facis

Zero Trust Architecture for the Aviation Ecosystem

Digital Collaboration by Federation – Enabling Federated Ecosystem Operations

eco

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

CLOUD·EDGE CONTINUUM

Version 1.1 (March, 18 2026)
ISBN: 978-3-9828074-6-1

Publisher eco - Association of the Internet Industry (editor responsible)
FACIS (eco)
AXIS (Airbus)
EdgeConnect (DTAG)

Authors:

Project	Name	Company
AXIS	Stephane Poulain	Airbus Operations GmbH
	Cristian Bertoldi	Airbus Operations SAS
	Yesmine Dhaouadi	Airbus Operations SAS
EdgeConnect	Ivan Gudymenko	Deutsche Telekom MMS GmbH
	Steffen Schulze	T-Systems International GmbH
	Hasan Yildirim	Deutsche Telekom MMS GmbH
FACIS	Andreas Weiss	eco – Association of the Internet Industry
	Thorsten Kraft	eco – Association of the Internet Industry

Table of Contents

Introduction **1**

Zero Trust Architecture (ZTA) Can Help Address the Challenge 1

Key Principles of Zero Trust Architecture **3**

Civil Aviation Ecosystem Principles **5**

Dealing with Identities **6**

Dealing with Policies **8**

Zero Trust in the Context of Self-Sovereign Identity (SSI) **13**

Integration of VCs into Basic Architecture **15**

User-based Flow 15

M2M Flow 15

Scenarios for Identity, ZTNA, and SSI Usage **17**

Cockpit Onboarding and Door Access (Scenario 1) 18

Ground Data Exchange & Maintenance Operations (Scenario 2) 20

Process Hardening **23**

Aviation Authority Licence Check via IATA 23

Shamir's Secret for Flight Security 23

Connected Aircraft in Federated Ecosystems (AXIS) **24**

Sampling the Aviation Ecosystem (AXIS, FACIS) 25

Conclusions **29**

Next Steps and Outlook 30

Glossary **31**

References **40**

Table of Figures

Figure 1 Level of Trust 5

Figure 2 Actor Relations 6

Figure 3 XACML Architecture Flow 9

Figure 4 Federated Policy Architecture with Decentralized Repository Nodes 12

Figure 5 Community Federation 14

Figure 6 User-based Flow 15

Figure 7 M2M Flow 16

Figure 8 Interaction between aircraft and ground 17

Figure 9 Cockpit Onboarding and Door Access (Scenario 1) 20

Figure 10 Ground Data Exchange & Maintenance Operations (Scenario 2) 21

Figure 11 Maintenance Operations Authorization Flow 22

Figure 12 Use case: aircraft refueling 24

Figure 13 PoC Scenario: Federated Aviation ecosystems 26

Introduction

The aviation industry is currently navigating a complex "security vs. connectivity" dilemma. While the drive for digital transformation requires seamless data exchange, every connection point introduces a potential vector for cyberattacks.

There are several areas to be considered. Modern aircraft must share real-time data with ground IT systems, maintenance crews, and air traffic control to optimize efficiency. Such interactions span various phases, from refueling and catering at the gate through to data sharing with service, maintenance, and travel partners for various use cases.

On the other side, such connections are introducing a wider threat landscape. Every external link – whether via Wi-Fi, cellular, satellite, or direct ground connections to external cloud services – represents a potential entry point for unauthorized access or data tampering.

In any case, the aviation industry operates on the mandate of operational integrity and physical safety.

Zero Trust Architecture (ZTA) Can Help Address the Challenge

Zero Trust is a security framework based on the principle of **"never trust, always verify."** Unlike traditional perimeter-based security, ZTA assumes that threats could already be inside the network and provides architecture patterns for detection and containment, blocking or limiting the propagation to critical systems, thus moving from a protective approach to a responsive one.

Identity-Centric Verification

ZTA moves security away from "trusted networks" to "trusted identities." Whether it is a ground technician's tablet or an automated refueling sensor, the system requires strict authentication and authorization for every single session.

Micro-Segmentation

In a connected aircraft, critical flight systems (avionics) are separated from non-critical systems (such as passenger Wi-Fi or catering logs). The following scenarios will focus on non-critical collaboration.

Continuous Monitoring and Least Privilege

The combination of least privilege – partners only receive access to the specific data they need – and real-time verification (trust is not granted once at the start of a flight; it is continuously re-evaluated based on the device's health, location, and behavior) provides continuous control about access, usage, and modification of data within supervised processes and applications.

Dynamic Policy Enforcement ZTA allows the aviation ecosystem to interact with a multitude of global partners by using policy engines in the context of collaboration. Such engines can grant or revoke access instantly across different jurisdictions and ground IT infrastructures without compromising the aircraft's core security.

ZTA provides the "logic" needed to balance high security with high connectivity. It allows the aircraft to remain a "connected node" in a global network while maintaining a hard shell of data integrity and confidentiality. Zero Trust doesn't just block access; it enables **secure collaboration** by ensuring that every interaction is authenticated, authorized, and encrypted, regardless of where it originates. By this, there is a shift from defending a static perimeter to protecting individual data transactions.

In any case, the usage of ZTA must enhance integrity and safety beyond the current level without compromising existing measures.

Key Principles of Zero Trust Architecture

Zero Trust flips the script of well-known perimeter concepts, such as Firewalls, Proxys, and Demilitarized Zones. It operates on a simple, rigorous motto: **"Never trust, always verify."**

According to the gold standard definition from **NIST Special Publication 800-207**, Zero Trust isn't a single product, but a set of guiding principles designed to prevent unauthorized access. It assumes that threats already exist both outside and *inside* the network. Every single request to access data or a system must be authenticated and authorized every time.

The Key Principles

To understand how Zero Trust works, think of it through these three core pillars:

1. **Continuous Verification:** Just because you logged in once doesn't mean you remain trusted. The system constantly checks who you are, what device you are using, and if your behavior looks normal.
2. **Least Privilege:** Users are only given the bare minimum access they need to do their jobs. A baggage handler doesn't need access to the flight navigation systems, and a gate agent doesn't need access to payroll.
3. **Assume Breach:** We design the system as if a hacker is already in the building. By segmenting data into small "rooms" with locked doors, we ensure that if a hacker gets into one room, they can't easily move to the next.

Zero Trust in the Aviation Industry

The aviation ecosystem is incredibly complex, involving airlines, airports, ground handling, and government agencies. This makes it a prime candidate for Zero Trust.

- **Supply Chain & Third Parties:** Airports rely on hundreds of external vendors (catering, fueling, cleaning). Zero Trust ensures that a vendor's compromised laptop cannot gain access to the airport's critical flight-scheduling systems.
- **Remote Maintenance & IoT:** Modern aircraft are "flying data centers." Mechanics often use tablets to run diagnostics on engines. Zero Trust ensures that the connection between the tablet and the aircraft is encrypted and verified, preventing "man-in-the-middle" attacks that could alter flight data and making sure that maintenance accesses are restricted to authorized personnel.

- **Passenger Privacy:** With the rise of biometric boarding (facial recognition), airports handle sensitive personal data. Zero Trust protects this by ensuring that only specific, authorized applications can access biometric databases, and only for the few seconds required to verify a traveller.

Civil Aviation Ecosystem Principles

There are many different types of interactions involving an aircraft: first, with humans on board, second with ground-based IT entities. Each of whom enjoys a different level of trust and corresponding access rights, as illustrated in the figure and the table below:

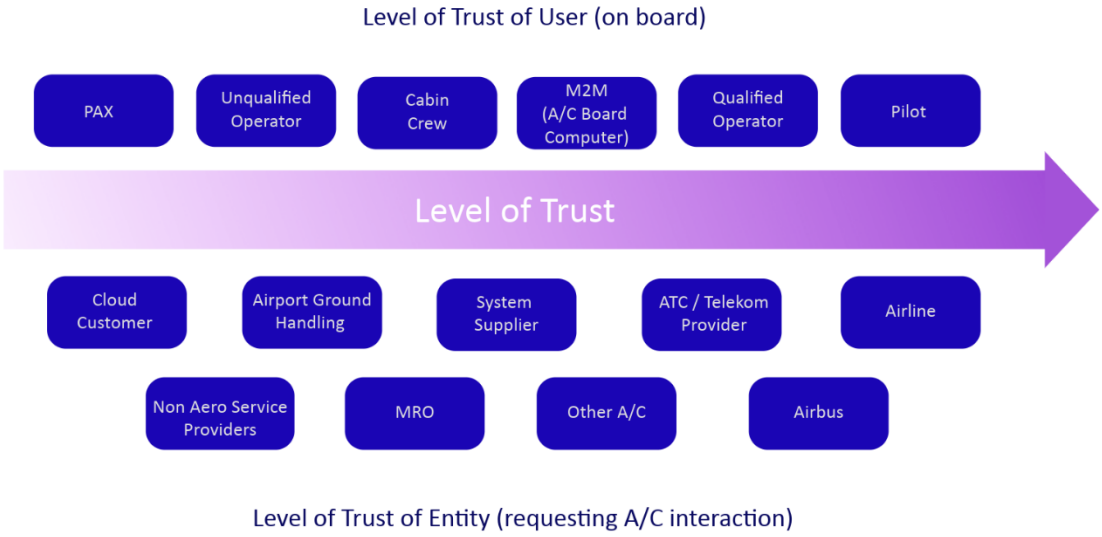

Figure 1 Level of Trust

The entire use case relies primarily on strong identities for the actors. Airlines own their aircraft, employ flight crews, and establish operational flight plans, allocating pilots to missions and coordinating schedules with airports via IATA (International Air Transport Association) timetables. Considering these responsibilities –namely, the assignment of pilots, the maintenance of trust with airports/IATA, and the ownership of aircraft – the airline functions as an "issuer" within the Self-Sovereign Identity (SSI) framework, suggesting the applicability of Verifiable Credentials (VCs).

