HORIZON –CL4-2022-TWIN-TRANSITION-01-06
HORIZON Innovation Actions
Grant Agreement No.: 101092295

Circular and Dynamic Manufacturing Supply Chain Orchestration and OptimiSation

D4.4 First version of Integrated Platform			
Report Identifier:	D4.4		
Work-package:	WP4	**Task:**	T4.4
Responsible Partner:	ED	**Version Number:**	1.0
Due Date	M16	**Document Date:**	12/12/2025
Distribution Security:	PU	**Deliverable Type:**	DEM
Keywords:	NGSI-LD, FIWARE, Orion-LD, Data Platform, Common Data Model, Interoperability, Context Broker		
Project website: https://circuloos.eu/			

Document History

Version	Content & Changes	Issue Date
0.1	**Document created**	**04/11/2024**
0.2	**Document sent for review**	**28/11/2025**
0.3	**Document reviewed**	**04/12/2025**
0.4	**Document reviewed**	**11/12/2025**
0.5	**Reviews are combined**	**11/12/2025**
0.6	**Sent for Quality Assurance**	**11/12/2025**
1.0	**Quality Assurance and Submission**	**12/12/2025**

Quality Control

	Organisation	Date
Editor	ED	04/11/2024
Peer review 1	MWCB	04/12/2025
Peer review 2	INCL	11/12/2025
Authorised by (Technical Coordinator)	ED	11/12/2025
Authorised by (Quality Manager)	ED	11/12/2025
Submitted by (Project Coordinator)	ED	11/12/2025

Legal Disclaimer

CIRCULOOS is an EU project funded by the Horizon Europe (HORIZON) research and innovation programme under grant agreement No. 101092295. The information and views set out in this deliverable are those of the author(s) and do not necessarily reflect the official opinion of the European Union. The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any specific purpose. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein. The CIRCULOOS Consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.

Copyright notice

Table of Contents

List of Figures

Abbreviations

Acronym	Description
AI	Artificial Intelligence
API	Application Programming Interface
CDM	Common Data Model
CE	Circular Economy
CIM	Context Information Management
CM	Circular Manufacturing
CMRA	Circular Manufacturing Reference Architecture
CSV	Comma-Separated Values
DT	Digital Twin
ETSI	European Telecommunications Standards Institute
EXD	Experiment Demonstrator (from the Open Calls)
GUI	Graphical User Interface
IAM	Identity and Access Management
IIoT	Industrial Internet of Things
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
JWT	JSON Web Token
KPI	Key Performance Indicator
LCA	Life Cycle Assessment
LD	Linked Data
MSME	Manufacturing Small and Medium Enterprise
NGSI	Next Generation Service Interfaces
NGSI-LD	Next Generation Service Interfaces – Linked Data
ORION-LD	FIWARE Context Broker implementing NGSI-LD
PEP	Policy Enforcement Point
SDM	Smart Data Model
SSO	Single Sign-On

SSL / HTTPS	Secure Sockets Layer / HyperText Transfer Protocol Secure
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business
URI / URN	Uniform Resource Identifier / Name
VC / VP	Verifiable Credential / Verifiable Presentation
WP	Work Package

Executive Summary

This deliverable presents the initial operational release of the CIRCULOOS Data Platform, which serves as the digital backbone for connecting, integrating, and managing data flows within the circular manufacturing ecosystem. The platform enables the secure exchange of contextual and process data among multiple stakeholders, factories, and digital tools developed in WP3, establishing the foundation for interoperability, traceability, and data-driven optimization across supply chains.

The first version of the CIRCULOOS Platform integrates all core FIWARE Generic Enablers namely Orion-LD, Mintaka, Keycloak, and Kong, together with custom services such as the CSV-to-Orion Agent and the Leather Board Outline Tool. All components are containerized, deployable via Docker Compose, and accessible either locally (on-premises) or through cloud installations, ensuring flexibility and scalability for pilot use cases.

A major achievement of this release is the implementation of the CIRCULOOS Common Data Model (CDM), a layered semantic framework built on the ETSI NGSI-LD standard and extended with domain-specific attributes from FIWARE Smart Data Models and CIRCULOOS-specific vocabularies.
This model ensures that all components, from factory sensors to higher-level applications such as GRETA, SCOPT, and the Digital Twin, can exchange and interpret data seamlessly.

The platform also implements centralized data exchange mechanisms, enabling real-time (Orion-LD) and historical (Mintaka) data retrieval through standardized NGSI-LD APIs.
A demonstration of the chain of transaction illustrates how products are created, offered, sold, and transferred between organizations using the same standardized framework.

This version of the integrated platform marks the completion of the first operational milestone of the CIRCULOOS architecture.
It provides a robust, standards-based foundation on which the next iterations will be built to include federated data exchange, enhanced interoperability with pilots, and advanced analytics capabilities.

The repository is publicly accessible (offered as Open Source code) in Github at this link: https://github.com/european-dynamics-rnd/circuloos-data-platform

1 Introduction

1.1 Project Introduction

The overall vision of CIRCULOOS is to deliver the tools to enable MSMEs to become full members of the Circular Manufacturing value chain. These tools orchestrate and continuously optimise the supply-chain end-to-end and integrate planning and execution monitoring to enable transparent and on-time communication. Combining these with direct calculation of the product sustainability and circularity profile, for both internal and external partners, this environment will enable them to configure and execute disruptive circular manufacturing processes for sustainable production that covers the entire life cycle of products; either by recovering the value of products that ended-up as waste or from recycled and remanufactured products.

To achieve this objective the project aims to deploy:

- Circular end-to-end supply chain orchestration for collaborative workflows which incorporates planning and execution metrics and integrates advanced and multimodal visualisation and analytics. The visualisation is delivered by comprehensive Digital Twins of the supply chains formulated, the factory processes and product design phases.
- Supply Chain Optimisation that monitors the global (across the supply chain) and local (within the factory) processes and execution, inputs and outputs and configuration parameters, to enable data-driven AI decision making, this way supporting continuous optimisation of targeted and measured performance and sustainability parameters.
- Dynamic Sustainability Assessment functionalities that investigate alternative supply-chain scenarios (varying in terms of materials used, processing technologies, suppliers involved and/or activated circular economy practices) in place of the existing schemes, quickly measuring their performance in terms of environmental sustainability and circular economy profile.
- Supply Chain Data Spaces for seamless, multi-level data flow across the supply chain partners, supporting the reuse of materials in novel products, the extension of the life-cycle of finished products (remanufacturing), and data-driven decisions for collaboration of parties offering matching services in the most dynamic and efficient way.
- Cybersecure and trustworthy data sharing across the supply chain by employing a distributed, trusted and efficient Identity and Access management system, that together with the associated trust framework will coordinate the identities of all IoT objects and ensure trustworthy data sharing among its members, aligned with the trust framework that is being implemented in EBSI.
- CM specific tools for the automatic recognition of recyclable parts by modern Machine Vision tools and Advanced Robotics, to enable optimised flows in the selection process.
- Novel circular business processes will be demonstrated supporting reusing, reducing, and recycling material in production and consumption systems. The new collaborative production models will provide quantifiable results on the sustainability increase across the supply chain, in terms of efficient use of raw materials, of by-products, of waste and energy and of emissions reduction. CIRCULOOS leverages the above with the RAMP integrated innovation IOT platform and the European network around it to deliver a CM ecosystem and platform for Manufacturing SMEs.
- Skills upskilling and reskilling will be provided in RAMP and through online courses, webinars, and best practice guides and success stories based on the pilots and Experiments for Demonstration (EXDs).

