FACIS FAP-IoT-AI

IoT-AI Pipeline over trusted zones

Version / Date: 28-Sep-2025

Status: *Idea / **Draft** / In Review / In Implementation / Released*

This FAP demonstrates a federated data pipeline with IoT sensing and data collection, data transfer via data space connectors, data aggregation with data lake and AI based data analysis for dashboard visualisation

Purpose & Value

The goal of the project is to connect any IoT data sources with federated data spaces, data platforms and AI systems – standardised, modular, multi-tenant and fully operable on-premise or air-gapped.

The importance of this FAP lies in:

- **Edge data gathering and multiple data channel management**
- **Standardized data transfer between sender and receiver, based on dataspace protocol**
- **Cloud enabled data aggregation**
- **AI supported data analysis**
- **Result visualisation as dashboard widget per data channel**

This FAP is essential to build a **trusted, distributed ecosystem data pipeline cross over domains with trust services**, making it easier for organisations to collect, transfer, aggregate and analyse data out of IoT environment and use services and resources across federated infrastructures.

Scope & Boundaries

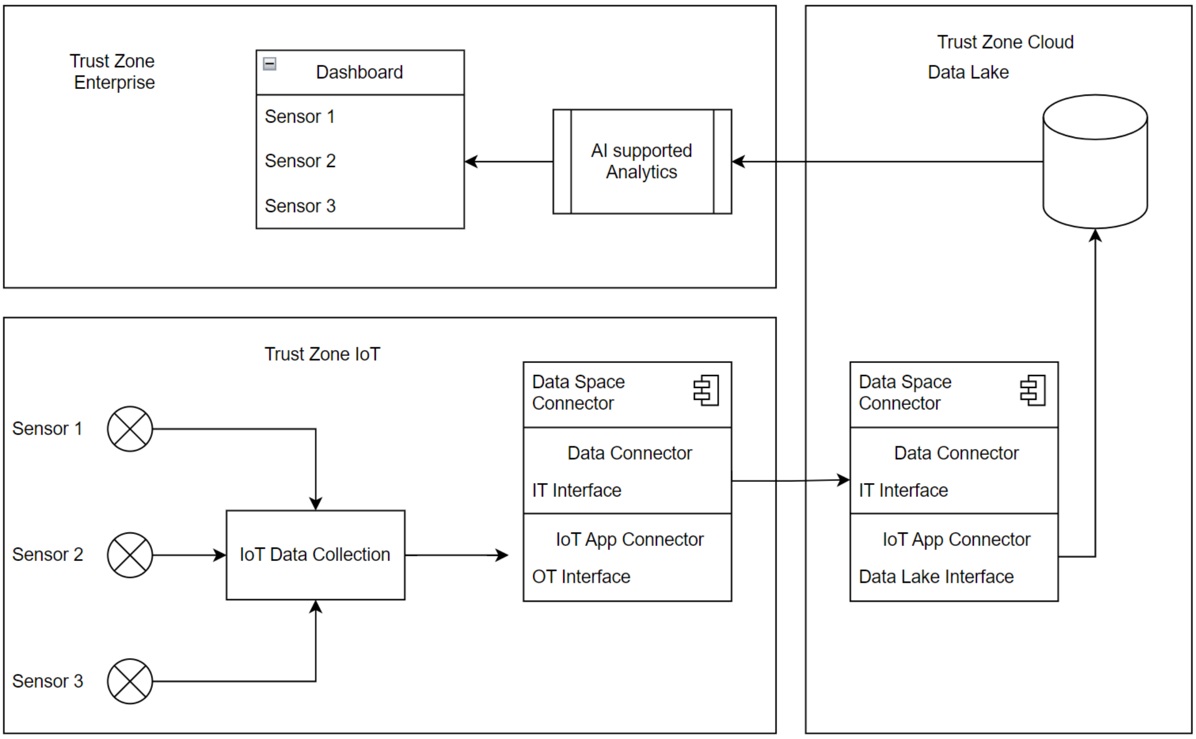
In Scope:

- Unified data pipeline from Sensor to dashboard in federated eco systems.
- Full abstraction of data gathering, data transfer, data aggregation, data analysis and data visualization
- Integration of Widgets and AI via Orchestration Engine (ORCE)
- Integration with identity credential and access management (ICAM) for secure asset access.
- Usage of Trust Anchor und Policy Engine for data flow security

Out of Scope:

- Data space management
- Complex data analytics
- Shopfloor protocols
- Deep Integration

Architecture Building Blocks



Feature-FAPs:

- IoT Data Collection.
- Data Lake Management
- AI and Visualisation

Micro-FAPs (examples):

- Data Space Connector

FAP Components:

- IoT Domain
  - Local Management of sensor data
- Data Lake
  - Consumption of IoT data streams and data persistance
- AI
  - Analytics of data and identification of thresholds
- Dashboard
  - Visual representation per data channel

XFSC Services:

- **CAT** (Catalogue): local data container for used services
- **ORCE** – orchestration of FAP service and simulation IoT backend.
- **AAS** – authentication & authorisation service.

