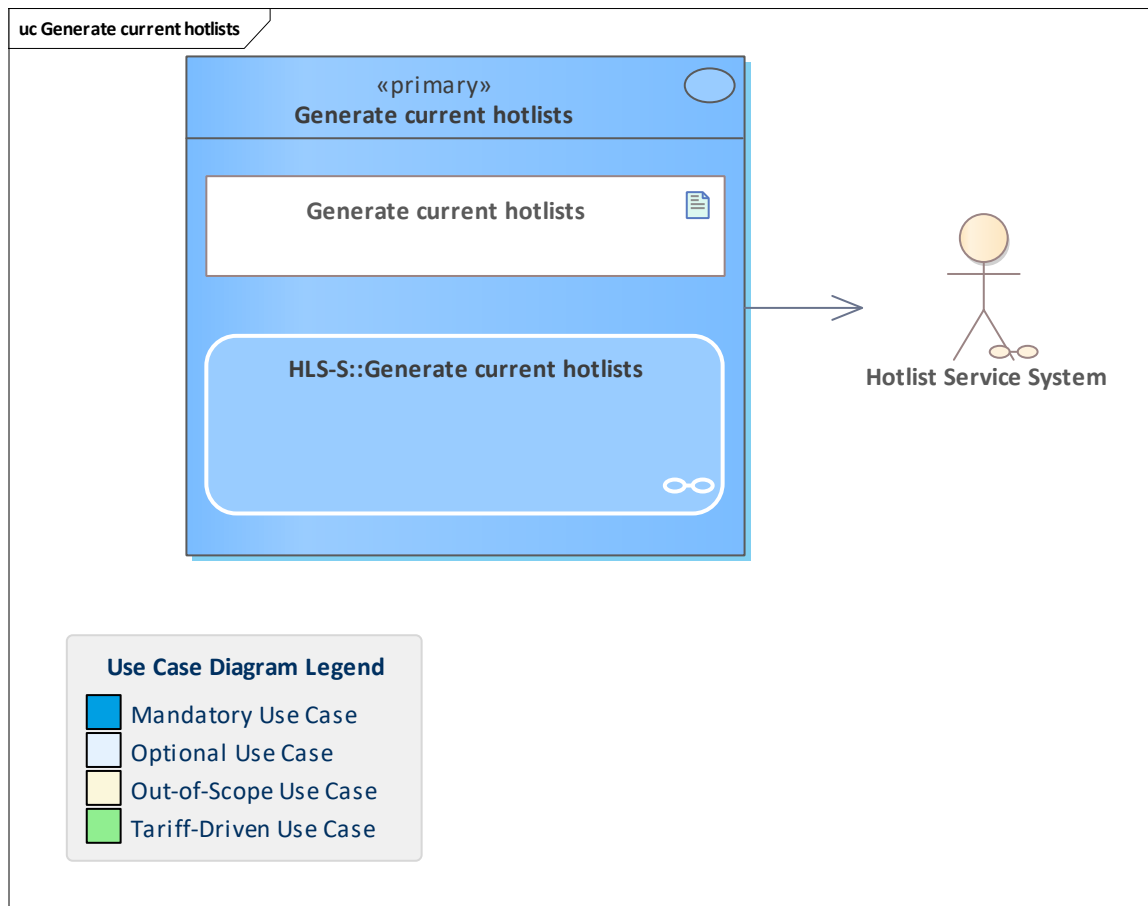
<b>Use Case</b>	<a href="#">Handle retrieval request for authentication key hotlist</a>
<b>Description</b>	After receiving a request for an authentication key hotlist, the hotlist service system processes the request by running checks (see activity diagram) and creating the current authentication key hotlist as a response.
<b>Initiating Actor</b>	<a href="#">Customer Contract Partner Back-Office Main Module</a> <a href="#">Service Operator Back-Office Main Module</a> <a href="#">(((eTicket Security Hub</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	





<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Get authentication key hotlist request :</a> <a href="#">getAuthenticationKeyHotlist</a>
<b>Outputs</b>	<a href="#">Get authentication key hotlist response :</a> <a href="#">getAuthenticationKeyHotlistResponse</a>
<b>Error Cases</b>	<a href="#">E HL STILL GENERATING HOTLIST</a> <a href="#">Get authentication key hotlist exception :</a> <a href="#">getAuthenticationKeyHotlistException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Handle retrieval request for authentication key hotlist</a>

## 4.2.11 Generate current hotlists



<b>Use Case</b>	<a href="#">Generate current hotlists</a>
<b>Description</b>	Current hotlist inventories are generated and stored in the database for a later hotlist compilation when requested.
<b>Initiating Actor</b>	
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	
<b>Base Activity</b>	
<b>Inputs</b>	
<b>Outputs</b>	
<b>Error Cases</b>	
<b>Activity Diagram</b>	<a href="#">HLS-S::Generate current hotlists</a>



## 5 Hotlist Service Configuration

Functionality bundle that covers all use cases for the configuration of the hotlist service.

