WORKSHOP 29.09.25

Deep Dive – Partner Onboarding with ORCE (Orchestration Engine)

Second FAP Workshop

info@facis.eu
www.facis.eu

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Energie
aufgrund eines Beschlusses des Deutschen Bundestages

Housekeeping Rules

Please note*

- This workshop will be recorded and made available online afterwards
- Please remain muted throughout the entire presentation
- Feel free to submit your questions in writing via the chat
- All questions will be addressed at the end of today's presentation.
- You are also welcome to ask your question/feedback verbally during the Q&A session following the presentation
  - To do so, please use the "raise hand" function.
  - We will call on you in turn – please unmute yourself, briefly state your name and company, and then ask your question.

01

Welcome & **Reference FAP Partner Onboarding Kick-off**

Nelia Zinatullina, FACIS Project Manager

AGENDA

01 Welcome & Reference FAP Partner Onboarding Kick-Off

02 Deployment & Environment

03 The Reference FACIS FAP

04 Deep Dive into Processes & Services

05 Implementing and Integrating the FAP

06 Wrap-Up & Final Q&A

BACKGROUND

FACIS is part of the Important Projects of Common European Interest – Cloud Infrastructure and Services (IPCEI-CIS)/8ra initiative. As cloud computing and edge technologies grow in importance, there is a pressing need for innovative solutions that can ensure service continuity and security across multiple providers.

FACIS leverages existing FOSS technologies and builds on previous data ecosystem initiatives. It excludes the development of proprietary solutions not aligned with FOSS principles.

The results will be made available as non-commercial contributions under Creative Commons (CC-BY), while the software components will be provided under the Apache 2.0 license and made accessible under the governance of the Eclipse Foundation: https://gitlab.eclipse.org/eclipse/xfsc

OVERVIEW

FACIS aims to stabilize the application of functional components for the establishment of cooperative data ecosystems with a focus on infrastructure-related service offerings. In particular, the operationalisation of networked cloud edge appliacations is being considered.

01

Federation Architecture Patterns

Generic, orchestrated architectural design patterns of recurring scenarios in the design of federated service groups (compute, data, identities, service catalogs, trust services) using FOSS components (8ra, XFSC, IX API, EDC, etc.)

02

SLA Governance Framework

Development of a governance framework for infrastructure and service providers for the bundled provision of multi provider/services/locations for cooperative data ecosystems.

03

Digital Contracting

Concept for dynamic service bundling with SLA statement; development of a digital contracting service with EUDI support

04

Community Empowerment

Rise the FOSS community for the accelerated adaptation of federated data infrastructure software component

WORKSHOP OBJECTIVES

By the end of this workshop, you will be able to:

- Understand the architectural structure of the **Reference FAP**
- Learn how ORCE is used to orchestrate the entire partner onboarding lifecycle.
- Get hands-on exposure to the **real FAP flow** used in FACIS.
- Understand how to integrate, customize, and reuse FAP processes for your domain.

STRATEGIC IMPORTANCE OF FAP ONBOARDING

Why this matters

- FACIS aims to standardize onboarding across domains (automotive, aviation, health, etc.).
- FAP1 is the reference onboarding flow.
- It combines Gaia-X compliance, VC/VP handling, and service trust anchors into one reusable orchestration model.
- This model is driven by real technical flows in ORCE

THE ROLE OF ORCE IN THE FEDERATION

ORCE = Orchestration Engine

- Visually configurable logic engine
- Executes onboarding tasks step-by-step (e.g., VC issuance, participant validation)
- Enables organizations to customize their onboarding path (e.g., add internal approval, SLA checks)
- The "brain" behind the entire **participant lifecycle**

Workshop Format

Presentation + Architecture Deep Dives
Hands-On with FAP Flows (real Node-RED flow)
We will analyze: VC issuance, Gaia-X compliance, registry submission
Focused on how you can replicate, extend or customize this onboarding

02

Deployment & Environment

Hossein Rafieekhah | FACIS WP1 Lead

WHAT IS THE “8RA” ENVIRONMENT?

- **Internal testbed environment** for FAP validation.
- **Based on:**
  - Cloud-managed Kubernetes
  - Gaia-X Trust Framework
  - TLS-secured ingress routing
  - Domain: *.facis.cloud
- **Why it matters:**
  - Ensures integration with Trust Anchors
  - Let’s you test VC issuance, registry access, and trust verification flows

TECHNICAL DEPLOYMENT STACK

Layer	Technology	Notes
Orchestration	ORCE	Manages the flow execution
Identity & Trust	Keycloak + Gaia-X Realm Config	Handles DID, VC, credential issuance
Federation Logic	Node-RED Flows	Implements FAP onboarding tasks
Deployment	Kubernetes + Helm	Each service is a Helm release
Communication	NATS + HTTPS	Real-time messaging and secure APIs

