

Software Requirements Specification

for

Federation Architecture for Composed Infrastructure Services (FACIS)

Federation Architecture Pattern Principal Credential Issuance (PCI)

FACIS.FAP_PCI

Version 1.0 (February 2026)

Published by
eco – Association of the Internet Industry (eco – Verband der Internetwirtschaft e.V.)
Lichtstrasse 43h
50825 Cologne, Germany

Copyright © eco Association of the Internet Industry (eco – Verband der Internetwirtschaft e.V.)

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA

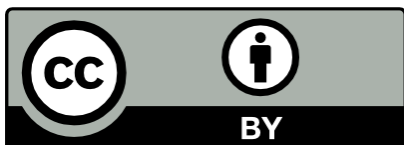


Table of Contents

Conformance Language	5
1. Executive Summary	5
2. Background & Context	5
3. Scope	6
4. Conceptual Architecture.....	7
5. Technical Architecture.....	9
6. Functional Requirements	10
6.1 Key Management.....	10
6.2 Public Endpoints	10
6.3 Participant Tenant Administration	11
6.4 Issuer Administration	11
6.5 Issuer Plugin	13
6.6 Participant Data Repository	14
6.7 Provider Tenant Administration.....	14
7. Interfaces & Data Models.....	15
7.1 Tenant Administration UX.....	15
7.2 Tenant Registration UX	16
7.3 Issuer Administration UX	16
7.4 Issuer Plugin (Participant Connection).....	17
8. Security & Trust.....	18
8.1 Transport Security	18
8.2 Identity & Authentication	18
8.3 Authorization & Policy Enforcement.....	19
8.4 Data Protection	19
9. Deployment & Operations	19
9.1 Deployment.....	19
9.2 Operational Requirements	20
10. Standards & Protocols.....	21
10.1 UX Framework.....	21
10.2 Participant Data Protocol	22

10.3 GitHub Requirements	22
11. Validation & Acceptance Criteria	23
11.1 Tenant Management Milestone	23
11.2 Issuer Management Milestone	24
11.3 E2E Issuance Milestone	24
11.4 Second Cluster Deployment Milestone	25
11.5 GitHub Finalization Milestone	25
12. Appendices	26
Appendix A: Glossary	26
Appendix B: Tenant-specific Ingress Rule	28

Conformance Language

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in RFC 2119¹.

1. Executive Summary

The purpose of this specification is to define the requirements for a multi-tenant principal issuer cloud SaaS solution utilizing the Eclipse XFSC tool stack², including the OCM W-Stack, PCM Mobile, Status List Provider, SD-JWT Service, Custom Policy Agent, Crypto Provider Service, and the Orchestration Engine (ORCE). The solution should provide a ready-to-deploy, white-labeled application that can be customized and used by multiple enterprise clients.

The solution shall simplify the OID4VCI issuance process across enterprise domains by providing an intuitive and integrated technology stack, reducing the technical barriers for small and medium-sized enterprises. The solution shall also optimize resource consumption and ensure ease of setup, deployment, and ongoing maintenance.

2. Background & Context

In the self-sovereign identity (SSI) ecosystem, verifiable credentials (VCs) are a fundamental component of every proof. However, the issuance of credentials involves a set of complex and interdependent activities that must be addressed to enable secure, compliant, and scalable usage.

In particular, the following challenges must be resolved prior to effective credential issuance:

- Definition and management of credential schemas,
- Maintenance of issuer metadata (e.g. OIDC configuration, cryptographic keys, branding assets),
- Establishment of standardized issuance flows,
- Traceability of credential issuance and lifecycle history,
- Management of credential revocation mechanisms,
- Reliable identification and binding of credentials to their holders,
- Compliance with applicable trust frameworks and regulatory requirements (e.g. eIDAS 2.0).

When addressed in isolation, these functional requirements must be re-implemented for each individual use case, resulting in increased complexity and reduced consistency across an organization. The solution defined in this specification shall address this gap by providing a standardized and scalable credential issuance framework that can be applied across multiple use cases and organizational contexts. Contextually, the solution shall operate within the following business environment:

¹ <https://rfc-editor.org/rfc/rfc2119>

² <https://github.com/eclipse-xfsc/docs>

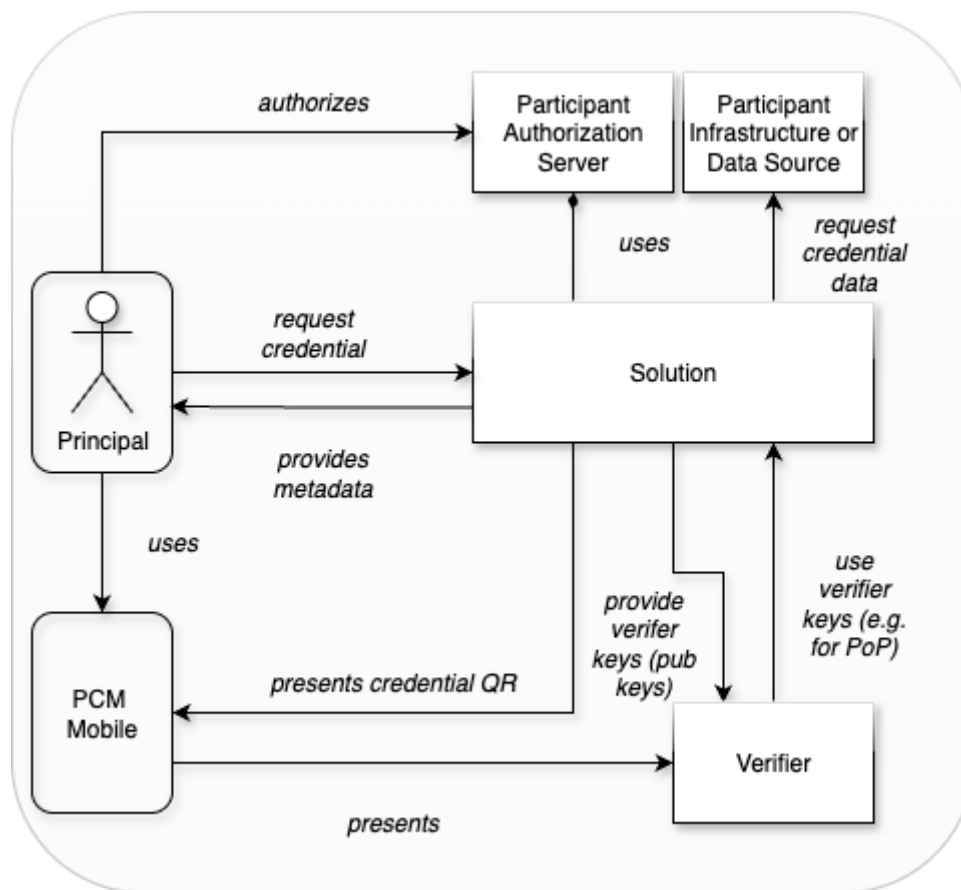


Figure 1 Solution Context

3. Scope

This section defines the scope of features of the requested solution. *In Scope* features must be implemented in accordance with this specification. *Out of Scope* features must not be implemented within this specification.