### 5.1 Overview

Handle request to add product acceptance entry to hotlist service configuration

Handle request to remove product acceptance entry from hotlist service configuration

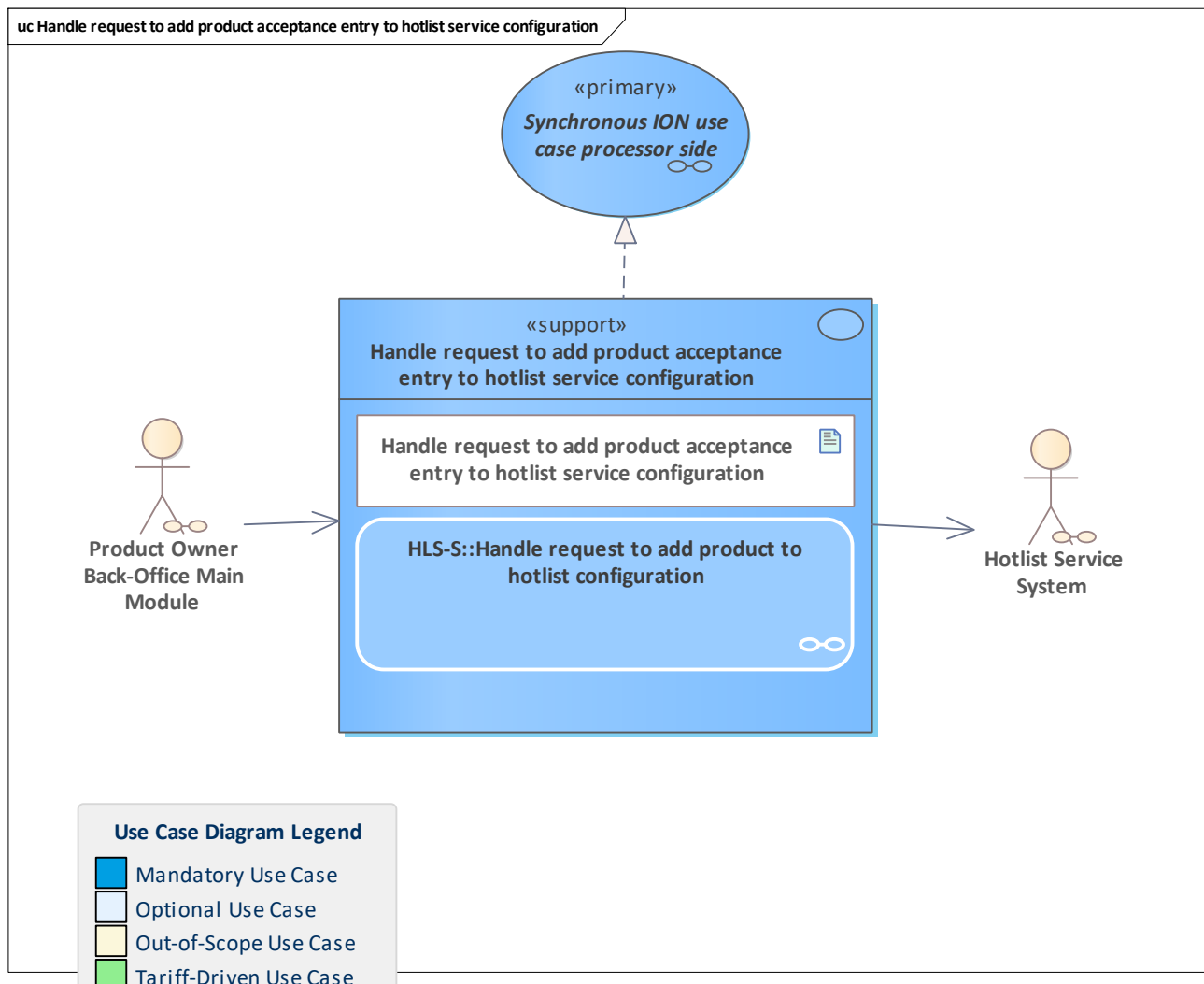
Handle request to remove product acceptance from participants