PRE-REQUIREMENTS FOR DEPLOYMENT

- Kubernetes Cluster (v1.27+)
- Ingress Controller (e.g., nginx) + wildcard DNS
- ORCE running (docker or cluster)
- Gaia-X Realm + Gaia-X JWK
- TLS cert (via FACIS or self-signed)

SERVICES TO BE DEPLOYED

- ORCE (Runtime)
- Catalogue (for SD storage)
- Keycloak
- Participant Registry Adapter
- VC/VP handling microservices
- Optional: PCM, TSA

HOW EASYSTACK BUILDER SOLVES IT

EasyStack Builder automates the deployment of all required components:

- Inject kubeconfig
- Pick component (e.g., ORCE or Catalogue)
- Fill domain and cert info

Handles:

- Helm injection
- Secret management
- Keycloak setup
- DNS + TLS bindings

Deploying the Catalogue takes ~2 minutes.

03

The Reference FACIS FAP

Hossein Rafieekhah | FACIS WP1 Lead

WHAT IS A FAP (FEDERATION ARCHITECTURE PATTERN)?

Definition:
A **reference model** that standardizes how federation services (Catalogue, OCM, PCM, TSA, etc.) interact.

Purpose:

- Provide a **common vocabulary** for developers & integrators
- Define reusable building blocks for onboarding flows

In FACIS:

- FAP1 = *Reference Participant Onboarding Flow*
- Backbone of federation trust model

FACIS FAP1
CORE COMPONENTS

- **ORCE Runtime** :Orchestrates the flow
- **Catalogue** : Stores Self-Descriptions (SDs)
- **OCM** :Manages organization credentials
- **PCM Cloud** :Manages personal credentials
- **TSA E1** : Timestamping & policy service
- **AA Service** : Authentication & authorization
- **Trusted Registry Adapter** : Connects to Trust Anchors (in this case Gaia-X Clearinghouses)

FAP1: END-TO-END ONBOARDING LIFECYCLE

- **Registration** : Organization submits SD
- **Validation** : Check against Gaia-X Trust Registry and Credential issuing
- **Credential Management** : OCM/PCM manages Verifiable Credentials for Organizations and Individuals
- **Integration** : Register in Catalogue
- **Federation Access** : Entity is entitled to act as part of the verified federation

WHY REFERENCE FAP MATTERS

- Provides **blueprint** to ramp up federations like aviation, automotive, manufacturing and others
- Avoids duplication – common orchestration reused across projects
- Accelerates onboarding process
- Allows AI-assisted flow generation
- Includes trust & compliance mechanism

HOW IT’S IMPLEMENTED IN ORCE

- Each FAP step is a **flow fragment** (inject → validate → register → audit)
- EasyStack Builder nodes map directly to federation services
- Message (msg) object carries onboarding status across steps
- Context storage used for **state tracking** (requested → approved → published)
- Outputs are standardized (URL, client secret, status) for chaining

EXAMPLE: REFERENCE ONBOARDING FLOW

Trigger: HTTP In → /onboard-participant
Validation: Function node → Schema check
External Calls: HTTP Request → Trusted Registry API
Decision: Switch node → Approved / Rejected / Pending
Action:

- Approved → Register SD in Catalogue
- Pending → Delay + Retry
- Rejected → Log & Error path

Output: ORCE Dashboard → Participant status visible

WHY A DEEP DIVE IS NEEDED

Reference FAP aggregates many subprocesses in a well orchestrated way.

We need to walk through these sub steps to allow adoption for individual needs and to

- understand VC/VP management, schema validation, registry sync
- reuse building blocks for the usage of **federation services** which must be orchestrated together
- include service which introduces **compliance + trust enforcement** rules

In practice: This is where most technical teams lack of experiance.

CORE PROCESSES IN FAP1

- **Authentication & Access Control**
  - AA issues tokens (OAuth2, OIDC) for services
  - Ensures only valid federated members access flows
- **Catalogue Registration**
  - SD stored in Federated Catalogue
  - Metadata accessible for search & discovery
- **Audit & Timestamping**
  - Signing credential issuance
  - Maintain Immutable audit logs
- **Self-Description (SD)Validation**
  - Upload SD → Validate against Trusted Shape Registry
  - JSON-LD schema check, signature verification
- **Credential Issuance (VC)**
  - OCM/PCM issue credentials once SD is validated
  - DID:WEB integration with Keycloak realm

04

Deep Dive into Processes & Services

Hossein Rafieekhah | FACIS WP1 Lead

SERVICE DEEP DIVE - CATALOGUE

Role: Central index of all participants & services
Stores **validated SDs** with search capabilities
Integrates with ORCE to:
- Accept new onboarding entries
- Update status (requested → approved → published)
- Expose data to federation participants

Pain points:
- Default Helm charts not optimized (memory-heavy Neo4j)
- EasyStack Builder solves → deploys easy with minimal resources

