kit-template Documentation

Eclipse Tractus-X

Generated: 2026-02-04

> **Source: README.md**

KIT Template

Remove this file before publishing

This directory contains the comprehensive template structure for creating a new KIT (Keep It Together) in the Eclipse Tractus-X ecosystem.

Quick Start

Prerequisites

Before creating a new KIT, ensure you have:

- A clear understanding of the business problem your KIT addresses
- Familiarity with the KIT Framework Documentation
- Read the TRG 10.01 - KIT Architecture
- Read the TRG 10.02 - KIT Content Structure
- Reviewed existing KITs for reference at Eclipse Tractus-X KITs

How to Use This Template

Follow these steps to create your KIT:

1. Copy and Rename

```
# From the docs-kits/kits/ directory
cp -r ../kit-template ./<kit-id>-kit
cd <kit-id>-kit
```

2. Update Placeholders

Search and replace all `[PLACEHOLDER]` text with your KIT-specific information:
- `[KIT_NAME]` → Your KIT's official name
- `[DESCRIPTION]` → Brief description of your KIT
- `[YOUR_COMPANY]` → Your company name
- `[YYYY]` → Current year
- `[GITHUB_USERNAME]` → Your GitHub username

3. Select Your Views

Based on your KIT's maturity level, implement the required views:
- **Sandbox**: Minimum - Adoption View
- **Incubating**: Adoption + Development Views
- **Graduated**: All Views (Adoption + Development + Operations + Industry Extensions)

4. Update Master Data

Register your KIT in `/data/kitsData.js`:
- Add KIT metadata (name, description, logo, routes)
- Assign to appropriate category (Foundation, Industry Core, Cross-Industry, Industry-Specific)
- Follow the JSON schema

5. Configure Sidebar

Add your KIT to `/sidebarsDocsKits.js` to make it navigable

6. Set Code Owners

Add maintainers to `/.github/CODEOWNERS`:

```
/docs-kits/kits/your-kit-name-kit/ @your-github-username
```

Directory Structure

```
kit-template/
├── README.md                    # Main guide - START HERE (REMOVE THIS FILE BEFORE PUBLISHING)
├── changelog.md                 # Version history template
│
├── adoption-view/               # Business documentation (MANDATORY for all)
│   ├── _category_.json
│   └── adoption-view.md         # Vision, mission, business value, use cases
│
├── development-view/            # Technical documentation (Required for Incubating+)
│   ├── _category_.json
│   └── development-view.md      # Architecture, APIs, semantic models
│
├── operations-view/             # Operations documentation (Required for Graduation)
│   ├── _category_.json
│   └── operations-view.md       # Deployment, monitoring, security
│
├── industry-extensions/         # Industry-specific adaptations (Required for Graduation)
│   ├── _category_.json
│   ├── README.md                # Overview of industry extensions
│   └── automotive/              # Example: Automotive industry extension
│       ├── _category_.json
│       └── overview.md          # Catena-X standards, automotive models
│
├── success-stories/             # Reference implementations (Recommended for Graduation)
│   ├── _category_.json
│   └── overview.md              # Case studies, reference implementations
│
├── documentation/               # Additional resources (Optional)
│   ├── _category_.json
│   └── overview.md              # External links, glossary, FAQ
│
└── resources/                      # Images, diagrams, files
        img/
         └── REUSE.toml              # License from each image
```

Maturity Levels

Your KIT will progress through different maturity levels, each with specific requirements:

Sandbox

Initial development phase with basic structure

Required Artifacts:

- CHANGELOG.md
- Copyright notices in all files
- Adoption View (Vision, Mission, Business Value, Use Case)

Incubating

Active development with growing feature set

Required Artifacts:

- All Sandbox requirements
- Development View (Architecture, APIs, Semantic Models)
- Standards documentation
- Business Process documentation
- Test cases
- Code Owner (recommended)

Incubating Sub-states:

- **Draft**: Initial structure and documentation
- **In Progress**: Active implementation of features
- **In Review**: Expert review and quality assurance

Graduated

Production-ready with complete documentation

Required Artifacts:

- All Incubating requirements

- Operations View (Deployment, Monitoring, Security)
- Industry Extensions (at least one)
- Reference Implementation
- Sample Data
- Code Owner (mandatory)
- Success Stories (recommended)

For detailed artifact requirements, see TRG 10.03 - KIT Lifecycle.

Required Files by Maturity Level

Mandatory for All Maturity Levels

File/Artifact	Description	Reference
`CHANGELOG.md`	Version history following semantic versioning	TRG 1.03
Copyright Notice	CC-BY-4.0 license in **every file**	TRG 7.07, TRG 7.08
Adoption View	At minimum: vision, mission, business value	TRG 10.02

Required for Graduation

File/Artifact	Description	Reference
`CODEOWNERS`	Maintainer assignments	TRG 10.02
Development View	Complete technical documentation	TRG 10.02
Operations View	Deployment and operations guides	TRG 10.02
Industry Extensions	At least one industry implementation	TRG 10.02
Reference Implementation	Working COTS or OSS implementation	TRG 10.04
Test Cases	Validation and testing documentation	TRG 10.03