In Scope

The scope of the solution includes the following components and activities:

- Setup of ready-to-use deployment on top of existing deployments,
- Implementation of an issuer plugin(s),
- Utilization of ORCE for web-based issuing flow, including potential implementation of new ORCE nodes and/or code extensions,
- Provision of issuance flow,
- Technical documentation for the architecture of the solution, components, and deployment,
- Documentation describing the steps required to achieve EUDI and ETSI TS 119 471 compliance, including identification of components requiring modification and estimation of the effort needed to reach full compatibility (e.g., relying party registration processes, key management, and handling procedures),
- Operations and SecDevOps documentation.

Out of Scope

The following items are explicitly out of scope for this specification:

- Modification of existing XFSC components, unless explicitly agreed with the client through a formal alignment process,
- Testing with external wallets (only PCM Mobile is required),
- Development of XFSC deployments for TSA, OCM W-Stack, Status List Service (except for required modifications), and SD-JWT Service
- Wallet implementation,
- Full EUDI- and ETSI TS 119 471-compliant implementation across all components.

4. Conceptual Architecture

The conceptual architecture of the solution comprises authentication, credential issuance flow orchestration, and the credential issuance service itself. The credential issuance flow shall be orchestrated by ORCE, which manages all steps between authorization and issuance. ORCE shall support tenant-specific and issuer-specific variations of the issuance process.

The solution shall enable configurable issuance scenarios, including single-credential and multiple-credential issuance, as well as optional intermediate steps such as forms, questionnaires, or additional validation procedures. These variations shall be implemented through appropriate ORCE nodes.

An issuer plugin shall complement the orchestration layer by providing issuer metadata, signing integration, and issuer administration capabilities.

The issuer administration page and authentication page shall be provided as static pages that support per-tenant customization, including branding elements such as logos and style configurations. During tenant registration, the solution shall allow configuration of the authentication server, TLS settings for principal data endpoints, and branding customization prior to deployment.

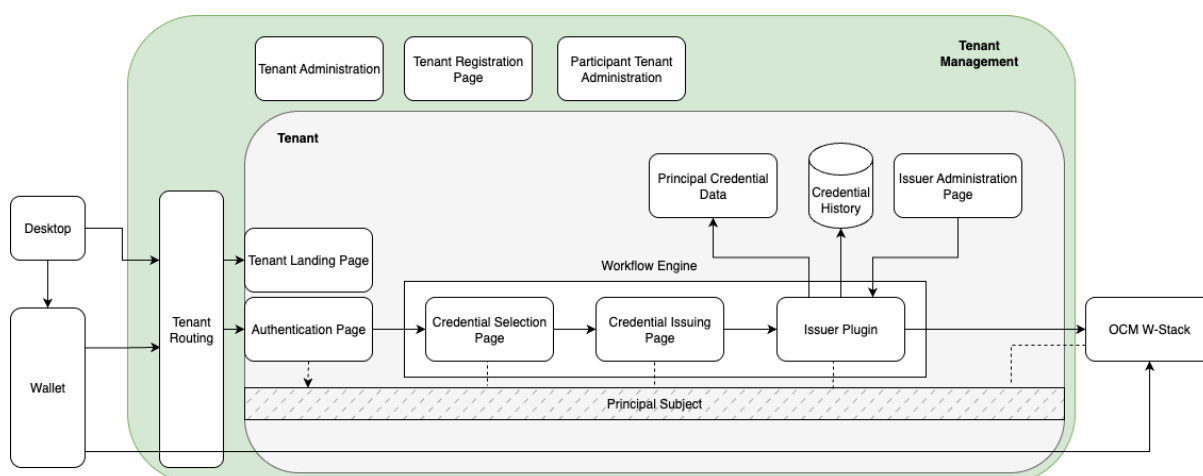


Figure 2 Conceptual Architecture Overview

The entire architectural flow shall be bound to the principal subject, which uniquely identifies the principal within the participant systems. This binding enables the use of

a participant-issued token in an end-to-end manner for external API access and for retrieval of credential data.

The operational flow of the solution shall be as follows:

1. A participant (organization administrator) shall register via the registration page by submitting required contact and organizational data.
2. Upon confirmation by the tenant administration, the tenant participant shall be activated via an email confirmation mechanism.
3. The participant shall be able to register a passkey for authentication within the tenant administration interface.
4. The participant shall be able to configure tenant-specific settings, including OIDC information, branding elements (logos and stylesheets) and TLS certificates for backend authentication. After that, claims with page administration and issuing rights for various credential types shall be configurable.
5. The participant shall configure the ORCE issuance flow to be deployed. A standard default flow shall be provided by the solution.
6. Upon completion of tenant configuration, the ORCE shall deploy the configured flow, including custom pages, integration of an issuer plugin, and issuer metadata.
7. The issuer flow pages and the tenant-specific issuer administration interface shall become accessible. Credential types shall be configurable within the tenant environment.
8. The issuer flow shall be triggerable through an authenticated login process.

The detailed credential issuance flow shall be implemented as follows:

1. The principal shall access the issuer page.
2. The principal shall authenticate via the OIDC login page configured by the participant.
3. If multiple credential types are available, a credential selection interface shall be presented.
4. The principal shall proceed through the ORCE-based flow configured by the issuer.
5. Upon successful completion of the flow, a QR code shall be generated and displayed.
6. The principal shall scan the QR code using a compatible wallet.
7. The wallet shall communicate with the OCM W-Stack.
8. OCM W-Stack shall invoke the issuer plugin.
9. The issuer plugin shall request required credential data from the participant backend system.
10. The issuer plugin shall return the credential data and corresponding signature and shall register the issued credential for revocation³ management and audit history.
11. OCM W-Stack shall transmit the issued credential to the wallet.

³ <https://github.com/eclipse-xfsc/statuslist-service>

5. Technical Architecture

The solution shall be deployed on a Kubernetes-based microservices architecture. The system shall consist of independently deployable services communicating through well-defined interfaces.

The Kubernetes cluster shall be logically organized into separate namespaces to ensure isolation of responsibilities and security boundaries. At minimum, the following namespaces for the cluster shall be defined:

- A public namespace,
- A tenant management namespace for tenant-specific services and configurations,
- An OCM W-Stack namespace for credential issuance infrastructure components.

Namespace separation shall ensure logical isolation, controlled access policies, and clear operational responsibility boundaries.

