Proof-of-Concept Specification for **Digital Collaboration by Federation**

www.facis.eu

Supported by:

Federal Ministry for Economic Affairs and Energy

on the basis of a decision by the German Bundestag

Funded by the European Union NextGenerationEU

Version 1.0 (Aug 18th, 2025)

Published by

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

Executive Summary

We are living in a decade of ongoing digital transformation, building digital twins, using the Internet of Things, adopting Cloud Native Services, and sharing cross-enterprise data. All of these trends are being accelerated by the fast-emerging use of artificial intelligence, and they require new conceptual strategies to deal with partners, data, and multi-provider services with interconnected systems. The primary challenges of collaboration and data sharing within the aviation ecosystem stem from the conflict between the immense value of collaboration and the risks inherent in a fragmented, competitive, and high-stakes environment. to Overcoming these challenges requires a fundamental shift from closed, proprietary systems open, trusted, and decentralized frameworks. In depth, we have to deal with a set of obstacles like

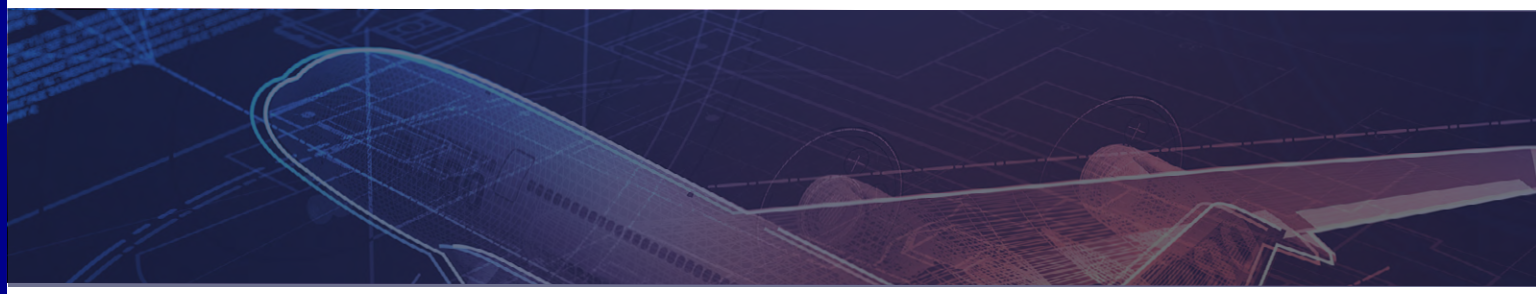
Data Silos and Lack of Interoperability:
Data is trapped in legacy systems with proprietary formats. An airline's booking system, an airport's ground operations software, and a manufacturer's maintenance platform don't speak the same language.

Lack of Trust and Verifiability:
How can an airline or airport ensure the accuracy and legitimacy of passenger data collected across competing platforms? There is no universal, digitally verifiable layer of trust for passenger information, leading to potential inconsistencies and disputes.

Cybersecurity Risks:
As critical national infrastructure, the aviation sector is a high-value target for cyberattacks. Opening up data sharing increases the potential attack surface if not managed with a fundamentally new security paradigm of Zero Trust versus the current stable but too static perimeter and network protection design.

Regulatory and Sovereignty Hurdles:
In Europe, the GDPR strictly governs the use of personal data (e.g., passenger information). Furthermore, nations and corporations are increasingly concerned about data sovereignty—maintaining control over their critical data and not being locked into non-European cloud providers.

New digital concepts, particularly those championed in Europe, offer a powerful toolkit to overcome these hurdles. They work together to create a secure and sovereign environment where data can be shared with granular control, without forcing participants to give up ownership or control.



The Framework: Dataspaces and the Gaia-X Trust Framework

The foundational solution is the creation **of aviation dataspaces**. A dataspace is not a giant, central database. Instead, it is a **decentralized, federated system** where participants agree on a common set of rules for data exchange while keeping the data in their own systems. It is like agreeing on the grammar and etiquette of a conversation without everyone having to speak the exact same language or be in the same room.

Gaia-X provides the essential trust framework for this. It is a European initiative that defines the rules of the road for these dataspaces, ensuring:

1 Interoperability:

Common technical standards and data formats that allows systems to connect.

2 Data Sovereignty:

Participants always retain control over their data, deciding precisely who can access what, for how long, and for what purpose.

3 Transparency and Trust:

Clear rules on governance, security, and compliance, creating a level playing field.

This document presents the technical design for the **Federated Aviation Domain Collaboration** as Proof of Concept (PoC) according to requirements, set by the 8ra project AXIS.

The PoC demonstrates secure, federated collaboration among aviation stakeholders using Federation Services, Trust Service Orchestration, and fine granular access control to systems, applications, networks, and devices.

In general, the PoC features modular, cloudnative software and service components deployed on 8ra-compliant platforms and secured through Trust Services. It showcases a scalable, interoperable federated security model aligned with common W3C standards and Gaia-X Trust Framework specifications.

The Security Model: Zero Trust Architecture

Due to open collaboration involving the exchange of services and data between a large number of partners, a Zero Trust approach must be adopted. This requires continuous verification of trusted partners (participants) and users (principals), as well as the enforcement of least-privilege access and the validation of transactions. The traditional "castle-and-moat" approach, whereby everything inside the network is trusted, is obsolete in a distributed ecosystem.

Zero Trust operates on a simple but powerful principle: "**Never trust, always verify.**"

In a Zero Trust architecture, every single request for data is treated as if it comes from an untrusted network. It must be authenticated and authorized before access is granted. There are emerging standards like W3C (World Wide Web Consortium to build a web based on the principles of accessibility, internationalization, privacy, and security) and **Self-Sovereign Identity (SSI)**, to give individuals complete ownership and control over their digital identity and **Verifiable Credentials (VCs)** as building block that makes the conceptual vision of SSI a practical reality. Unlike traditional models where identity is managed by centralized authorities (like governments or corporations), SSI empowers the user to manage their own personal data and decide what information to share, with whom, and when. It is a fundamental shift from a user being a data subject to a data controller.

→ **Authentication: The system uses the SSI in combination with VCs to verify the identity of the user or system making the request.**

→ **Authorization: Policies are applied to grant the minimum level of access necessary. For example, a ground handler's application can see a flight's arrival gate and time, but not its passenger list or maintenance history.**

This granular, per-request security model is a perfect fit for a complex dataspace. It allows an airline, an airport, and a service company to share specific data points for a specific purpose without exposing their entire internal systems to each other, drastically reducing the attack surface.

A Practical Scenario: An Aircraft on Ground (AOG) Event

Imagine a flight is grounded due to a technical issue. Today, this triggers a flurry of phone calls, emails, and manual data lookups. With these new concepts, the process is transformed:

1. **Identification: The pilot and ground crew use their SSI wallets to access the system. Their VCs prove their identity and qualifications.**
2. **Dataspace Access: They connect to the Aviation Dataspace, governed by Gaia-X rules.**
3. **Secure & Granular Data Pull: The mechanic's application, authenticated via Zero Trust, retrieves the specific error code from the aircraft's data stream. It automatically cross-references this with producer's technical documentation via a secure API.**
4. **Automated Action: The system identifies the required spare part, checks its availability in the airport's MRO inventory (another participant in the dataspace), and verifies the part's authenticity using its own VC.**
5. **Resolution: The right part is dispatched to the right gate, and the certified mechanic is given access to the specific digital work instructions.**

The result is a dramatic reduction in downtime, achieved by sharing the right data with the right entities at the right time, all without compromising security, control, or commercial confidentiality.

The Demonstrator

To demonstrate the core concepts of federated aviation ecosystems and their technical implementation, the FACIS team has developed an implementation within a federated environment. Partners are onboarded with verified digital identities and compliance credentials, enabling them to publish services to a shared catalogue. Access to these services is governed by attribute-based access control (ABAC) policies, which are defined and enforced by the federation. Operational users receive role-specific verifiable credentials which are evaluated at runtime to authorize access to protected systems. All access decisions are dynamically enforced based on trust policies to ensure secure, compliant, and auditable interactions across the aviation ecosystem.

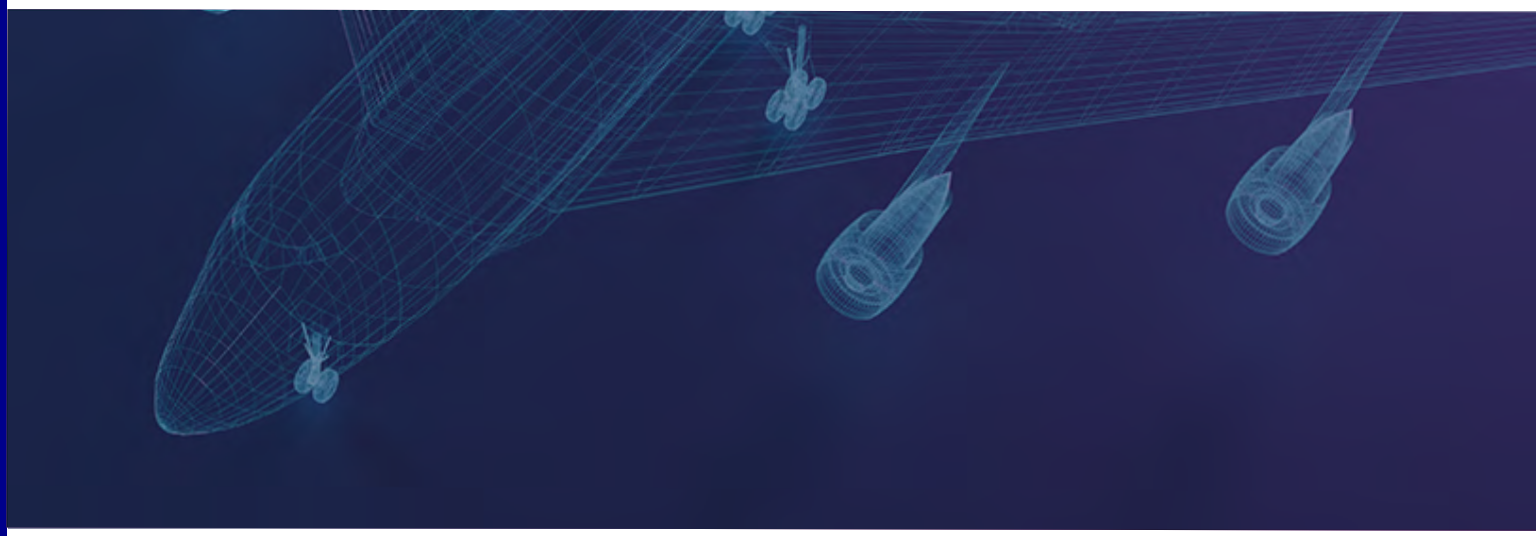


Table of Contents

List of Figures

List of Tables

1. Introduction

The PoC demonstrates secure, federated interactions among participants in the aviation industry, such as airlines, airports, travel agencies, and manufacturers. The primary use cases include partner onboarding, partner services listing, and partner services access based on verifiable credentials and principal attribute-based access control. This PoC emphasizes the use of modular, portable components and cloud-native deployment on 8ra-ready IaaS offerings and Kubernetes support.