Documentation Guidelines

Adoption View

Target Audience: Business stakeholders, decision-makers, non-technical users

Focus: Business value, use cases, and strategic benefits

Required Content:

- **Vision & Mission**: Strategic objectives and inspiration for solution providers
- **Business Value**: 3-5 key benefits with descriptions (ROI, cost savings, market access)
- **Use Case Explanation**: Industry problem, current challenges, and benefits by stakeholder type
- **Business Processes**: Process flows, access policies, and data sovereignty considerations
- **Semantic Models**: Data structure definitions for interoperability
- **Standards**: Industry standards and compliance requirements
- **Tutorials** (Optional for Sandbox/Incubating): Videos or step-by-step guides

Best Practices:

- Use clear, non-technical language
- Include visual diagrams and flowcharts
- Provide real-world examples
- Highlight competitive advantages
- Show value from multiple stakeholder perspectives (OEM, SME, Solution Provider)

Development View

Target Audience: Software developers, architects, technical implementers

Focus: Technical implementation, APIs, and development resources

Required Content:

- **Architecture Overview**: System design, components, and architectural patterns

- **Component/Sequence Diagrams**: Visual representations of system interactions
- **API Specifications**: OpenAPI/Swagger files with endpoint documentation
- **Standards**: Technical standards and protocol compliance
- **Logic/Schema**: Business logic definitions and data flow diagrams
- **Semantic Models**: Detailed data structures and relationships
- **Test Cases**: Unit tests, integration tests, and validation scenarios
- **Sample Data**: Example datasets and payloads
- **Tutorials**: Developer quick-start guides and code examples

Best Practices:

- Include working code examples
- Provide API endpoint examples with request/response samples
- Document error handling and edge cases
- Include architecture diagrams (C4, UML, etc.)
- Link to live API documentation
- Provide sample data in multiple formats (JSON, XML, CSV)

Operations View

Target Audience: DevOps engineers, system administrators, operators

Focus: Deployment, operations, and maintenance

Industry Extensions

Target Audience: Industry-specific implementers

Focus: Industry-specific adaptations while maintaining core interoperability

Required Content:

- **Industry Overview**: Specific industry context and requirements
- **Industry Standards**: Compliance with industry-specific standards (e.g., Catena-X, IDTA, ISO/DIN)
- **Semantic Models**: Industry-specific data models and extensions
- **Use Cases**: Industry-specific scenarios and implementations
- **Code Owner**: Industry extension maintainer(s)

Implementation Checklist

Phase 1: Setup (Sandbox)

- [ ] Copy template directory to `/docs-kits/kits/your-kit-name-kit/`
- [ ] Update README.md with KIT-specific information
- [ ] Create CHANGELOG.md with initial version
- [ ] Add copyright notices to all files
- [ ] Implement basic Adoption View (vision, mission, business value)
- [ ] Create KIT logo and banner in `/assets/`
- [ ] Register KIT in `/data/kitsData.js`
- [ ] Add sidebar configuration in `/sidebarsDocsKits.js`
- [ ] Create initial GitHub issue in sig-release

Phase 2: Development (Incubating)

- [ ] Complete Adoption View documentation
- [ ] Implement Development View structure
- [ ] Document architecture and components
- [ ] Create API specifications (OpenAPI)
- [ ] Document semantic models and data structures
- [ ] Add business process documentation
- [ ] Create developer tutorials
- [ ] Implement test cases
- [ ] Add sample data
- [ ] Document standards compliance
- [ ] Set recommended code owner in CODEOWNERS

Phase 3: Production (Graduated)

- [ ] Complete Operations View documentation
- [ ] Create deployment guides and scripts
- [ ] Document monitoring and security
- [ ] Implement at least one Industry Extension
- [ ] Add reference implementations

- [ ] Document success stories
- [ ] Set mandatory code owner in CODEOWNERS
- [ ] Complete expert review process
- [ ] Submit graduation request in sig-release

Compliance Requirements

Your KIT **MUST** comply with these Tractus-X Release Guidelines (TRGs):

TRG	Title	Requirement
TRG 10.01	KIT Architecture	KIT category classification and registration in master data
TRG 10.02	KIT Content Structure	Required content sections and structure
TRG 10.03	KIT Lifecycle	Maturity levels and artifact requirements
TRG 10.04	KIT Graduation Process	Requirements and process for graduation
TRG 10.05	KIT Deprecation Process	Deprecation criteria and procedures
TRG 7.07	Legal Notice for Non-Code	Image and media licensing (CC-BY-4.0)
TRG 7.08	Legal Notice for Documentation	Documentation licensing (CC-BY-4.0)
TRG 1.03	CHANGELOG.md	Semantic versioning and changelog format

Additional Resources

Documentation

- KIT Framework Documentation - Complete KIT structure and artifacts guide

- KIT Getting Started Guide - Step-by-step KIT creation guide
- KIT Lifecycle Guide - Maturity levels and progression
- KIT Master Data Overview - Master data structure and registration

Examples

Browse existing KITs for reference:

- Connector KIT - Dataspace Foundation
- Digital Twin KIT - Industry Core Foundation
- PCF Exchange KIT - Cross-Industry Use Case
- MaaS KIT - Industry-Specific Use Case