Dealing with Identities

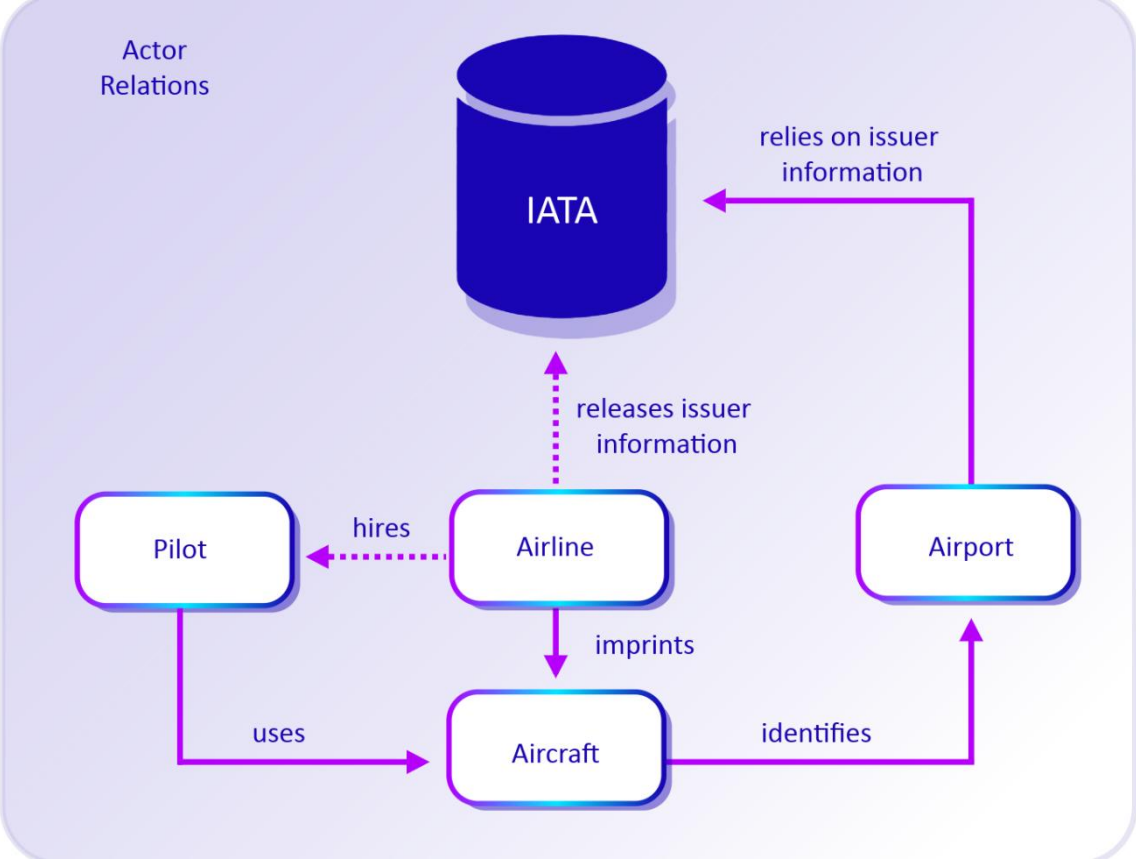

Figure 2 Actor Relations

To implement the relationship described in the picture above, the following is required:

1. Creation of a global registry to store issuer information, including aircraft identities.
2. The airline and the pilot have an internal trust relationship. This means it can be granted or revoked at any time.
3. Four identities are required: an airline issuer identity, a pilot holder identity, an aircraft holder identity, and, optionally, an airport verifier identity.

The aircraft must be capable of actively monitoring the pilot's status. Furthermore, a solution is needed for transferring control when the aircraft itself is operating as an independent identity.

Identity Management

To enable this use case, the preceding chapter mandates the provision of four distinct identities. It is important to note that each of these identities has unique requirements.

Aircraft Identity

The aircraft's identity is established through its onboard computer capabilities. This identity has two parts: a static network component, represented by an imprinted X.509 certificate, and a dynamic component, which is an EdDSA key pair generated by the onboard computer itself. The binding between these two identity components is established through a certificate chain or attestation mechanism: the onboard computer signs its dynamically generated EdDSA public key with the private key associated with the X.509 certificate, creating a verifiable link between the static network identity and the dynamic credential identity. This ensures that a valid X.509 certificate cannot be combined with an unauthorized EdDSA key pair.

The binding is required for:

1. Establishing well-defined network connections via TLS (Transport Layer Security) for ground operations, maintenance, data exchange, and others.
2. Securing data transmission.
3. Authorizing Onboard Computer transactions (e.g., pilot login).

Airport/Airline Identity

The airport and airline identities can be fully fledged wallet identities for organizations, which are able to issue, revoke, and present credentials. These identities shall be controlled just over a limited number of staff, because they are important for connections to trust registries and the authorization of personnel (e.g., for maintenance). The usage of the identities can be standard wallets. Thus, a cockpit can verify via a live connection if a maintenance worker is allowed to take actions on the aircraft, such as updating key lists.

Crew Identities

The crew identity is a human-controlled identity. This means it must be possible to use a human-controlled device with strong personal binding. Ideally, it is issued and managed by the airline itself. The managed device contains a wallet software which manages the keys for door access, aircraft functionality, login, etc.

Dealing with Policies

Each Zero Trust Architecture depends on a set of policies that frame the decision-making within the verification process. The word policy is quite overloaded in this context (the term "policy" is often used for technical policies), so a differentiation must be made between:

- Legal policies
- Business policies
- Technical policies

Legal Policies

These policies are, in most cases, custom or legal Trust Framework (e.g., eIDAS), which define a legal boundary for signature/credential verification on a government level. This can be also defined for an entire sector, such as the aviation sector, where the EASA/IATA defines a framework for the civil aviation. Legal policies can sometimes be enforced technically, but in most cases they are enforced on organization and auditing level by each member of the sector to fulfill the legal boundaries.

Business Policies

Business policies are policies which are enforced by the sector participants. This can be, for example, Business-to-Business requirements such as payment conditions, onboarding criteria, rules of business, certifications, etc. These kinds of business policies are partially technically enforceable – for example, through 9114 implementations or automatic contracting – but in most cases this kind of policies are enforced through business processes, such as ordering processes or contractual negotiations.

Technical Policies

Technical policies are the policies which are enforceable by concrete technologies such as OPA and can be modified directly within a code editor. For the following chapters, technical policies are the main focus. To implement technical policies, the XACML (eXtensible Access Control Markup Language) architecture can be used with regards to **RFC 2904, which** established the "AAA" (Authentication, Authorization, and Accounting) framework. It was one of the first documents to formally argue that security should be **decoupled**. Instead of every application having its own hard-coded rules, the logic (PDP) should be separate from the application (PEP). This allows an

organization to change a policy in central domains (the PAP) and have it applied instantly across every application in the domain network.

The XACML Architecture Flow

The pattern follows a logical sequence where a user's request is intercepted, evaluated against business rules, and then either permitted or denied.

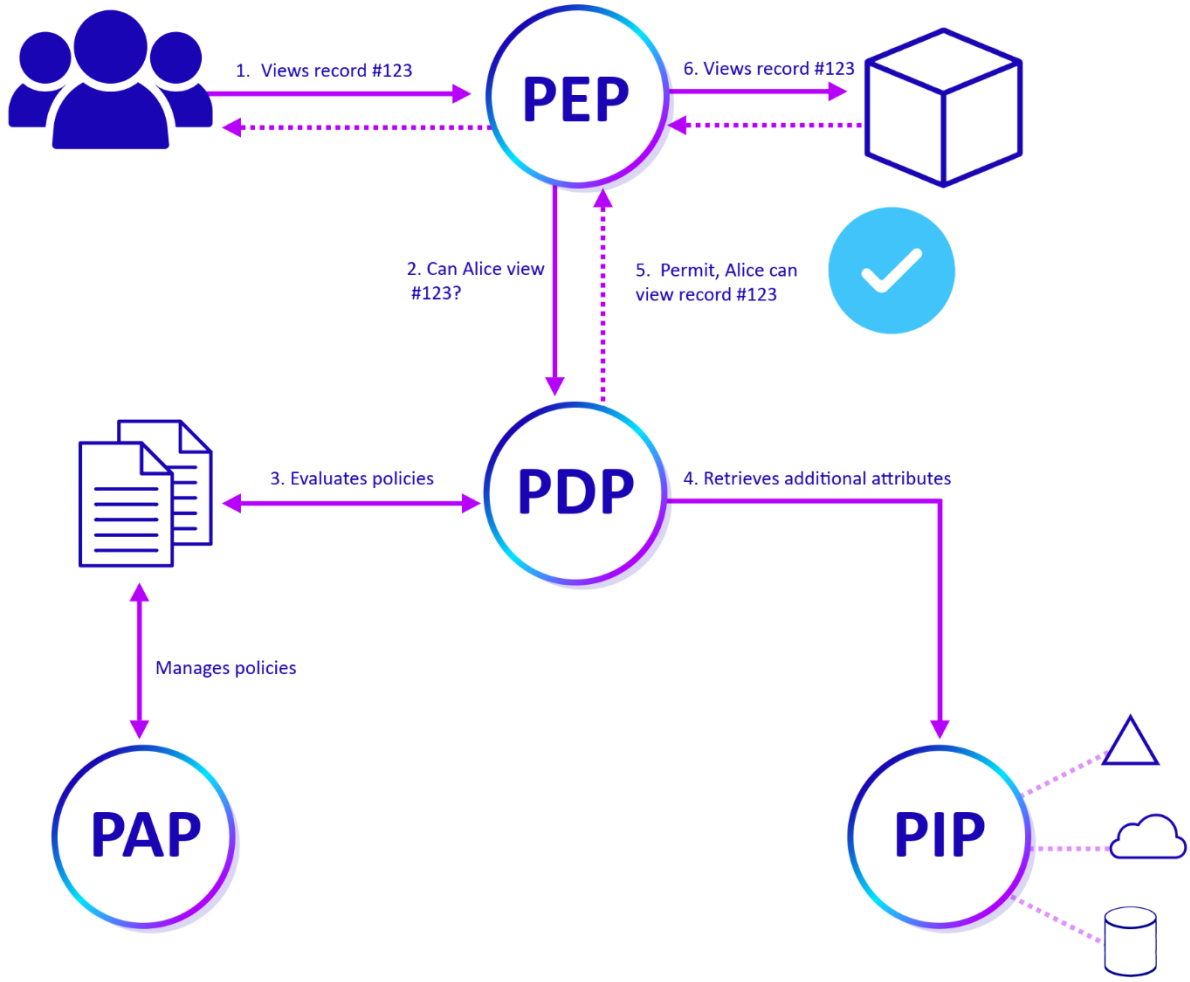

Figure 3 XACML Architecture Flow

Table 1 XACML Architecture Flow Components

Component	Full Name	Role according to RFC 2904 / XACML
PEP	**Policy Enforcement Point**	The "guard" at the gate. It intercepts a user's request, sends it to the PDP for a decision, and then acts on that decision (e.g., opens the door or stays locked).
PDP	**Policy Decision Point**	The "brain" of the system. It receives the request from the PEP, retrieves policies, and evaluates them to return a **Permit** or **Deny** response.
PIP	**Policy Information Point**	The "source of truth." It provides extra information (attributes) that the PDP might need to make a decision, such as a user's current location or their job title from a database.
PAP	**Policy Administration Point**	The "editor." This is where administrators write, manage, and update the security policies (the rules of the game).
PRP	**Policy Retrieval Point**	The "library." It is a storage location (such as a database or file system) where the PAP stores policies and where the PDP goes to fetch them.