1.2 Deliverable Purpose

The purpose of this deliverable is to deliver the first integrated, functional version of the CIRCULOOS Platform, consolidating all essential components developed under WP4 and integrating them with the

tools and models from WP3.
It serves as proof of technical readiness of the platform architecture, data model, and API-based communication mechanisms between the various services and pilot factories.

Specifically, this deliverable:

- Describes the CIRCULOOS Common Data Model (CDM) and its alignment with NGSI-LD, which ensures semantic consistency and interoperability across all tools and data sources.
- Presents the technical architecture of the platform, detailing the roles and interconnections of Orion-LD, Mintaka, Keycloak, and Kong.
- Demonstrates custom CIRCULOOS services, including the CSV-to-Orion Agent and the Leather Board Outline Tool, that enable integration of factory data and production metadata into the platform.
- Explains the data ingestion and retrieval mechanisms, including both *push* (subscription-based) and *pull* (query-based) interactions.
- Showcases the sale process (chain of transaction) as an end-to-end example of how the platform supports traceable data sharing and ownership transfer.
- Provides guidance for local deployment using Docker Compose, ensuring that partners and pilots can independently install and test the platform in their environments.

This deliverable represents a key integration milestone, validating that the CIRCULOOS platform is technically operational, interoperable, and ready for deployment within the pilot sites and demonstration environments.

2 Semantic Framework and Common Data Model

2.1 Description of the CIRCULOOS Common Data Model (CDM)

The CIRCULOOS CDM defines how all data are structured, described, and exchanged within the CIRCULOOS platform. It establishes a unified semantic layer that ensures interoperability between every system component, ranging from factory-level data collection (via sensors or CSV agents) to higher-level tools such as GRETA, SCOPT, SCDT, the Orchestrator, and the Computer Vision modules. The CDM provides a shared "language" through which these heterogeneous systems communicate, enabling data to be understood and reused consistently across the entire supply chain ecosystem.

The CDM is based on the NGSI-LD open standard developed by ETSI for context information management. NGSI-LD introduces a structured and semantically rich data representation model built on linked data principles. Each real-world object, process, or observation is represented as an entity with unique identifiers (URIs), typed attributes, and temporal and spatial metadata. This allows data from different domains—such as energy consumption, material flows, emissions, or product characteristics—to be linked together meaningfully and queried dynamically across distributed systems.

In practice, the CIRCULOOS CDM is implemented through a layered JSON-LD @context architecture that combines standard and project-specific vocabularies.

- The first layer uses the official NGSI-LD core context (ngsi-ld-core-context-v1.7.jsonld), which provides the foundational vocabulary for NGSI-LD entities, properties, relationships, timestamps (observedAt, createdAt, modifiedAt), and geospatial attributes (location, GeoProperty, coordinates).
- The second layer integrates the FIWARE Smart Data Models (SDM), a collection of open, domain-specific vocabularies covering Devices, Energy, and Environment. These provide well-established definitions for concepts such as Device, DeviceMeasurement, AirQualityObserved, ACMeasurement, and attributes like voltage, co2, relativeHumidity, and temperature.
- The third layer is the CIRCULOOS-specific context (circuloos-context.jsonld), which extends these standard models with project-relevant attributes tailored to circular manufacturing domains, for example, material-specific properties such as color, thickness, surfaceSize, tannedProcess, and recycledTimes for leather-based products.

For operational convenience, these three layers are also consolidated in a single merged file (merge_data_model.jsonld), which serves as the unified reference context used by the platform's agents and services. This ensures that every CIRCULOOS component can resolve all terms—standard or project-specific—from one authoritative location.

Together, these layers form the complete CIRCULOOS CDM, ensuring full semantic alignment between all platform services. This structure guarantees that data generated by one component (e.g., the Computer Vision system) can be directly interpreted by another (e.g., the Digital Twin or GRETA sustainability tool) without additional mapping or transformation. It also allows all services to share and query data in a consistent, standards-compliant way.

All entities in the CIRCULOOS platform follow a uniform modeling convention. Each has a globally unique id in URN format (e.g. urn:ngsi-ld:Product:leather:KB-2025-000123), a type indicating its category (e.g. Product, Process, Factory, Device), and a set of 'Properties', 'Relationships', and 'GeoProperties'. Each

'Property' includes its value, optional unitCode (from UN/CEFACT), and observedAt timestamp representing when the measurement occurred. Relationships are defined with object references that link entities (e.g., a Product related to a Factory through Relationsip). Geospatial information is captured using GeoJSON geometry within a GeoProperty. This modeling pattern is universal across all data ingested, whether it originates from IoT devices, CSV uploads, or simulation tools.

The CDM also defines a clear data lifecycle. During ingestion, data are captured as NGSI-LD JSON objects either through direct API submissions or via the CSV-to-Orion Agent, which converts tabular files into compliant NGSI-LD entities. These entities are stored in the Orion-LD Context Broker, which maintains the latest entity state, while the Mintaka service handles temporal data storage and time-series retrieval using observedAt as the temporal key. Data can then be retrieved through standard NGSI-LD APIs for real-time monitoring, historical analysis, or visualization in the Digital Twin interface. This lifecycle ensures that all measurements and updates remain traceable, timestamped, and queryable according to NGSI-LD specifications.

To simplify partner adoption, the CDM aligns directly with the structure of commonly used datasets and CSV templates. For example, CSV headers such as id, type, name, color, thickness, and observedat map directly to NGSI-LD attributes within the model. This mapping enables the CSV Agent to automatically generate valid entities that comply with the CDM, eliminating the need for manual coding or schema adjustments. The use of UN/CEFACT unit codes, ISO 8601 timestamps, and standardized relationships ensures data consistency across all pilots and tenants in the platform.