Tools

- JSON Schema for KIT Master Data - Validation schema for master data
- Artifact Requirements Data - Detailed artifact requirements by maturity level

Support and Community

Get Help

- **KIT Community Office Hours**: Weekly alignment meetings - Join Meeting
- **Matrix Chat**: Daily discussions and support - #tractusx-kits:matrix.eclipse.org
- **GitHub Issues**: Report bugs or request features - Create Issue

Alignment Mechanisms

Channel	Purpose	Frequency	Link
Alignment Day	Quarterly refinement sessions	Quarterly	Release Process
Open Planning Day	Open planning sessions	Quarterly	Release Process
KIT Office Hours	Community alignment	Weekly	Join Meeting
Matrix Chat	Daily support	Ongoing	Join Chat

Contributing

Before creating a KIT, follow the contribution process:

1. **Discuss Idea**: Share in GitHub Discussions (optional but recommended)
2. **Create Ticket**: Submit ticket in sig-release
3. **Present**: Present your KIT in Open Planning Day of target release
4. **Build**: Develop your KIT in your fork following this template
5. **Submit PR**: Create pull request to main repository with changelog and updated master data
6. **Review**: Get approval from responsible committer
7. **Publish**: KIT published in target release changelog

For detailed contribution guidelines, see KIT Getting Started Guide.

Happy KIT Building!

For questions, reach out via Matrix Chat or KIT Office Hours.

NOTICE

> Source: adoption-view/adoption-view.md

Adoption View

Welcome to the **[KIT_NAME] KIT Adoption View**. This view provides business value, strategic benefits, and use cases for business stakeholders and decision-makers.

info: Target Audience

Business Managers, Product Owners, Solution Architects, Industry Experts, and Decision Makers.

Business Value

Value Proposition #1: [Title]

Benefit: [Primary benefit description]

Target Stakeholders: [OEMs | SMEs | Solution Providers | etc.]

Measurable Outcomes: [Key metrics]

Value Proposition #2: [Title]

Benefit: [Second benefit description]

Target Stakeholders: [Target audience]

Measurable Outcomes: [Key metrics]

Value Proposition #3: [Title]

Benefit: [Third benefit description]

Target Stakeholders: [Target audience]

Measurable Outcomes: [Key metrics]

Use Case Context

Industry Challenge

[Describe current industry problems and pain points]

Current Challenges:

- **Challenge 1**: [Problem description and impact]
- **Challenge 2**: [Problem description and impact]
- **Challenge 3**: [Problem description and impact]

The Solution

[Explain how this KIT addresses the challenges]

Solution Components:

1. **[Component 1]**: [Description]
2. **[Component 2]**: [Description]
3. **[Component 3]**: [Description]

Business Processes

tip: For industry-specific business processes, see the Industry Extensions documentation.

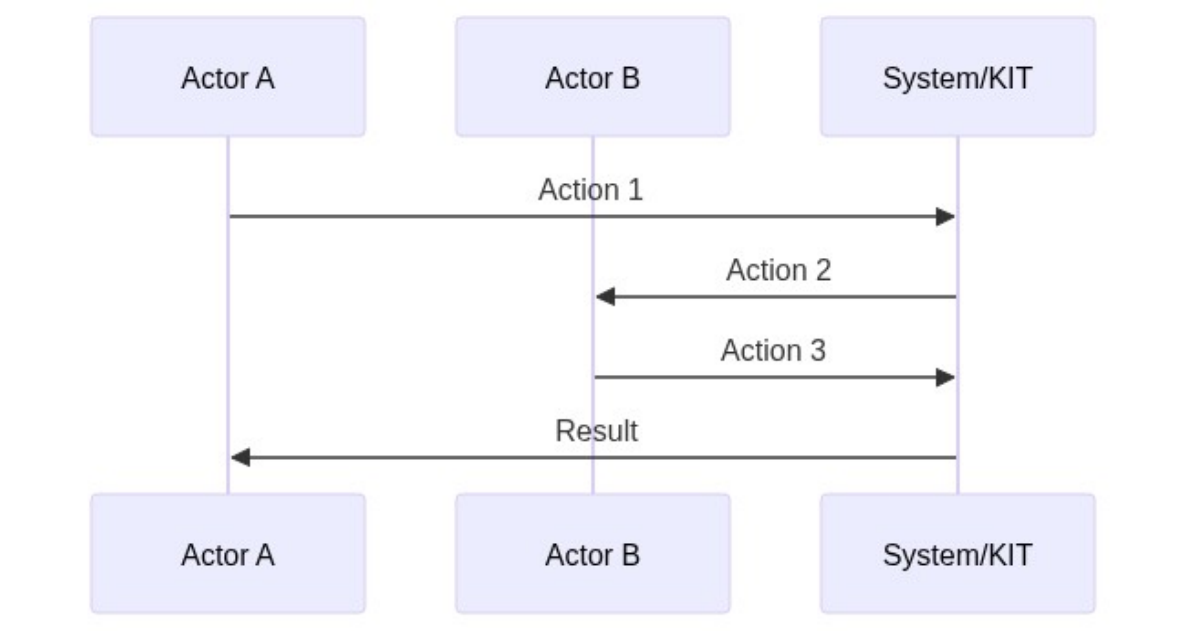
Core Business Process: [Process Name]

Purpose: [Business goal]

Stakeholders: [List key stakeholders]

Process Steps:

📊 **Diagram**



Source Code:

```
sequenceDiagram
    participant A as Actor A
    participant B as Actor B
    participant C as System/KIT
A->>C: Action 1
C->>B: Action 2
B->>C: Action 3
C->>A: Result
```

Process Description: [Brief description of key steps]

Access & Usage Policies

warning: Industry-Specific Policies

For industry-specific policy requirements, refer to the Industry Extensions section.