How it Works (Step-by-Step)

1. **Request:** A user attempts to access a resource (e.g., "Open Document A").
2. **Intercept:** The **PEP** stops the request and converts it into a standardized XACML request.
3. **Consult:** The **PEP** sends this request to the **PDP**.
4. **Evaluate:** The **PDP** fetches the relevant rules from the **PRP**. If it needs more information (e.g., "Is the user currently on shift?"), it queries the **PIP**.

5. **Decision:** The **PDP** combines the rules and attributes to reach a decision (Permit, Deny, NotApplicable, or Indeterminate).
6. **Enforce:** The **PDP** sends the decision back to the **PEP**, which then allows or blocks the user's access.

When moving it to a more federated (decentralized) concept, the architecture changes slightly, as new "PxP" are now required and introduced:

Policy Verification Point: The policy verification point checks on the client side whether a policy is allowed to be executed, either by checking hashes of the policies itself, searching for policies, and providing information to requested polices – for instance, when the Service Provider offers a checkup according to policy. Overall, the task of this point is to provide policy-related information to the client for interaction.

Policy Trust Client: The policy trust client provides capabilities to the client, for talking authorization flows, managing attestations, or generating and managing key materials. The trust client is mostly called a "Wallet."

Policy Authorization Point: The policy authorization point provides authorizations to the clients, if required. This can be either Access Tokens, Credentials, or both.

Policy Trust Point: The policy trust point has the capability to provide trusted information like registered key materials. It's also the point for registering information or updating them. Depending on the use case, this can be either JWKS files or TRAIN components, etc. This point can also be distributed, depending on the technology – for instance, a distributed ledger steward by using technologies such as Hyperledger-based networks or other decentralized trust registries.

The other points are just modified for decentralized usage, especially policy retrieval point and policy administration point, because they must be enabled to retrieve the information from the decentralized data registry or writing to it.

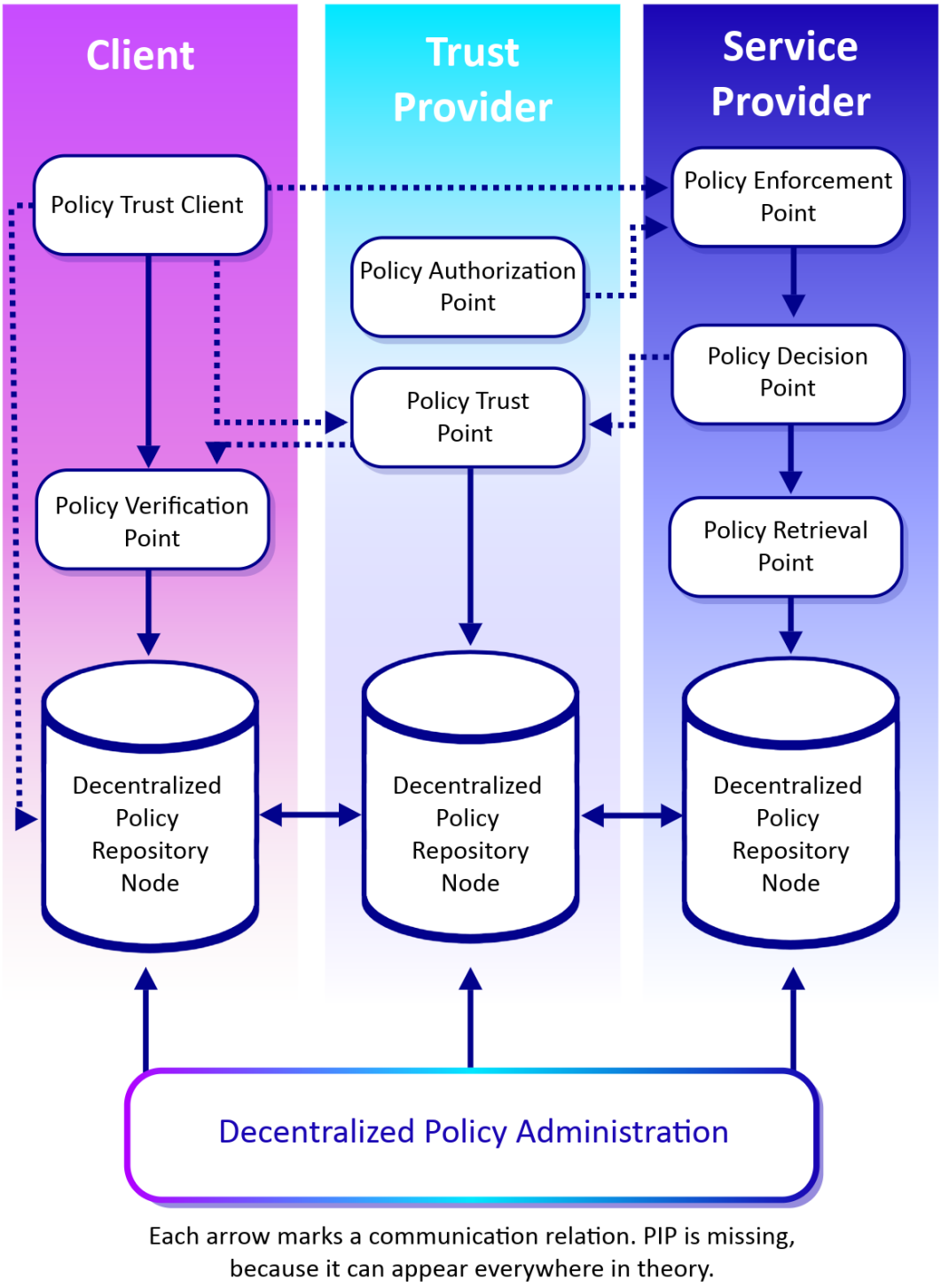

Figure 4 Federated Policy Architecture with Decentralized Repository Nodes

Zero Trust in the Context of Self-Sovereign Identity (SSI)

As mentioned before, with the rise of Web 3.0, decentralized approaches are increasingly required for various use cases. In this context, SSI (Self-Sovereign Identity) is particularly popular. But how does Zero Trust fit into the SSI Triangle of Trust when we actually talk about "Zero Trust"? There is a simple answer: Zero Trust principles are just acting on the verifier side. The holder and the issuer side may also use Zero Trust approaches, but it depends on the use case. On the verifier side, there is actually no other choice but to decide on presentations. And depending on the scenario, this decision should ideally be made during each transaction. Sounds familiar? It is. The basic Zero Trust Architecture described above serves as a blueprint for an SSI verifier system that supports various scenarios of the SSI world.

Federation Building with Zero Trust and SSI

With the knowledge about the basic ZTA and its main use case "bringing anonymous users together," some interesting use cases are emerging which were already explored by projects like XFSC to build federated digital ecosystems. The FACIS project enhances the functional components to use generic Blueprints for Federations – so-called Federation Architecture Patterns (FAP).

Within this project some federation scenarios were explored, for instance the "community federation," which does not rely on restrictions within the business domain. In this context, business domain means that the domain of the business as such is not limited to one single company or legal jurisdiction. Or in more simpler words: a group of participants agree on a set of governance rules for data exchange without the need of being a member of any legal construct. It's a group of loosely coupled actors which interact with each other. Actually, they just need a set of governance rules and similar technology to trust each other, which is a good fit for Zero Trust combined with SSI concepts and the triangle of trust.

An overview of a community federation can be seen in this picture:

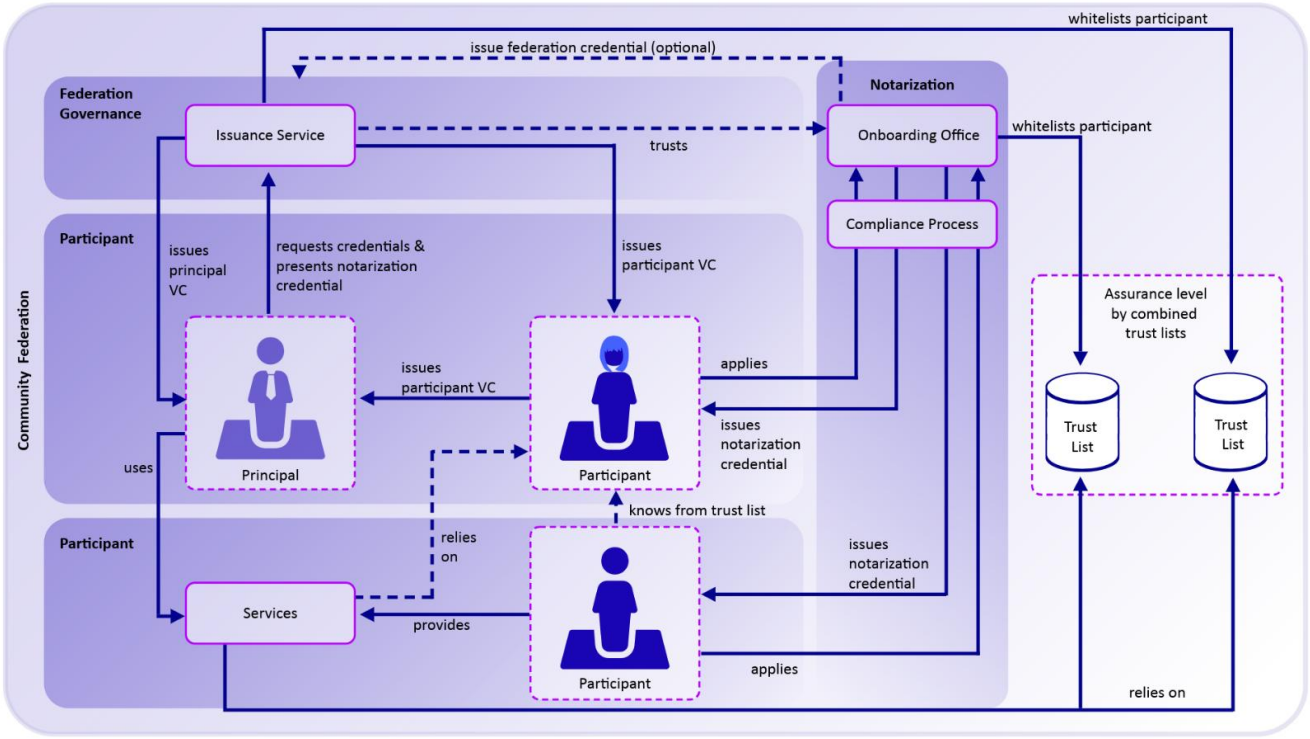

Figure 5 Community Federation

As shown in the picture, the community federation works mostly over a couple of issued VCs and a defined process of establishing trust in them. The "onboarding office" can be everything – for instance, a lawyer, a regulator, or any kind of standardization board. All VCs are later verified by the services via one or more trust lists. The basic Zero Trust architecture fits well into that process because it can be easily extended for a VC setup that grants actors more permissions.

Integration of VCs into Basic Architecture

Depending on the basic architecture, integrating VCs can be adopted quite easily in the OAuth 2.0 / OpenID Connect authorization flow (e.g., using Ory Hydra), but this must be separated in user-based flows and Machine-to-Machine (M2M) flows (client assertion, client credentials).