Architectural deviations from this structure shall only be permitted where technically justified and shall not compromise security, multi-tenancy isolation, or operational stability.

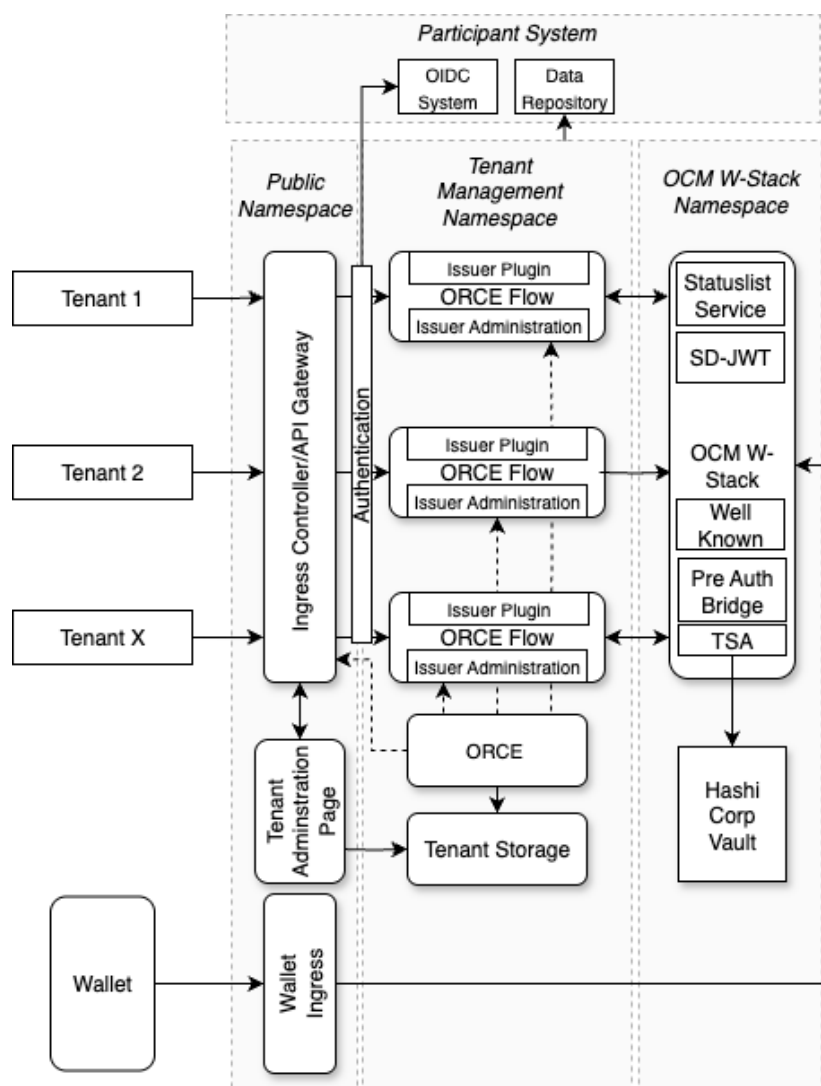


Figure 3 Technical Overview

6. Functional Requirements

6.1 Key Management

[FR-PCI-01] Key Creation over TSA Crypto Provider

Priority: MUST

Description: The key creation and key usage MUST be handled over the TSA Crypto Provider⁴. The crypto provider supports currently HashiCorp Vault, but it MUST be tested with OpenBao to ensure compatibility.

Acceptance criteria:

- Token signing key for Pre-Authorization Bridge can be configured in the issuer administration page,
- Credential signing key for issuer can be configured in the issuer administration page.

[FR-PCI-02] DID Provisioning over TSA Crypto Provider

Priority: MUST

Description: For providing DID documents, the TSA Crypto Provider MUST be used to host DID web documents via Ingress rules⁵.

Acceptance criteria:

The issuer keys are represented over a public resolvable DID document (via the universal resolver).

6.2 Public Endpoints

[FR-PCI-03] Credential Endpoint

Priority: MUST

Description: The credential endpoint of the OCM W-Stack MUST be public per tenant in the deployment for the wallet to receive its credential. This MUST be configured in the ingress configuration/API gateway.

Acceptance criteria:

Demonstration via API request (e.g., Curl or Postman).

[FR-PCI-04] Pre-Authorization Bridge Endpoints

Priority: MUST

Description: The endpoints of the Pre-Authorization Bridge MUST be public per tenant to retrieve tokens for the issuance. This MUST be configured in the ingress configuration/API gateway.

Acceptance criteria:

Demonstration via API request (e.g., Curl or Postman).

[FR-PCI-05] TSA Endpoints

Priority: MUST

Description: The endpoints of the TSA MUST be public per tenant for the JWKS for the issuance. This MUST be configured in the ingress configuration/API gateway.

Acceptance criteria:

Demonstration via API request (e.g., Curl or Postman).

⁴ <https://github.com/eclipse-xfsc/docs/tree/main/tsa>

⁵ <https://github.com/eclipse-xfsc/deployment/blob/main/OCM%20W-Stack/Well%20Known%20Ingress%20Rules/templates/ingress.yaml#L18>

6.3 Participant Tenant Administration

[FR-PCI-06] Issuer Management Delegation

Priority: MUST

Description: The tenant administration MUST provide a feature where the participants in tenant administration can delegate credential creation and credential configuration management to several principals of the organization. This MAY be realized by defining a role which enables users to log in as a credential administrator and create credentials.

Acceptance criteria:

- Principals can be enabled to create, modify, and delete credential configuration definitions.
- Principals can be enabled to manage the issuance flows.

[FR-PCI-07] OID4VCI Issuer Metadata Configuration

Priority: MUST

Description: The tenant administration MUST be able to configure the OID4VCI issuer metadata and distribute that metadata to the OCM W-Stack. The issuer administration MUST NOT have access to this information as this information is static.

Acceptance criteria:

- Configured metadata are configurable and distributed over NATS,
- Editable by participant tenant administration only,
- Metadata is visible in the well-known configuration of the tenant.

6.4 Issuer Administration

[FR-PCI-08] Issuing Management

Priority: MUST

Description: The issuing management MUST provide a feature where the principals can manage issuance flows and details about credential configurations. This MAY be realized by defining a role which enables users to log in as a credential administrator and create credentials.

Acceptance criteria:

- Principals can be enabled to create, modify, and delete credential configuration definitions.
- Principals can manage issuance flows

[FR-PCI-09] OID4VCI Credential Configuration Permissioning

Priority: MUST

Description: The issuer administration MUST consider defined roles which are able to create, modify, and delete credential configurations.

Acceptance criteria:

Standard principal cannot access the administration page.