Handle request for product acceptance configuration list

Retrieve and distribute organisation list

### 5.2 Use Cases

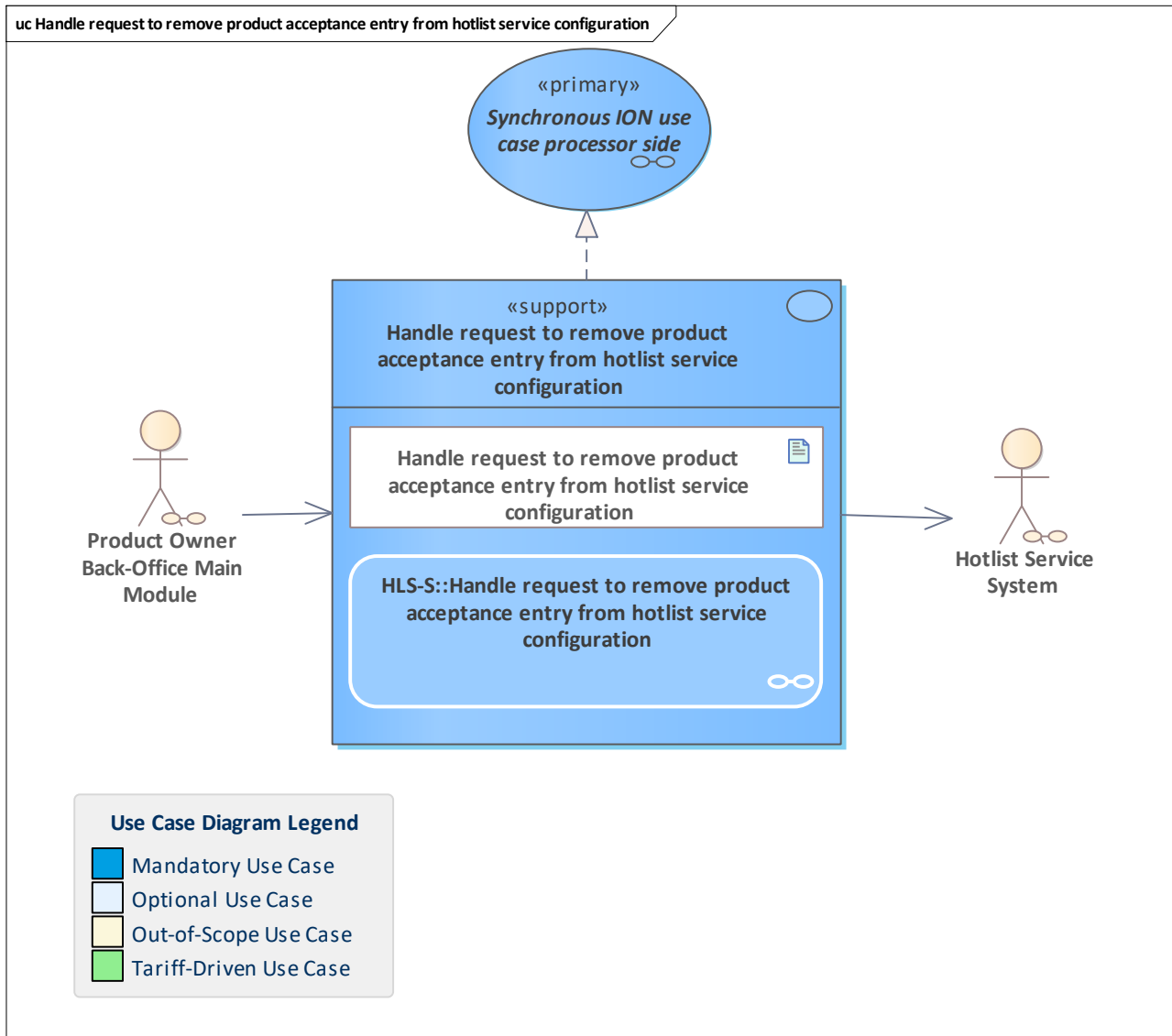
#### 5.2.1 Handle request to add product acceptance entry to hotlist service configuration