2. Aviation Partner Ecosystem

In the aviation ecosystem, the aircraft is a key resource which belongs to an airline. The aircraft can be considered as a system of systems (more than 100), connected within an on-board network, which can communicate with the Internet through a "Communication Manager". The communication manager may select different physical communication channels depending on the situation (Gatelinks, Satcom, Cellular, WiFi).

The main objective for the airplane industry is to simplify the management of digital interfaces between the aircraft and all stakeholders that may interact with them during their operation (airline, airport, ground services providers, crews, passengers, suppliers, ATC).

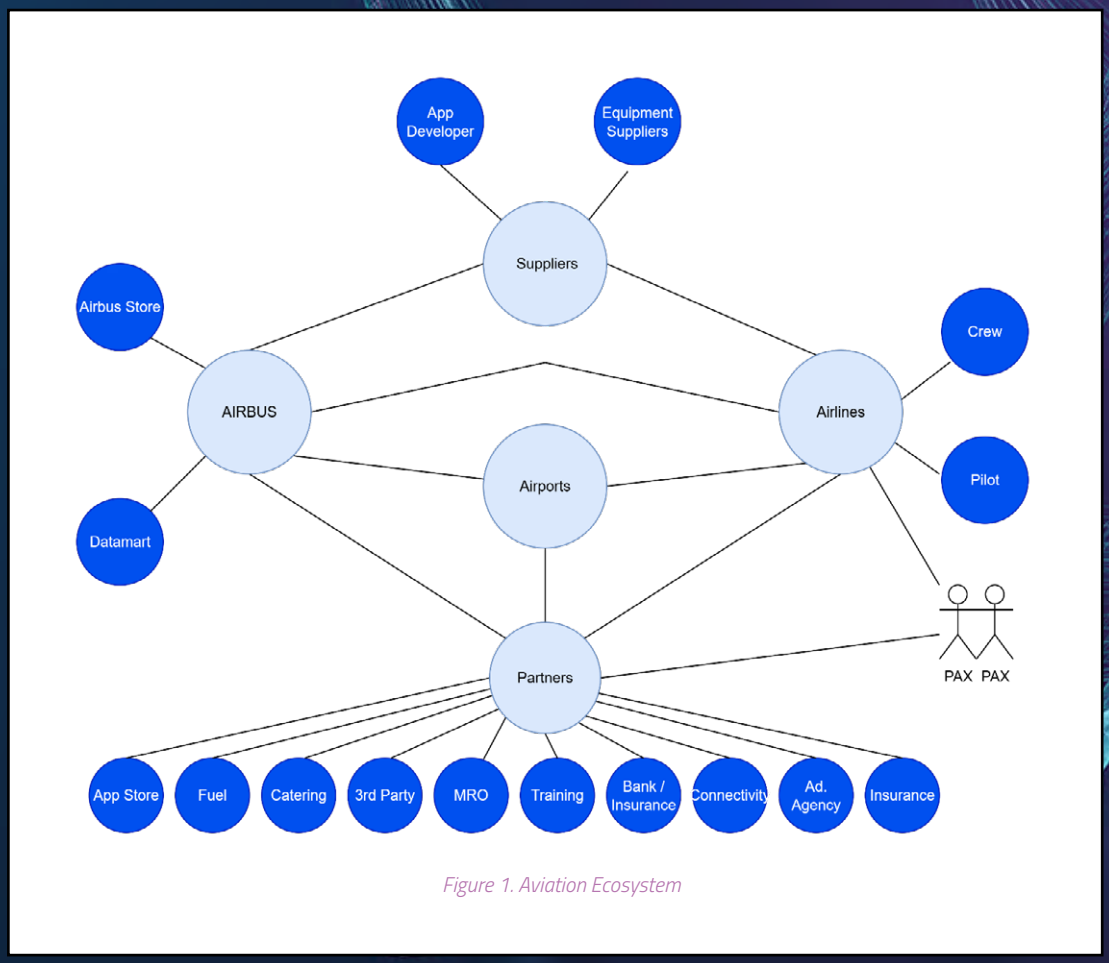


Figure 1. Aviation Ecosystem

2.1 Aviation Ecosystem Interdependencies

Deploying services on aircraft has always been a difficult exercise. The aviation IT culture, safety first, encourages determinist approaches which are incompatible with cloud IT techniques. Aviation software is designed peer-to-peer, which makes it difficult for functions to interact with each other if that was not initially planned. Implementing new service optimization during operation based on context is slow and expensive.

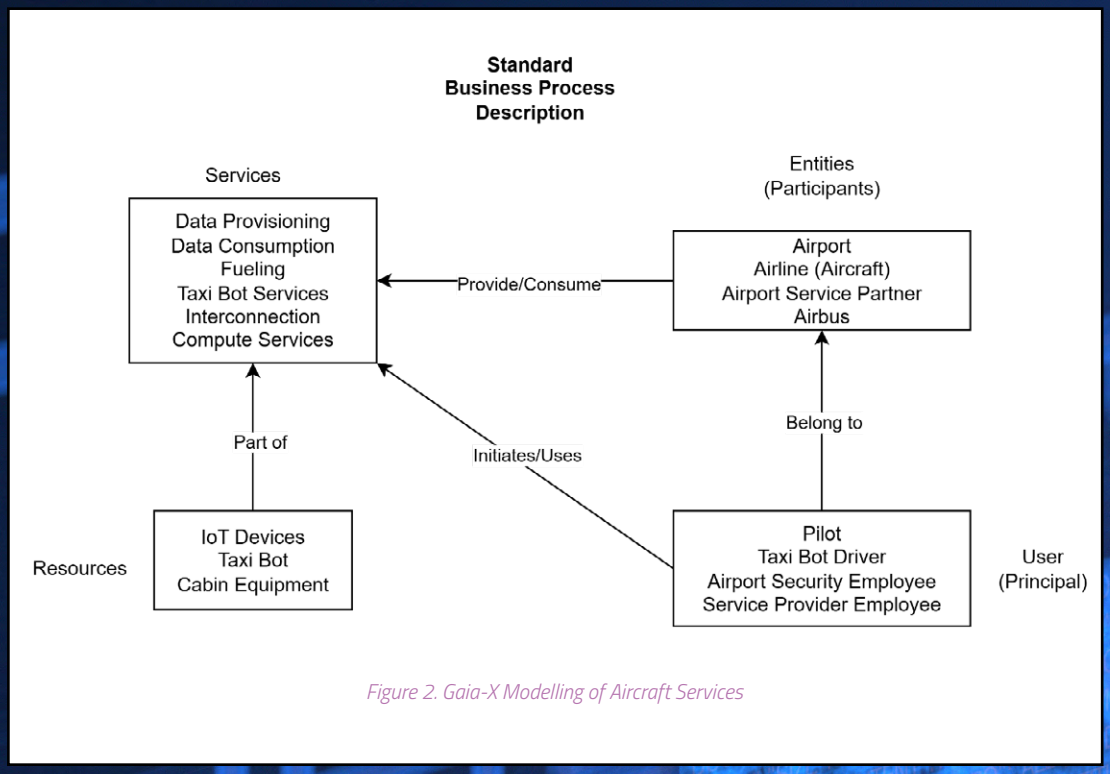


Figure 2. Gaia-X Modelling of Aircraft Services

A conceptual model of interactions between Participants, Services, Resources, and Principals is a first step to build a federation model with a set of federation services to make it operational.

Federation services are key components to establish and manage federated digital ecosystems. Due to the complexity of many-to-many relations between the various actors in aerospace (like manufactures, airlines, airports, 3rd party service suppliers), it is hard to manage new connections, implement agile business service scenarios, and to safeguard security and compliance requirements at the same time. The concept of Gaia-X allows to build individual trust areas and connect them by state-of-the-art trust mechanism in conjunction with rules for cooperation, mainly for service and data sharing. The motivation of adopting federation services is to open aviation to new actors, build synergies with other industries, sharing IT infrastructure, development tools, and open software source. Within IPCEI-CIS, this should also ease compliance with European data regulations and infrastructure sovereignty towards overseas approaches subject to export control limitations.

Within IPCEI-CIS, the Gaia-X specification foresees a Standard Business Process as shown, where Users belonging to an Entity can initiate and use Services. Services are provided by an Entity and consumed by the Entities to which the initiating User belongs. To provide a Service, it is physically enabled by a collection of Resources.

Such key activities as finding services, negotiating usage of services, and capturing agreements as contracts—which leads to entitlement to exchange data between related participants—are managed with an overall Trust Framework with Trusted Entities and, where applicable, certification authorities. A guiding principle is Zero Trust strategy for all involved participants, principles, and services.

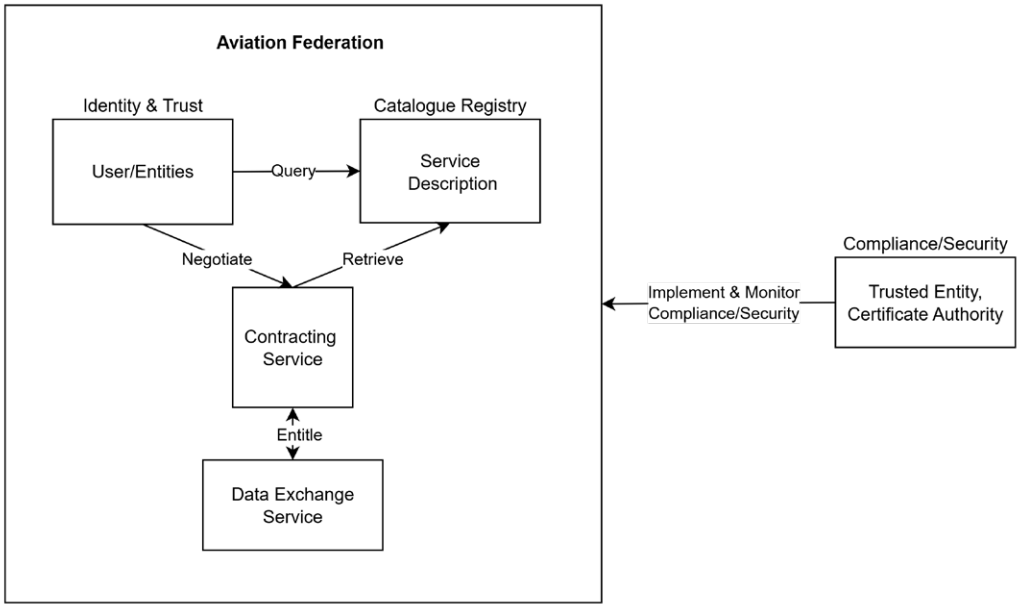


Figure 3. Modelling of Federated Interactions for Service Booking with Trust Anchor.