[FR-PCI-10] Credential Logos

Priority: MUST

Description: The issuer administration MUST allow the uploading and storing of credential card logos for the credential configuration data. These logos MUST be resolvable over a tenant-specific URL from the storage for a wallet to download it via the well-known openID issuer configuration.

Acceptance criteria:

Wallet can load and display the credential card logo.

[FR-PCI-11] Issuance Flow Configuration

Priority: MUST

Description: The issuer administration MUST allow for a specific issuance flow with various steps to be configured per credential configuration. The steps MUST be configured via ORCE. Special UX frameworks MAY be used but ORCE is preferred.

Acceptance criteria:

For each credential configuration ID, a separate flow can be configured.

[FR-PCI-12] Credential History per Configuration

Priority: MUST

Description: The issuer administration MUST provide a credential history UI and backend which display the latest ID of the credential issuing, the status (“offered”, “expired”, “revoked”, “issued”) and the information on the principal (JWT format).

Acceptance criteria:

UI with overview of issuances per credential configuration which is only visible for credential configuration administrator.

[FR-PCI-13] Credential Revocation

Priority: MUST

Description: Within the issuer administration, there MUST be an option where an issued credential can be searched and revoked. The revocation MUST happen over the OCM W-Stack Status List Provider. An issued credential MUST be searchable over principal details, date or ID per credential configuration. UI history MAY be integrated.

Acceptance criteria:

For each credential configuration, a revocation option is available (the credential can be searched and revoked). Revocation links are embedded in the credential.

[FR-PCI-14] Credential Configuration Deletion

Priority: MUST

Description: A credential configuration MUST be deletable, but it MAY be archived if technically required.

Acceptance criteria:

Credential configuration can be deleted.

[FR-PCI-15] Credential Configuration Creation/Update

Priority: MUST

Description: A credential configuration MUST be creatable and modifiable according to the metadata credential configuration specification of OpenID4VCI.

Acceptance criteria:

Credential configuration can be deleted.

[FR-PCI-16] Participant Backend Connection

Priority: MUST

Description: A credential configuration setup MUST consist of participant backend connection settings which allow it to contact the participant backend for credential data. This requires all necessary settings for mTLS.

Acceptance criteria:

mTLS can be configured per credential configuration.

6.5 Issuer Plugin

[FR-PCI-17] OCM W-Stack Integration

Priority: MUST

Description: The issuer plugin MUST be integrated into the OCM W-Stack issuance flow according to the current architecture⁶. This requires the setup of a structure which speaks NATS messages. The programming language SHOULD be goLang, but for easier ORCE flow integration others MAY be used.

Acceptance criteria:

- Code review and demonstration to crosscheck the integration pattern and functionality,
- Full-working issuance flow.

[FR-PCI-18] NATS Provisioning of Configured Metadata

Priority: MUST

Description: The issuer plugin MUST provide the configured metadata via NATS to the OCM W-Stack well-known service according to the messaging library⁷.

Acceptance criteria:

Demonstration of an issuer configuration and credential configuration, crosscheck via well-known API of the tenant.

[FR-PCI-19] History Logging

Priority: MUST

Description: The issuer plugin MUST log all credential issuance requests in a GDPR-compliant way with status progress, e.g., “offered” or “accepted”. It MUST contain date, holder binding DID, revocation list address and other relevant data grouped by credential configuration.

Acceptance criteria:

Presentation of log entries.

[FR-PCI-20] Participant Backend Connection

Priority: MUST

Description: The issuer plugin MUST be able to either get data from the participant data repository to sign a credential via TSA, or to send a credential issuance request to the participant backend to get a signed credential back. An external issuer plugin MUST be configured per credential configuration for the purpose either for signing a credential or for providing the data for it. Communication MUST happen via TLS-protected GRPC calls.

Acceptance criteria:

- Demonstration of data collection for TSA signing,
- Demonstration of credential signing via a participant backend mock.

[FR-PCI-21] ORCE Integration

Priority: MUST

⁶ <https://github.com/eclipse-xfsc/oid4-vci-issuer-service?tab=readme-ov-file#introduction>

⁷ <https://github.com/eclipse-xfsc/nats-message-library/blob/main/wellknown.go>

Description: The issuer plugin MUST be triggerable via ORCE to request offering links from the OCM W-Stack. The integration MUST also provide a way to render the link as QR code.

Acceptance criteria:

- Code review,
- ORCE flow demonstration.

6.6 Participant Data Repository

[FR-PCI-22] Golang Service

Priority: MUST

Description: The participant data repository MUST be implemented as go lang microservice to reuse existing libraries, e.g., Crypto Provider Core. The service MUST be implemented with the Goa framework.

Acceptance criteria:

Code review.

[FR-PCI-23] Data Format Library

Priority: SHOULD

Description: Data format definitions for the used participant SD-JWT and W3C credentials SHOULD be collected in a go library and contributed to XFSC. It is RECOMMENDED to make use of W3CVCDM v2.0 or its profile in ETSI TS 119 472-1 as the credential formats belong to EAA.

Acceptance criteria:

Code review.

[FR-PCI-24] Data Request

Priority: MUST

Description: The core data request for a credential MUST be abstracted via GRPC to allow a participant to dock various data pickup providers.

Acceptance criteria:

- Demonstration of a simple data request via GRPC,
- Code review of the protobuf contract and demo review.

6.7 Provider Tenant Administration

[FR-PCI-25] Tenant Management

Priority: MUST

Description: The provider tenant administration MUST provide a functionality where the provider administrator can see, approve, and reject tenant registration requests made by participants.

Acceptance criteria:

Demonstration of the procedure.

[FR-PCI-26] Tenant Approved

Priority: MUST

Description: Once a tenant is approved, the Kubernetes resources, domains etc. can be provided, and the tenant can be unlocked to log in by participant tenant administration.

Acceptance criteria:

- Demonstration of the procedure,
- Email verification for the participant tenant administration.

[FR-PCI-27] Tenant Deletion

Priority: MUST

Description: The tenant MUST be deletable by the provider tenant administration and the participant tenant administration. In both cases, confirmation MUST be triggered before all resources are deleted.

Acceptance criteria:

- Deletion after confirmation,
- Deletion of resources (Kubernetes logs),
- Tenant gone, no URLs reachable anymore, and issuer DIDs are not resolvable anymore.

[FR-PCI-28] Tenant Rejection

Priority: MUST

Description: The tenant MUST be rejectable by the provider tenant administration. An email MUST be sent out to communicate the decision. The decision is logged, and the registration request is stored.

Acceptance criteria:

- Rejection is stored,
- Email is sent out.

7. Interfaces & Data Models

7.1 Tenant Administration UX

[FR-PCI-29] Confirmation/Deletion/List Section

Priority: MUST

Description: The administration page needs a protected section for the provider tenant administration, where tenants can be confirmed, listed, and deleted. This is used when a participant requests a tenant creation. Only one tenant per participant is allowed. This UX MUST be initially configured during the SaaS setup.