Quality assurance and validation are intrinsic to the CDM design. The use of shared URIs and published contexts guarantees that every attribute name resolves to an authoritative definition. Syntax and structure are validated against the NGSI-LD specification, and entity payloads are tested using automated scripts to verify correct typing, temporal attributes, and geospatial properties. Each tenant in the CIRCULOOS ecosystem operates under a unique NGSILD-Tenant identifier to ensure data isolation, while all share the same CDM to maintain cross-pilot interoperability.

Finally, the CDM is openly governed and version-controlled in the CIRCULOOS GitHub repository (https://github.com/european-dynamics-rnd/CIRCULOOS_Data_model).
This repository hosts the NGSI-LD core context, the CIRCULOOS-specific context, and the merged operational context, along with documentation for deployment, validation, and updates. All changes to the CDM follow semantic versioning and are reviewed jointly by WP4 (Data Platform) and WP3 (Tool Integration) partners to ensure stability and backward compatibility. This transparent governance model enables continuous improvement of the data model, promotes reuse beyond the project's duration, and positions the CIRCULOOS CDM as a reference for circular manufacturing data interoperability.

2.2 NGSI-LD Standard and Its Role in CIRCULOOS

NGSI-LD (Next Generation Service Interfaces – Linked Data) is a standard specification developed by the European Telecommunications Standards Institute (ETSI) under the Context Information Management (CIM) initiative. It defines a common API and data model for representing, linking, and exchanging context information across diverse domains and systems.

Built on Linked Data principles and using JSON-LD (JavaScript Object Notation for Linked Data), NGSI-LD enables data about real-world entities—such as devices, materials, factories, or processes—to be described in a machine-understandable and semantically interoperable format. Each entity can include attributes (descriptive properties like temperature or color), relationships (links to other entities), and temporal metadata (observedAt, createdAt, modifiedAt) that capture how information evolves over time.

The goal of NGSI-LD is to provide a unified, semantically rich framework that allows heterogeneous systems to:

- Exchange and interpret contextual data consistently.
- Link distributed information sources through semantic relationships.
- Support dynamic updates, queries, and subscriptions on real-world events.
- Integrate IoT, AI, and digital twin applications through a shared model.

In CIRCULOOS, NGSI-LD serves as the semantic backbone of the platform. It defines how context data—originating from factories, sensors, or digital tools—is structured, described, and linked across the system. By adopting NGSI-LD, CIRCULOOS ensures that all platform components operate on a common semantic model, enabling interoperability, traceability, and seamless data integration across circular manufacturing processes and partners.

2.3 Data lifecycle (ingestion, storage, retrieval)

The term 'data lifecycle' in the CIRCULOOS platform represents the complete process through which contextual information is created, managed, and made available across the ecosystem. It covers the stages of data ingestion, storage, temporal management, and retrieval, all implemented in full compliance with the NGSI-LD standard and the CIRCULOOS CDM.

Data ingestion occurs through several standardized entry points:

- Direct NGSI-LD API submission: Partners and connected systems can send NGSI-LD-compliant JSON payloads directly to the Orion-LD Context Broker, which manages entity creation and updates.
- CSV-to-Orion Agent: A dedicated tool provided by CIRCULOOS that allows partners to upload tabular data (CSV files) and automatically convert them into NGSI-LD entities. Each CSV must include at least id and type columns, and optionally observedAt timestamps. The agent generates JSON-LD entities following the CDM and sends them to Orion-LD via authenticated requests.
- Federation from local brokers: Local Orion-LD instances at pilot sites can automatically forward selected data entities to the central CIRCULOOS platform using NGSI-LD federation mechanisms, preserving data ownership while contributing relevant context for analysis.

Once ingested, all contextual information is stored and managed within the CIRCULOOS platform's core FIWARE components:

- The Orion-LD Context Broker maintains the current state of all entities. Each entity update replaces or modifies the existing attributes, while system metadata (createdAt, modifiedAt) are automatically recorded.

- The Mintaka service provides the temporal layer, enabling storage of historical states and time-series data for all attributes that include the observedAt field. This allows users and services to query how a parameter (e.g., temperature, energy consumption, production rate) evolved over time.
- All entities are organized within isolated tenants using the NGSILD-Tenant header, ensuring that data from each partner or pilot remains securely separated while conforming to the shared CDM.

Data retrieval is handled through standard NGSI-LD API queries, providing both real-time and historical access:

- Real-time retrieval: Users or connected applications can retrieve the latest entity states directly from Orion-LD using filters such as entity type, attributes, or relationships.
- Temporal retrieval: Historical or time-series data can be queried through Mintaka, allowing analyses of trends, performance evolution, and production behavior.
- API access control: All requests are routed through Kong, which acts as an API gateway enforcing access policies, and authenticated via Keycloak, ensuring that only authorized users and services can access specific data streams.

Throughout the lifecycle, all exchanged data remain traceable and standardized. Each dataset is timestamped (observedAt), versioned (modifiedAt), and semantically described according to the CDM. This ensures that data from various factories, materials, or processes can be cross-referenced and reused by different tools such as GRETA (for sustainability assessment), SCOPT (for supply chain optimization), or the Digital Twin for visualization and monitoring.

2.4 Example of entity structure (Factory, Process, Product)

In the NGSI-LD model adopted by CIRCULOOS, every piece of contextual information is represented as an Entity. Each entity corresponds to a real-world object (such as a product, process, or device) and is described through a combination of Properties, Relationships, and, where relevant, GeoProperties.

Entities are uniquely identified by a URN (id), semantically defined (type), and connected to others through relationships, forming a dynamic, queryable graph of contextual data. This semantic structure ensures full interoperability across all tools and services in the CIRCULOOS ecosystem.

The following example illustrates a Product/Material entity as implemented in the CIRCULOOS platform. It is based on the orion_create_leather.json file from the official GitHub repository, representing a leather material used in the project's supply chain pilots.

```
[
    {
        "id": "urn:ngsi-ld:leather:apm5zima95",
        "type": "leather",
        "leather_type": {
            "type": "Property",
            "value": "animal",
            "observedAt": "2024-08-02T09:26:35Z"
        },
        "kind_of_animal": {
            "type": "Property",
            "value": "pig",
            "observedAt": "2024-08-02T09:26:35Z"
        },
        "leather_type_tanned": {
            "type": "Property",
            "value": "chrome",
            "observedAt": "2024-08-02T09:26:35Z"
        },
        "ownedBy": {
            "type": "Relationship",
            "object": "urn:ngsi-ld:Company:001",
            "observedAt": "2024-08-02T09:26:35Z"
        }
    }
]
```

Figure 1 Product/ Material example

Description:

- This NGSI-LD entity represents a specific leather material produced in a CIRCULOOS factory.
- The id uniquely identifies the product using a URN.
- The type defines the entity category (leather).
- kind_of_animal and leather_type are Properties that describe the material's attributes.
- The ownedBy Relationship links the product to the producing factory (urn:ngsi-ld:Company:001).
- The @context field connects the entity to the CIRCULOOS CDM and ETSI NGSI-LD core vocabulary, ensuring semantic compatibility and machine readability.