Example Access Policy

```
{
  "policy": {
    "permission": {
      "action": "use",
      "constraint": {
        "leftOperand": "UsagePurpose",
        "operator": "isAnyOf",
        "rightOperand": [
            "mx.core.digitalTwinRegistry:1"
        ]
      }
    }
  }
}
```

[Brief policy explanation]

Standards

warning: Industry-Specific Standards

For industry-specific standards, refer to the Industry Extensions section.

Supported Standards

Standard	Version	Description	Compliance Level	Link
[Standard 1]	X.Y	[Description]	Mandatory/Optional	[Link]
[Standard 2]	X.Y	[Description]	Mandatory/Optional	[Link]

Tutorials & Resources

Getting Started Tutorial

[Link to tutorial or brief description]

Video Resources

Title	Duration	Link
[Video 1]	[X min]	[Link]

Whitepaper

[Link to whitepaper if available]

NOTICE

> Source: changelog.md

Changelog

All notable changes to this KIT will be documented in this file.

The format is based on Keep a Changelog,
and this project adheres to Semantic Versioning.

[Unreleased]

Added

Changed

Deprecated

Removed

Fixed

Security

Version History

[1.0.0] - YYYY-MM-DD

Added

- Initial release of [KIT_NAME] KIT
- Complete adoption view documentation including:
- Vision and mission statement
- Business value propositions

- Use case documentation
- Business process descriptions
- Semantic models
- Standards compliance documentation
- Complete development view documentation including:
- Architecture overview and diagrams
- API specifications (OpenAPI/Swagger)
- Component and sequence diagrams
- Logic and schema definitions
- Sample data and test cases
- Developer tutorials
- Complete operations view documentation including:
- Deployment guides and scripts
- Configuration management
- Monitoring and logging guidelines
- Security best practices
- Troubleshooting guides
- Industry extensions for [INDUSTRY_NAME]
- Reference implementation documentation
- Success stories and case studies