Acceptance criteria:

- Demonstration of registration of a tenant with confirmation within the page,
- Demonstration of the registered tenant by using the tenant administration page for participants.

[FR-PCI-30] Participant Tenant Administration

Priority: MUST

Description: The tenant administration for participants is a part of the administration page where a participant can configure its issuer metadata, OIDC login information, styles, and other relevant tenant information. This functionality MUST be unlocked after confirmation by tenant administrator. The login for this page is created just for this registration page and will be sent via email. After first login the password MUST be changed and a 2FA registered.

Acceptance criteria:

- Demonstration of tenant configuration and direct usage of it,
- Documentation of tenant configuration,

- Demonstration of tenant login.

7.2 Tenant Registration UX

[FR-PCI-31] Compliance

Priority: SHOULD

Description: The implementation of the UX SHOULD consider the fulfilment of ETSI EN 301 549 as it is referenced by IA 2025/2162.

[FR-PCI-32] Registration Data Form

Priority: MUST

Description: Under a given domain, the deployment MUST provide a registration data form which can be either private or public (per config). The form SHOULD contain the contact data of a participant who wants to create a tenant. The registration itself MUST be confirmed by tenant administrator before anything is created.

Acceptance criteria:

- Demonstration of registration and confirmation,
- Demonstration of how resources are created in the cluster.

7.3 Issuer Administration UX

[FR-PCI-33] Compliance

Priority: SHOULD

Description: The implementation of the UX SHOULD consider the fulfilment of ETSI EN 301 549 as it is referenced by IA 2025/2162.

[FR-PCI-34] Credential Configuration Management

Priority: MUST

Description: UX MUST enable creation, modification, and deletion of credential configurations according to the given OID4VC specification. This includes card layouts, data repositories (e.g., participant backend), logos, texts, and other metadata. This also includes the connection to the data repository/backend and/or the issuing plugin configuration.

Acceptance criteria:

- Demonstration of management by creating a configuration,
- Demonstration of docking a data repository/backend.

[FR-PCI-35] Credential Issuance Permissioning

Priority: MUST

Description: UX MUST enable “docking” existing permissions like roles or groups to a credential configuration so that only a defined range of principals can request a credential. For instance, a group of directors only receive a director’s credential.

Acceptance criteria:

Demonstration of how two different credential requests can result in the issuance of two different kinds of credentials.

[FR-PCI-36] Credential History and Revocation

Priority: MUST

Description: UX MUST provide a feature to see the history of a credential configuration, e.g., how often it was issued, or which principal received which credential. Additionally, each history record SHALL have an option to revoke the credential via the Status List Service to block the credential for the principal and to block the re-issuance.

Acceptance criteria:

Demonstrate credential history and revocation.

[FR-PCI-37] Steps Configuration

Priority: MUST

Description: The issuer administration MUST be able to configure UI steps via ORCE for the issuance process, beginning with a landing page followed by other steps for issuance.

Acceptance criteria:

- Steps can be configured and different kinds of flow can be produced,
- Tenant-specific layout is applied to the pages.

[FR-PCI-38] ORCE Utilization

Priority: SHOULD

Description: The issuer administration SHOULD utilize ORCE to define issuance flows on UX level. If this is not possible, another low-code design MUST be provided.

Acceptance criteria:

- Custom flows can be configured for credential issuances,
- Two custom flows have been demonstrated.

7.4 Issuer Plugin (Participant Connection)

The issuer plugin connects itself to the participant data repository to get the credential data belonging to the subject. This interface MUST be therefore additionally protected to become enterprise ready. It is RECOMMENDED that the interface follows ETSI TS 119 478 as participant data repository.

[FR-PCI-39] mTLS Connection

Priority: MUST

Description: The connection to the participant data repository MUST be established by the issuer plugin via an mTLS connection. The certificates for this connection MUST be configured in the tenant administration of the participant.

Acceptance criteria:

- Demonstration of mTLS configuration,
- Documentation of configuration.

[FR-PCI-40] Participant JWT Reusage

Priority: MUST

Description: During the call of the participant data repository service/backend, the issuer plugin MUST use the JWT issued by the participant backend to have access to participant backend resources, e.g., for accessing data repositories.

Acceptance criteria:

- Code review,
- Data log.

[FR-PCI-41] Issuing Settings

Priority: MUST

Description: The tenant administration SHALL be able to limit the issuer plugin with parameters like rate limits.

Acceptance criteria:

- Code review,
- Data log.

8. Security & Trust

8.1 Transport Security

[FR-PCI-42] TLS Security

Priority: MUST

Description: The transport security for external connections MUST support TLS 1.3 following the „European Cybersecurity Certification Group, Sub-group on Cryptography: ‘Agreed Cryptographic Mechanisms’ published by the European Union Agency for Cybersecurity⁸”. All components which establish TLS connections MUST have pre-configuration.

Acceptance criteria:

Documentation of TLS settings of the network components in the code.

[FR-PCI-43] TLS Certificates

Priority: MUST

Description: The deployment setup MUST be configured so that either Let’s Encrypt can be activated per route, or a static TLS secret can be used.

Acceptance criteria:

- Review of the configuration,
- Demonstration of TLS certificate.

8.2 Identity & Authentication

[FR-PCI-44] Authentication with Keycloak in Tenant

Priority: MUST

Description: The solution MUST be protected via an OIDC provider which identifies the principle and defines its roles for a certain scope. In general, it SHALL follow sections 7.4 ETSI TS 119 471 as well as 4.2.2.1 if applicable.

Acceptance criteria:

Demonstration of login with Keycloak and usage of the issuance process.

[FR-PCI-45] Authentication with Microsoft Entra in Tenant

Priority: MUST

Description: The solution MUST be able to use Microsoft Entra to start the issuance process. A process on how to connect an Entra Tenant MUST be documented.

Acceptance criteria:

- Demonstration of login with Microsoft Entra and usage of the issuance process,
- Demonstration of Entra Tenant Connection.

⁸ <https://digital-strategy.ec.europa.eu/en/policies/cybersecurity-certification-group>

[\[FR-PCI-46\] Authentication in Tenant Administration/Registration](#)

Priority: MUST

Description: The solution MUST use an authentication solution to reach tenant administration features and registration. This contains email verification flows, passkey registration and a confirmation flow by the cluster administrator.

Acceptance criteria:

Demonstration of Tenant Registration flow with email confirmation, cluster administrator confirmation and registration of passkey.