<b>Use Case</b>	<a href="#">Handle request to add product acceptance entry to hotlist service configuration</a>
<b>Description</b>	<p>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.</p> <p>Please note that interoperable products are not allowed to be in the configuration list.</p>
<b>Initiating Actor</b>	<a href="#">Product Owner Back-Office Main Module</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Add product acceptance configuration : addProductAcceptanceConfiguration</a>
<b>Outputs</b>	<a href="#">Add product acceptance configuration response : addProductAcceptanceConfigurationResponse</a>
<b>Error Cases</b>	<a href="#">E HL UNKNOWN ORGANISATION FOR PRODUCT ACCEPTANCE</a> <a href="#">E HL PRODUCT ACCEPTANCE ALREADY EXISTS</a> <a href="#">E HL INTEROPERABLE PRODUCT NOT ALLOWED IN PRODUCT ACCEPTANCE</a> <a href="#">Add product acceptance configuration exception : addProductAcceptanceConfigurationException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Handle request to add product to hotlist configuration</a>

## 5.2.2 Handle request to remove product acceptance entry from hotlist service configuration

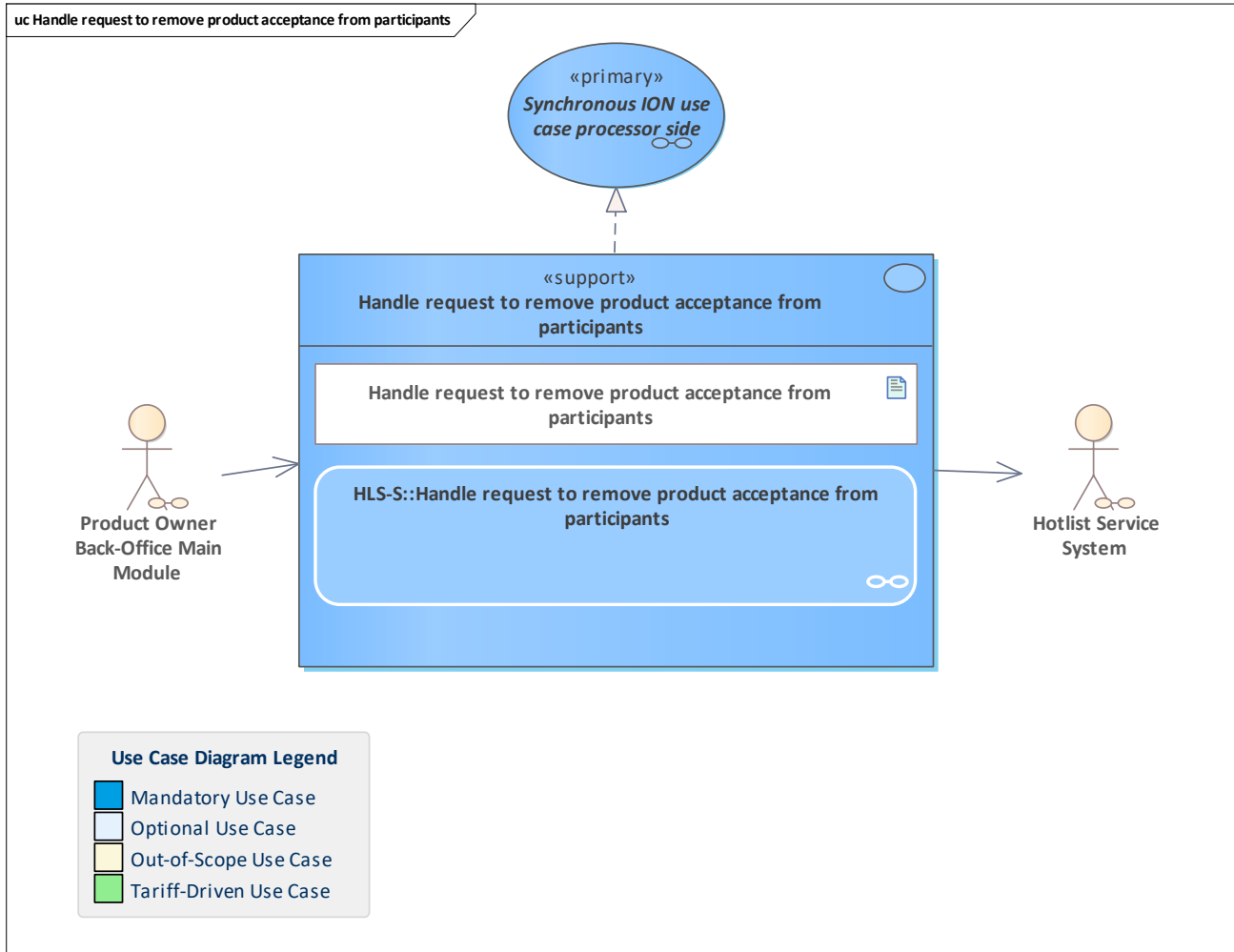


<b>Use Case</b>	<a href="#">Handle request to remove product acceptance entry from hotlist service configuration</a>
<b>Description</b>	The hotlist service system receives a request to remove a product acceptance entry from its configuration. Only product owners are allowed to remove an acceptance configuration 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.