Changed

Deprecated

Removed

Fixed

Security

> Source: development-view/architecture.md

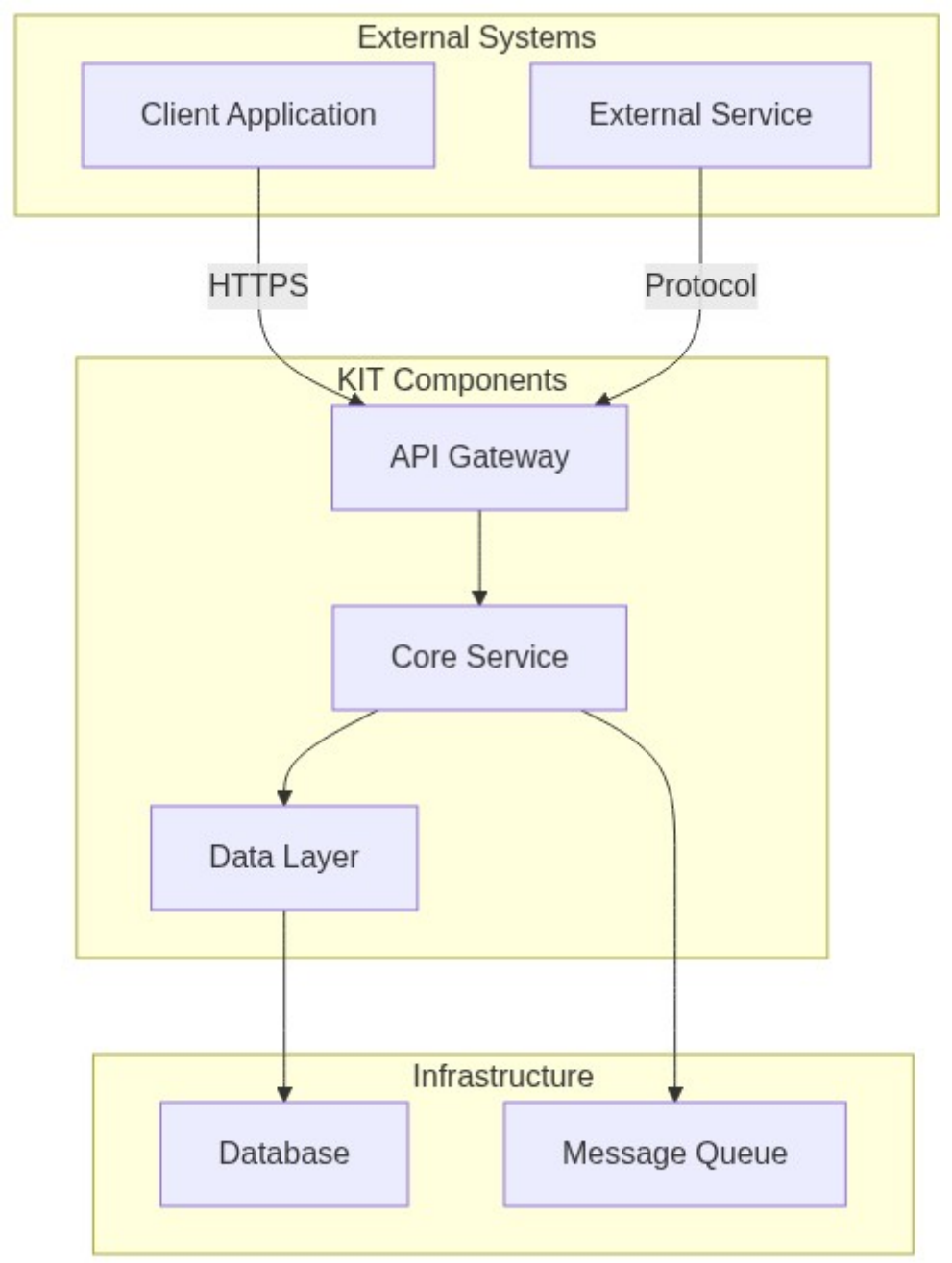
Architecture View

Architecture Overview

System Architecture

[High-level architecture diagram and explanation]

📊 Diagram



Source Code:

```
graph TB
    subgraph External Systems
        A[Client Application]
        B[External Service]
    end
subgraph KIT Components
    C[API Gateway]
    D[Core Service]
    E[Data Layer]
end

subgraph Infrastructure
    F[Database]
    G[Message Queue]
end

A -->|HTTPS| C
B -->|Protocol| C
C --> D
D --> E
E --> F
D --> G
```

Architecture Principles

1. **Modularity**: Loosely coupled components
2. **Scalability**: Horizontal scaling support
3. **Security**: End-to-end encryption
4. **Interoperability**: Standards-based APIs
5. **Observability**: Built-in monitoring and logging

NOTICE

> Source: development-view/development-view.md

Development View

Technical documentation for developers, architects, and implementers.

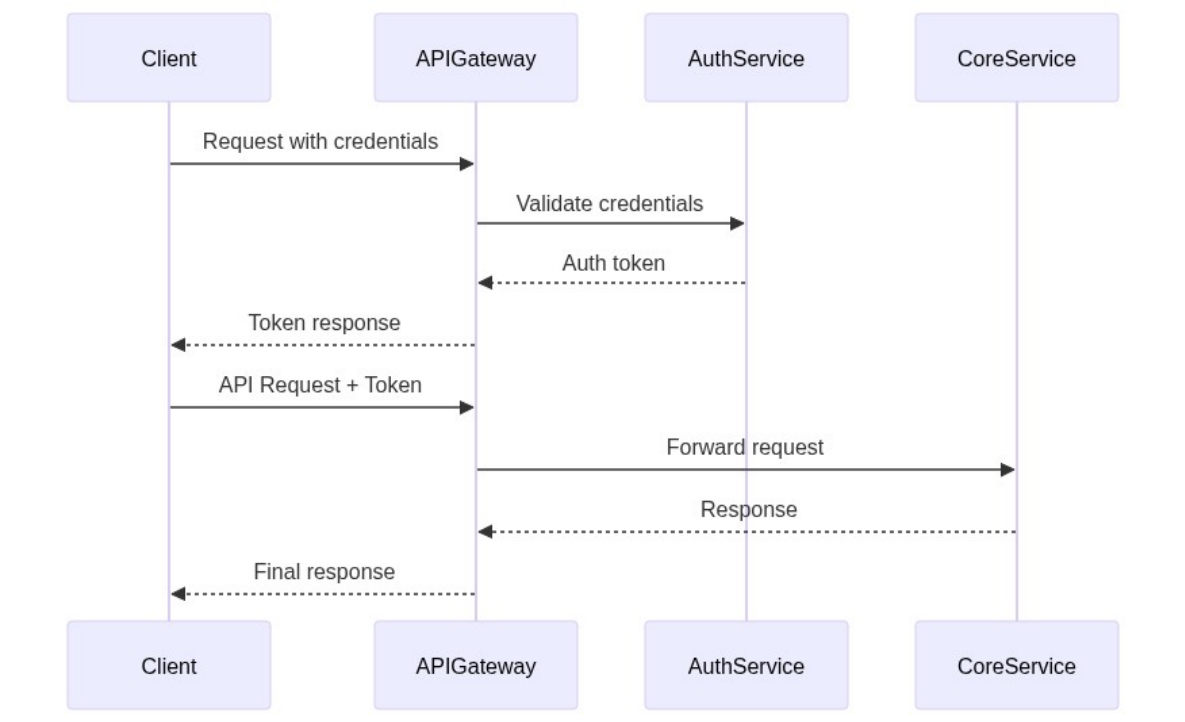
info: Target Audience

Software Developers, Solution Architects, Technical Leads, API Developers, Integration Engineers.