User-based Flow

The user-based flow provides, via Hydra, a consent screen through a consent provider, which can be used to request more permissions via VCs from the user. The token is directly enriched with permissions if required, after the user's consent.

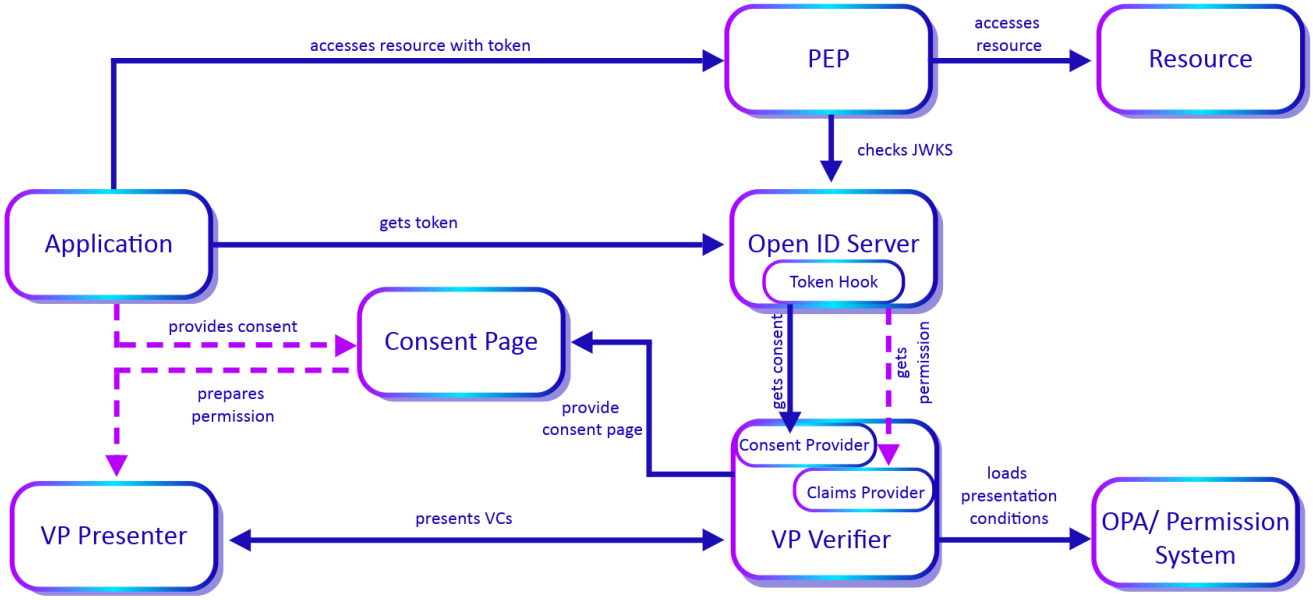

Figure 6 User-based Flow

M2M Flow

The M2M flow can use a VC presentation directly before getting a token from Hydra. If other permissions are required, the VP presentation must be triggered again. How it is retriggered depends on the application or the ecosystem. For example, a DIDComm protocol could be used, which provides a special message format. It should be noted that in aviation environments with limited or intermittent connectivity, the M2M flow must support offline or cached credential verification. Pre-provisioned credential sets or local verification caches may be required for degraded-mode operation.

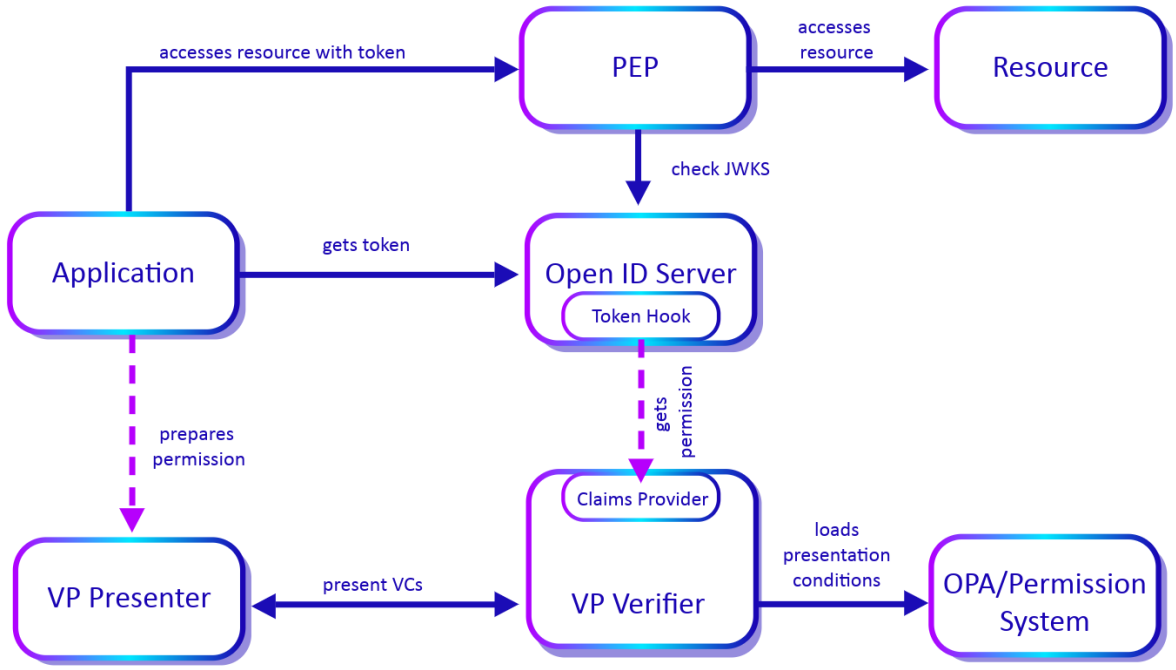

Figure 7 M2M Flow

Scenarios for Identity, ZTNA, and SSI Usage

Basic Aviation Knowledge

The interaction between aircraft and ground depends on the aircraft's operational status. We have two main scenarios, depending whether the engines are "on" or "off." The following pictures details the main aircraft state depending on this trigger.

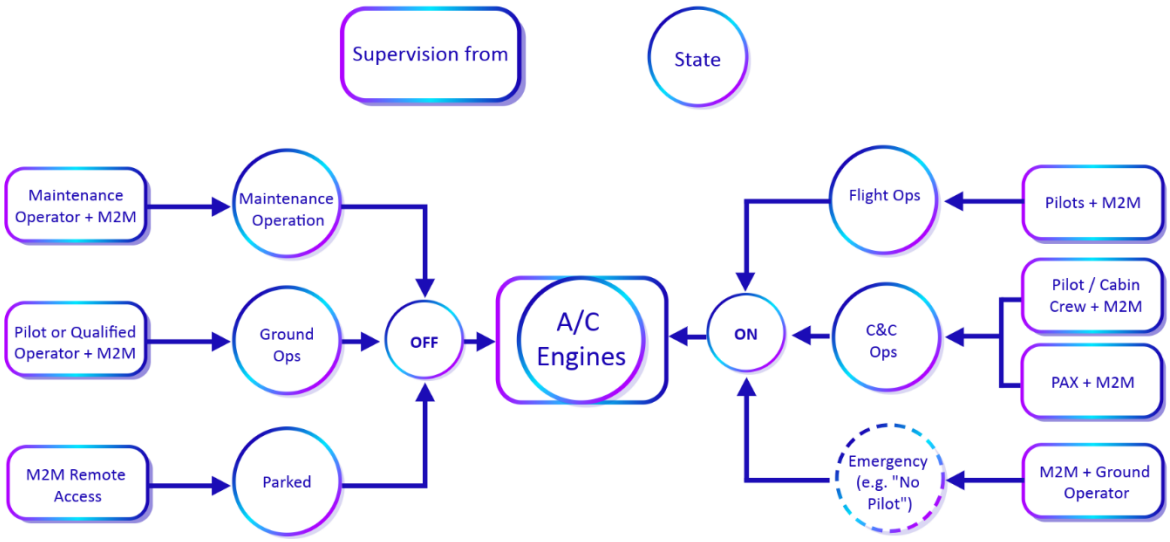

Figure 8 Interaction between aircraft and ground

Both User and Machine-to-Machine (M2M) flows are supported in either mode.

In legacy systems, M2M flows must not alter the aircraft's status, such as activating new software images. Such critical operations are restricted to periods when the engines are off and must be overseen by a qualified operator. M2M interactions are strictly controlled and subject to stringent qualification tests.

While M2M-driven operations, such as auto-pilot, significantly enhance operational efficiency, they are always supervised by a "human in the loop" (a pilot or certified qualified operator). External devices are currently not permitted to initiate such operations during flight. However, on the ground, with the engines off, remote maintenance activities are permissible.

A critical design consideration for any Zero Trust implementation in aviation is resilience under limited or no connectivity. During flight, at remote airports, or during network outages, real-time access to trust registries, PIPs, or credential verification services may be unavailable. The architecture must support graceful degradation, including pre-cached trust lists, locally stored and pre-validated credentials, and fallback policies that maintain security while ensuring

operational continuity. As a baseline, the following principles apply: (1) Safety-critical operations (e.g., engine start authorization) must fail-safe – if credential verification is unavailable, the operation is denied. (2) Security-relevant but non-safety-critical operations (e.g., cabin crew door access) may fall back to locally cached credentials with a defined maximum cache validity period. (3) Monitoring and logging operations may continue without real-time verification but must be reconciled once connectivity is restored. The detailed fallback decision trees, cache invalidation policies, and failure scenario handling will be described in future technical specifications.

Cockpit Onboarding and Door Access (Scenario 1)

Building upon the preceding theoretical chapters, we now examine a practical, real-world aviation scenario. Current aviation processes are heavily reliant on a foundation of trust built on standardized procedures, particularly concerning who is authorized to operate an aircraft. The simple answer is a pilot, but the selection process is more complex.

Existing processes are designed to maintain a robust "chain of trust." This includes security onboardings for crew members to verify their integrity, authentication against airline staff to receive flight plans, and the crew taking control of the aircraft from ground operators. While these well-defined procedures aim to prevent imposters, past events demonstrate that they are not entirely foolproof.

Specific vulnerabilities can exist:

- **License Verification:** Aviation authority licenses (e.g., EASA or FAA) are not continuously checked for validity.
- **Aircraft Type Authorization:** The pilot's authorization to fly that specific aircraft type may not be 100% verified (in an academic sense).
- **Cockpit Access:** Control over who has access to the cockpit remains a concern.
- **Unattended Aircraft:** The chain of trust becomes sensitive to security issues even when an aircraft has been parked overnight without staff present.

In short, there is significant potential for security improvement within this sensitive process. The most promising solution lies in leveraging **Self-Sovereign Identity (SSI)** capabilities. SSI is ideally suited for decentralized decision-making problems, such as bringing "unknown" staff together with "aircrafts" for authentication and authorization of actions.