The architecture and components are designed for portability and independence from legacy systems, supporting scalability and extensibility to other regulated sectors.

2.3 Zero Trust

In response to the collapse of the perimeter model, a new strategic paradigm has emerged: Zero Trust. As a concept and an architecture, Zero Trust represents a complete inversion of the traditional security philosophy. Its foundational principle is "never trust, always verify". It assumes that the network is always hostile and that a breach is inevitable, if not already present. Consequently, Zero Trust eliminates the concept of implicit trust based on network location. Every user, device, and application—regardless of whether it is inside or outside a legacy perimeter—must continuously prove its identity and security posture to be granted access. The focus of security shifts decisively from protecting network segments to protecting the resources themselves—the data, applications, and services—wherever they may reside.

2.4 Scope Clarification & Context

The PoC covers:

- Federated interactions among airlines, airports, travel agencies, manufacturers, and other aviation domain participants.
- Use of Gaia-X Trusted Framework and Cross Federation Services Components (XFSC) for sovereign identity, verifiable credentials, trust anchoring, and federated catalogue management.
- Enforcement of secure, fine-grained, cross-domain access control via ABAC policies.
- Deployment on FACIS testbed using XFSC ORCE (IPaaS) and EasyStack (smart Kubernetes deployment) technologies.
- Aviation domain use cases (e.g, set by the 8ra AXIS project):
  - Distributed digital services collaboration
  - Airport operations
  - Passenger services
  - Aircraft maintenance

2.4.1 Not in Scope of the PoC & Demo

Please note:

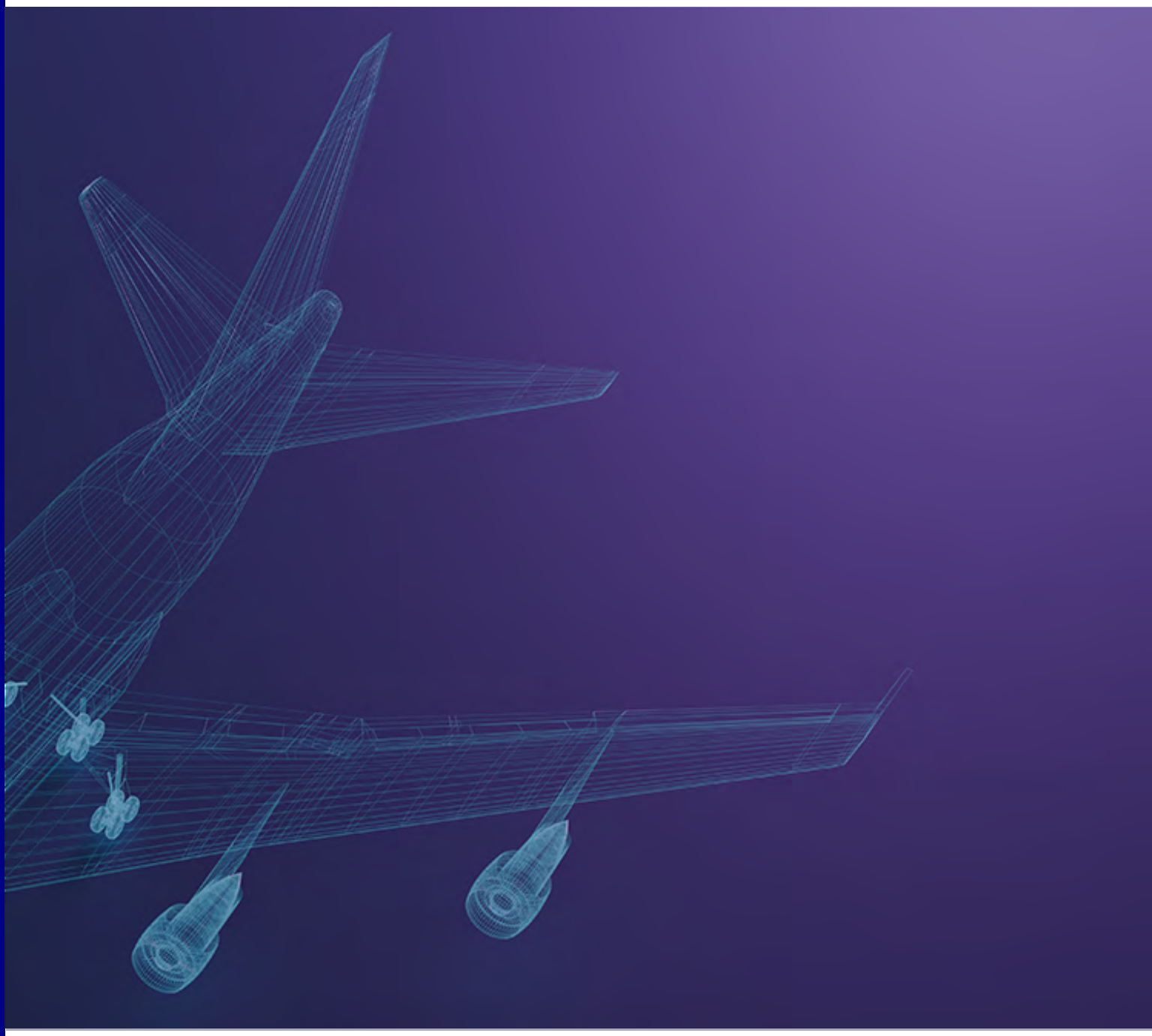
- The formal creation of a Gaia-X-compliant Aviation Federation (governance, trust anchors, federator assignment) **is a formal process** outside the scope of this PoC.
- For the PoC, the Federation **is considered an established prerequisite.**
- The PoC **focuses on operationalization, functional workflows, and interoperability,** not on Federation setup.
- Tooling, reference materials, and guidance **are provided** to support independent Federation setup if needed.

2.4.2 What the Demo Does Show

The demo illustrates the **user journey inside a simulated Federation** via the **Federation Portal** and demonstrates how key **Federation Services** operate within a federated environment.

- **The demo is deployed at: https://aviation.facis.cloud**
- **All services are pre-integrated and operational**
- **All required service endpoints are accessible via the XFSC Orchestrator & Runtime Component Environment (ORCE).**

The **PoC Federator** is represented by our FACIS integration platform, facilitating federation operations and onboarding logic.



2.5 Federation Context (For Reference Only)

A **Federation** can be seen as a group of participants operating under **shared governance** with the goal of solving a common domain challenge, meeting joint legal obligations, or unlocking new business opportunities.

The key aspects of establishing and operating such a federation include:

Key Aspects of a Federation

- **Shared Governance and Purpose**
- **Being founded on a clear mission statement and a defined governance structure. This ensures all participants are aligned to a common goal, whether it is for business, legal, or domain-specific challenges. A "Federator" is appointed to manage the federation's operations and to provide a set of trust anchors, which are set by the governance as reliable resources for Trust Verification.**

A **trust anchor** is a foundational, authoritative entity or a cryptographic object that is implicitly trusted and serves as the starting point for a "chain of trust." In a digital security context, it is a known, verifiable root from which all other trust is derived.

The entire model is built on trust, which is technically established through a trust anchor. The federation has its own Decentralized Identifier (DID) and issues initial Verifiable Credentials (VCs), such as membership credentials, to form the basis of all trusted interactions.

Core Technical Services

A federation is enabled by a set of common, deployable technical services that manage its core functions. The key components mentioned are:

- **TSA (Trust Service API):** Issues and verifies credentials and enforces access control policies.
- **CAT (Federated Catalogue):** A registry where participants can publish and discover services.
- **AAS (Authentication & Authorization Service):** Manages user authentication and access sessions.
- **OCM/PCM (Credential Managers):** Digital wallets for organizations (OCM) and individuals (PCM) to store and present their credentials.
- **ORCE (Orchestrator):** Automates workflows like partner onboarding and service integration.

A Common Interaction Portal

A **Federation Portal** acts as the user-facing entry point for participants. It is used to manage the user journey, including onboarding new partners, publishing services to the catalogue, and accessing the federation's resources.

While the full process of federation creation is crucial in real-world settings, **this PoC demonstrates what happens after a federation has been set up**—enabling focus on how services, identities, and policies interact in a live environment.

Federation Setup—General Concept

While not demonstrated live, the following outlines how a Federation is typically instantiated:

1. Federation Foundation

- Mission statement
- Governance structure
- Federator appointed
- Trust anchor established

---

2. Core Services Deployment

→ **Technical services such as:**

- **TSA** (Trust Service), **CAT** (Catalogue), **AAS** (Authentication & Authorization), **PCM** (Personal Credential Manager), **OCM** (Organizational Credential Manager), **ORCE** (Orchestration & Runtime Environment)