<b>Initiating Actor</b>	<a href="#">Product Owner Back-Office Main Module</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	



<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Remove product acceptance configuration : removeProductAcceptanceConfiguration</a>
<b>Outputs</b>	<a href="#">Remove product acceptance configuration response : removeProductAcceptanceConfigurationResponse</a>
<b>Error Cases</b>	<a href="#">E HL UNKNOWN PRODUCT ACCEPTANCE Remove product acceptance configuration exception : removeProductAcceptanceConfigurationException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Handle request to remove product acceptance entry from hotlist service configuration</a>

## 5.2.3 Handle request to remove product acceptance from participants



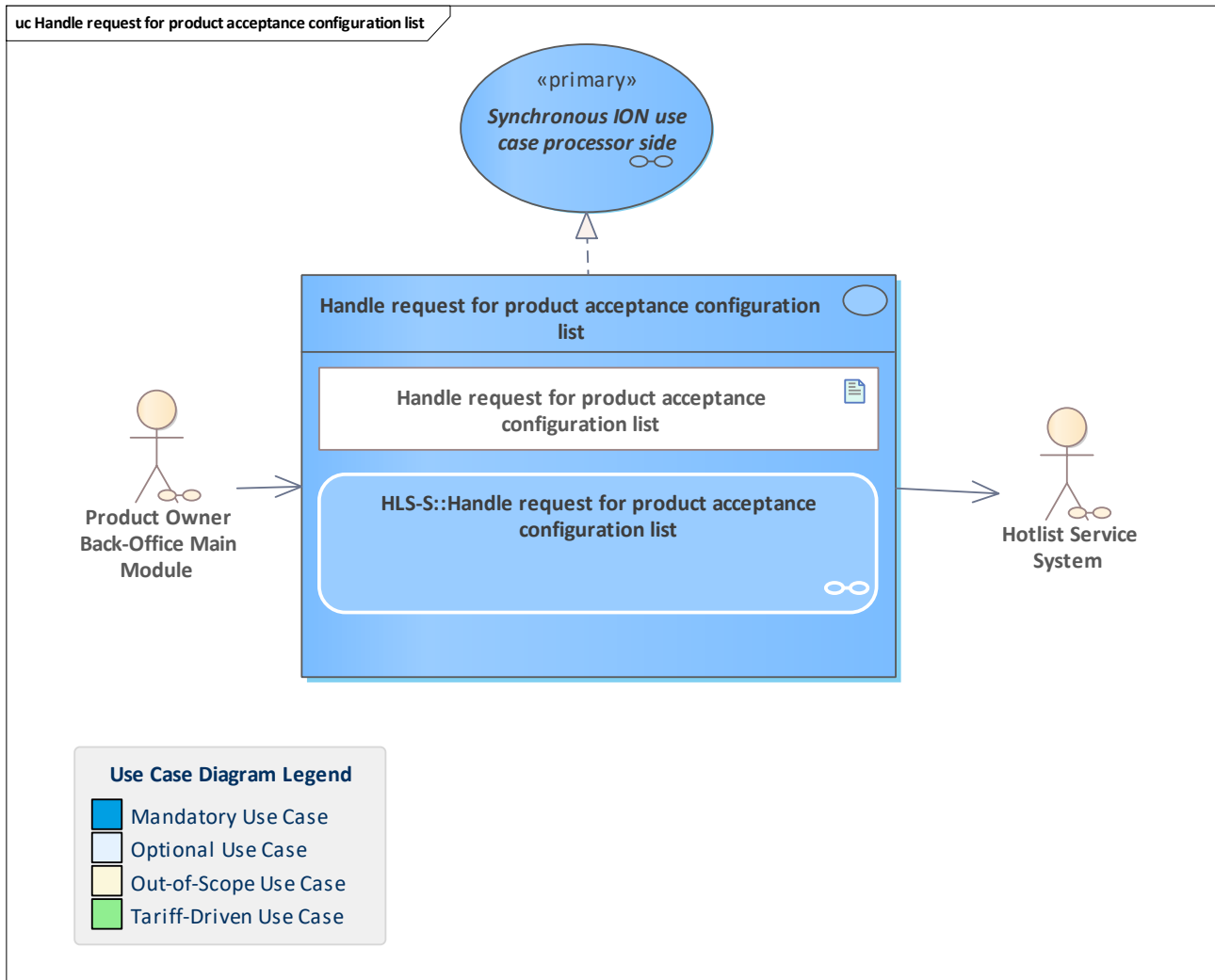
<b>Use Case</b>	<a href="#">Handle request to remove product acceptance from participants</a>
<b>Description</b>	The hotlist service system receives a request to remove all acceptance entries for a product from its configuration for all acceptance candidates. Only product owners are allowed to remove an acceptance configuration for their own products. The request contains a product expiry date which must be taken into account when handling the request.
<b>Initiating Actor</b>	<a href="#">Product Owner Back-Office Main Module</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>