SERVICE DEEP DIVE – OCM & PCM

- **OCM (Organizational Credential Manager)**
  - Issues/verifies org-level VCs
  - Handles revocation & renewal
- **PCM (Personal Credential Manager)**
  - Manages credentials to individual users
  - Stores securely
- **ORCE** orchestrates both to:
  - Trigger issuance after validation
  - Update federation state (active/revoked)
  - Route credentials into partner onboarding workflows

SERVICE DEEP DIVE – AA (AUTHENTICATION AUTHORITY)

- Provides **OAuth2 / OIDC tokens**
- Controls session, role, and policies across federation services
- ORCE integration:
  - Ensures only **approved participants** receive valid tokens
  - Tokens flow through onboarding pipelines
- Best practice: Always integrate AA into flows, even for testbeds

PUTTING IT ALL TOGETHER

- ORCE orchestrates processes + services as a **single flow**
- Message (msg) object travels across:
  - Validation → Credential Issuance → Catalogue → Audit → Access
- Context storage keeps participant state across async calls
- Dashboard widgets show **real-time participant status**

05

Implementing and Integrating the FAP

Hossein Rafieekhah | FACIS WP1 Lead

WHY IMPLEMENTATION MATTERS

Reference FAP is only valuable if it's **replicable & integratable**
Organizations must be able to:

- Deploy FAP modules **in their own clusters**
- Integrate with **federation-level registries & trust anchors**
- Customize flows to match **internal policies**

Goal: Balance **standardization** (federation rules) and **flexibility** (domain-specific needs).

IMPLEMENTATION APPROACH WITH ORCE

Implementation Approach with ORCE

- Use **EasyStack Builder nodes** for each service (Catalogue, OCM, PCM, TSA, AA)
- Deploy ORCE centrally in Kubernetes or locally for PoC
- Import the **Reference FAP Flow (JSON)** into ORCE workspace
- Customize flow fragments (e.g., add approval subflow, customized policies)
- Connect to external Trust Services

INTEGRATION STEPS

- **Initial Setup**
  - Deploy ORCE via Docker/K8s
  - Install federation service nodes (via EasyStack Builder)
- **Configure Environment**
  - Set domain, TLS, Keycloak realm
  - Connect to federation APIs
- **Load Reference Flow**
  - Import FAP JSON flow (from FACIS GitHub repo)
  - Verify all endpoints & credentials
- **Test Integration**
  - Run sample onboarding (dummy SD)
  - Check outputs in Catalogue + logs in TSA
- **Go Live**
  - Onboard first real participant
  - Monitor dashboard for status updates

FEDERATION VALUE OF INTEGRATION

Trust: Every onboarding step logged, timestamped, auditable

Scalability: Same flow reused across domains (health, automotive, industry)

Compliance: Gaia-X rules enforced by design

Automation: No manual DevOps steps → minutes instead of days

Innovation: AI FAP Builder can generate variations from the Reference Flow

KEY TAKEAWAYS

- **ORCE** is the orchestration backbone of FACIS — enabling automation, policy enforcement, and integration.
- **Reference FAP1** provides the **blueprint for onboarding**: validation, credential issuance, catalogue registration, audit, and access control.
- **EasyStack Builder** simplifies deployment → turning multi-day manual steps into minutes.
- **Gaia-X compliance** is embedded at every stage: schemas, trust anchors, timestamps, and audit logs.
- **Customizability**: Flows can be extended for local requirements without breaking federation rules.

PRACTICAL GAINS FROM WORKSHOP #2

You can now:

- Deploy ORCE in a real environment (Docker/K8s)
- Import and understand the Reference FAP1 flow
- Identify and explain the function of each service (Catalogue, OCM, PCM, TSA, AA)
- Run a complete onboarding cycle (dummy SD → validated → VC issued → published)
- Visualize participant status in ORCE dashboard

STRATEGIC IMPACT FOR PARTICIPANTS

- Faster partner onboarding
- Clear governance & auditability for compliance teams
- Easier collaboration across federated ecosystems
- Foundation for **multi-domain scaling**
- Pathway to **AI-assisted orchestration** via AI FAP Builder

COFFEE BREAK

06

Wrap-up & Final Q&A

Hossein Rafieekhah | FACIS WP1 Lead

Ask about

Technical deployment issues
Integration scenarios
Compliance requirements
Next steps for your own organization

FAP1 is not just a demo, it is your starting point to operational federation.

Get in touch with us

Homepage

Newsletter

info@facis.eu
www.facis.eu

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Energie
aufgrund eines Beschlusses des Deutschen Bundestages

WORKSHOP 29.09.25

THANK YOU FOR JOINING US!

info@facis.eu
www.facis.eu

CLOUD·EDGE CONTINUUM

Finanziert von der Europäischen Union
NextGenerationEU

Gefördert durch:
Bundesministerium für Wirtschaft und Energie
aufgrund eines Beschlusses des Deutschen Bundestages