---

3. Portal & Identity Setup

→ **Federation Portal created for participants and onboarding**

→ **Initial digital identity created as trust anchor**

→ **Federation issues VCs, e.g., membership credentials**

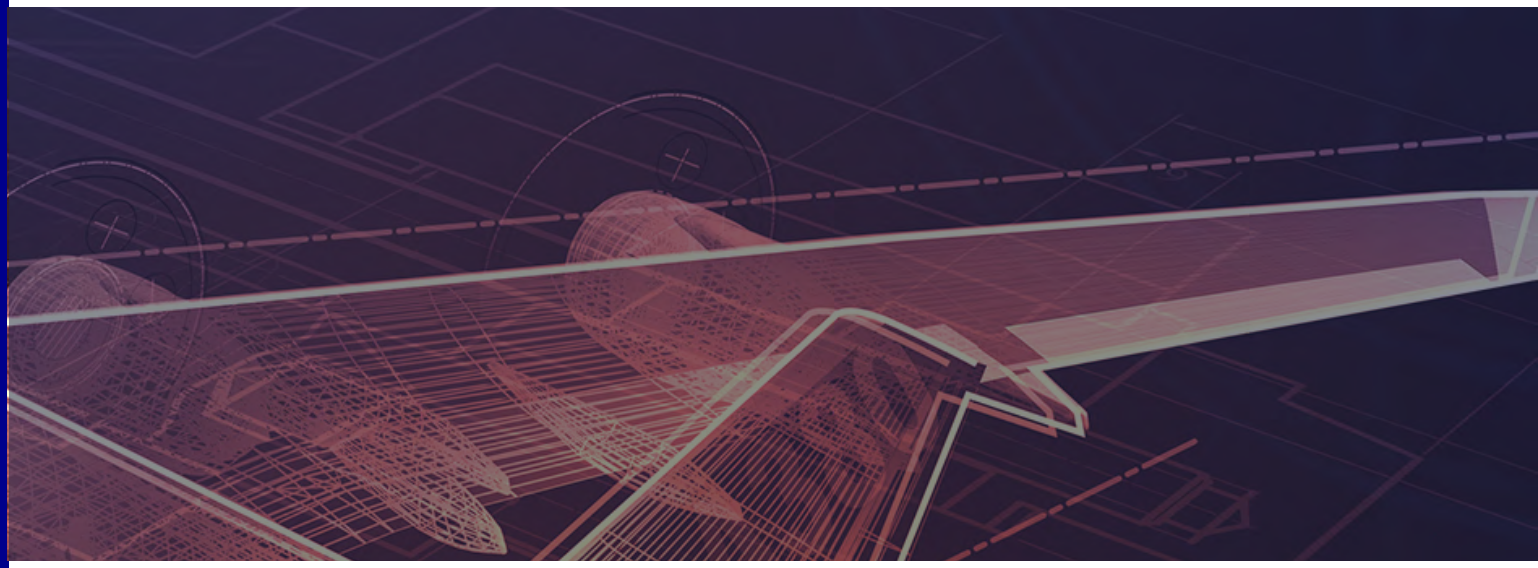
---

4. Participants Preloaded for Demo

→ **One Federation Admin**

→ **One partner service in the catalogue**

→ **Test participant available for onboarding demonstration**



3. Stakeholders and Roles

With the kickstart of Gaia-X back in 2019, a large community of stakeholders has been established to drive the conceptual adoption of federated ecosystems, dataspaces, and multi provider cloud/edge service operations. Several light house projects (Cooperants, Fair Data Spaces) and new formal groups (e.g., Catena-X, Manufacturing-X) were built for various domains.

To materialize this momentum, a so-called IPCEI (Important Project of Common European Interest) for Next Generation Cloud Infrastructure and Services has been defined by the European Member States, together with the European Commission, and is currently driven with the brand name 8ra. The Initiative envisions a future where European businesses, public institutions, and citizens benefit from sovereign, accessible, and high-performance cloud-edge infrastructure. By connecting providers and users across Europe, the 8ra Initiative fosters innovation, strengthens economic resilience, and accelerates the continent's technological leadership.

The eco – Association of the Internet Industry represents the community of digital infrastructure and services with a scope on Internet governance and regulation, further on the evolution of digital business models, cybersecurity and cloud adoption. By this, eco facilitates the 8ra project FACIS, which is at the forefront of shaping the future of digital ecosystems, addressing critical challenges in interoperability, governance, and the increasing demand for flexible, decentralized infrastructures. By combining cutting-edge technologies like Federation Architecture Patterns (FAPs), machine-readable Service Level Agreements (SLAs), and low-code solutions with robust Governance Frameworks, FACIS fosters a seamless Multi-Provider Cloud-Edge Continuum.

Airbus Operations GmbH's AXIS—Aero Concept for Cloud Information Services—project connects aircraft as "flying edge devices" to an IPCEI-CIS-compliant infrastructure. The project focuses on the integration and testing of technological components for communication and data processing in an aircraft on the one hand and on the prototypical definition, implementation and networking of connectivity services on the other. This creates new integrable services for passenger transport processes and "smart services".

Federation Model Overview

A lightweight Aerospace Federation is simulated as prerequisites, involving:

- **Airlines** as service consumers operating aircraft.
- **Airports** providing infrastructure and fueling services.
- **Travel Agencies** managing booking and passenger interactions.
- **Manufacturers** issuing aircraft credentials and digital twins.

These participants operate within a trust framework leveraging Verifiable Credentials, federated catalogue-based service discovery, and cross-domain policy enforcement. In the long run, there are many more ecosystem participants as illustrated in Figure 1.

Use Case Scenarios

The PoC is built around a realistic scenario in which **mobile computing devices** (e.g., crew tablets, EFBs, passenger PEDs) interact securely with **aircraft infrastructure** and backend IT services.

Scenarios:

1. **Aviation Partner Onboarding (Gaia-X Trust Framework)**
2. **Partner Services Listing (Gaia-X Trust Framework)**
3. **Attribute Based Access control (Federation, Participant, Principal)**
4. **Use Case credential Issuing (e.g., Fuel Service) for individual user (Principal)**

4. Architecture

4.1 Architecture Overview

The PoC architecture integrates aerospace participants into a federated, attribute-driven security ecosystem. The following components are central to the design:

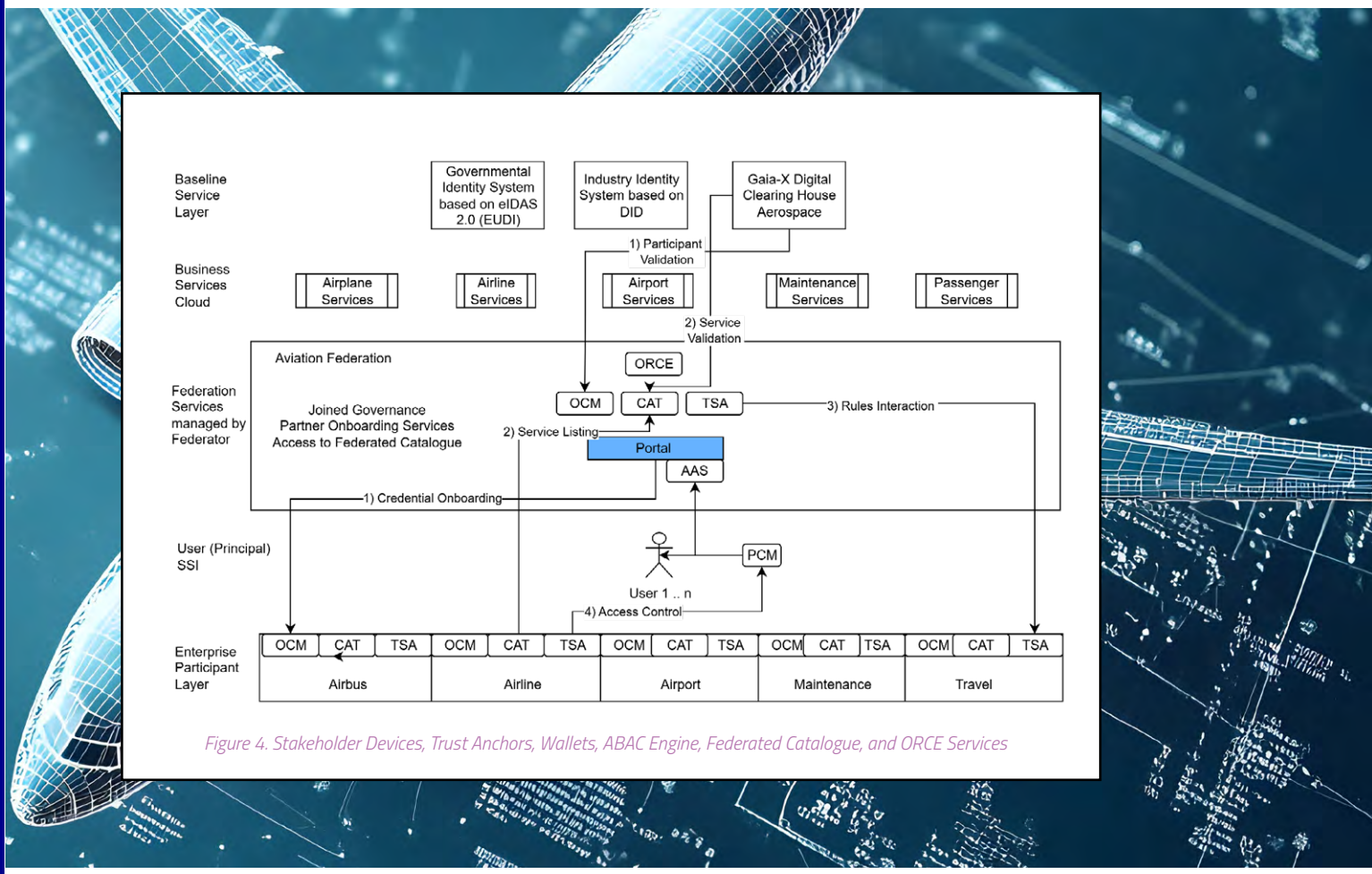


Figure 4. Stakeholder Devices, Trust Anchors, Wallets, ABAC Engine, Federated Catalogue, and ORCE Services

4.2 Core Component

This section outlines the essential components required for secure, federated service orchestration, based on SSI, VCs, and Gaia-X principles. It covers credential management, access control, service cataloguing, orchestration, and secure runtime execution, and maps each component to the relevant standards that enable interoperability, trust, and data sovereignty.

4.2.1 Digital Wallets (OCM, PCM)

Function: Manage, store, and present verifiable credentials for organizations and individuals with support for selective disclosure and ZKPs.

Role in Federation: Act as holders and presenters of trust-based credentials (identity, attributes, membership).

Table 1. Digital Wallets Technical Artifacts

Aspect	Description
OCM	**Manages organizational credentials (e.g., Gaia-X participant credential)**
PCM	Manages personal/self-sovereign credentials
Standards	**- W3C VCs:** Data model for credentals **- DID:** Identifier format for issuers/subjects **- OpenID Connect for Verifiable Presentations (OIDC4VP):** Authentication and selective disclosure **- ISO/IEC 18013-5** (optional): mDL interaction models **- ZKP Support:** Enables privacy-preserving disclosure
Interfaces	- Trust Service API (for issuance and validation) - Policy engine (for ABAC evaluations) - VC APIs and DIDComm
Features	**- Offline VC presentation** **- Web/mobile UX support** **- Holder binding and credential expiry tracking**

4.2.2 Trust Service API (TSA)

Function: Core trust broker that issues, verifies, and manages lifecycle of credentials, schemas, and ABAC-based access control.

Role in Federation: Provides interoperability anchor for identity and trust exchanges.

Table 2. Trust Services API Technical Artifacts

Aspect	Description
Functions	**- Issuance of VCs** **- Credential Verification** **- Schema Registry (for VC types)** **- ABAC Policy Evaluation** **- Revocation Registry**
Standards & Tools	**Gaia-X Trust Framework:** Governance and identity alignment - **DIDComm:** Secure credential exchange - **SD-JWT:** Selective disclosure token format - **XACML:** Policy definition and enforcement - **Open Policy Agent (OPA):** Runtime policy engine
Integration	**- Wallets (OCM/PCM) for credential flow** **- Catalogue (CAT) for policy-linked resource access** **- Backend service APIs for trust enforcement**

4.2.3 Federated Catalogue (CAT)

Function: Distributed asset and service registry with verifiable metadata and access policies.

Role in Federation: Acts as a decentralized catalogue of trusted participants, services, and data offerings.