This technology is particularly critical for securing highly sensitive areas such as cockpit door access and the onboard computer. To introduce SSI into this scenario, we propose utilizing standard protocols such as **OID4VCI**, **OID4VP**, and **TOTP**.

1. To enter an aircraft, all personnel – especially crew and operator staff – should be equipped with a device featuring an OID4VCI wallet implementation.
2. Access to the onboard computer should necessitate a successful authentication via OID4VP.
3. Door access should be secured using cryptographically protected "time-based one-time passwords" (TOTP), which refreshes at a configurable interval (e.g., every 20 seconds, reduced from the RFC 6238 default of 30 seconds to limit replay windows, noting that this requires tighter time synchronization between devices).

Assuming that everyone has a valid OID4VCI/VP app with a TOTP authenticator, the scenario can be improved as following:

1. Pilots can authenticate within an airline website, where they can request their access credentials for the certain flight. The airline then issues a verifiable credential for the onboard computer access and a TOTP QR code for the door authenticator. The pilot loads this into their mobile app via QR code scan or NFC.
2. After issuance, the airline data center unlocks the pilot's public keys for the flight.
3. The ground operator uses an airline website to obtain a set of access credentials for the aircraft, but with a limited number of rights (e.g., no engine start).
4. Before the crew is entering the aircraft, the ground operator activates with his/her credentials the board computer by scanning a QR code on the screen. The operator then updates the set of public keys within the onboard computer. This stage is a new "imprinting" of the onboard computers key storage to activate the new keys of the crew. After that procedure, the onboard computer falls back in the onboarding screen with the QR code, and the cockpit door can theoretically be closed and activated, just to be opened with the crew's new TOTP codes (which is not the case in the current process).
5. When the crew enters the cockpit, the pilots will activate the cockpit by scanning the QR code on the screen.

6. During the flight, the pilots can continuously generate new door access codes to enter the cockpit. These codes are valid for one TOTP interval only (e.g., 20 seconds as configured above), preventing replay.

The entire process looks schematically as follows:

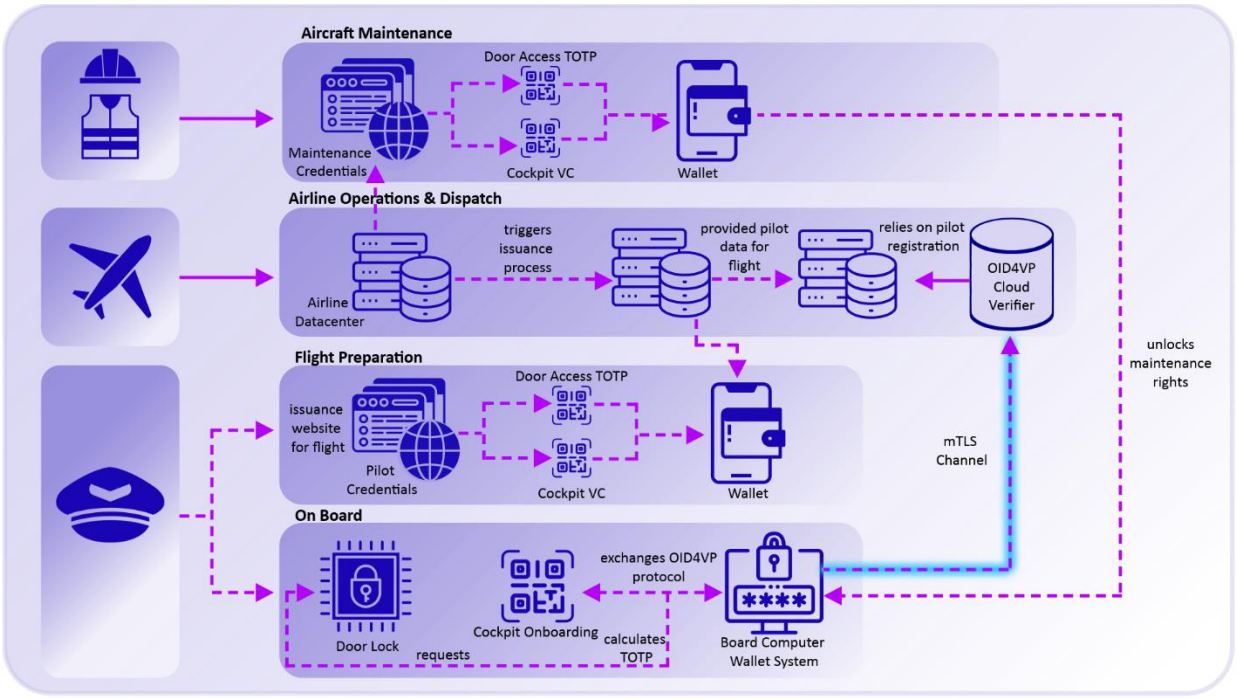

Figure 9 Cockpit Onboarding and Door Access (Scenario 1)

The TOTP procedure works as well without Smartphone – example, via TOTP card, token, or any other hardware device which implements the RFC 6238 standard.

Ground Data Exchange & Maintenance Operations (Scenario 2)

For the second scenario, the focus shall be more on Zero Trust-based mechanisms between machines and data exchange, because an aircraft in operation is producing a lot of data:

- Application or security logs
- Performance data
- Airline or model specific data

These data are the property of the airline and need to be collected for security and operational purposes (reference FAA AC-119).

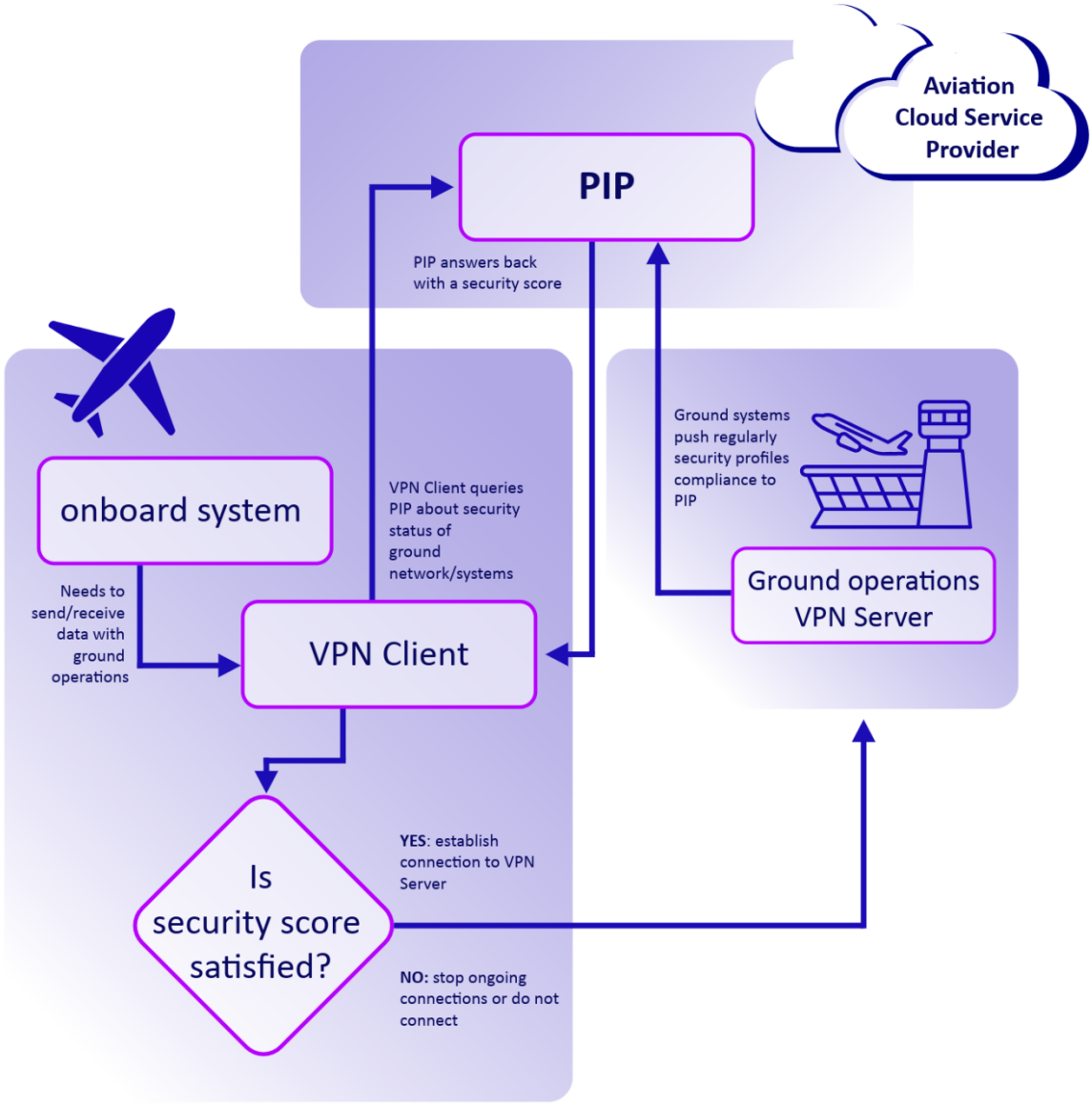

Figure 10 Ground Data Exchange & Maintenance Operations (Scenario 2)

The ground segment must maintain an adequate security level to prevent threat propagation to the onboard ZTA components. Some examples of the mechanisms used to verify the integrity of data exchanges on the aircraft side are detailed in the following section.

A key consideration in this architecture is the integrity of the Policy Information Point (PIP) data itself. Since ground systems push their security profiles to the PIP, a compromised ground segment could potentially supply falsified compliance data. To mitigate this, security profiles should be cryptographically signed by the ground system's hardware trust anchor and independently validated by the PIP before being accepted.

Another type of data exchange operation is represented by maintenance operations performed from an airline ground operational segment (which could be subcontracted to a third party), where the operator either connects to an onboard maintenance interface or sends configuration data to the aircraft.