3 Platform Architecture Overview

The CIRCULOOS Platform provides a modular, scalable, and standards-based infrastructure that enables secure and interoperable data management for circular manufacturing ecosystems.
Its architecture is built upon FIWARE Generic Enablers and NGSI-LD technologies, allowing seamless integration between sensors, factory systems, digital tools, and higher-level analytics services.

The platform is designed to operate both on-premises (factory level) and in the cloud, ensuring flexibility for pilot partners while maintaining data sovereignty. It integrates data ingestion, context management, temporal storage, access control, and API exposure into one coherent framework.

3.1 General architecture

At the core of the CIRCULOOS Platform lies a set of interoperable FIWARE components, each responsible for managing a specific layer of data operations. The architecture (as depicted in Figure 2) follows a service-oriented and containerized design, deployed via Docker and Docker Compose, and can be replicated across different environments (local development, factory installations, or cloud production).

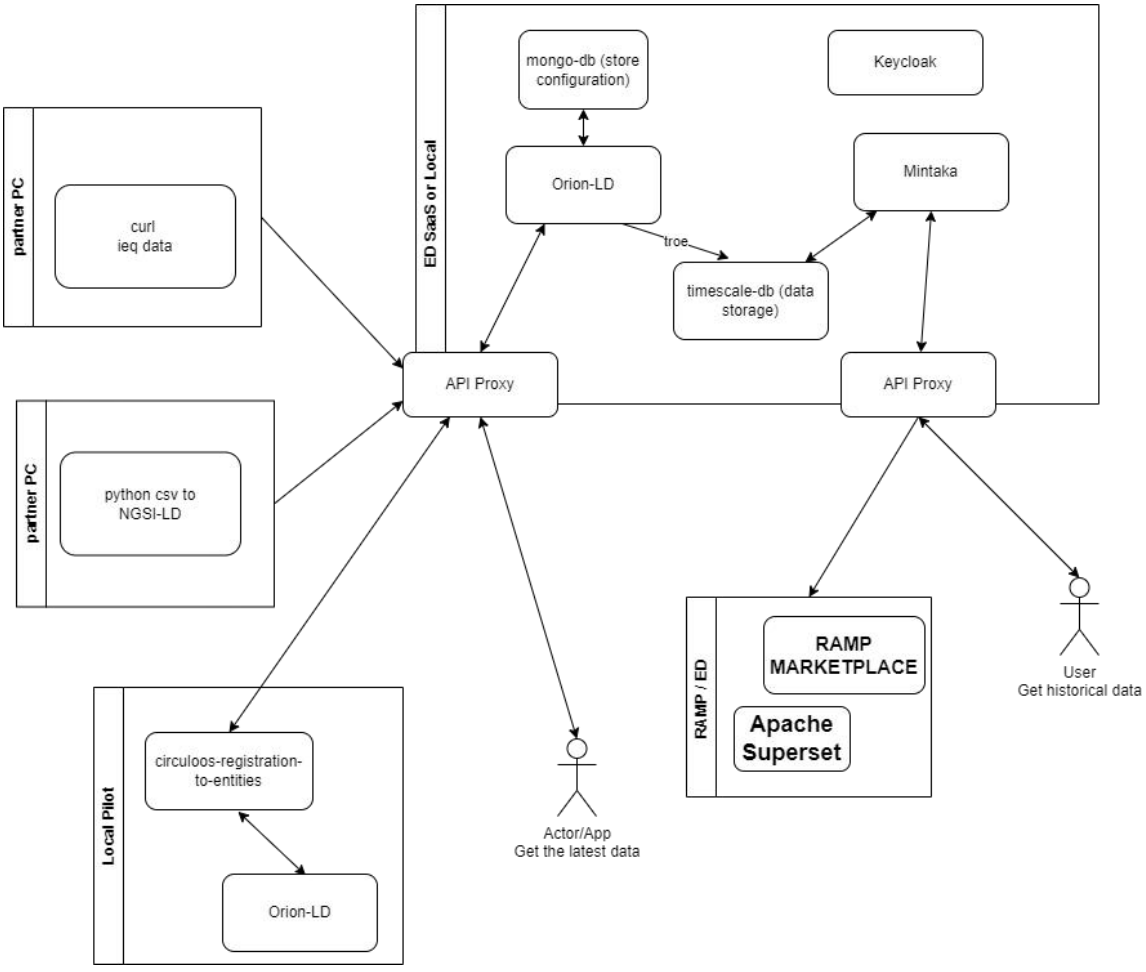

Figure 2 CIRCULOOS Integrated Platform Architecture

3.1.1 Orion-LD (context broker)

Orion-LD [1]is the core context broker of the CIRCULOOS platform. It serves as the central hub where all contextual information, representing entities, properties, and relationships, is created, updated, and queried in real time.

Main functions:

- Data ingestion and state management: Orion-LD receives NGSI-LD entities from various sources (IoT sensors, CSV Agent, or other systems) through RESTful API calls (POST, PATCH).
- Context awareness: It maintains the latest state of all entities and updates them as new information arrives. Each change automatically updates the entity's metadata (createdAt, modifiedAt).
- Semantic linking: Orion-LD handles entity relationships (e.g., linking a product to its process or factory), enabling data to form a connected knowledge graph.
- Subscriptions and notifications: Applications can subscribe to specific entities or attributes to receive real-time updates when data changes occur.
- Multi-tenancy: Orion-LD supports the NGSILD-Tenant header, allowing each partner organization or pilot to have isolated data spaces within the shared infrastructure.

In the CIRCULOOS architecture, Orion-LD acts as the main entry point for data ingestion and as the real-time interface for all WP3 tools and pilot data. It ensures that all contextual data are compliant with the NGSI-LD specification and the CIRCULOOS Common Data Model.

3.1.2 Mintaka (historical data)

Mintaka[2] is the temporal data component of the CIRCULOOS platform. It complements Orion-LD by storing the historical evolution of NGSI-LD entities and enabling time-series queries.