Table 3. Federated Catalogue Technical Artifacts

Aspect	Description
Functions	- Metadata publication (self-descriptions) - Access policy linkage to services and assets - SLA templating
Standards	- **JSON-LD:** Structured metadata format - **RDF:** Ontology support and graph linkage - **Gaia-X Self-Descriptions (SOT/SUP):** Schema templates for assets, organizations, services
Interfaces	- Public APIs for participants and orchestration - Interaction with TSA for access policy reference - Linkage with Portal and ORCE runtime

4.2.4 Authentication and Authorization Services (AAS)

Function: Enforces access control by verifying credential-based identities and establishing session contexts.

Role in Federation: Serves as bridge between Verifiable Credentials and ABAC-driven access authorization.

Table 4. Authentication and Authorization Services Technical Artifacts

Aspect	Description
Functions	- Personal credential validation - Session key generation - Session-bound policy enforcement
Standards	- **W3C VC / OIDC4VP:** Identity verification - **ABAC / XACML:** Context-based decisioning
Integration	- PCM Wallet for holder VCs - TSA for policy and schema validation - Federated resource endpoints for access session management

4.2.5 Service Orchestration (ORCE)

Function: Automates component deployment, onboarding processes, and runtime coordination of services.

Role in Federation: Enables lifecycle automation, multi-tenant deployment, and secure workflows.

Table 5. Service Orchestration Technical Artifacts

Aspect	Description
Functions	- **Automated onboarding and orchestration** - **Credential validation pipelines** - **Policy-driven flow execution**
Tools	- **Kubernetes** for container orchestration - **Helm & Terraform** for declarative infrastructure - **OpenTelemetry** for observability
Standards	- **Gaia-X Compliance** for service interactions - **CNF (Cloud-Native Functions)** for modularity
Integration	- Interacts with all federation components - Drives user flow in the Portal UI prototype

4.2.6 Portal (UI Prototype)

Function: Demonstrates federated interaction logic: onboarding, VC issuance, and secure service access.

Role in Federation: User-facing component for operational flows and PoC navigation.

Table 6. Portal Technical Artifacts

Aspect	Description
Features	- **Participant registration** - **Credential exchange and acceptance** - **Federated service access and policy display**
Technologies	- React / Web UI Frameworks - RESTful and VC APIs - Integration with TSA, AAS, ORCE

4.2.7 Secure Execution Layer Deployed by XFSC EasyStack

Function: Provides infrastructure isolation, secure deployment, and runtime integrity for federated components.

Role in Federation: Trusted base layer ensuring that all workloads run securely and with tenant isolation.

Table 7. Secure Execution Layer Technical Artifacts

Aspect	Description
Functions	**- Federation-compliant IaaS provisioning** **- Workload sandboxing and lifecycle management** **- Logging and telemetry tracking**
Platform	Hosted on IONOS Sovereign Cloud testbed
Standards & Tools	**- Helm, Terraform for IaC** **- OpenTelemetry, Prometheus, Loki, Jaeger for observability and traceability**
Integration	- Fully aligned with ORCE - Trusted deployment target for all federation services

4.3 Standards-to-Functional Clusters Mapping

Table 8. Standards-to-Functional Clusters Mapping Overview

Functional Cluster	Relevant Standards & Specifications
Identity & Credential Management	- **W3C VC** – structured credential data - **Decentralized Identifiers (DID)** – subject identifier - **DIDComm** – secure communication protocol for VCs - **OpenID Connect (OIDC)** – user authentication - **SD-JWT** – selective disclosure of identity attributes
Access Control & Policy	- **ABAC** – contextual policy evaluation based on attributes - **XACML** – policy language and decision model - **OPA (Open Policy Agent)** – distributed policy evaluation runtime - **Rego** – policy definition language
Service Metadata & Discovery	- **JSON-LD** – structured metadata for service descriptions - **RDF** – semantic relationships and linked data - **Gaia-X Self-Descriptions** – standard templates for participants and assets
Credential Issuance & Verification	- **W3C VC Status List** – credential revocation and status - **SD-JWT** – selective disclosure in JWT-based tokens - **DIDComm** – secure credential transmission
Security & Privacy	- **TLS 1.3** – encrypted data transport - **GDPR** – compliance with European privacy regulation - **Zero-Knowledge Proofs ZKPs** – privacy-preserving authentication
Deployment & Orchestration	- **Helm, Terraform** – infrastructure-as-code for Kubernetes - **Kubernetes** – container orchestration - **XFSC ORCE** – service orchestration for federated workloads
Observability & Monitoring	- **OpenTelemetry** – unified telemetry framework - **Prometheus** – metrics collection - **Loki** – log aggregation - **Jaeger** – distributed tracing

4.4 Summary of Key Interactions

Table 9. Summary of Key Interactions

Component	Core Role	Interacts With
OCM/PCM	**Credential storage & presentation**	**Trust anchor, policy engine, Portal**
TSA (Trust anchor + Policy engine)	Credential lifecycle & policy evaluation	Wallets, AAS, Catalogue
CAT	**Service registry and metadata publishing**	**TSA, ORCE, External Participants**
ORCE	Orchestration of services and policies	XFSC EasyStack, CAT, TSA
AAS	**SSI-based user authentication and access**	**PCM, TSA**
XFSC EasyStack	Federated infrastructure execution layer	ORCE
Portal	**User interaction and process visualization**	**All of the above**

4.5 Key Entities and Actors

Table 10. Key Entities and Actors

Entity/Role	Description
Federation (Aviation Federation)	**The Gaia-X-compliant network for aviation service providers. It has its own DID and issues credentials.**
Federation Admin (Federator)	Manages onboarding, service listing, and credential issuance. Sets trust rules in TSA.
Partner Organization	**An external aviation provider (e.g., FuelCo) applying for onboarding.**
Partner Admin/Officer	Represents the partner during onboarding. Issues internal credentials to staff.
Principal (User)	**An end-user with a VC (e.g., fuel technician) accessing services via AAS.**

5. Implementation Steps & Demo Walkthrough

5.1 Prerequisites

Before any onboarding or interaction can take place, the following prerequisites for the initialization of a federation must be fulfilled:

- Federation legal and governance entity established
- Federation DID created
- Initial Federation Admin identity created (Federator)
- Federation deployed using XFSC EasyStack with the following services:
  - ORCE: Onboarding and orchestration workflows
  - OCM: Organizational credential storage and validation
  - PCM: Personal credential wallet and management
  - AAS: Authorization and session handling for SSI users
  - CAT: Federation-compliant service and asset catalogue
  - TSA: Rule definition and credential issuance authority
- Federation Admin UI is operational
- Federation trust anchor and credential schemas defined

All these components are fully domain agnostic and can be used to build any other type of federation.

5.2 Step-by-Step Guide

5.2.1 Step 1: Partner Onboarding

Goal

Create Gaia-X compliant partner profiles for aviation stakeholders and onboard them into the Aviation Federation's Organizational Credential Manager (OCM).

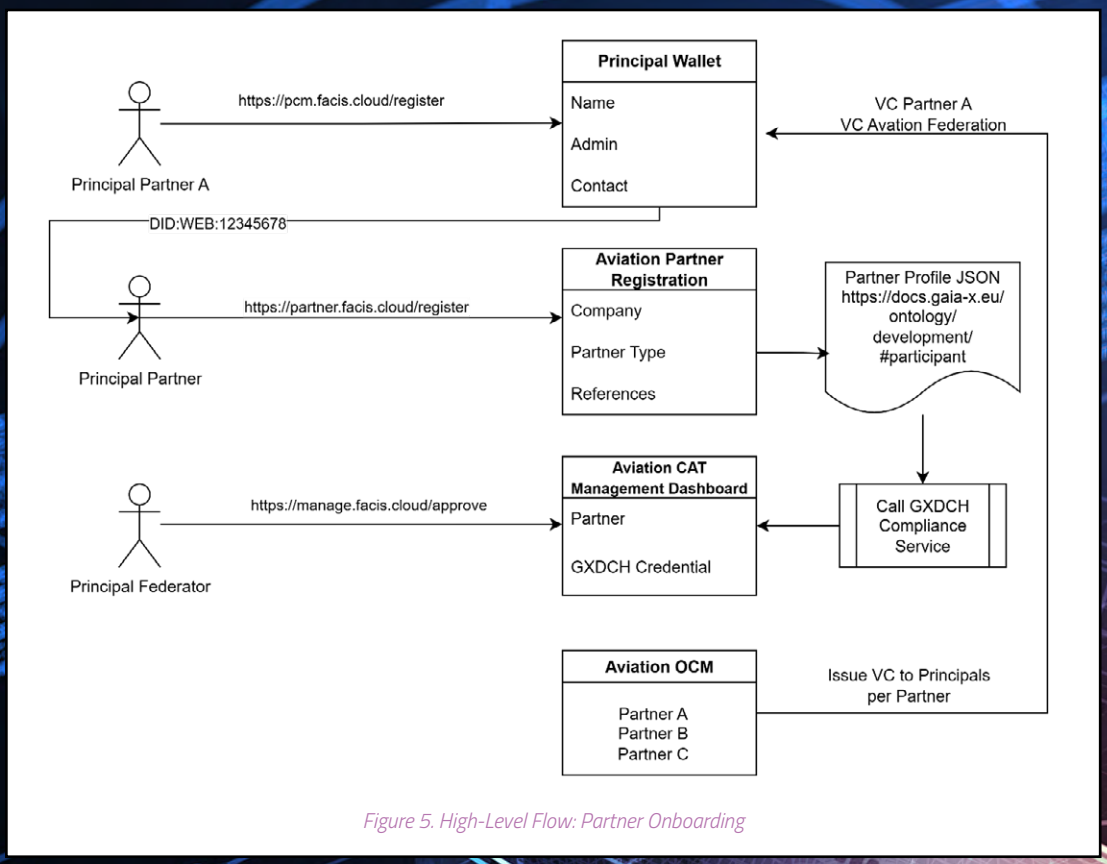


Figure 5. High-Level Flow: Partner Onboarding

Potential Partners

- Airbus (Aircraft Manufacturer)
- Airport (Infrastructure Provider)
- Airline (Carrier)
- Fuel Partner (Fuel Services)
- Maintenance Partner

Components

ORCE (Orchestrator), OCM (Organizational Credential Manager), TSA (Trust Service API)

Actors

- Partner Representative (onboarding user)
- ORCE (workflow engine)
- OCM (partner profile repository)
- TSA (compliance validator & VC issuer)

Steps

1. **Create DID:** Partner generates a Decentralized Identifier (DID) to establish a self-sovereign identity.
2. **Populate Partner Profile:** Partner uploads organizational self-description including certifications, compliance documents, and metadata.
3. **Verification:** TSA validates compliance (e.g., ISO 27001 certification).
4. **Issue Onboarding VC:** TSA issues a Gaia-X compliant VC capturing the partner's compliance status.
5. **Store Profile:** OCM stores the verified profile linked with the partner DID.
6. **Confirmation:** Partner receives confirmation and onboarding badge in a digital wallet.