8.3 Authorization & Policy Enforcement

[\[FR-PCI-47\] IDP Authorization Reusage](#)

Priority: MUST

Description: The participant's IDP will provide identity and optional authorizations like roles and permissions. The solution SHOULD re-use this kind of existing authorization. Documentation MUST be provided as well.

Acceptance criteria:

Solution is compatible with the participant's login solution.

8.4 Data Protection

[\[FR-PCI-48\] GDPR Compliance](#)

Priority: MUST

Description: To have a GDPR-compliant system, the issuer flow SHALL never store data which was retrieved by the participant backend system.

Acceptance criteria:

- Review of "fire and forget" logic during issuing: review of data deletion logic and documentation after issuance and review of the running system,
- Documentation of processes and preparation for auditing,
- Review of example implementation for participant data repository.

9. Deployment & Operations

9.1 Deployment

[\[FR-PCI-49\] Deployment Playbook](#)

Priority: MUST

Description: The deployment MUST be reproducible in other Kubernetes environments. The setup MUST be executable over any generic mechanism. For instance, scripts, terraform, ansible, Kubernetes operators or similar.

Acceptance criteria:

Demonstration of automatic setup in a second cluster.

[\[FR-PCI-50\] Route Management](#)

Priority: MUST

Description: For each issuance workflow, there MUST be appropriate route management within ingresses which separates the tenants from each other. This includes tenant headers for OCM W-Stack and others⁹ (see Appendix B).

⁹ <https://github.com/eclipse-xfsc/deployment/blob/main/OCM%20W-Stack/Well%20Known%20Ingress%20Rules/templates/ingress.yaml#L9>

Acceptance criteria:

Code review

[FR-PCI-51] Deployment Publication

Priority: MUST

Description: All scripts and deployment helm charts MUST be integrated into the XFSC deployment repository¹⁰.

Acceptance criteria:

Pull Requests created and merged by maintainer according to milestones.

[FR-PCI-52] Deployment Tool Stack

Priority: MUST

Description: The deployment tool stack MUST be open-source to avoid technology lock-in for users. The tools used for deployment MAY be ORCE, Ansible, Terraform, Crossplane or similar standard tools which simplify the setup of the solution.

Acceptance criteria:

Presentation of choice and documentation.

[FR-PCI-53] Resource Limitation

Priority: MUST

Description: All helm charts and templates MUST be limited to consumed resources. Setup resource limitations (scale out or scale downs) MAY be realized over dynamic processes/additional tooling like KEDA or any other tool stack.

Acceptance criteria:

- Documentation on how to set a limit,
- Documentation on how to scale down and scale up.

[FR-PCI-54] Web Documentation

Priority: MUST

Description: The documentation for the deployment MUST be created on GitHub. Additionally, there MUST be a web-formatted GitHub support page where the documentation is rendered in a way that people can search for FAQ, deployment topics and guides over GitHub Pages. The page can be rendered out of the existing documentation, but it MUST contain well-formatted hints, FAQs, and other helpful things for setting up the deployment.

Acceptance criteria:

Presentation of the support page for deployment.

9.2 Operational Requirements

[FR-PCI-55] Jaeger Tracing

Priority: MUST

Description: The solution MUST provide a pre-configured Jaeger instance for tracing events in its deployment.

Acceptance criteria:

- Documentation,
- Presentation of demo logs.

¹⁰ <https://github.com/eclipse-xfsc/deployment>

[\[FR-PCI-56\] Prometheus](#)

Priority: MUST

Description: The solution MUST provide Prometheus Instances to collect metrics.

Acceptance criteria:

- Documentation,
- Presentation of demo logs.

[\[FR-PCI-57\] OTEL Collection](#)

Priority: MUST

Description: The solution MUST provide an OTEL Collector with interfaces to Prometheus and Jaeger.

Acceptance criteria:

Demonstration of working logging.

[\[FR-PCI-58\] Health Endpoints/Logging](#)

Priority: MUST

Description: The solution MUST provide health endpoints to reflect the status of the application and appropriate logging.

Acceptance criteria:

Demo logs of Kubernetes.

10. Standards & Protocols

10.1 UX Framework

[\[FR-PCI-59\] Low Code](#)

Priority: MUST

Description: The UX framework MUST be a low-code variant where a pre-defined page can be easily adopted by the participant tenant administrator. This can be a CMS, ORCE, or something else. The entire UX framework MUST be consistent across the entire solution. This SHALL enable the participant tenant to customize the white label UX.

Acceptance criteria:

Demonstration of modification without coding with the same UX framework in all UIs.

[\[FR-PCI-60\] Layout Customization](#)

Priority: MUST

Description: The UX framework MUST allow modifications to logos, colors, and the entire page design according to FACIS style sheets and other configurations.

Acceptance criteria:

- Demonstration of website modification to FACIS layout,
- Live demonstration of website style ad-hoc during the presentation.

[\[FR-PCI-61\] UX Tests](#)

Priority: MUST

Description: The UX steps MUST be tested E2E via BDD tests by using the BDD Executor¹¹ similar to the Cloud PCM tests¹².

¹¹ <https://github.com/eclipse-xfsc/bdd-executor>

¹² https://github.com/eclipse-xfsc/cloud-wallet-integration-tests/blob/main/steps/presentation_selection_steps.py

Acceptance criteria:

Demonstration of the automated tests.

[FR-PCI-62] OID4VCI/VP Version

Priority: MUST

Description: OID4VCI/VP Version Draft 13 and 1.0 MUST be supported.

Acceptance criteria:

- Code review,
- UX can represent both versions.

10.2 Participant Data Protocol

[FR-PCI-63] Protocol Structure

Priority: MUST

Description: The protocol to communicate with the participant’s system/data source for the issuance is based on two requests: credential data request and credential request notification. The protocol MUST communicate via REST or GRPC. During the credential data request, there MUST be an option to get either just the data of a credential configuration (unsigned VC and SD-JWT) or the entire signed credential. The credential signing within the data repository service MUST be implemented via the Crypto Provider Core GRPC interface to support various providers. If there are any additional requests required, they MAY be introduced.