Main functions:

- Temporal persistence: Mintaka stores all attribute values that include the observedAt timestamp, allowing the reconstruction of an entity's state over time.
- Time-based queries: Users and tools can query past states of entities using temporal filters such as timeAt, endTimeAt, and timerel.
- Integration with Orion-LD: Mintaka works in parallel with the Context Broker. When Orion-LD receives entity updates, the corresponding time-stamped attributes are forwarded to Mintaka for temporal storage.
- Support for analytics: By preserving the time dimension, Mintaka enables performance monitoring, process optimization, and sustainability tracking within the CIRCULOOS ecosystem.

[1] https://github.com/FIWARE/context.Orion-LD

[2] https://github.com/FIWARE/mintaka

In CIRCULOOS, Mintaka ensures that both current and historical contexts are accessible, supporting data-driven insights across production stages. This is particularly valuable for evaluating process efficiency, energy trends, and lifecycle impacts in circular manufacturing.

3.1.3 Keycloak (authentication and access)

Keycloak [3]provides Identity and Access Management (IAM) for the CIRCULOOS Platform. It secures all services by managing user authentication, authorization, and token-based access to APIs.

Main functions:

- User and role management: Keycloak defines users, roles, and groups, assigning different permissions depending on organizational roles (e.g., admin, data provider, tool developer).
- Single Sign-On (SSO): Users can authenticate once and access all CIRCULOOS components and services seamlessly.
- OAuth2 and OpenID Connect protocols: These standard security frameworks are used to issue tokens for secure API access.
- Integration with Kong: All NGSI-LD API calls pass through the Kong API gateway, which validates Keycloak-issued tokens before granting access.

Keycloak ensures that data access follows the principles of least privilege and accountability, allowing secure interaction between multiple partners and tools within the CIRCULOOS federation. This guarantees compliance with EU data protection and sovereignty requirements.

3.1.4 Kong (API gateway and security)

Kong [4]functions as the API gateway and Policy Enforcement Point (PEP) of the CIRCULOOS Platform. It acts as the front-facing layer for all API traffic, ensuring secure, traceable, and controlled communication between users, applications, and FIWARE components.

Main functions:

- API routing and load balancing: Kong manages and routes HTTP requests to the appropriate internal services (Orion-LD, Mintaka, or custom applications).
- Access control: Before forwarding requests, Kong validates the authentication tokens issued by Keycloak, enforcing authorization policies.
- Rate limiting and auditing: It can monitor, throttle, or log API requests to ensure fair use and maintain system performance.

[3] https://www.keycloak.org/

[4] https://konghq.com/

- Policy enforcement: Kong ensures that only authorized users and services can access specific endpoints or tenants.
- Gateway for external access: It enables secure exposure of CIRCULOOS APIs to partners, ensuring interoperability with external NGSI-LD-compliant systems.

In the overall architecture, Kong forms the secure communication layer, ensuring that every data transaction is authenticated, authorized, and monitored. Together with Keycloak, it provides the trust and governance framework necessary for multi-tenant, cross-organizational collaboration.

3.2 Data Ingestion & Integration

The CIRCULOOS Platform incorporates a dedicated component that enables data stored in CSV format to be prepared and submitted in a structure compliant with the NGSI-LD standard. This capability supports partners who manage information in tabular form by providing a reliable method for transforming and uploading such data into the platform's context broker.

The following subsection presents this component in detail.

3.2.1 CSV-to-Orion Data Ingestion Agent

The CSV-to-Orion Agent is a custom CIRCULOOS service developed to facilitate data ingestion from tabular (CSV) sources into the NGSI-LD-based data platform.
Its purpose is to transform structured CSV data into NGSI-LD-compliant entities and automatically upload them to the Orion-LD Context Broker, ensuring that legacy or external datasets can be seamlessly integrated into the CIRCULOOS ecosystem.

This service is particularly important for pilot partners that do not directly generate NGSI-LD data but maintain information in spreadsheets, laboratory files, or production reports.
By converting these datasets into standardized NGSI-LD entities, the agent enables full interoperability with the platform's semantic and contextual data model.

3.2.1.1 Purpose and Functionality

The agent automates the process of:

1. Reading CSV files containing entity information (e.g. materials, sensors, products, processes).
2. Mapping columns to NGSI-LD attributes based on the CIRCULOOS CDM.
3. Generating NGSI-LD JSON entities that include identifiers, types, attributes, relationships, and optional timestamps.
4. Posting the generated entities to the Orion-LD Context Broker via authenticated REST API calls.

This enables partners to upload both static and time-stamped data in a few simple steps, with no need for programming or schema configuration.

3.2.1.2 Technical Overview

The CSV-to-Orion Agent is implemented in Python and deployed via Docker Compose as part of the CIRCULOOS platform under the /csv_NGSILD_Agent directory of the official repository: https://github.com/european-dynamics-rnd/circuloos-data-platform/tree/master/csv_NGSILD_Agent

Its main components include:

- load_csv_ngsild.py – the primary script that parses the CSV file, constructs NGSI-LD entities, and handles API communication with Orion-LD.
- csv_ngsild_agent_utils.py – a utility module providing helper functions for JSON-LD conversion, timestamp validation, and request handling.
- circuloos-csv-ngsild-agent.yml – the Docker Compose configuration file defining the service parameters, environment variables, and authentication credentials for deployment.
- leatherProducts.csv – an example input file demonstrating correct CSV formatting for two sample entities.

The agent exposes a lightweight web interface (running locally at http://localhost:5000) that allows users to perform all operations interactively.

3.2.1.3 Operation Workflow

1. Prepare the CSV file:

   The CSV must include at least two mandatory columns:

   - id – the unique entity identifier in URN format (e.g., urn:ngsi-ld:Product:leather:001).
   - type – the entity type (e.g., Product, Factory, Process).

Additional columns correspond to NGSI-LD properties such as name, color, thickness, surfaceSize, energyUsed, etc.

If a timestamp column named observedAt is provided (in ISO 8601 format, e.g. "2024-01-31T12:03:02Z"), the value is assigned as the observation time. Otherwise, the agent automatically inserts the current date and time.

2. Upload the CSV file: Access the local interface at http://localhost:5000, click “Browse…” to select the file, and then “Upload”.
3. Generate NGSI-LD entities: After uploading, click “Generate NGSI-LD entities” to convert the CSV content into a structured JSON-LD representation. The interface displays the resulting entities, showing how each record maps to NGSI-LD properties and relationships.
4. Send data to Orion-LD: Click “Post NGSI-LD entities to Orion-LD”. The system sends a REST POST request to the configured Orion-LD endpoint using authentication tokens provided by Keycloak. Once completed, a message confirms the IDs of all entities successfully created or updated in the context broker.
5. Verify data ingestion: Users can verify uploaded data either through the web interface or by executing command-line scripts such as: ./getDataOrionSensors.sh leather which retrieves the most recent measurements and attributes stored in Orion-LD.