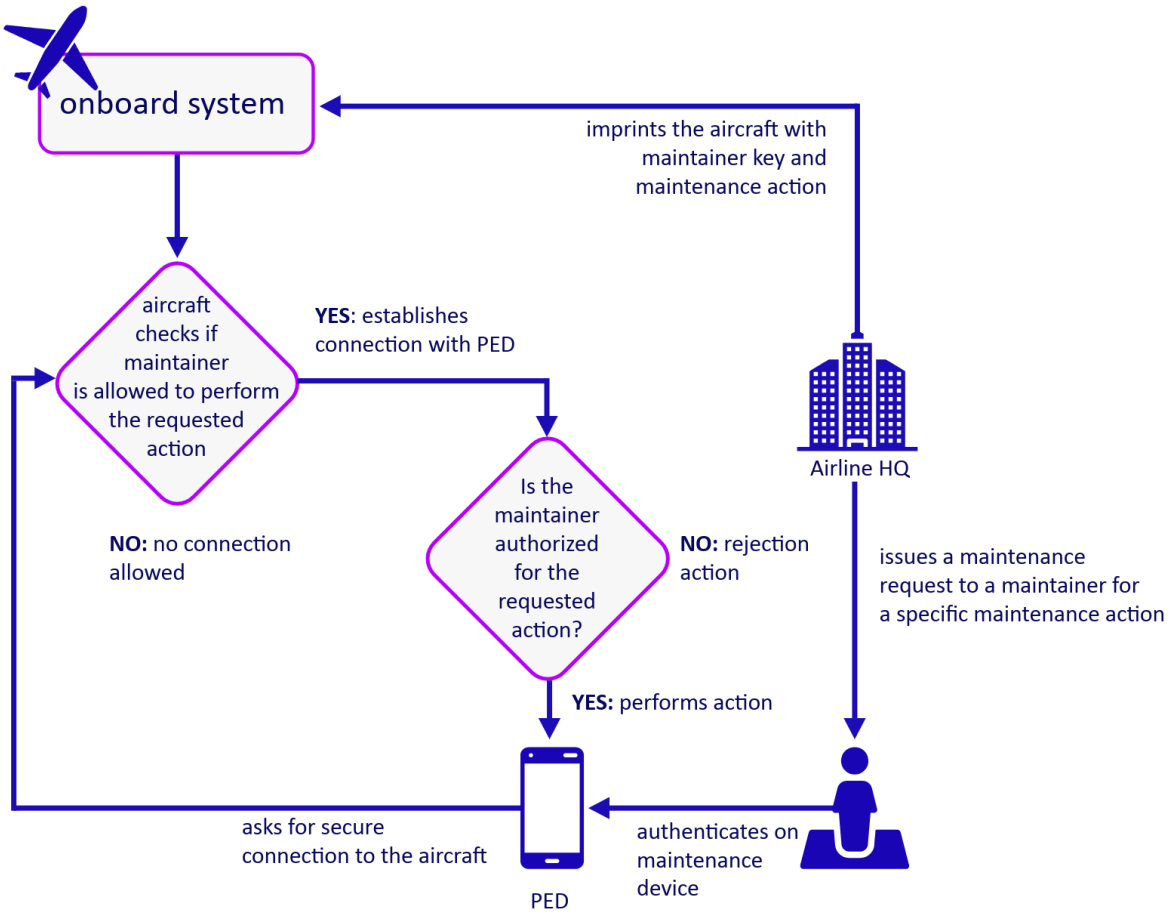

Figure 11 Maintenance Operations Authorization Flow

Today, both data exchanges happen via secure communication tunnels that ensure mutual authentication as well as the integrity/confidentiality protection of the exchanged data. The tunnels are established by the aircraft toward the ground over public radio networks (LTE/5G/...) and always stay on.

Operational data download represent a machine-to-machine flow that runs continuously (when connectivity is available), while maintenance operations are a human to machine operation where the human could be an airline employee or a subcontracted third-party airport maintenance shared operator.

The challenge: as ground operational segments security may degrade over time, we want to improve the situation leveraging the following capabilities that Zero Trust can easily enable:

- Check and validate the security profiles provided by the ground endpoint and ground target network services before establishing the VPN connection.
- Check and validate the security properties of the ground operator's machine or ground segment before providing access to critical maintenance operation, and limit access to non-critical ones if security prerequisites are not met.
- React to a security event detecting the loss of one or some of the expected security properties (e.g., obsolete OS version, unpatched software, security event detected, etc.) and either:
    - close the established connection
    - limit the type of data being exchanged or the available maintenance functions selectable by the maintainer
- Handle third-party subcontracted maintainers connecting from non-managed devices (e.g., not enrolled by the airline owning the aircraft) by limiting the access to a specific function on a specific aircraft on a specific time window.

All of the above challenges can be addressed by a Zero Trust Architecture and provisioned/orchestrated by a federated ecosystem like AXIS (see below).

Process Hardening

Aviation Authority Licence Check via IATA

During the issuance of aircraft credentials, the process may check aviation authority licences (EASA, FAA, or equivalent) before issuing any credential for the aircraft. This works only when the IATA provides a kind of digital ID specifically for that purpose, which isn't the case yet. This could improve the process and create a robust identification of pilots and their capabilities, which would simplify the identification and authorization for certain flights.

Shamir's Secret for Flight Security

Following the Germanwings 9525 incident, it is now mandated that a crew member must be present in the cockpit whenever one pilot leaves. With electronically protected TOTP access, this

physical presence rule can be complemented – though not replaced – by introducing threshold cryptography for cockpit door access. Using a scheme such as FROST (RFC 9591), which builds on Shamir's Secret Sharing, a door-unlock signature can be generated cooperatively by a minimum of X crew members without any single participant ever possessing the full signing key.

Connected Aircraft in Federated Ecosystems (AXIS)

In 8ra (IPCEI-CIS), Airbus has investigated the implementation of European GAIA-X federated services [Ref 1 - DASC Paper on "Next aircraft hyperconnected system platform"]. The idea is to associate each stakeholder with a trust anchor that can be verified by a trusted entity. The picture below shows the principle in the context of an aircraft refueling use case.

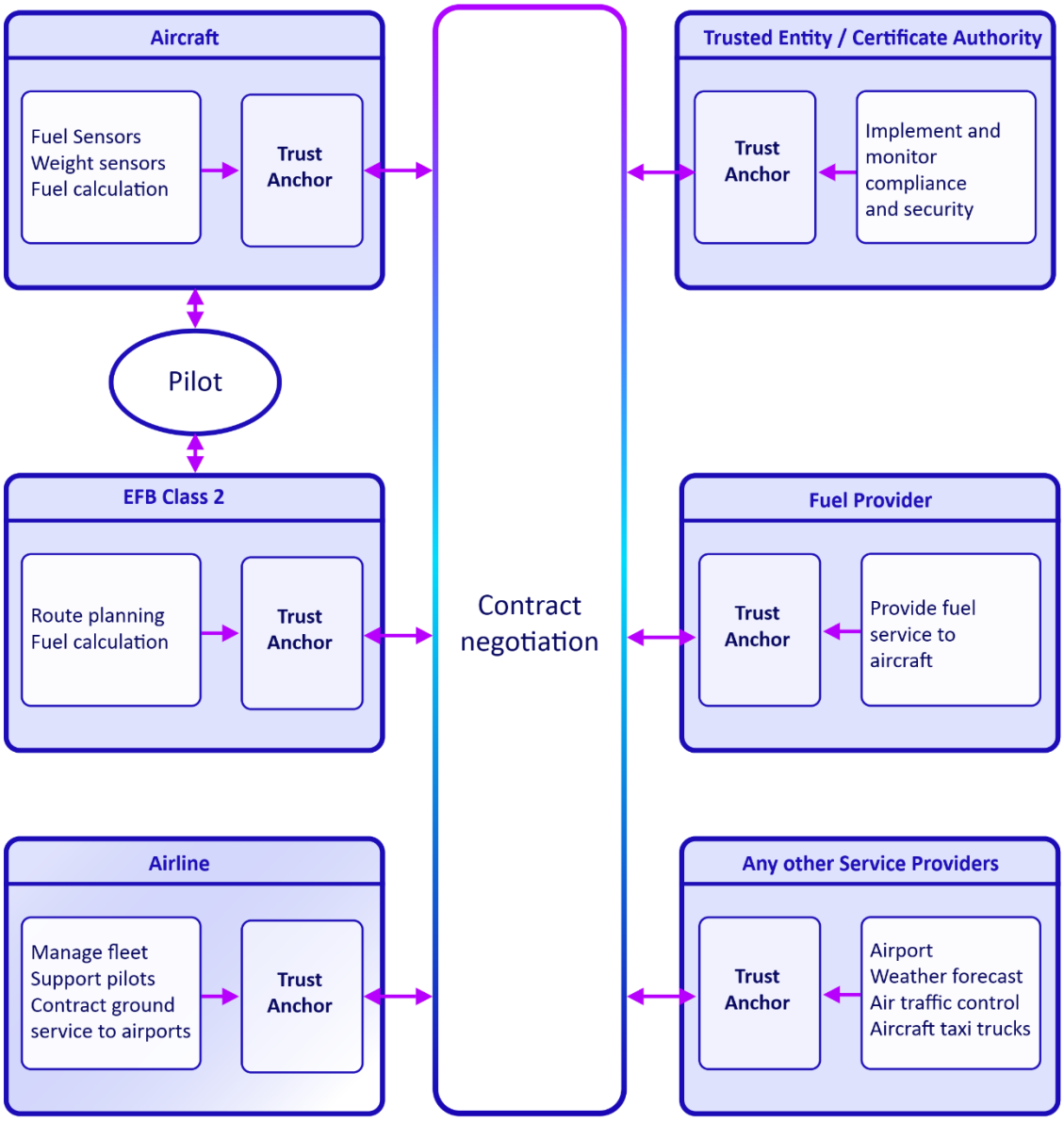

Figure 12 Use case: aircraft refueling

The advantage of trust frameworks (e.g., Gaia-X Compliance) is to decouple each stakeholders and provide standards to inboard each of them in the aviation ecosystem without requiring complex point-to-point contract negotiations.

For our ZTNA approach, the idea is to ensure that proper SLA is implemented for each use cases without introducing aircraft safety risks once in operation (e.g., no disruption of pilot access to aircraft systems as soon as engines are on). In other words, the user authentication/authorization mechanism should have the same level of safety qualification that the aircraft system that this user will interact with.

Sampling the Aviation Ecosystem (AXIS, FACIS)

To demonstrate the core concepts of federated aviation ecosystems and their technical implementation with Zero Trust capabilities, 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.