Demo Flow

Table 11. Demo Flow; Partner Onboarding

Step	Action	Interface	Expected Outcome
1	**Partner logs into Federation Portal**	**aviation.facis.cloud/ federation-portal**	**DID creation interface shown**
2	Upload self-description and compliance docs	Federation Portal UI	Upload confirmation
3	**TSA validates compliance and issues onboarding VC**	**Backend service**	**VC issued, stored in OCM**
4	Partner views issued Gaia-X Federation Badge in wallet	Demo wallet app UI	Federation badge displayed

Partner Profile JSON Schema (simplified example)

```
{
 „$schema“: „http://json-schema.org/draft-07/schema#“,
 „title“: „AviationPartnerProfile“,
 „type“: „object“,
 „properties“: {
  „did“: {
   „type“: „string“,
   „description“: „Decentralized Identifier of the partner“
  },
  „organizationName“: {
   „type“: „string“
  },
  „organizationType“: {
   „type“: „string“,
   „enum“: [„Manufacturer“, „Airport“, „Airline“, „FuelSupplier“, „Maintenance“]
  },
  „contactEmail“: {
   „type“: „string“,
   „format“: „email“
  },
  „complianceCertifications“: {
   „type“: „array“,
   „items“: {
    „type“: „object“,
    „properties“: {
     „certificationName“: {„type“: „string“},
     „issuedBy“: {„type“: „string“},
     „validFrom“: {„type“: „string“, „format“: „date“},
     „validTo“: {„type“: „string“, „format“: „date“}
    },
    „required“: [„certificationName“, „issuedBy“, „validFrom“]
   }
  },
  „serviceEndpoints“: {
   „type“: „array“,
   „items“: {
    „type“: „string“,
    „format“: „uri“
   }
  }
 },
 „required“: [„did“, „organizationName“, „organizationType“, „contactEmail“]
}
```

5.2.2 Step 2: Partner Services Listing

Goal

Partners publish their offered services to the Aviation Federation's Service Catalogue.

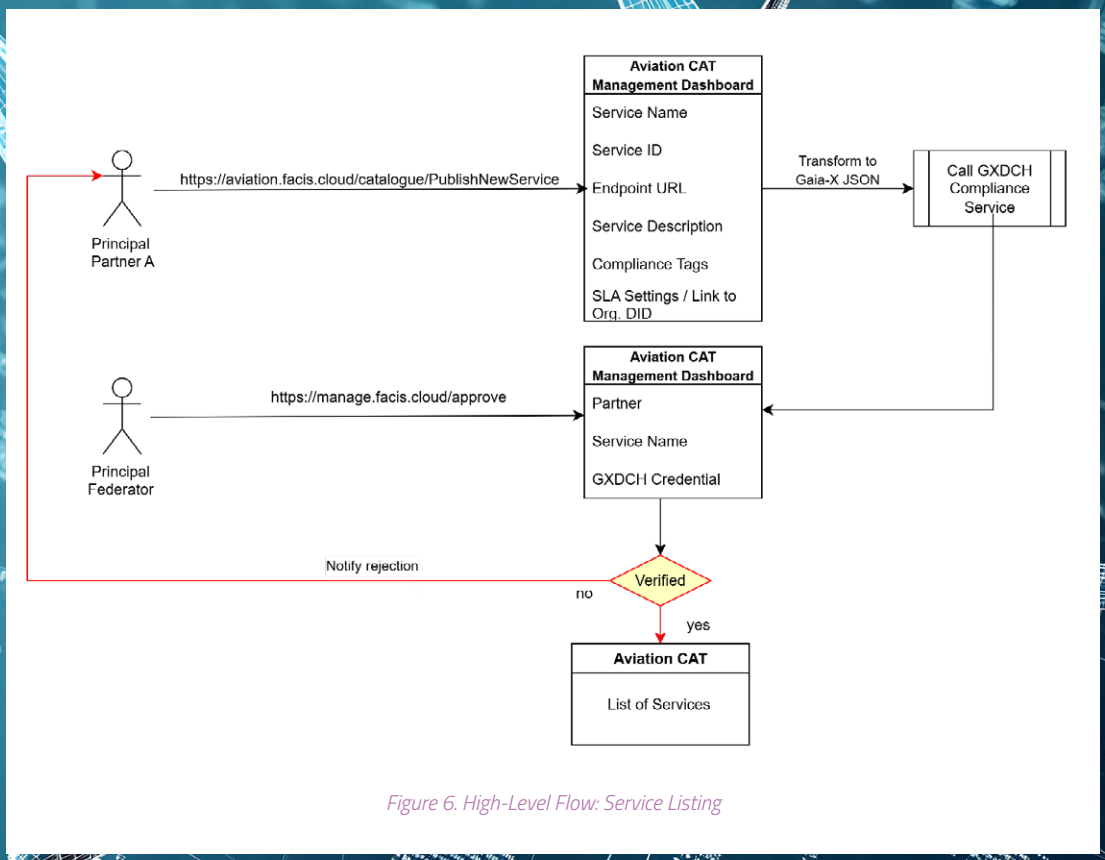


Figure 6. High-Level Flow: Service Listing

Actors

- Partner Admin (service owner)
- Catalogue Service (CAT)
- OCM (identity and proof store)

Steps

1. Partner defines service metadata (e.g., API endpoint, SLA, compliance tags).
2. Partner publishes service to the catalogue via CAT.
3. CAT registers the service, links metadata to partner DID.
4. OCM provides cryptographic proofs of the partner's identity and compliance status.
5. Service appears in a searchable, Gaia-X compliant federation-wide catalogue.

Demo Flow

Table 12. Demo Flow; Partner Services Listing

Step	Action	Interface	Expected Outcome
1	**Partner enters service details**	**aviation.facis.cloud/catalogue-management**	**Form to input metadata**
2	Publish service to the federation catalogue	CAT UI	Service listed with Gaia-X metadata
3	**View service listing**	**Federation-wide catalogue UI**	**Searchable entry with metadata**

Service Metadata JSON Schema (simplified)

```
{
 „$schema“: „http://json-schema.org/draft-07/schema#“,
 „title“: „AviationServiceMetadata“,
 „type“: „object“,
 „properties“: {
  „serviceId“: {„type“: „string“},
  „serviceName“: {„type“: „string“},
  „description“: {„type“: „string“},
  „providerDid“: {„type“: „string“},
  „endpointUrl“: {„type“: „string“, „format“: „uri“},
  „sla“: {
   „type“: „object“,
   „properties“: {
    „availability“: {„type“: „string“, „pattern“: „^\\d{1,3}%$“},
    „responseTimeMs“: {„type“: „integer“}
   }
  },
  „complianceTags“: {
   „type“: „array“,
   „items“: {„type“: „string“}
  },
  „publishedAt“: {„type“: „string“, „format“: „date-time“}
 },
 „required“: [„serviceId“, „serviceName“, „providerDid“, „endpointUrl“]
}
```

5.2.3 Step 3: Rule Definition & Trust Policy Setup

Goal

Define access control policies that govern service consumption in the federation.

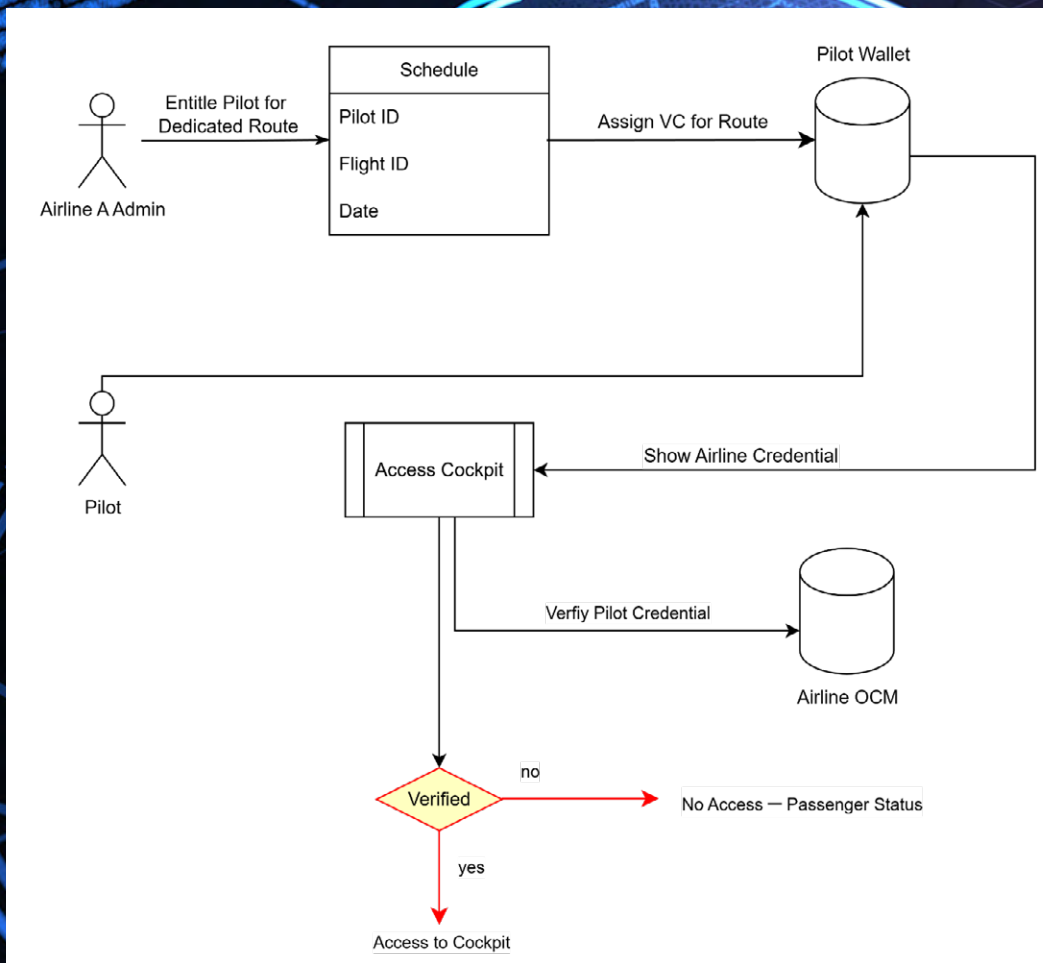


Figure 7. High-Level Flow: ABAC

Actors

- Federation Admin
- TSA (Policy engine)
- ORCE (Workflow orchestration)

Steps

1. Admin defines policies specifying which partners/users can access specific services (e.g., "Only ISO 27001 certified partners may access Fuel Planning API").
2. Policies are expressed in ABAC format and uploaded to TSA.
3. TSA enforces policies and integrates with ORCE for runtime enforcement.