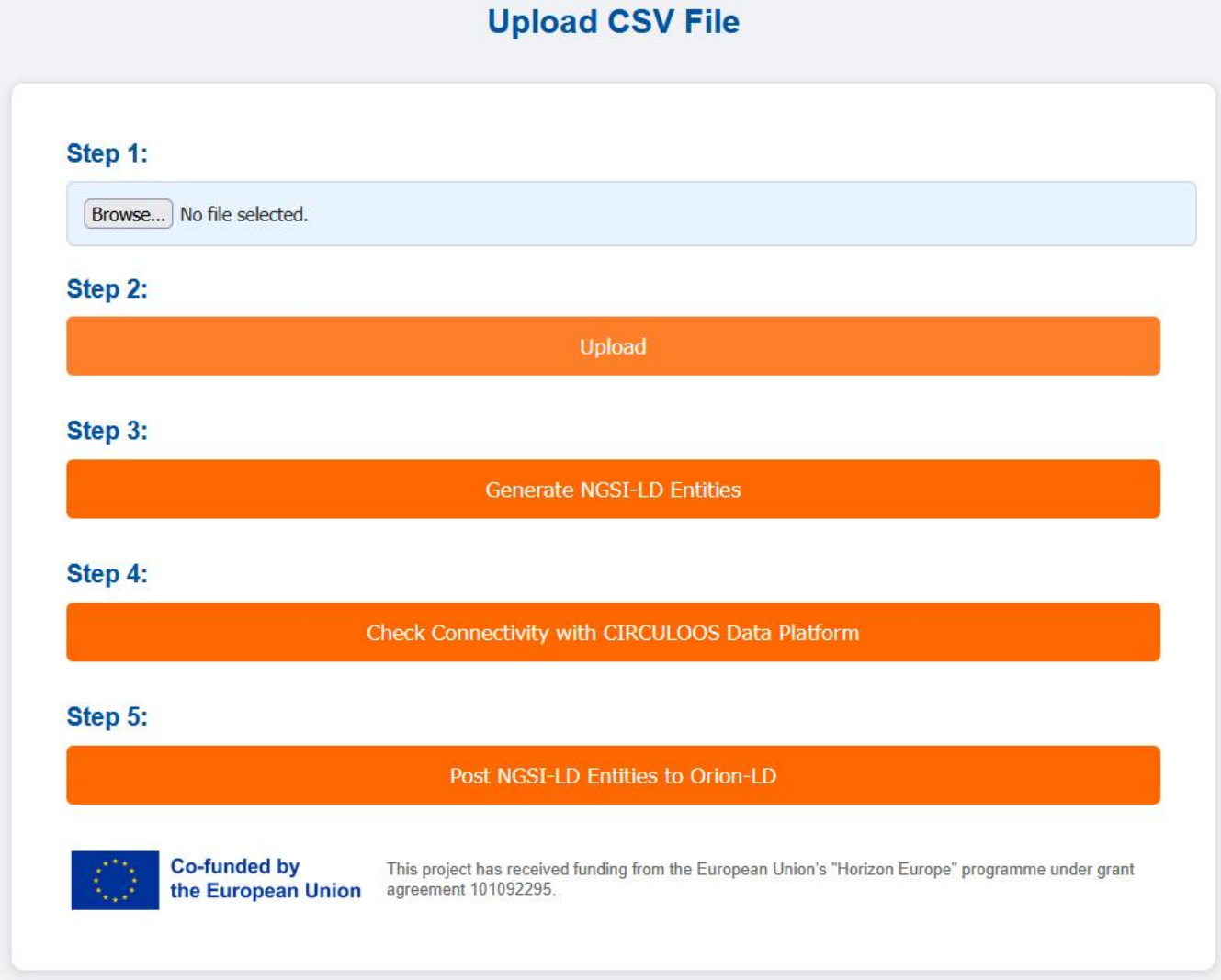

Figure 3 User Interface for CSV File Upload

3.2.1.4 Integration with the Central Platform

To send data directly to the official CIRCULOOS cloud platform rather than a local test instance, the user must configure the agent with valid partner credentials.

Steps:

1. Stop any running CIRCULOOS Docker services:./service.sh stop
2. Edit the file csv_NGSILD_Agent/circuloos-csv-ngsild-agent.yml and update the following fields with the credentials provided by the CIRCULOOS technical team:
    - PARTNER_USERNAME
    - PARTNER_PASSWORD
3. Start the agent service:
4. docker compose -f circuloos-csv-ngsild-agent.yml up

5. Follow the same upload and generation steps as in the local setup.

The agent will automatically authenticate via Keycloak and send the data through the Kong API Gateway to the CIRCULOOS Orion-LD Context Broker.

3.2.1.5 Design and Data Validation

The CSV-to-Orion Agent incorporates several built-in validation and conversion mechanisms (see code under csv_ngsild_agent_utils.py in this link: https://github.com/european-dynamics-rnd/circuloos-data-platform/blob/a8cb7e2fd0dc36245b0dece8cecc8084aa8c67ba/csv_NGSILD_Agent/csv_ngsild_agent_utils.py#L164):

- Syntax and schema checks ensure that each entity has a valid id and type, and that timestamps comply with ISO 8601.
- Automatic data typing converts numeric, textual, and boolean values into the appropriate NGSI-LD Property format.
- Real-time feedback is provided through the web interface for successful uploads or detected errors.
- Integration with the CDM: The attribute names in the CSV must correspond to terms defined in the merged CDM context (merge_data_model.jsonld), ensuring semantic alignment with all other platform components.

3.2.1.6 Role in the CIRCULOOS Architecture

The CSV-to-Orion Agent acts as a data onboarding tool within the CIRCULOOS ecosystem. It allows non-technical users or data providers to contribute information, such as production batches, quality indicators, or material properties, without interacting directly with the NGSI-LD APIs.

By standardizing CSV uploads into semantically rich entities, the agent:

- Facilitates rapid population of the platform with pilot data.
- Enables consistent integration of legacy datasets.
- Ensures all uploaded data are NGSI-LD-compliant and immediately usable by higher-level tools (SCOPT, GRETA, SCDT, and others).

This service therefore represents a key bridge between human-readable data formats and the machine-interpretable semantic layer of the CIRCULOOS platform.

3.3 Deployment

The CIRCULOOS Platform has been designed with a modular and containerized architecture that enables local (on-premises) deployment at factory or pilot level.