Base Activity	
Inputs	<a href="#">Remove product acceptance from participants :</a> <a href="#">removeProductAcceptanceFromParticipants</a>
Outputs	<a href="#">Remove product acceptance from participants response :</a> <a href="#">removeProductAcceptanceFromParticipantsResponse</a>
Error Cases	<a href="#">E HL UNKNOWN PRODUCT ACCEPTANCE</a> <a href="#">E HL PRODUCT EXPIRY DATE TOO FAR</a> <a href="#">Remove product acceptance from participants exception :</a> <a href="#">removeProductAcceptanceFromParticipantsException</a>
Activity Diagram	<a href="#">HLS-S::Handle request to remove product acceptance from participants</a>

## 5.2.4 Handle request for product acceptance configuration list

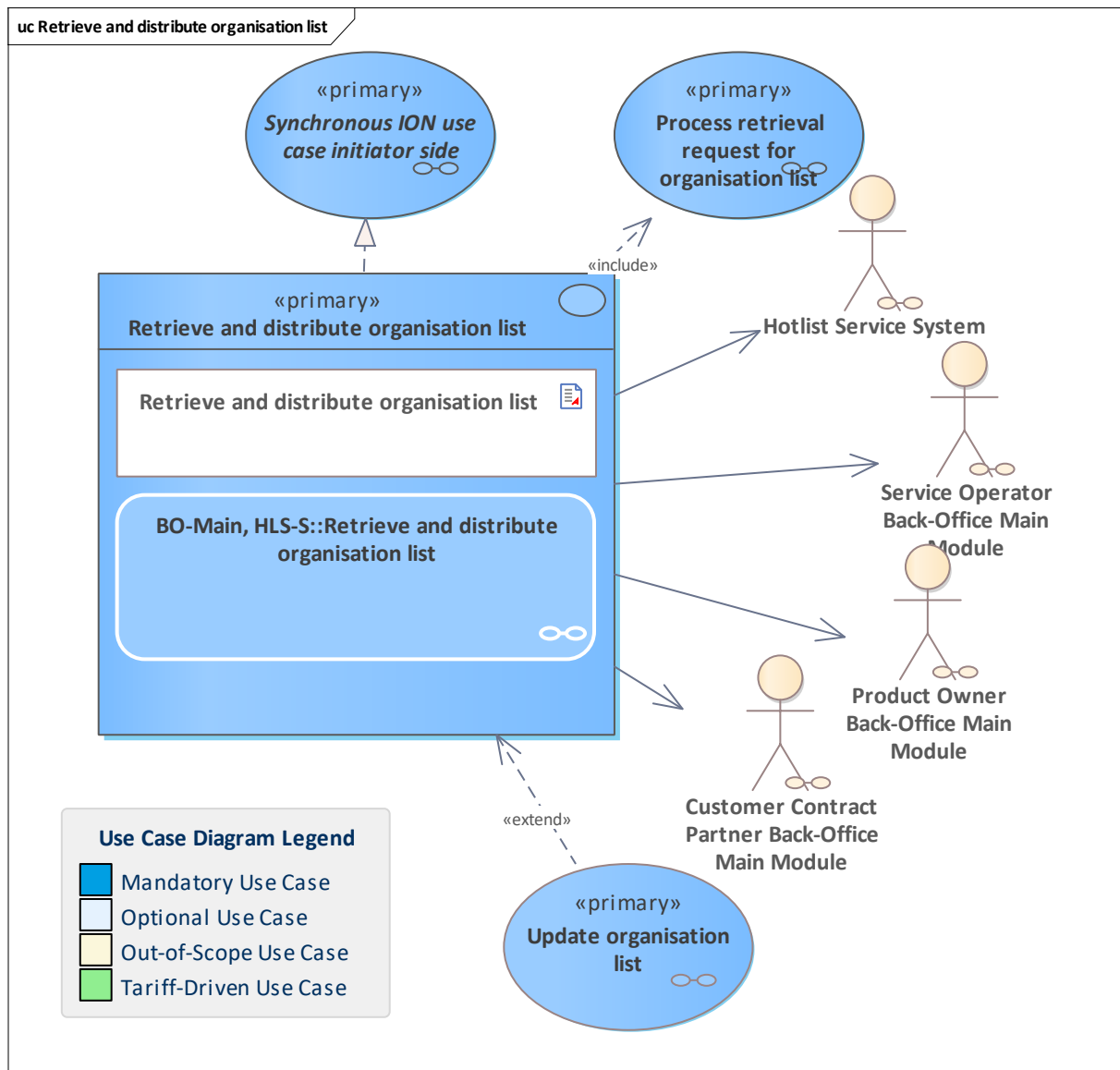


<b>Use Case</b>	<a href="#">Handle request for product acceptance configuration list</a>
<b>Description</b>	The hotlist service system provides its product acceptance configuration as a compressed list of product acceptance entries.
<b>Initiating Actor</b>	<a href="#">Product Owner Back-Office Main Module</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Get product acceptance configuration list :</a> <a href="#">getProductAcceptanceConfigurationList</a>



<b>Outputs</b>	<a href="#">Get product acceptance configuration list response : getProductAcceptanceConfigurationListResponse</a>
<b>Error Cases</b>	<a href="#">Get product acceptance configuration list exception : getProductAcceptanceConfigurationListException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Handle request for product acceptance configuration list</a>

## 5.2.5 Retrieve and distribute organisation list



<b>Use Case</b>	<a href="#">Retrieve and distribute organisation list</a>
<b>Description</b>	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.
<b>Initiating Actor</b>	
<b>Reacting Actor</b>	<a href="#">Customer Contract Partner Back-Office Main Module</a> <a href="#">Service Operator Back-Office Main Module</a> <a href="#">Product Owner Back-Office Main Module</a> <a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases</b>	<a href="#">Update organisation list</a>



<b>(Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	<a href="#">Process retrieval request for organisation list</a>
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case initiator side</a>
<b>Base Activity</b>	
<b>Inputs</b>	
<b>Outputs</b>	
<b>Error Cases</b>	
<b>Activity Diagram</b>	<a href="#">BO-Main, HLS-S::Retrieve and distribute organisation list</a>