Sequence Diagrams

Authentication Flow

📊 Diagram



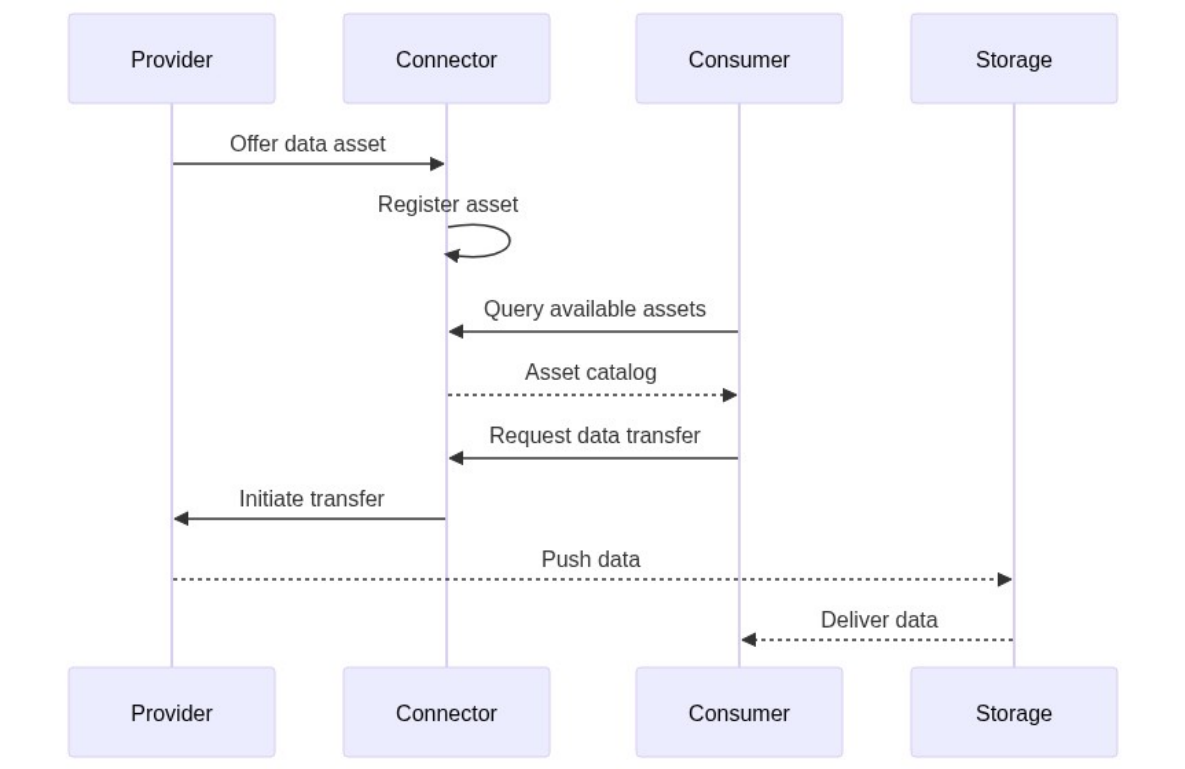
Source Code:

```
sequenceDiagram
    participant Client
    participant APIGateway
    participant AuthService
    participant CoreService
Client->>APIGateway: Request with credentials
APIGateway->>AuthService: Validate credentials
AuthService-->>APIGateway: Auth token
APIGateway-->>Client: Token response
Client->>APIGateway: API Request + Token
APIGateway->>CoreService: Forward request
CoreService-->>APIGateway: Response
APIGateway-->>Client: Final response
```

[Brief flow description]

Data Exchange Flow

📊 Diagram



Source Code:

```
sequenceDiagram
    participant Provider
    participant Connector
    participant Consumer
    participant Storage
Provider->>Connector: Offer data asset
Connector->>Connector: Register asset
Consumer->>Connector: Query available assets
Connector-->>Consumer: Asset catalog
Consumer->>Connector: Request data transfer
Connector->>Provider: Initiate transfer
Provider-->>Storage: Push data
Storage-->>Consumer: Deliver data
```

[Brief flow description]

Standards Compliance

Standard	Version	Compliance	Description
[Standard 1]	X.Y	Mandatory	[Description]
[Standard 2]	X.Y	Optional	[Description]

Standard Details

[Standard Name]

Compliance Level: [Mandatory | Optional | Recommended]

Implementation: [Brief description]

Reference: [Link]

Semantic Models

Model: [Model Name]

Version: X.Y.Z

Namespace: `urn:samm:org.eclipse.tractusx.[domain]:[version]#`

Description: [Model description]

Key Properties:

Property	Type	Required	Description
`property1`	string	Yes	[Description]
`property2`	integer	No	[Description]

Example:

```
{
  "@context": {
    "@vocab": "urn:samm:org.eclipse.tractusx.[domain]:[version]#"
  },
  "property1": "value1",
  "property2": 42
}
```

Reference: [Link to SAMM specification]

Sample Data

Sample Dataset: [Dataset Name]

Purpose: [Sample purpose]

Format: JSON

Download: [Link]

Example:

```
{
  "sampleData": [
    {
      "id": "sample-001",
      "field1": "value1"
    }
  ]
}
```

Integration Examples

Integration with [System Name]

Java Example:

```
public class KitIntegration {
    private final KitClient client;

    public KitIntegration(String apiUrl, String apiKey) {
        this.client = new KitClient(apiUrl, apiKey);
    }

    public Resource getResource(String resourceId) {
        return client.resources().get(resourceId);
    }
}
```

Python Example:

```
from kit_sdk import KitClient

client = KitClient(api_url="https://api.example.com", api_key="your-key")
resource = client.resources.get("resource-id")
```

NOTICE

> **Source: documentation/sample-data.md**

Sample data

```
{
    ...
}
```

NOTICE

> **Source: industry-extensions/automotive/overview.md**

Automotive Industry Extension

This extension adapts the **[KIT_NAME] KIT** for the automotive industry with Catena-X standards compliance.

info: Extension Purpose

Adds: Catena-X standards, automotive semantic models, business processes, and policies.