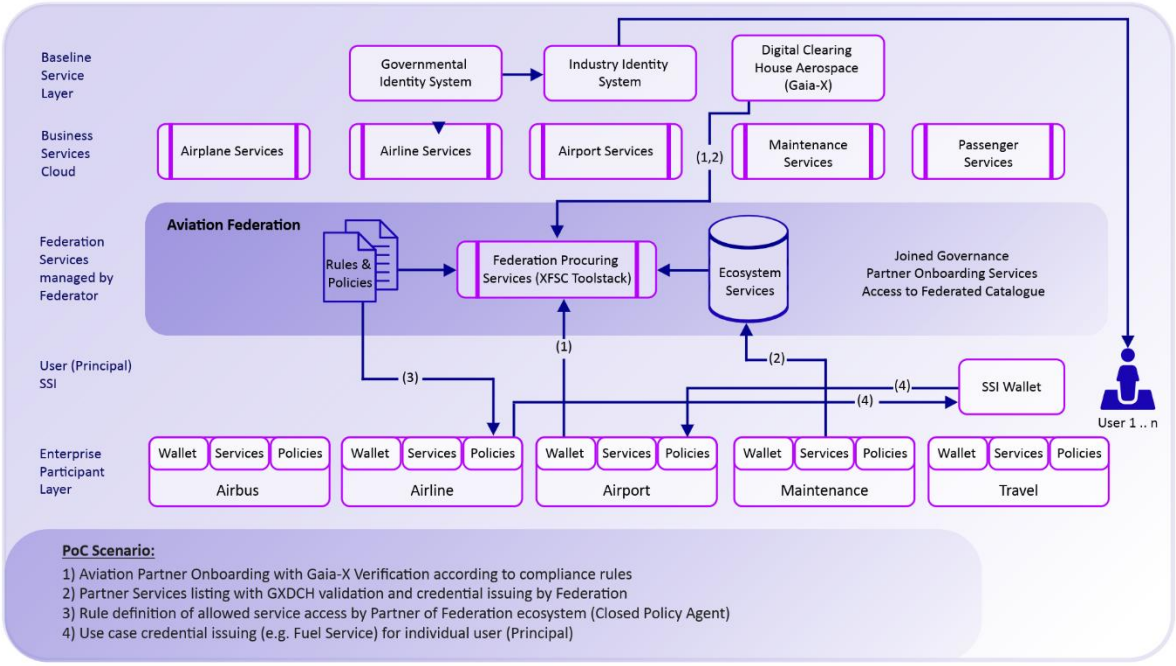

Figure 13 PoC Scenario: Federated Aviation ecosystems

This illustration outlines the essential components required for secure, federated service orchestration, based on Self-Sovereign Identities (SSI), Verifiable Credentials (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.

With SSI, users control and own their digital identities and other verifiable digital credentials locally. It is not required to use a predominant cloud service provider, nor is the establishment of a central Gaia-X identity provider necessary. Users are thus completely independent of third parties and decide themselves which identity data they share and with whom, as all identity data is securely stored only with the individual user in their SSI wallet. As a result, a trustworthy and straightforward peer-to-peer exchange between users and applications do not require a mediator.

There is a set of key functions to operate such concepts, which are exposed as Open Source Software under the governance of Eclipse Foundation.

Eclipse XFSC (Cross Federation Services Components)[1] develops the software components necessary to set up a federated system that interconnects several participants within a data and service infrastructure, aiming to develop new data-driven services and innovative products. Such ecosystems consist of joined interconnected data and infrastructure ecosystems, aggregated into so-called Federations that are individually orchestrated and operated with the help of Federation Services as part of Gaia-X.

It consists of several components (mainly microservices), enabling federations in data ecosystems and providing interoperability across federations.

The Eclipse XFSC Toolbox provides a set of services for the functional implementation of Self-Sovereign Identities, W3C credential management, Trust Services, database functions for knowledge graphs, usage policy negotiation, and a core Low-Code Engine. The main purpose is the operational uptake of federations as decentralized ecosystems.

Eclipse XFSC is to be seen as the implementation of a suite of solutions – providing for the minimum technical requirements to empower Federations to become operational and to allow organizations to participate in a world of Self-Sovereign Identity and data ecosystems. The Eclipse XFSC toolbox defines a range of components necessary to fulfil the Gaia-X's objective of building trust and interoperability, while ensuring that participants retain sovereignty over their data.

Concretely, the first set of services delivered are:

ICAM & Trust Over IP (Identity, Credential and Access Management)

These services empower federated ecosystems, like federations, to authenticate and authorize users and systems in a decentralized, self-sovereign manner, ensuring trust without depending on a central authority. These services utilize credential validation and technology functionalities to maintain a consistent level of trust among all participants in the federation.

- **Authentication/Authorization Service (AAS)**
- **Organization Credential Manager (OCM)**

[1] https://projects.eclipse.org/projects/technology.xfsc

- **Personal Credential Manager (PCM)**
- **Trust Services API (TSA)**
- **Notarization Service (NOT)**
- **Trust Management Infrastructure (TRAIN)**

Decentralized Catalogue and Contracting Service (CCS)

The Decentralized Catalogue and Contracting Service act as an inventory, allowing participants to discover, understand, and use available data in an ecosystem. The Federated Catalogue serves as a repository for a Federation, enabling participants to find each other's information and services through self-descriptions. The toolbox provides code for groups to create their own catalogue and includes services for contract negotiation and tracking data transactions, giving participants control over their data usage within the group.

- **Federated Catalogue (CAT)**
- **Self-Description Wizard (SD-Wizard)**
- **Data Contract Services (DCS)**
- **Data Exchange Logging Service (DELS)**

Orchestration & Monitoring

Orchestration and monitoring services are vital for managing complex ecosystems, ensuring seamless and compliant operations. The distinction lies in their focus and functionality: ORCE (Orchestration Engine) is tailored for orchestrating tasks, particularly emphasizing complex application networking, while ORC (Orchestration) is specifically designed for managing the lifecycle of infrastructure services, including deployment, updates, and deletion based on actions from consumers or providers. Additionally, the Continuous Automated Monitoring (CAM) service ensures transparency through automated compliance monitoring, providing insights into service adherence to federation rules.

- **Orchestration Engine (ORCE)**
- **Service Mesh Orchestrator, (ORC)**
- **Continuous Automated Monitoring (CAM)**

Conclusions

The digital transformation of the aviation industry is unlocking new levels of connectivity, but it also needs to be aligned with existing security models. Moving from a perimeter approach to a "never trust, always verify" model provides a more flexible option to deal with a multitude of ecosystem partners.

Zero Trust Architecture (ZTA) ensures that security is not a one-time check (e.g., at the front door), but a continuous verification process for every interaction.

Key Strategic Advantages

Transitioning to this identity-centric model offers three distinct competitive advantages for our ecosystem:

- **Hardened Security & Compliance:** By integrating **Self-Sovereign Identity (SSI)** and **Verifiable Credentials (VCs)**, we move away from vulnerable centralized passwords to decentralized, tamper-proof digital IDs. This allows for fine-grained access control – ensuring that the right person accesses the right data at the right time. Sensitive scenarios like **Cockpit Access** and **Maintenance Operations** become significantly more secure, reducing the risk of unauthorized entry or data breaches.
- **Operational Agility & Resilience:** Traditional security often bottlenecks operations. A Zero Trust model is dynamic. It automates access decisions based on real-time context (e.g., user location, device health) rather than static rules. This reduces the manual burden on security operations teams, allowing for smoother, faster, and more resilient workflows even in a hyper-connected environment.
- **Trusted Ecosystem Collaboration:** Adopting a federated model (such as **AXIS**, guided by **Gaia-X** principles) provides a standardized governance framework. This enables secure, auditable data sharing across the entire supply chain and partner network. It transforms our security posture from a barrier into an enabler of trusted collaboration.

The Bottom Line

Adopting Zero Trust is not merely a technical upgrade; it is a business imperative. It shifts the industry from static, perimeter-based protection to **dynamic, identity-centric verification**, ensuring that infrastructure is robust enough to handle the demands of the modern, hyper-connected aircraft.

Next Steps and Outlook

To move this paradigm from concept to operational reality, the next critical steps involve the development of a joint demonstrator. This proof-of-concept is necessary to validate the technical interoperability and security claims of the ZTA and SSI integration across multiple stakeholders in the aviation ecosystem. Furthermore, this practical development must be paired with the writing of detailed technical specifications and governance rules. Clear, standardized specifications are essential to ensure scalable implementation, regulatory compliance, and broad industry adoption of the federated architecture.

A significant challenge in this implementation will be addressing the impact on aircraft certification caused by the shift to dynamic configurations and policies inherent in ZTA, which contrasts with the determinism required by safety demonstrations. This challenge can be addressed by formally demonstrating that policies consistently guarantee safety requirements, or by utilizing a restricted set of pre-validated, safety-compliant policies.

As a guiding principle: "Expect the unexpected and, in any case, the final decision must stay under human control."

Glossary

Abbreviation	Full Name	Definition
AAA	Authentication, Authorisation, and Accounting	A security framework that controls access to resources by verifying identity (Authentication), granting appropriate permissions (Authorisation), and tracking usage (Accounting).
AAS	Authentication / Authorisation Service	A dedicated service responsible for verifying the identity of entities and determining their access rights within a Zero Trust or federated architecture.
ABAC	Attribute-Based Access Control	An access control paradigm where authorisation decisions are based on attributes of users, resources, actions, and environment rather than on static roles or identities.
CAM	Continuous Automated Monitoring	An ongoing, automated process for assessing and reporting on the security and compliance posture of systems in real time, enabling rapid detection and response to policy violations.
CAT	Federated Catalogue	A distributed registry within the Gaia-X / XFSC ecosystem that aggregates and publishes self-descriptions of services, data offerings, and participants to support discovery and trust.
CCS	Decentralized Catalogue and Contracting Service	An XFSC component enabling decentralised management of service catalogues and the automated negotiation and execution of data-sharing contracts between participants.
Chain of Trust	Chain of Trust	A hierarchical sequence of cryptographic validations in which each entity's authenticity is verified by a higher-level trusted authority, ultimately anchoring to a root trust anchor. It ensures that trust is traceable and verifiable at every level.

Abbreviation	Full Name	Definition
DCS	Data Contract Services	A service managing the lifecycle of data contracts, including creation, negotiation, storage, and enforcement of terms governing data exchange between parties.
DELS	Data Exchange Logging Service	A service that records and audits data exchange events within a federated ecosystem, providing traceability and accountability for all data transactions.
DIDComm	Decentralized Identifier Communication	A messaging protocol built on top of Decentralized Identifiers (DIDs) enabling secure, private, and interoperable peer-to-peer communication without reliance on a central intermediary.
Digital ID	Digital Identity	A machine-readable representation of an entity (person, device, or organisation) in a digital system, typically expressed through credentials, certificates, or DIDs, enabling verifiable identification without physical documents.
EASA	European Union Aviation Safety Agency	The European regulatory authority responsible for civil aviation safety, including the certification of aircraft, components, and the oversight of aviation organisations and personnel across EU member states.
EdDSA	Edwards-curve Digital Signature Algorithm	A modern public-key digital signature scheme based on twisted Edwards curves (commonly Ed25519), offering high performance, strong security, and compact key and signature sizes.
FAA	Federal Aviation Administration	The national aviation authority of the United States, responsible for regulating and overseeing all aspects of civil aviation, including airworthiness certification, air traffic control, and aviation security standards.
FAA AC-119	FAA Advisory Circular 119	An FAA Advisory Circular providing guidance on the certification and operational requirements for connected