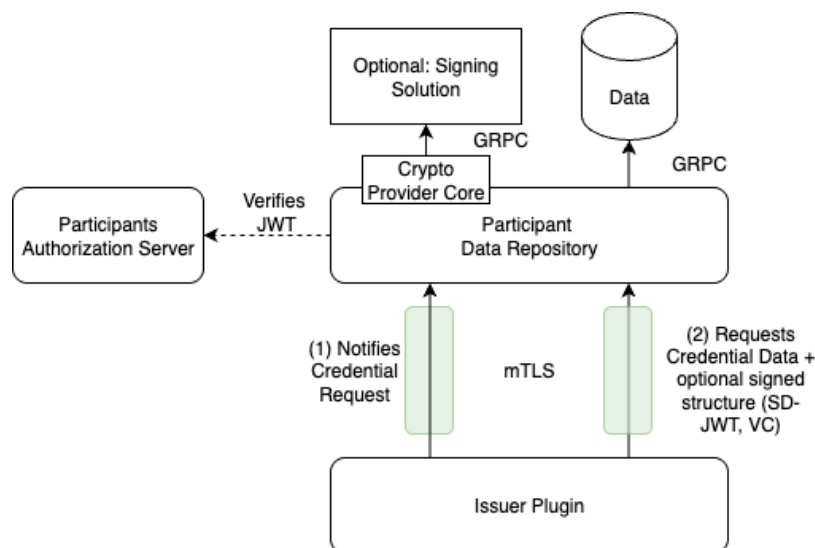


Figure 4 Credential Data Protocol

Acceptance criteria:

- Demonstrate the protocol functionality via example setups,
- Demonstrate a setup with credential signing,
- Demonstrate a setup without credential signing but with signing within OCM.

10.3 GitHub Requirements

[FR-PCI-64] Standard Workflows

Priority: MUST

Description: All GitHub repositories which contain code MUST contain standard workflows from the GitHub DevOps¹³ section. Especially the docker build, test, eclipse IP scan and SBOM creation. The workflows MUST be integrated as remote workflow¹⁴.

Acceptance criteria:

- Code review,
- Run action log.

11. Validation & Acceptance Criteria

Besides the technical acceptance criteria in the requirements, there are more validation rules and top-level acceptance criteria which MUST be fulfilled during acceptance presentations milestone by milestone. All these criteria MUST be implemented as well via the BDD Executor as BDD test steps.

In general, it MUST also be clarified which parts of the implementation are performed by AI and how this affects code quality and project performance.

11.1 Tenant Management Milestone

Table 1 Tenant Management Milestone

No.	Validation criteria	Definition of done
FR-PCI-65	Tenant can be registered	A tenant can be registered, activated per email and confirmed by the provider tenant administration. The tenant is visible in the tenant listing.
FR-PCI-66	Tenant can be deleted	The tenant can be deleted by the tenant administrator (provider and participant). All resources in the cluster are deleted. A confirmation was triggered.
FR-PCI-67	Tenant can be confirmed	After registration the provider tenant admin must be able to confirm a registration by reviewing the application details. All resources will be provided afterwards.
FR-PCI-68	Tenant can be rejected	After registration, the provider tenant admin must be able to reject a registration by reviewing the application details. An email is sent out. No resources are provided.
FR-PCI-69	Configured participant login works	After the participant IAM system was configured by the participant tenant admin, the login for the issuer page works with the principal credentials.
FR-PCI-70	Multiple tenants can be created	Multiple tenants under multiple domains/routes can be created and work properly from landing page to authentication.
FR-PCI-71	Tenants can be customized with layouts, logos, and custom texts for landing page	After tenant creation, each tenant landing page must be configurable (e.g., logos, styles, custom texts of the participant).
FR-PCI-72	Issuer management can be delegated	Within the tenant administration, roles or user accounts can be unlocked for issuer management.
FR-PCI-73	Eclipse IP scans	Eclipse IP scans are integrated and triggered, no license conflicts with Eclipse license.

¹³ <https://github.com/eclipse-xfsc/dev-ops/tree/main/.github/workflows>

¹⁴ <https://github.com/eclipse-xfsc/email-service/blob/main/.github/workflows/sbom.yml#L11>

		compatibility (e.g., no GPL, GPL 2, AGPL, LGPL).
FR-PCI-74	Deployment integration	All scripts and changes are available as PRs in the XFSC deployment repository.
FR-PCI-75	Version tagging	Running version tagged and documented.

11.2 Issuer Management Milestone

Table 2 Issuer Management Milestone

No.	Validation criteria	Definition of done
FR-PCI-76	Credential configuration can be created	Any delegated user can create a credential configuration by configuring fields, background image, description etc. according to the OID4VCI metadata specification.
FR-PCI-77	Credential configuration can be modified	An existing credential configuration can be modified by changing the background images, changing texts, claims etc.
FR-PCI-78	Credential configuration can be deleted	An existing credential configuration can be deleted, but existing credentials remain resolvable.
FR-PCI-79	Credential logo can be configured	Logo images can be uploaded for a credential configuration. The image size is optimized and/or highlighted and the image resolvable over the well-known openID issuer configuration.
FR-PCI-80	Flow per credential configuration can be configured	For each credential configuration, a custom flow consisting of multiple page steps and layouts (e.g., survey, data protection information, introduction of process, QR code page) can be created.
FR-PCI-81	History per configuration shows latest issuances and status	A credential configuration shows a list of credential history events with ID of the credential. The ID can be searched, and the credential can be revoked.
FR-PCI-82	Issuance based on participant backend	The credential must be issued over the usage of the participant signing and credential creation capabilities.
FR-PCI-83	Issuance based on participant data repository	The credential must be issued over the usage of the participant data repository and internal signing capabilities.
FR-PCI-84	Eclipse IP scans	Eclipse IP scans are integrated and triggered, no license conflicts with Eclipse license compatibility (e.g., no GPL, GPL 2, AGPL, LGPL).
FR-PCI-85	Deployment integration	All scripts and changes are available as PRs in the XFSC deployment repository.
FR-PCI-86	Version tagging	Running version tagged and documented.

11.3 E2E Issuance Milestone

Table 3 E2E Issuance Milestone

No.	Validation criteria	Definition of done
FR-PCI-87	Multiple issuance flows can be used	Multiple issuance flows in various layouts can be used, and credentials can be imported in PCM Mobile.

FR-PCI-88	Multiple tenants and users can be used	Multiple tenants and users can be used for various issuances in the PCM mobile app.
FR-PCI-89	Version tagging	Running version tagged and documented.
FR-PCI-90	Eclipse IP scans	Eclipse IP scans are integrated and triggered, no license conflicts with Eclipse license compatibility (e.g., no GPL, GPL 2, AGPL, LGPL).

11.4 Second Cluster Deployment Milestone

Table 4 Second Cluster Deployment Milestone

No.	Validation criteria	Definition of done
FR-PCI-91	Solution is deployed on a second cluster	E2E credential issuance works.
FR-PCI-92	Deployment integration	All scripts and changes are available as PRs in the XFSC deployment repository.
FR-PCI-93	Version tagging	Running version and fixes tagged and documented.
FR-PCI-94	Eclipse IP scans	Eclipse IP scans are integrated and triggered, no license conflicts with Eclipse license compatibility (e.g., no GPL, GPL 2, AGPL, LGPL).