This setup allows partners to run their own instance of the platform within their internal infrastructure, ensuring full control over data, security, and performance while maintaining interoperability with the central CIRCULOOS ecosystem.

Local deployment is primarily intended for pilot demonstrations, factory environments, or individual partner installations.

In this setup, the entire platform stack runs on a local machine or server, allowing organizations to retain full control and ownership of their data while benefiting from the platform's NGSI-LD interoperability.

Key characteristics:

- Deployed via Docker and Docker Compose using the repository's main configuration file (docker-compose.yml) and supporting services (temporal.yml, keycloak.yml, circuloos_custom_apps.yml).
- Runs all FIWARE components locally, including Orion-LD, Mintaka, Keycloak, and Kong.
- Allows local tools (e.g., the CSV-to-Orion Agent) to connect directly to the local Orion-LD instance for data testing and validation.
- Suitable for partners wishing to federate selected data to the central cloud instance while keeping sensitive information within their premises.

System requirements:

- Linux environment (Ubuntu recommended) or Windows Subsystem for Linux (WSL).
- Docker and Docker Compose installed.
- Minimum configuration: 4 vCPUs, 8 GB RAM, and 20 GB free disk space.

Deployment steps (simplified):

1. Clone the official repository: *git clone* https://github.com/european-dynamics-rnd/circuloos-data-platform.git *cd circuloos-data-platform*
2. Start the local platform:*./service.sh start* (This script initializes all containers, downloads required images, and starts the FIWARE components.)
3. Wait for Docker to download and initialize all services (approx. 10 minutes on first launch).
4. Verify that services are running using health check scripts (*e.g., getOrionVersion.sh, getMintakaVersion.sh).*

This deployment mode is particularly useful during testing or integration phases, allowing developers to simulate data ingestion, transformation, and retrieval without requiring access to the central CIRCULOOS infrastructure.

3.4 Cybersecurity and data sovereignty

Cybersecurity and data sovereignty are central design principles of the CIRCULOOS platform. Since the platform integrates industrial, environmental, and process data across multiple partners and pilot sites, strong security mechanisms and clear data governance policies ensure the integrity, confidentiality, and ownership of all shared information. The architecture follows a "security-by-design" approach, embedding protection and access control features into every component.

The CIRCULOOS platform applies a multi-layer security model that combines authentication, authorization, encryption, and access control. The Keycloak identity management service provides centralized user authentication and role-based authorization. Each user, tool, or system authenticates using secure tokens based on OAuth2 and OpenID Connect standards, ensuring that access is traceable

and compliant with European security requirements. Roles and permissions are clearly defined for different user categories—such as administrators, data providers, or service integrators—so that each entity only accesses the data necessary for its operation.

All interactions between services are routed through the Kong API Gateway, which enforces access policies and verifies Keycloak tokens before forwarding requests. Kong also manages encryption, rate limiting, and traffic monitoring to prevent unauthorized access or service overuse. All data exchanges between external systems and the CIRCULOOS APIs are encrypted using HTTPS, guaranteeing secure communication across all network layers.

Data traceability are built into the platform's context management model. Every NGSI-LD entity stored or updated in the system automatically includes metadata such as createdAt, modifiedAt, and observedAt, ensuring that the history of each dataset can be reconstructed at any point. The combination of Orion-LD and Mintaka provides complete data persistence, with audit-ready logs of all entity updates and temporal changes. This traceability allows the platform to meet both operational monitoring needs and compliance requirements related to transparency and accountability.

From an infrastructure perspective, the platform's services are containerized and isolated using Docker. Each core service—Orion-LD, Mintaka, Keycloak, Kong, and custom agents—runs within a separate container defined in Docker Compose files, limiting dependencies and reducing potential attack surfaces. Sensitive parameters such as credentials and tokens are managed through environment variables and secure configuration files, avoiding exposure in the source code. The system is regularly updated, pulling the latest stable FIWARE releases to maintain resilience against emerging vulnerabilities.

Data sovereignty is ensured through a multi-tenant and federated architecture that respects partner autonomy. Each organization operates under a unique NGSILD-Tenant identifier, ensuring that its data remain logically isolated within the shared environment. Access to tenant-specific information is strictly controlled by Keycloak's role assignments and Kong's API policies. At the same time, federation mechanisms allow partners to share selected data with the central platform. This ensures that sensitive information remains within local control while still supporting collective analytics and cross-pilot insights.

In practice, each partner can deploy a local instance of the platform—on factory premises or private infrastructure—mirroring the cloud setup. Local brokers can federate entities with the central Orion-LD instance based on configurable rules, allowing the exchange of anonymized or aggregated data only. This hybrid model balances data sovereignty with collaboration, enabling secure participation in a shared ecosystem without compromising proprietary or confidential data.

All data handling and security operations within CIRCULOOS comply with the General Data Protection Regulation (GDPR), the EU Data Act, and the principles of emerging European Data Spaces. The use of open standards such as NGSI-LD, JSON-LD, OAuth2, and HTTPS ensures compatibility with existing and future industrial data-sharing frameworks, including the International Data Spaces (IDS) and Gaia-X initiatives.

Overall, cybersecurity and data sovereignty in CIRCULOOS are not treated as separate layers but as an integral part of the system's architecture. Through standardized security protocols, multi-tenant control, federated sharing, and transparent governance, the platform provides a trustworthy digital environment where data remain secure, traceable, and fully under the control of their rightful owners.

4 Platform services

4.1 Registration of new manufacturers

The CIRCULOOS platform allows new manufacturing partners to join the ecosystem either by submitting data directly to the central platform (centralized onboarding) or by deploying their own local instance that federates data to the central environment (federated onboarding).

The following section illustrates the centralized onboarding workflow, showing how a factory authenticates through Keycloak and submits production data to the central Orion-LD Context Broker.

Participants:

- Factory: The local system or application generating production or process data.
- Keycloak: The identity and access management service issuing authentication tokens.
- Kong (API Gateway): The gateway that validates authentication tokens and routes secure API requests.
- Central_OrionLD: The main CIRCULOOS Context Broker that stores and manages NGSI-LD entities.