Demo Flow

Table 13. Demo Flow; Rule Definition & Trust Policy Setup

Step	Action	Interface	Expected Outcome
1	**Create new access rule with conditions**	**aviation.facis.cloud/ trust-config**	**Policy editor with validation**
2	Save policy linked to service	TSA backend	Policy stored and active
3	**ORCE shows enforcement during orchestration**	**ORCE dashboard**	**Active enforcement logs**

Sample ABAC Policy Templates

Fuel Planning API Access Policy

```
{
 „policy“: „fuel_planning_api_access“,
 „rules“: [
  {
   „effect“: „permit“,
   „condition“: {
    „and“: [
     {„credential.type“: „PartnerCredential“},
     {„credential.complianceCertifications[].certificationName“: „ISO 27001“},
     {„resource.serviceName“: „Fuel Planning API“}
    ]
   }
  }
 ]
}
```

5.2.4 Step 4: Pilot Access Control Scenario

Goal

Demonstrate dynamic, fine-grained access control to aircraft cockpit systems using verifiable credentials and ABAC policies.

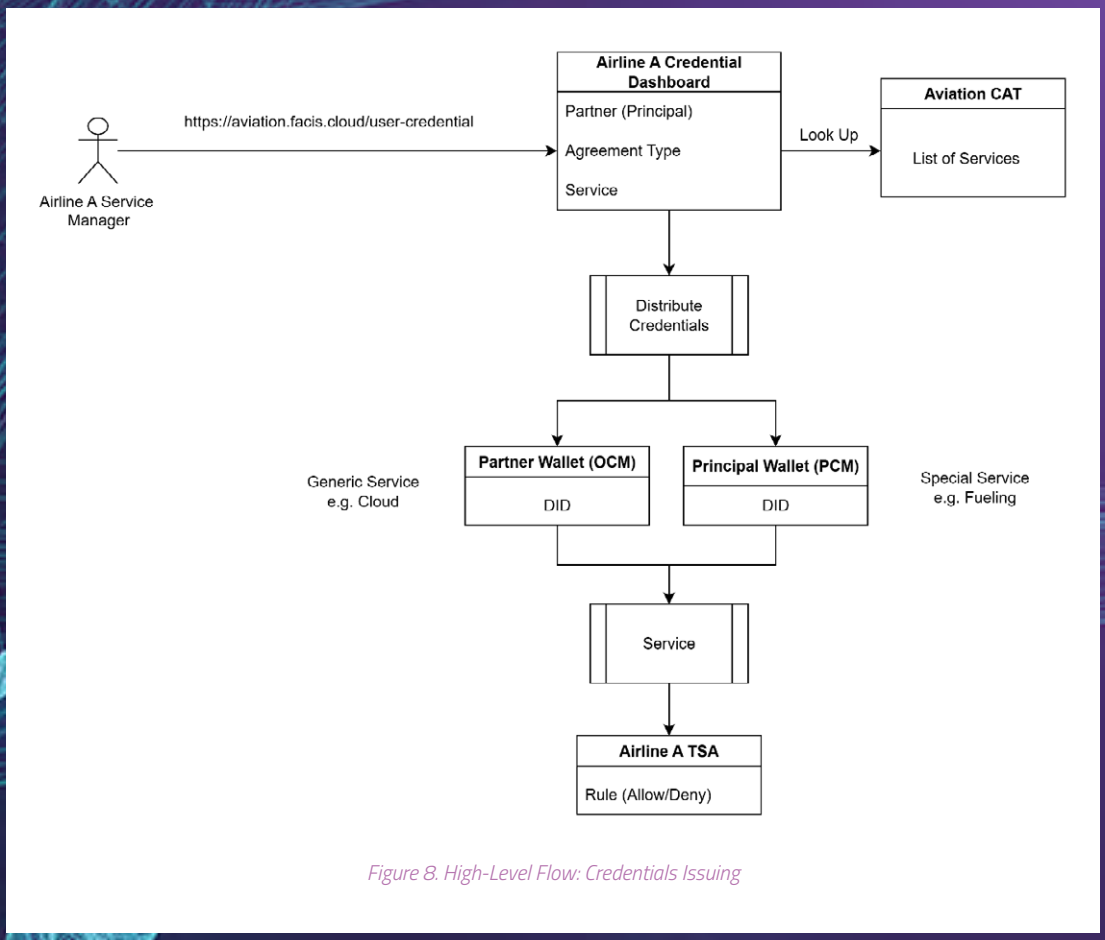


Figure 8. High-Level Flow: Credentials Issuing

Actors

- Credential Issuer (Airline or Aviation Authority)
- Personal Credential Manager (Pilot's Wallet App)
- Verifier (Aircraft On-board Computer / Gate Access System)
- Policy Decision Point (ABAC engine in TSA)

Steps

1. Verifiable Credential Issuance:

Airline issues a route- and role-specific Pilot License VC to the pilot.


```
{
 „@context“: [„https://www.w3.org/2018/credentials/v1“],
 „type“: [„VerifiableCredential“, „PilotLicense“],
 „issuer“: „did:example:airline123“,
 „issuanceDate“: „2024-01-15T10:00:00Z“,
 „credentialSubject“: {
  „id“: „did:example:pilot456“,
  „licenseNumber“: „ATP-12345“,
  „aircraftTypes“: [„A320“, „A321“],
  „validRoutes“: [
   {
    „from“: „FRA“,
    „to“: „MUC“,
    „flightNumber“: „LH2050“,
    „date“: „2025-06-29“,
    „role“: „pilot“
   },
   {
    „from“: „MUC“,
    „to“: „FRA“,
    „flightNumber“: „LH2051“,
    „date“: „2025-06-29“,
    „role“: „passenger“
   }
  ]
 }
}
```


2. Assign Credential:

→ VC is pushed to pilot's secure digital wallet.

3. Define Policy in TSA (ABAC Rule):


```
{
 „policy“: „cockpit_access“,
 „rules“: [
  {
   „effect“: „permit“,
   „condition“: {
    „and“: [
     {„credential.type“: „PilotLicense“},
     {„flight.number“: „credential.validRoutes[].flightNumber“},
     {„credential.validRoutes[].role“: „pilot“},
     {„current.date“: „credential.validRoutes[].date“},
     {„flight.route“: „credential.validRoutes[].from->to“}
    ]
   }
  }
 ]
}
```

5.2.5 Step 5: Access Request and Evaluation

- Pilot presents VC at gate or cockpit system.
- Verifier extracts attribute and sends to ABAC engine.
- Policy decision grants or denies access dynamically (e.g., cockpit access if role=pilot on flight date, cabin access if role=passenger).

Implementation Details

Table 14. Implementation Details; Access Request and Evaluation

Step	Description
Credential issuance	Airline digitally signs route-specific pilot VCs with DID-based identities
Access request	Pilot presents credential via wallet to verifier
Policy evaluation	TSA ABAC engine evaluates conditions dynamically
Authorization	Access granted or denied per policy
Wallet integration	Automatic VC selection, offline validation enabled

Security Aspects

- Zero-Knowledge Proofs: Only minimal required attributes disclosed to verifier.
- Temporal Constraints: VCs valid only for specified flight dates.
- Revocation: Distributed revocation registry prevents use of invalid credentials.
- Audit Trail: All access attempts logged for compliance and forensic analysis.

5.2.6 Step 6: Use Case Credential Issuing (e.g., Ground Staff Access)

Goal

Issue use-case-specific VCs for operational roles such as fuel refueling staff.

Actors

- TSA (Credential issuer)
- Personal Credential Manager (User's Wallet)
- AAS (Authentication & Session management)

Steps

1. User requests VC via portal for specific use case (e.g., "Fuel Refueling Access").
2. TSA verifies user identity and compliance.
3. VC issued and pushed to user wallet.
4. User receives QR code or token to access protected services.

Demo Flow

Table 15. Demo Flow; Use Case Credential Issuing

Step	Action	Interface	Expected Outcome
1	**User requests credential**	**aviation.facis.cloud/ user-credential**	**Request form submitted**
2	TSA issues use-case VC	Backend service	VC appears in wallet
3	**User accesses service with credential**	**aviation.facis.cloud/ service-access**	**Access granted or denied**

5.2.7 Step 7: Accessing Protected Services

Goal

Demonstrate runtime enforcement of access control policies based on presented VCs.

Actors

- User presenting VC
- AAS (Verifier & session manager)
- ORCE (Orchestration of validation workflows)

Steps

1. User presents credential for service access.
2. AAS verifies credential integrity and authenticity.
3. ORCE invokes TSA to evaluate applicable policies.
4. Access granted or denied based on ABAC evaluation.
5. Access logs recorded and session token issued

Demo Flow

Table 16. Demo Flow; Accessing Protected Services

Step	Action	Interface	Expected Outcome
1	**User initiates access request**	**aviation.facis.cloud/ service-access**	**Credential submission UI**
2	Credential verification	AAS backend	Credential validity confirmed
3	**Policy evaluation and access decision**	**TSA & ORCE**	**Access granted or denied**
4	Session management and logging	ORCE	Session token & audit log updated

5.3 Walkthrough Demo Guide with Descriptions

This section highlights what the demo should show:

Success Criteria

- **Federation deployed with DID and services**
- **Partner onboarded and issued org VC**
- **Partner service listed in CAT**
- **User issued VC for technician role**
- **User accessed a service with valid VC**
- **TSA rule enforced access control**

Demo Walkthrough Guide

5.3.1 Step 1: Aviation Partner Onboarding

UI: Aviation Federation Portal — Partner Registration
URL: aviation.facis.cloud/federation-portal

Layout:

- **Header:** "Aviation Federation Portal — Partner Onboarding"
- **Left Panel:** Federation Overview and Contact
- **Main Form Panel:**
  - Organization Name (Input)
  - Organization Type (Dropdown)
  - Decentralized Identifier (DID) (Input)
  - Compliance Certifications (Add multiple — e.g., ISO 27001)
  - Upload Self-Description File (JSON-LD) (Upload button)
  - Submit Profile (Primary button)

Feedback on Submission:

- **Success Modal:** "Profile Submitted. TSA Verification in Progress."
- **After TSA validation:**
  - Notification: "Your Organization has been verified."
  - Badge: Federation Trust Badge appears near profile
  - Credential stored in backend OCM

Interactions:

→ **Upload parsing validates JSON-LD format**

→ **Live DID lookup using DID resolver endpoint**

5.3.2 Step 2: Service Publishing

UI: Federated Catalogue Management Dashboard
URL: aviation.facis.cloud/catalogue-management

Layout:

→ **Header: "Service Management"**

→ **Left Navigation:**

- My Services
- Publish New

→ **Main Panel:**

- Service Name (Input)
- Service ID (Auto-generated or entered)
- Endpoint URL (Validated URI field)
- Service Description (Textarea)
- Compliance Tags (Multi-select)
- SLA Settings:
  - Availability (%)
  - Max Response Time (ms)
- Link to Organizational DID (auto)
- Publish Service (Button)

Feedback:

- Inline validation of fields
- On success: "Service published to Federated Catalogue."
- Entry becomes visible on aviation.facis.cloud/catalogue

5.3.3 Step 3: Access Policies Setup

UI: Trust Service Authority (TSA) Admin Panel
URL: aviation.facis.cloud/trust-config

Layout:

- **Header:** "ABAC Policy Management"
- **Left Panel:** Policies List
- **Main Panel:**
  - JSON Editor Panel with syntax highlighting
  - Policy Test Engine
  - "Apply Policy" (Button)

Workflow:

- Admin selects the service
- Loads existing policy or creates new JSON template
- Uses test interface with example credentials to preview decisions
- Saves and activates policy

5.3.4 Step 4: Credential Issuing

UI: Credential Request Portal
URL: aviation.facis.cloud/user-credential

Layout:

- **Header:** "Issue Access Credentials"
- **User View:**
  - Select Role (Pilot, Crew, Ground Staff)
  - Select Credential Type (Dropdown)
  - Flight Route / Assignment (Dynamic Fields)
  - Request Credential (Button)

Backend Flow:

- TSA issues VC
- PCM stores it in wallet
- AAS pre-validates keys

Feedback:

- Credential card appears in dashboard
- "Credential ready for use"

5.3.5 Step 5: Access Service

UI: Protected Service Access Page
URL: aviation.facis.cloud/service-access

Layout:

- **Header:** "Secure Access Portal"
- **User View:**
  - Credential Selector: Lists issued VCs
  - Context Selection (flight, device, timestamp)
  - Present Credential (Button)
- **Result Panel:**
  - Access Decision: Permit / Deny
  - Reason or Match Trace
  - Access Token (if granted)
  - Link to secure service endpoint

Verifier System:

- AAS extracts and evaluates policy conditions
- Logs match or failure reason (e.g., missing route attribute)

5.3.6 Optional: Admin Dashboard View

A unified dashboard for Federation Admin includes:

- Partner Status Overview
- Credential Issuance Metrics
- Service Registry View
- Policy Violation Logs
- Export for audit trails

6. Deployment Architecture

6.1 Objectives

The deployment architecture is designed to support a modular, secure, and scalable federation model tailored for aviation stakeholders. Key objectives include:

- Enabling **modular and automated deployment** of federation services.
- Utilizing **Kubernetes orchestration** through XFSC ORCE.
- Supporting **multi-tenant, domain-isolated deployments** in a secure cloud environment.

6.2 Target Environment

- **Cloud Region:** Germany (within the European Union sovereign zone, IONOS testbed).
- **Orchestration Tools:** Kubernetes with **Helm** and **Terraform**, integrated via XFSC ORCE.
- **Security & Observability:**
  - End-to-end encryption (TLS)
  - Identity and Access Management (IAM)
  - Federated observability using the **OpenTelemetry** stack (metrics, logs, traces)

6.3 Logical Domain Composition (Per Participant)

Each participating organization (e.g., airline, service provider, ground operator) deploys the following logical services (as isolated pods or containers):

- **Digital Wallet Interface** – for verifiable credential management (mobile/web)
- **Trust Anchor Service** – for identity verification and credential issuance
- **Federated Catalogue Node** – for service registration and discovery
- **ABAC Policy Engine** – for attribute-based access control
- **Business APIs** – e.g., Aircraft Gateway, DXP Gateway (external system connectors)
- **Secure Gateway & Edge Controller** – for ingress control and routing
- **Observability Stack** – Prometheus, Loki, Jaeger (telemetry collection and visualization)

6.4 Deployment Model

Table 17. Deployment Model

Component	Deployment Type	Technology Stack
Digital Wallet	Mobile / Web Client	OIDC, VC APIs
Trust Anchor	Pod / Microservice	DIDComm, SD-JWT
Catalogue Node	Microservice	JSON-LD, GXFS APIs
Policy Engine	Sidecar / Pod	OPA (Open Policy Agent), XACML
ORCE Services	Kubernetes Stack	Helm, Terraform

7. Requirements

7.1 Functional Requirements

Table 18. Functional Requirements

ID	Requirement Description	Priority	Notes
FR-01	**The system shall support issuance, storage, and presentation of VCs compliant with W3C standards.**	**High**	**Essential for identity and attribute verification.**
FR-02	The digital wallet must allow selective disclosure of attributes, including support for Zero-Knowledge Proofs (ZKP).	High	Ensures privacy-preserving authentication.
FR-03	**The trust anchor shall issue, verify, and revoke credentials using standardized protocols (e.g., DIDComm, SD-JWT).**	**High**	**Critical for trust anchor functionality.**
FR-04	The federated catalogue shall publish service metadata, SLAs, and access policies in machine-readable formats (JSON-LD, RDF).	High	Enables dynamic service discovery and policy enforcement.
FR-05	**The ABAC engine shall enforce access policies based on credential attributes, including dynamic rule updates.**	**High**	**Provides fine-grained, flexible authorization control.**
FR-06	The system shall support federated authentication and authorization across all participating organizations.	Medium	Supports compliance and incident investigation.
FR-07	**All communication between components (wallets, trust anchors, policy engines, services) shall be encrypted (e.g., TLS).**	**High**	**Ensures confidentiality and data integrity.**
FR-08	The system shall enable logging and audit trails for all access requests and credential verifications.	Medium	Supports compliance and incident investigation.
FR-09	**The system shall provide APIs for integration with legacy aerospace systems and third-party platforms (e.g., Amadeus DXP, Diehl AdvantagE).**	**Medium**	**Facilitates interoperability with existing infrastructures.**

7.2 Non-Functional Requirements

Table 19. Non-Functional Requirements

ID	Requirement Description	Priority	Notes
NFR-01	**The system shall be deployable on Kubernetes clusters, supporting automated provisioning via Helm and Terraform.**	**High**	**Ensures scalable and repeatable deployments**
NFR-02	The system shall maintain high availability and fault tolerance for critical components such as trust anchors and policy engines.	High	Ensures operational resilience in federated environments.
NFR-03	**The system shall comply with Gaia-X sovereignty and data privacy principles, including GDPR requirements.**	**High**	**Ensures legal and regulatory compliance.**
NFR-04	The system shall be modular and portable to allow reuse and extension to other domains and cloud providers.	High	Facilitates adaptability and future scalability.
NFR-05	**The system shall provide observability capabilities, including metrics, logging, and tracing, compatible with OpenTelemetry standards.**	**Medium**	**Enables monitoring and troubleshooting**
NFR-06	The digital wallet interface shall be user-friendly and accessible on both mobile and desktop devices.	Medium	Supports user adoption and ease of use.
NFR-07	**The system shall perform authentication and authorization checks with low latency to enable real-time access control.**	**Medium**	**Enables monitoring and troubleshooting**

8. Strategic Impact

This Proof-of-Concept establishes a replicable pattern for secure, federated operations in the aviation domain, demonstrating:

1. Operational resilience through federated trust models.
2. Identity-driven interoperability across multiple stakeholders.
3. Compliance with Gaia-X standards, enhancing European sovereignty and trust.
4. Portability of trust frameworks to other regulated sectors.

9. Abbreviations

Table 20. Abbreviations

Abbreviation	Full Form
8ra	European initiative for a resilient, open, and future-proof digital infrastructure
AAS	Authentication & Authorization Service
ABAC	Attribute-Based Access Control
API	Application Programming Interface
ATC	Air traffic control
CAT	Federated Catalogue
CNF	Cloud-Native Functions
CRD	Custom Resource Definition (Kubernetes)
DID	Decentralized Identifier
DIDComm	Decentralized Identifier Communication Protocol
DLT	Distributed Ledger Technology
DNS	Domain Name System
DXP	Digital Experience Platform
EBSI	European Blockchain Services Infrastructure
EFBs	Electronic flight bag; information management device
eIDAS	Electronic Identification, Authentication and Trust Services
EU	European Union
GAIA-X	European Data Infrastructure Project
GDPR	General Data Protection Regulation

Abbreviation	Full Form
GXDCH	Gaia-X Digital Clearing House
GXFS	Gaia-X Federation Services
IaaS	Infrastructure as a Service (cloud computing model)
IaC	Infrastructure as Code
IAM	Identity and Access Management
IPaaS	Integration Platform as a Service
IPCEI-CIS	EU project 'Next Generation Cloud Infrastructure and Service'
Jaeger	Distributed Tracing
JSON-LD	JavaScript Object Notation for Linked Data
Loki	Log Aggregation
mDL	Main Deck Loader
MRO	Maintenance, Repair and Overhaul
OCM	Organizational Credential Manager
OIDC	OpenID Connect
OIDC4VP	OpenID Connect for Verifiable Presentations
OPA	Open Policy Agent
OpenTelemetry	Observability Framework
ORCE	Orchestrator for Federation Components / Orchestrator & Runtime Component Environment
PCM	Personal Credential Manager
PEDs	Personal/ Portable Electronic Devices
PoC	Proof of Concept
Prometheus	Metrics Collection and Monitoring
Rego	OPA Policy Definition Language
RDF	Resource Description Framework
SD-JWT	Selective Disclosure JSON Web Token
SLA	Selective Disclosure JSON Web Token
SOT	Service Offering Template
SUP	Service Usage Policy
TLS	Transport Layer Security
TSA	Trust Service API

Abbreviation	Full Form
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience
VC	Verifiable Credential
W3C	World Wide Web Consortium
XACML	Extensible access control markup language
XFSC	Cross-Federation Service Components
ZKP	Zero-Knowledge Proof

10. References

- **W3C Verifiable Credentials Data Model 1.0** https://www.w3.org/TR/2019/REC-vc-data-model-20191119/?utm_source=chatgpt.com
- **OpenID Connect Core 1.0** https://openid.net/specs/openid-connect-core-1_0-final.html
- **ISO/IEC 18013-5 (Mobile Driver's License)** https://www.iso.org/standard/69084.html
- **XACML 3.0 – eXtensible Access Control Markup Language** https://www.oasis-open.org/standard/xacmlv3-0/
- **Gaia-X Federation Services Specifications** https://github.com/eclipse-xfsc/docs
- **XFSC ORCE Technical Documentation** https://github.com/eclipse-xfsc/orchestration-engine

VERBAND DER INTERNETWIRTSCHAFT

CLOUD·EDGE CONTINUUM

Funded by the European Union NextGenerationEU

Supported by: Federal Ministry for Economic Affairs and Energy on the basis of a decision by the German Bundestag