## 6 Hotlist Reports

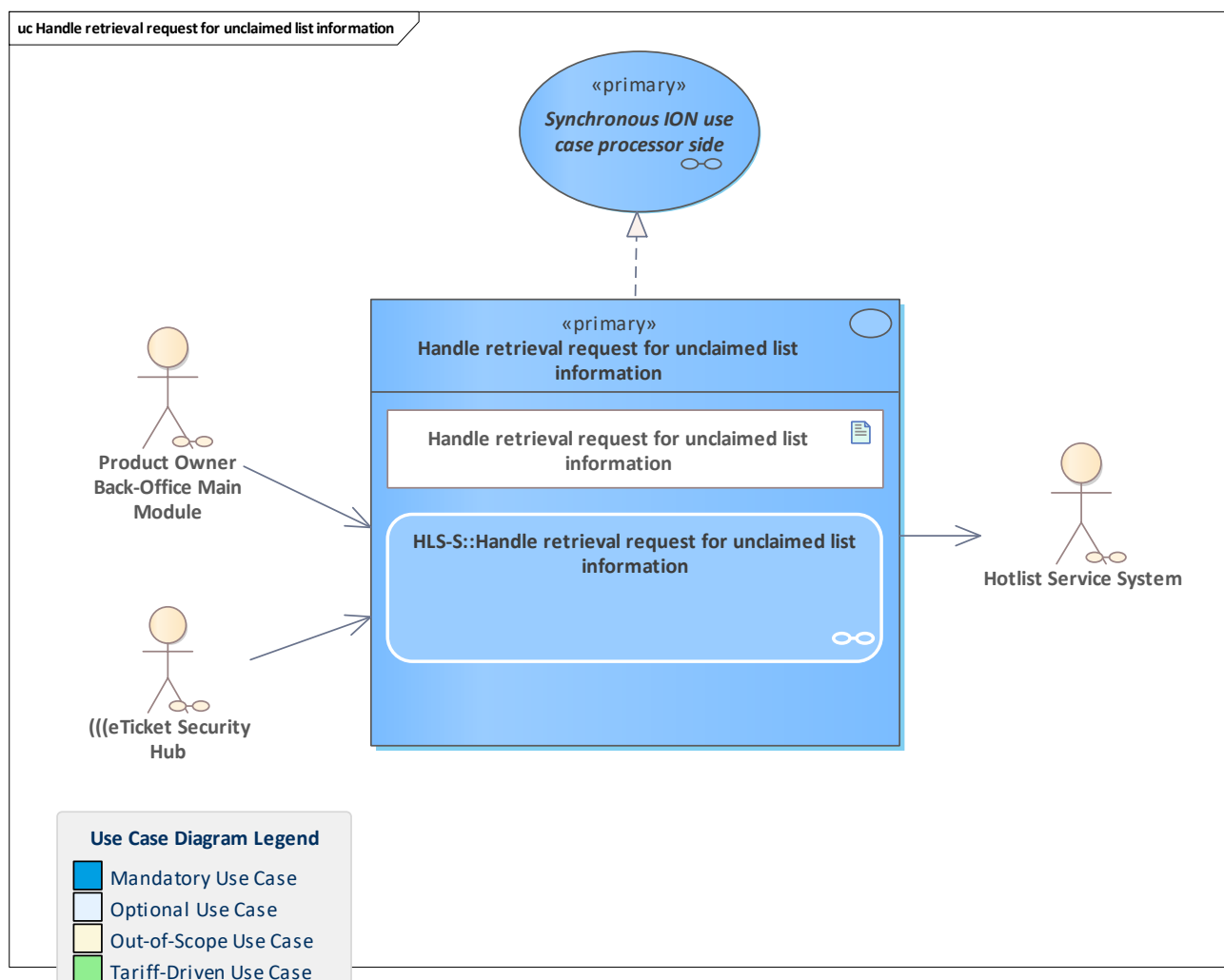
Functionality bundle that covers the available reports provided by the hotlist service.

### 6.1 Overview

Handle retrieval request for unclaimed list information

### 6.2 Use Cases

#### 6.2.1 Handle retrieval request for unclaimed list information



<b>Use Case</b>	<u>Handle retrieval request for unclaimed list information</u>
<b>Description</b>	<p>The hotlist service system creates and sends the unclaimed list information to the requesting PO.</p> <p>The hotlist service system gathers the information for all related</p>



	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.
<b>Initiating Actor</b>	<a href="#">Product Owner Back-Office Main Module</a> <a href="#">(((eTicket Security Hub</a>
<b>Reacting Actor</b>	<a href="#">Hotlist Service System</a>
<b>Preconditions</b>	
<b>Postconditions</b>	
<b>Linked Use Cases (Extended By)</b>	
<b>Linked Use Cases (Includes)</b>	
<b>Linked Use Cases (Realises)</b>	<a href="#">Synchronous ION use case processor side</a>
<b>Base Activity</b>	
<b>Inputs</b>	<a href="#">Get unclaimed list information : getUnclaimedListInformation</a>
<b>Outputs</b>	<a href="#">Get unclaimed list information response : getUnclaimedListInformationResponse</a>
<b>Error Cases</b>	<a href="#">E_CO_CYCLE_NUMBER_UNKNOWN_OR_TOO_OLD</a> <a href="#">Get unclaimed list information exception : getUnclaimedListInformationException</a>
<b>Activity Diagram</b>	<a href="#">HLS-S::Handle retrieval request for unclaimed list information</a>

