Federation Architecture Pattern

facis

A Practical Pre-built Blueprint for Operating Federated Digital Ecosystems

eco
ASSOCIATION OF THE INTERNET INDUSTRY

8ra 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

FAP in a Nutshell

Federation Architecture Pattern (FAP) is a modular, extensible, and standards-based reference architecture for operating secure, interoperable, and policy-compliant federated digital ecosystems.

FAP addresses critical trust and interoperability challenges that often arise when setting up federated ecosystems by defining a **federation-native architecture**. Under this model, each actor operates independently but adheres to shared protocols, trust models, and policy enforcement standards. Such an approach promotes **modularity, trust by design, and interoperability at the edge** – enabling dynamic collaboration between autonomous entities without surrendering control. This is indispensable in trust-based ecosystems like **federated healthcare systems or cross-organizational supply chains.**

Why FAPs are Essential for Innovative Collaboration

Modern digital ecosystems demand cross-organizational interoperability – but without sacrificing data control, system autonomy, or regulatory compliance.

Existing approaches are insufficient because of:

- **Centralized brokers**, which introduce risk and bottlenecks
- **Tightly coupled APIs**, which limit flexibility and resilience
- **One-size-fits-all identity providers**, which erode sovereignty

What FAPs are for

FAPs are designed to **solve critical trust and interoperability challenges in setting up multi-organizational ecosystems** where participants operate independently, each with their own infrastructure but with shared policies and rules as well as governance.

It allows stakeholders to **retain sovereignty** over their data, processes, and services while still engaging in trusted digital collaboration using open, cryptographically verifiable protocols.

The FAP aims to abstract complexity for implementers by providing clearly defined roles, reusable components, and integration blueprints, supporting both cross-domain reuse and domain-specific customization (e.g., healthcare or supply chain metadata).

Characteristics

- **Decentralized identity and credential exchange**
- **Secure federation without tight coupling**
- **Dynamic policy enforcement using verifiable data**
- **Scalable architectures that remain composable and modular**

The Technical Approach

Technology-agnostic, modular, and container-friendly. The architecture is built around:

- **Self-Sovereign Identity (SSI)**: Participants use decentralized identifiers (DIDs) and verifiable credentials (VCs) to assert identity and trust.
- **Composability**: Each module (e.g., identity, credential issuance, access control) is independently deployable and integratable.
- **Standardized protocols:** FAP aligns with standards like W3C VC/VP, OIDC4VC, and DIDComm v2 to ensure compatibility across implementations.
- **Open-source tooling:** Components from the Eclipse XFSC project (e.g., OCM, PCM, TSA, TRAIN) are used to build trusted, policy-driven systems.

The Key Principles and Guidelines

1. **Federation by Design**
   - All components are loosely coupled and designed for distributed governance, enabling organizations to retain control over identity, data, and service policies.

2. **Modular and Pluggable Components**
   - Architecture supports replaceable components via standard APIs. For example, credential issuance can be implemented via OCM or third-party SSI agents.

3. **Self-Sovereign Identity (SSI)**
   - Users and organizations use decentralized identity frameworks (e.g., DID, VC, VP) instead of centralized authentication.

4. **Scalability and Extensibility**
   - FAP supports scaling across sectors by reusing credential models and orchestration workflows (e.g., via GitOps or event-driven flows with NATS).

5. **Interoperability and Compliance**
   - Adherence to W3C, eIDAS, GDPR, and Gaia-X Trust Framework ensures trust is verifiable, policies are enforceable, and data sovereignty is preserved.

A Dynamic Concept | Open and Community Driven

FAP is not static. It is a living pattern that encourages feedback, reuse, and alignment with best practices, regulatory frameworks, emerging standards, and federation requirements.

Contributors from industry, academia, and government can propose enhancements (e.g., new modules, integrations), submit policy templates (e.g., Rego rules for new sectors), and improve documentation, developer tooling, and deployment patterns.

Exemplary Use Cases

Federated Service Catalogue Synchronization

Organizations maintain separate service or product catalogues; updates are automatically synchronized across partners, ensuring consistent information without relying on a central registry.

Zero-Trust Access for Collaborative Ecosystems

Cross-organizational services enforce continuous authentication and attribute-based access control; users and systems access resources securely without assuming trust from network location.

Automated Partner Onboarding in Federated Ecosystem

New partners register and verify credentials digitally; access to federation services, resources, and catalogues is provisioned automatically, enabling scalable and secure ecosystem growth.

All use cases emphasize **autonomy, decentralized governance, and trust by verifiable evidence** rather than central control.

August 2025

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

Image Source: Adobe Stock

Email: info@facis.eu
Website: www.facis.eu

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