Semantic Models

Serial Part (CX-0002)

Version: 3.0.0

Aspect Model: `urn:samm:io.catenax.serial_part:3.0.0#SerialPart`

Key Attributes: `catenaXId`, `localIdentifiers`, `manufacturingInformation`, `partTypeInformation`

Example:

```
{
  "catenaXId": "urn:uuid:ed2ace5b-b25d-4e64-9b54-c2fb13c35a5c",
  "localIdentifiers": [
    {
      "key": "manufacturerPartId",
      "value": "95657362-83"
    }
  ],
  "manufacturingInformation": {
    "date": "2023-02-04T14:48:54",
    "country": "DEU"
  },
  "partTypeInformation": {
    "manufacturerPartId": "95657362-83",
    "nameAtManufacturer": "High Voltage Battery"
  }
}
```


Semantic Hub

Access & Usage Policies

Catena-X Framework Policy

```
{
  "@context": {"odrl": "http://www.w3.org/ns/odrl/2/"},
  "@type": "PolicyDefinitionRequestDto",
  "@id": "cx-policy",
  "policy": {
    "@type": "Policy",
    "odrl:permission": [{
      "odrl:action": "USE",
      "odrl:constraint": {
        "odrl:leftOperand": "BusinessPartnerNumber",
        "odrl:operator": {"@id": "odrl:eq"},
        "odrl:rightOperand": "BPNL00000003CRHK"
      }
    }]
  }
}
```

Compliance

Regulation	Region	Relevance
GDPR	EU	Data protection
Battery Regulation	EU	Battery passport
Supply Chain Due Diligence	DE	ESG reporting

Certifications: ISO/TS 16949, VDA 6.3, TISAX

Resources

- Catena-X Standard Library
- Tractus-X Open Source

NOTICE

This work is licensed under the CC-BY-4.0.

- SPDX-License-Identifier: CC-BY-4.0
- SPDX-FileCopyrightText: [YYYY] [YOUR_COMPANY]
- SPDX-FileCopyrightText: [YYYY] Contributors to the Eclipse Foundation
- Source URL: https://github.com/eclipse-tractusx/eclipse-tractusx.github.io

> Source: industry-extensions/shop-floor/overview.md

Shop-Floor Industry Extension

Overview

This extension adapts the **[KIT_NAME] KIT** for shop-floor operations and manufacturing environments.

info: Extension Purpose

Adds: Manufacturing-X standards, shop-floor semantic models, MES processes, and real-time data exchange.

Semantic Models

Production Order

Version: 1.0.0

Aspect Model: `urn:samm:io.manufacturingx.production_order:1.0.0#ProductionOrder`

Key Attributes: `orderId`, `orderType`, `productIdentifier`, `quantity`, `scheduledTime`, `status`, `assignedResources`

Example:

```
{
  "orderId": "PO-2025-11-21-001234",
  "orderType": "STANDARD",
  "productIdentifier": {
    "productId": "PART-A-12345",
    "productName": "Gear Housing Assembly"
  },
  "quantity": {
    "planned": 500,
    "produced": 342,
    "unit": "pieces"
  },
  "status": "IN_PROGRESS",
  "assignedResources": [
    {"resourceType": "MACHINE", "resourceId": "CNC-MILL-03"}
  ]
}
```


Machine State

Version: 1.0.0

Aspect Model: `urn:samm:io.manufacturingx.machine_state:1.0.0#MachineState`

Key Attributes: `machineId`, `machineType`, `operationalStatus`, `performance`, `alarms`, `maintenance`, `timestamp`

Example:

```
{
  "machineId": "CNC-MILL-03",
  "machineType": "5-Axis CNC Milling Center",
  "operationalStatus": "PRODUCING",
  "performance": {
    "oee": 78.5,
    "availability": 92.3,
    "quality": 95.5
  },
  "alarms": [
    {"severity": "WARNING", "type": "TOOL_WEAR", "message": "Tool #5 approaching wear limit"}
  ],
  "timestamp": "2025-11-21T12:00:00Z"
}
```


Access & Usage Policies

Manufacturing-X Framework Policy


```
{
  "@context": {"odrl": "http://www.w3.org/ns/odrl/2/"},
  "@type": "PolicyDefinitionRequestDto",
  "@id": "mx-policy",
  "policy": {
    "@type": "Policy",
    "odrl:permission": [{
      "odrl:action": "USE",
      "odrl:constraint": {
        "odrl:leftOperand": "DataspaceParticipant",
        "odrl:operator": {"@id": "odrl:eq"},
        "odrl:rightOperand": "did:web:example.com"
      }
    }]
  }
}
```

Compliance

Regulation	Region	Relevance
GDPR	EU	Data protection
ISO 9001	Global	Quality management

Certifications: ISO/TS 16949, VDA 6.3

Resources

- Manufacturing-X
- Tractus-X Open Source

NOTICE

> **Source: operations-view/operations-view.md**

Operations View

Guidance for deploying, operating, and maintaining this KIT in production environments.

info: Target Audience

DevOps Engineers, System Administrators, Site Reliability Engineers, Cloud Architects, Infrastructure Teams.