- **OCM/PCM (Credential Managers):** For participant and principal credentials.
- **TSA Trust Service API:** Specify Policies for data connector and user actions

Standards & Protocols

- **W3C DID/VC:** Decentralized identifiers and verifiable credentials for asset provenance.
- **OIDC4VC:** Standardized flows for credential issuance.
- **DIDComm v2:** Secure communication between participants.
- **Gaia-X Trust Framework:** Compliance, trust anchor, and catalogue integration.
- **JSON-LD:** Linked data for standardized service metadata.
- **OpenAPI / GraphQL** – service discovery and integration APIs.
- **GDPR** – data minimisation and lawful processing of metadata
- **Data Space Protocol**

Reuse & Variants

The key abstraction are

- IoT Management Platform/ Shopfloor
    - Node-Red (https://nodered.org/)
- Data Space Ready Data Connectors
    - https://dssc.eu/
    - (https://github.com/International-Data-Spaces-Association/ids-specification)
    - https://projects.eclipse.org/projects/technology.edc
    - https://simpl-programme.ec.europa.eu/
- Data Lake Service for data collection and aggregation
    - https://cloud.ionos.com/solutions/big-data
    - Apache Spark, Kafka, Trino, HDFS, Superset
- AI supported Analytics
    - https://cloud.ionos.de/managed/ai-model-hub
    - Sovereign AI
- BI Dashboard
    - XFSC ORCE (Node-Red)
    - https://github.com/eclipse-xfsc/orchestration-engine

Reusable Modules:

- Data Connector.
- AI Analytics.
- Dashboard UI.

Variants:

- Asset Adminstration Shell
- Multiple Data Space Implementations

Next Steps & Involvement

- **Testing & QA** IONOS Data Lake Services and AI Model Hub.
- **Pilot integrations** with Node RED IoT Platform.
- **Community involvement:** OSS contributors, Data Space Community, Manufacturing-X

Sample Scenario (Industrial IoT data pipeline)

🏭 Scenario: Predictive Maintenance for Manufacturing Robots

A **large-scale automotive manufacturing plant** uses a fleet of sophisticated **welding robots** on its assembly line. Downtime is extremely costly, so the company wants to implement **predictive maintenance** using sensor data.

⚙️ Data Collection and Edge Processing

Data Source

- **Sensors:** Each welding robot is equipped with various IoT sensors collecting real-time data:
  - **Vibration Sensors:** Monitoring motor and joint health.
  - **Temperature Sensors:** Tracking the welding torch and hydraulic system.
  - **Current/Voltage Sensors:** Measuring power consumption and motor load.
- **Data Format:** The sensor data is collected at the robot's edge gateway.

Edge Connector

- **Component:** An **Edge Data Connector** is deployed on the plant's local network (the **Data Space**).
- **Protocol:** This connector adheres to the **Dataspace Connector protocol**. It acts as a trusted intermediary, packaging the raw data into standardized **Data Assets** and defining the associated **Usage Policies** (e.g., "Data can only be used for predictive maintenance analytics for 3 months").
- **Transmission:** The DC securely sends the data, respecting the established policies, across the internet to the cloud environment.

☁️ Cloud Environment and Data Lake

Data Ingestion and Storage

- **Cloud Platform:** The data is received by a **Cloud Environment** .
- **Data Connector Endpoint:** The cloud environment hosts a receiving **DC Endpoint** that validates the incoming data assets against the agreed-upon policies and contracts.
- **Data Lake Services:** The validated data is then ingested into the **Data Lake**.

  - **Aggregation:** Data Lake services (like streaming ingestion tools) process the incoming high-velocity data, perhaps aggregating high-frequency sensor readings into 5-minute averages to reduce volume.
  - **Storage:** The data is stored in its raw and semi-processed formats (e.g., as partitioned **Parquet** or **Delta Lake** files) within the Data Lake's inexpensive object storage. This provides a single source of truth for all historical and real-time robot data.

🧠 AI Analysis and Visualization

AI Agent Analysis

- **Agent Deployment:** A dedicated **AI Agent** (a Machine Learning model service) is deployed, which has read access to the Data Lake.
- **Analysis:** The AI Agent is trained to perform **time-series analysis** and **anomaly detection**.
  - It analyzes the aggregated data (vibration, temperature, power) against historical baselines and known failure signatures.
  - **Outcome:** When the vibration frequency in a specific robot's joint motor exceeds a calculated threshold for three consecutive intervals, the AI agent flags it as an **"Imminent Failure Risk"** with a high confidence score.

Visual Result Presentation

- **BI Dashboard:** The AI Agent's analysis results (e.g., Robot ID, Anomaly Type, Confidence Score, Estimated Time to Failure) are fed into a **Business Intelligence (BI) Dashboard** service.
- **Visualization:** The dashboard presents:
  - An **overview** of the entire robot fleet's health (a "Fleet Health Score").
  - **Detailed charts** showing the recent spike in vibration for the flagged robot.
  - A **clear, visual alert** indicating which robot requires immediate maintenance, allowing the maintenance team to schedule a fix *before* the component actually fails, thus achieving **zero unscheduled downtime**.