11.5 GitHub Finalization Milestone

Table 5 GitHub Finalization Milestone

No.	Validation criteria	Definition of done
FR-PCI-95	Version tagging	All repositories are tagged with the latest versions. Releases are created. Helm charts and Docker Images are uploaded in Harbor in the latest versions.
FR-PCI-96	Documentation	Documentation feedback is refined in the GitHub readmes. All readmes are up to date.
FR-PCI-97	Eclipse IP scans	Eclipse IP scans are integrated and triggered, no license conflicts with Eclipse license compatibility (e.g., no GPL, GPL 2, AGPL, LGPL etc.).
FR-PCI-98	EUDI compliance statement	The solution contains documentation describing which parts must be modified and/or developed to reach EUDI and ARF compliance in later versions of the solution.
FR-PCI-99	No open PRs and issues	All GitHub repositories for the solution have no open issues/no open PRs.
FR-PCI-100	Standard workflows integrated	All standard workflows in GitHub are integrated (Docker, Helm, Eclipse scan, test).
FR-PCI-101	Successful workflow runs	All workflow runs must be successful without any error.

12. Appendices

Appendix A: Glossary

Table 6 Glossary

Term	Category	Definition	Relevant specifications/links
DID (Decentralized Identifier)	SSI building block	Globally unique identifiers that do not depend on a central authority and can be cryptographically resolved	W3C DID Core: https://www.w3.org/TR/did-core/
Eclipse XFSC (Cross Federation Services Components)	Software framework / component set	An open-source software toolbox for building and operating federated data ecosystems. It provides microservices for identity/trust, credential management, cataloging, and interoperability across federations. Originally part of Gaia-X and now under the Eclipse Foundation. ([projects.eclipse.org][1])	Eclipse XFSC project: https://github.com/eclipse-xfsc Eclipse XFSC on Eclipse.org: https://projects.eclipse.org/projects/technology.xfsc Eclipse XFSC documentation: https://github.com/eclipse-xfsc/docs
ETSI TS 119 471	Standard	Electronic Signatures and Trust Infrastructures (ESI); Profiles for Electronic Attestation of Attributes; Part 1: General requirements	https://www.etsi.org/deliver/etsi_ts/119400_119499/119471/01.01.01_60/ts_119471v010101p.pdf
ETSI TS 119 472-1	Standard	Electronic Signatures and Trust Infrastructures (ESI); Profiles for Electronic Attestation of Attributes; Part 1: General requirements	https://www.etsi.org/deliver/etsi_ts/119400_119499/119472/01/01.01.01_60/ts_11947201v010101p.pdf
European Cybersecurity Group	Standard	The European Cybersecurity Certification Group was established to help ensure the consistent implementation and application of the Cybersecurity Act.	https://digital-strategy.ec.europa.eu/en/policies/cybersecurity-certification-group
OpenID4VC (OpenID for Verifiable Credentials)	Protocol framework	OpenID Foundation specs that define how verifiable credentials are issued and presented using OAuth2/OIDC flows	OpenID4VCI: https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html OpenID4VP: https://openid.net/specs/openid

			d-4-verifiable-presentations-1_0.html
OpenID4VCI (credential issuance)	Protocol	OAuth2/OpenID-based flows for issuing verifiable credentials from an issuer to a wallet	<p>OAuth 2.0 RFC 6749: https://www.rfc-editor.org/rfc/rfc6749</p> <p>PKCE RFC 7636: https://www.rfc-editor.org/rfc/rfc7636</p> <p>OpenID: https://openid.net/specs/openid-d-4-verifiable-credential-issuance-1_0-ID1.html</p>
Organization Credential Manager (OCM)	XFSC component	Manages digital credentials for organizations in SSI-based ecosystems (e.g., connections, DID management, credential issuance/verification)	https://github.com/eclipse-xfsc/docs/tree/main/ocm-w-stack
Personal Credential Manager (PCM)	XFSC component	Client-side service that holds user credentials and enables selective disclosure and interaction in SSI contexts. ([projects.eclipse.org][1])	https://github.com/eclipse-xfsc/docs/tree/main/pcm-cloud
SD-JWT	Standard	Selective Disclosure JWT	https://datatracker.ietf.org/doc/rfc9901/
SSI (Self-Sovereign Identity)	Identity paradigm	Decentralized model where users control identities and credentials using cryptography, DIDs, and VCs. ([GXFS.eu][6])	<p>W3C DID Core: https://www.w3.org/TR/did-core/</p> <p>W3C Verifiable Credentials: https://www.w3.org/TR/vc-data-model/</p>
Token Status List	Standard	Token Status List describes how to embed information for revocation in SD-JWT tokens or W3C Credentials	<p>https://www.ietf.org/archive/id/draft-ietf-oauth-status-list-02.html</p> <p>https://github.com/eclipse-xfsc/statuslist-service</p>
Trust Services API (TSA)	XFSC component	Provides trust validation, policy enforcement, and cryptographic verification across ecosystem participants. ([eclipse.dev][5])	https://github.com/eclipse-xfsc/docs/tree/main/tsa
VC (Verifiable Credential)	SSI data model	Cryptographically secure, machine-verifiable statements about a subject that can be selectively disclosed	<p>W3C VC Data Model: https://www.w3.org/TR/vc-data-model/</p>
Orchestration Engine (ORCE)	Framework	XFSC orchestration engine based on Node Red	https://github.com/eclipse-xfsc/orchestration-engine

Issuer	SSI role	Entity that creates and signs VCs asserting attributes about a subject	W3C VC Data Model: https://www.w3.org/TR/vc-data-model/
Holder	SSI role	Entity (often the user) storing and presenting credentials	W3C VC Data Model: https://www.w3.org/TR/vc-data-model/
Verifier	SSI role	Entity that requests and verifies credentials/presentations	OpenID4VP: https://openid.net/specs/openid-4-verifiable-presentations-1_0.html

Appendix B: Tenant-specific Ingress Rule

```

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: {{ .Release.Name }}-diddocument-ingress
  annotations:
    cert-manager.io/cluster-issuer: letsencrypt-prod
    nginx.ingress.kubernetes.io/configuration-snippet: |
      proxy_set_header X-DID did:web:{{ .Values.ingress.hostname }};
      proxy_set_header X-NAMESPACE {{ .Values.ingress.tenantName }};
    nginx.ingress.kubernetes.io/rewrite-target: /v1/did/document
spec:
  ingressClassName: nginx
  rules:
  - host: {{ .Values.ingress.hostname }}
    http:
      paths:
      - path: /.well-known/did.json
        pathType: ImplementationSpecific
        backend:
          service:
            name: {{ .Values.ingress.service.backend }}
            port:
              number: {{ .Values.ingress.service.port }}
    {{ if .Values.ingress.tls }}
    tls:
      - hosts:
        - {{ .Values.ingress.hostname }}
        secretName: {{ .Values.ingress.tls.secret }}
    {{ end }}

```