Deployment

Docker Compose

```
git clone https://github.com/eclipse-tractusx/[repository-name].git
cd [repository-name]
docker-compose up -d
```

docker-compose.yml:

```
version: '3.8'
services:
  app:
    image: [registry]/[kit-name]:latest
    ports:
      - "8080:8080"
    environment:
      - DB_HOST=postgres
      - DB_NAME=[kit_db]
      - DB_USER=[kit_user]
      - DB_PASSWORD=[CHANGE_ME]
    depends_on:
      - postgres

  postgres:
    image: postgres:14-alpine
    environment:
      - POSTGRES_DB=[kit_db]
      - POSTGRES_USER=[kit_user]
      - POSTGRES_PASSWORD=[CHANGE_ME]
    volumes:
      - postgres-data:/var/lib/postgresql/data

volumes:
  postgres-data:
```

Kubernetes

```
helm repo add tractusx https://eclipse-tractusx.github.io/charts
helm install [kit-name] tractusx/[kit-name] --namespace [kit-name]
```

deployment.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: [kit-name]
spec:
  replicas: 2
  selector:
    matchLabels:
      app: [kit-name]
  template:
    metadata:
      labels:
        app: [kit-name]
    spec:
      containers:
      - name: [kit-name]
        image: [registry]/[kit-name]:1.0.0
        ports:
        - containerPort: 8080
        resources:
          limits:
            cpu: 1000m
            memory: 1Gi
```

Monitoring

Health Checks

Endpoint	Purpose
`/health`	Overall health status
`/health/liveness`	Liveness probe
`/health/readiness`	Readiness probe

Metrics

Prometheus metrics: `/actuator/prometheus`

Troubleshooting

Common Issues

Application Won't Start

- Check database connectivity
- Verify environment variables
- Check logs: `docker-compose logs` or `kubectl logs`

High Memory Usage

- Monitor: `kubectl top pods`
- Increase memory limits in deployment config
- Tune JVM settings

Diagnostic Commands

```
# Docker
docker-compose logs [service-name]

# Kubernetes
kubectl logs -n [kit-name] [pod-name]
kubectl describe pod -n [kit-name] [pod-name]
```

Scaling

```
# Manual scaling
kubectl scale deployment [kit-name] --replicas=5

# Auto-scaling
kubectl autoscale deployment [kit-name] --cpu-percent=70 --min=2 --max=10
```

NOTICE

- SPDX-FileCopyrightText: [YYYY] [YOUR_COMPANY]
- SPDX-FileCopyrightText: [YYYY] Contributors to the Eclipse Foundation
- Source URL: https://github.com/eclipse-tractusx/eclipse-tractusx.github.io

> Source: success-stories/my-app.md

[Your Application Name] - Success Story

or [Application Logo]

Organization: [Company/Organization Name]

Industry: [Industry Sector]

Implementation Type: COTS / Open Source / Custom

Go-Live Date: [Month Year]

KIT Version: [e.g., 1.2.0]

Link to Repo: [if exists]

The Challenge

Business Problem

[Describe the business problem or opportunity that led to implementing this KIT]

Key Pain Points:

- [Pain point 1]
- [Pain point 2]
- [Pain point 3]

Technical Requirements

- [Technical requirement 1]

- [Technical requirement 2]
- [Technical requirement 3]

Demos

[Include here demo videos]

Results & Impact

Business Outcomes

Metric	Before	After	Improvement
[Metric 1]	[Value]	[Value]	[X%]
[Metric 2]	[Value]	[Value]	[X%]
[Metric 3]	[Value]	[Value]	[X%]

Quantified Benefits

- **Cost Savings**: [e.g., "Reduced operational costs by 25%"]
- **Time Savings**: [e.g., "Decreased processing time from 2 hours to 15 minutes"]
- **Quality Improvement**: [e.g., "Reduced error rate by 40%"]
- **Revenue Impact**: [e.g., "Enabled new revenue stream worth €X million"]

Qualitative Benefits

- [Benefit 1 - e.g., "Improved customer satisfaction"]
- [Benefit 2 - e.g., "Enhanced data transparency"]
- [Benefit 3 - e.g., "Better compliance management"]

Testimonial

> *"[Quote from project sponsor or key stakeholder about the implementation and its impact]"*

> — ***[Name]**, [Title], [Company]*

Future Plans

- [ ] [Planned enhancement 1]
- [ ] [Planned enhancement 2]
- [ ] [Planned expansion to other business units/regions]

Related Resources

- [Link to implementation guide]
- [Link to technical documentation]
- [Link to presentation or webinar]

NOTICE