Abbreviation	Full Name	Definition
		aircraft systems, including cybersecurity considerations for avionics and aircraft networks.
FAP	Flight Attendant Panel	An onboard control interface used by cabin crew to manage in-flight systems such as lighting, passenger services, and emergency functions.
Federated Identity	Federated Identity Management	A model in which identity information and trust are shared across multiple organisations or domains, allowing users and systems to authenticate once and be recognised across a federation of participants without requiring centralised identity management.
IATA	International Air Transport Association	A trade association representing the world's airlines, responsible for setting industry standards, facilitating safe and efficient air transport operations, and coordinating regulatory compliance across the global aviation industry.
ICAM	Identity, Credential, and Access Management	A comprehensive framework encompassing the policies, processes, and technologies used to manage digital identities, credentials, and access rights across an organisation or federated ecosystem.
Identity (ZTA)	Identity in Zero Trust Architecture	In a ZTA context, identity is the primary security perimeter. It encompasses not only human users but also devices, services, and workloads, each requiring a cryptographically verifiable identity before any access is granted.
JWKS	JSON Web Key Set	A JSON-formatted file or endpoint that publishes a set of public cryptographic keys, used by relying parties to verify the signatures of JSON Web Tokens (JWTs) issued by an identity or authorisation server.
Low-Code Engine	Core Low-Code Engine	A development platform that enables the rapid creation and deployment of applications and workflows through visual configuration and minimal hand-coding, used to

Abbreviation	Full Name	Definition
		accelerate the implementation of policy, identity, and integration components in complex architectures.
M2M	Machine-to-Machine	Direct automated communication between devices or systems without human intervention, enabling data exchange, remote monitoring, and control in distributed architectures including avionics and ground systems.
Microservices	Microservices Architecture	An architectural style in which an application is composed of small, independently deployable services that communicate over well-defined APIs. In ZTA, each microservice is treated as a separate trust boundary requiring its own authentication and authorisation.
mTLS	Mutual Transport Layer Security	An extension of the TLS protocol in which both the client and server authenticate each other using digital certificates, ensuring bidirectional identity verification and encrypted communication.
NFC	Near Field Communication	A short-range wireless communication technology enabling data exchange between devices within approximately 4 cm, used for contactless authentication, access control, and digital credential presentation.
NOT	Notarization Service	An XFSC trust service that provides tamper-evident timestamping and notarization of digital artefacts, ensuring their integrity and non-repudiation for audit and compliance purposes.
OCM	Organization Credential Manager	A component managing the lifecycle of digital credentials on behalf of an organisation, including issuance, storage, presentation, and revocation within an SSI or federated identity ecosystem.
ODRL	Open Digital Rights Language	A W3C policy expression language used to represent permissions, prohibitions, and obligations over digital assets and data, enabling machine-readable data usage policies in federated and Zero Trust environments.

Abbreviation	Full Name	Definition
OID4VCI	OpenID for Verifiable Credential Issuance	An OpenID-based protocol defining how Verifiable Credentials are issued by an issuer to a holder's digital wallet in a secure and standardised manner.
OID4VP	OpenID for Verifiable Presentations	An OpenID-based protocol defining how holders can present Verifiable Credentials to verifiers in a privacy-preserving and interoperable way.
OPA	Open Policy Agent	An open-source, general-purpose policy engine that enables unified, context-aware policy enforcement across services, APIs, and infrastructure using the declarative Rego policy language. In ZTA, OPA acts as a PDP, evaluating access requests against defined policies at runtime.
ORC	Service Mesh Orchestrator	A component responsible for coordinating and managing communication between microservices within a service mesh, including traffic routing, load balancing, and policy enforcement.
ORCE	Orchestration Engine	A component that automates and coordinates complex multi-step workflows across distributed services, managing sequencing, dependency resolution, and error handling in federated or cloud-native environments.
PAP	Policy Administration Point	The component in an access control architecture responsible for creating, managing, and maintaining security policies.
PCM	Personal Credential Manager	A user-controlled digital wallet or agent that stores, manages, and presents personal Verifiable Credentials, enabling individuals to selectively disclose identity attributes in SSI-based systems.
PDP	Policy Decision Point	The component responsible for evaluating access requests against applicable policies and returning an authorisation decision (permit, deny, or not applicable).

Abbreviation	Full Name	Definition
PEP	Policy Enforcement Point	The component that intercepts access requests, forwards them to the PDP for evaluation, and enforces the resulting decision by granting or denying access to the requested resource.
PIP	Policy Information Point	The component that provides the attributes and contextual data needed by the PDP to evaluate policies (e.g., user attributes, environmental conditions, resource metadata).
PRP	Policy Retrieval Point	The component responsible for retrieving applicable policies and making them available to the PDP for evaluation.
Public Radio Networks	Public Radio Networks (LTE/5G/...)	Commercially operated cellular networks used for broadband wireless communication. In aviation contexts, LTE and 5G networks may serve as communication channels for ground-to-aircraft data exchange, requiring appropriate security controls within a Zero Trust framework.
RFC 6238	RFC 6238 — TOTP Standard	An IETF standard defining the Time-Based One-Time Password (TOTP) algorithm, specifying how a shared secret and the current time are combined to generate short-lived, single-use authentication codes.
SD-WIZARD	Self-Description Wizard	An XFSC tool that guides participants in creating machine-readable self-descriptions of their services and data offerings, compliant with the Gaia-X Trust Framework and verifiable by the federated catalogue.
Shamir Secret Sharing	Shamir's Secret Sharing	A cryptographic algorithm that splits a secret (e.g., a private key) into a defined number of shares, such that only a minimum threshold of shares is required to reconstruct the secret. It enforces multi-party authorisation for sensitive operations but does not address physical presence requirements.

Abbreviation	Full Name	Definition
SSI	Self-Sovereign Identity	An identity model in which individuals or entities fully own and control their digital identities and credentials, independent of any centralised authority or third-party provider, typically implemented using DIDs and Verifiable Credentials.
Steward (Hyperledger Indy)	Hyperledger Indy Steward	A trusted node operator within the Hyperledger Indy permissioned blockchain network, responsible for validating transactions and maintaining the distributed ledger used for decentralised identity and credential verification.
TLS	Transport Layer Security	A cryptographic protocol providing encrypted, authenticated communication over a network. TLS is the foundation of secure data exchange in aviation networks and is extended to mTLS for mutual authentication in Zero Trust environments.
TOTP	Time-based One-Time Password	A form of multi-factor authentication that generates a temporary numeric password based on a shared secret and the current time (per RFC 6238), typically valid for 30 seconds and often presented via QR code for initial enrollment.
TRAIN	Trust Management Infrastructure	An XFSC component providing the mechanisms to establish, publish, and verify trust relationships within a federated ecosystem, enabling participants to discover and validate each other's credentials and trust anchors.
Trust Anchor	Trust Anchor	A trusted and authoritative entity or artefact (such as a root CA certificate, Access Token, Verifiable Credential, or API endpoint) from which trust is derived and propagated through a chain of trust. Trust anchors serve as the foundation for all identity and access verification in a Zero Trust architecture.

Abbreviation	Full Name	Definition
Trust Over IP	Trust Over IP (ToIP)	A Linux Foundation framework defining a dual-stack model for establishing digital trust, combining cryptographic trust at the technical layer with human and legal trust at the governance layer, enabling interoperable and verifiable identity across organisations.
TSA	Trust Services API	An API component exposing trust-related services such as credential verification, status checks, and trust anchor resolution, enabling relying parties to programmatically validate the trustworthiness of identity claims.
VC	Verifiable Credential	A tamper-evident digital credential conforming to the W3C Verifiable Credentials Data Model, containing cryptographically signed claims about a subject issued by a trusted authority, enabling third-party verification without contacting the issuer.
VP	Verifiable Presentation	A packaging of one or more Verifiable Credentials, optionally with additional proofs, presented by a holder to a verifier in a way that can be cryptographically verified, supporting selective disclosure.
W3C	World Wide Web Consortium	The international standards body responsible for developing and maintaining open web standards, including the Verifiable Credentials Data Model, Decentralized Identifiers (DID), ODRL, and other specifications foundational to SSI and Zero Trust architectures.
Web 3.0	Web 3.0 (Decentralized Web)	The next generation of the internet characterised by decentralisation, user ownership of data, and trust established through blockchain and cryptographic mechanisms rather than centralised intermediaries, underpinning SSI and federated identity models.
X.509 Certificate	X.509 Certificate	An ITU-T standard defining the format of public key certificates used in PKI to bind a public key to an entity's

Abbreviation	Full Name	Definition
		identity. In aviation ZTA, X.509 certificates serve as the static network identity component for onboard systems and ground infrastructure.
XACML	eXtensible Access Control Markup Language	An OASIS standard XML-based language and processing model for defining and evaluating fine-grained, attribute-based access control policies, providing a standardised interface between PAP, PDP, PEP, and PIP components.
XFSC	Cross Federation Services Components	A suite of open-source software components developed under the Gaia-X initiative to enable federated identity, trust, catalogue, and data exchange management across different organisational domains.
ZTA	Zero Trust Architecture	A security model based on the principle of never trust, always verify, requiring continuous authentication, authorisation, and validation of every user, device, and connection regardless of network location or prior access.
ZTNA	Zero Trust Network Access	A technology solution implementing Zero Trust principles at the network access layer, granting least-privilege, identity-verified access to specific applications or services rather than broad network segments.

References

Reference	Title	Release Year	Source
AC 119-1	Airworthiness and Operational Approval of Aircraft Network Security Program (ANSP)	2015	Federal Aviation Administration (FAA)
CISA ZT Maturity Model v2.0	ZT Maturity Model v2.0	2023	Cybersecurity and Infrastructure Security Agency (CISA)
DASC'25 paper	Next aircraft hyperconnected system platform	2025	Digital Avionics Systems Conference (DASC)
DO-326A / ED-202A	Airworthiness security process	2014/2018	Radio Technical Commission for Aeronautics (RTCA) European Organisation for Civil Aviation Equipment (EUROCAE)
EASA AMC 20-42B	Aircraft cybersecurity	2020	European Union Aviation Safety Agency (EASA)
eIDAS -electronic identification and trust services	electronic identification and trust service	2018	European Union Regulation (eIDAS)
NIST SP 800-63B	Digital identity & authentication	2017	National Institute of Standards and Technology (NIST)

Reference	Title	Release Year	Source
NIST SP 800-207	Zero Trust Architecture	2020	National Institute of Standards and Technology (NIST)
ODRL	Information Model 2.2	2018	National Instituteof Standards and Technology (NIST)
RFC 2904	AAA Authorization Framework	2000	Internet Engineering Task Force (IETF)
RFC 6238	TOTP: Time-Based One-Time Password Algorithm	2011	Internet Engineering Task Force (IETF)
RFC 7519	JSON Web Token (JWT)	2015	Internet Engineering Task Force (IETF)
RFC 8705	OAuth 2.0 mTLS	2020	Internet Engineering Task Force (IETF)
RFC 9114	HTTP/3	2020	Internet Engineering Task Force (IETF)
RFC 9591	The Flexible Round-Optimized Schnorr Threshold (FROST) Protocol for Two-Round Schnorr Signatures	202 4	Internet Engineering Task Force (IETF)