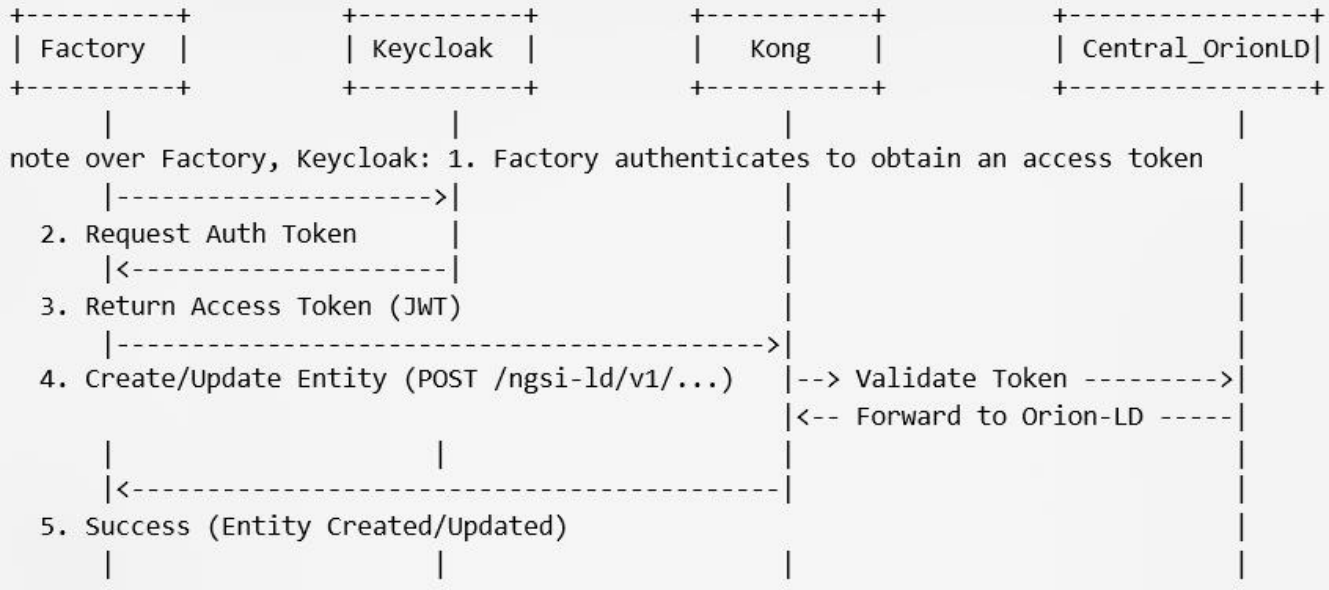

Figure 4 ACSII UML Sequence Diagram Factory Data Submission

The factory authenticates through Keycloak to obtain an access token, which it then uses to submit or update NGSI-LD entities in the Central Orion-LD broker. The token is then included in all API requests sent through Kong, which acts as the secure API gateway. Kong validates the token and forwards authorized requests to the Central Orion-LD context broker. This flow ensures that all operations, such as entity creation or updates, are securely authenticated, traceable, and isolated per tenant (NGSILD-Tenant header).

4.2 Retrieving Data ("pull mechanism")

Data within the CIRCULOOS Platform can be accessed in two primary ways: through a **push** or a **pull** mechanism. The push mechanism is event-driven meaning that data updates are automatically sent ("pushed") to subscribed applications whenever a relevant change occurs. In contrast, the pull mechanism is request-based which means that applications explicitly retrieve data from the platform when needed.

In the pull approach, client applications can:

- Access real-time entity states by sending GET requests directly to the Orion-LD Context Broker, which provides the most recent values of entities and attributes.
- Retrieve historical time-series data by querying the Mintaka service, which interfaces with the TimescaleDB database to return sequences of past measurements or events.

Pulling of data can be done either by sending direct requests to the Orion Context Broker requesting individual snapshots of the entities' values or by interfacing with Mintaka for entire time-histories. Retrieving real-time contextual data directly from the Orion-LD Context Broker represents the simplest form of the "pull" mechanism, where an application sends an HTTP GET request to the /ngsi-ld/v1/entities endpoint of Orion-LD to obtain the current state of entities registered within the platform.

Each request includes the necessary NGSI-LD headers that define the tenant namespace, context, and data format. For example, the NGSILD-Tenant header identifies the pilot or organization workspace (e.g., circuloos_demo), while the Link header points to the JSON-LD context used to interpret the entity attributes semantically.

A typical command for retrieving entity data is provided in the platform repository under commands/getDataOrionSensors.sh. The script uses curl to query the Context Broker and return results in a readable JSON-LD format, as shown below:

```
curl -s -G -X GET  'http://'"${HOST}"':'"${ORION_LD_PORT}"'/ngsi-ld/v1/entities' \
-H 'NGSILD-Tenant: circuloos_demo' \
-H 'NGSILD-Path: /' \
-H 'Link: <'"${CONTEXT}"'>; rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json"' \
-H 'Accept: application/ld+json' \
-d 'type='"${type}"'' |jq
```

This command retrieves all entities of the specified type (for example, *Product, Device*, or *Factory*), returning their latest values in NGSI-LD-compliant JSON.
The response contains each entity's identifier (*id*), type (*type*), attributes (such as *temperature, color, or thickness*), and corresponding timestamps (*observedAt, createdAt, modifiedAt*).

```
[
  {
    "id": "urn:ngsi-ld:Product:leather:001",
    "type": "Product",
    "name": { "type": "Property", "value": "Recycled Leather Sheet" },
    "color": { "type": "Property", "value": "Brown" },
    "thickness": { "type": "Property", "value": 2.5, "unitCode": "MMT" },
    "observedAt": "2025-03-10T09:00:00Z"
  }
]
```


This real-time retrieval method is particularly useful for monitoring applications, dashboards, or analytics tools that require instant access to the current state of factory processes, materials, or environmental sensors. When combined with the Mintaka-based historical retrieval, it provides a complete view of both current and past data across the CIRCULOOS ecosystem. A practical example of the pull operation is provided in the repository under *commands/getDataOrionSensors.sh* (circuloos-data-platform/commands/getDataOrionSensors.sh at master · european-dynamics-rnd/circuloos-data-platform · GitHub) , which demonstrates the standard structure of an HTTP GET request used to access contextual data. When an application needs historical information, such as environmental readings, process measurements, or production records, it interacts with the NGSI-LD Temporal API using Mintaka. The Kong API Gateway acts as the secure access point, validating authentication tokens and enforcing access control policies before forwarding valid requests to Mintaka. Mintaka then retrieves the requested information from TimescaleDB and returns it in NGSI-LD-compliant JSON format, preserving all temporal metadata.

The main components involved in this interaction are:

- Application – the client system requesting data (e.g., a digital twin, visualization, or analytics service).
- Keycloak – the identity and access management service responsible for user authentication and token issuance.
- Kong – the API Gateway enforcing access control and validating authentication tokens.
- Mintaka – the FIWARE Temporal Data Service handling historical data queries.
- TimescaleDB – the time-series database that stores the historical records linked to NGSI-LD entities.

--

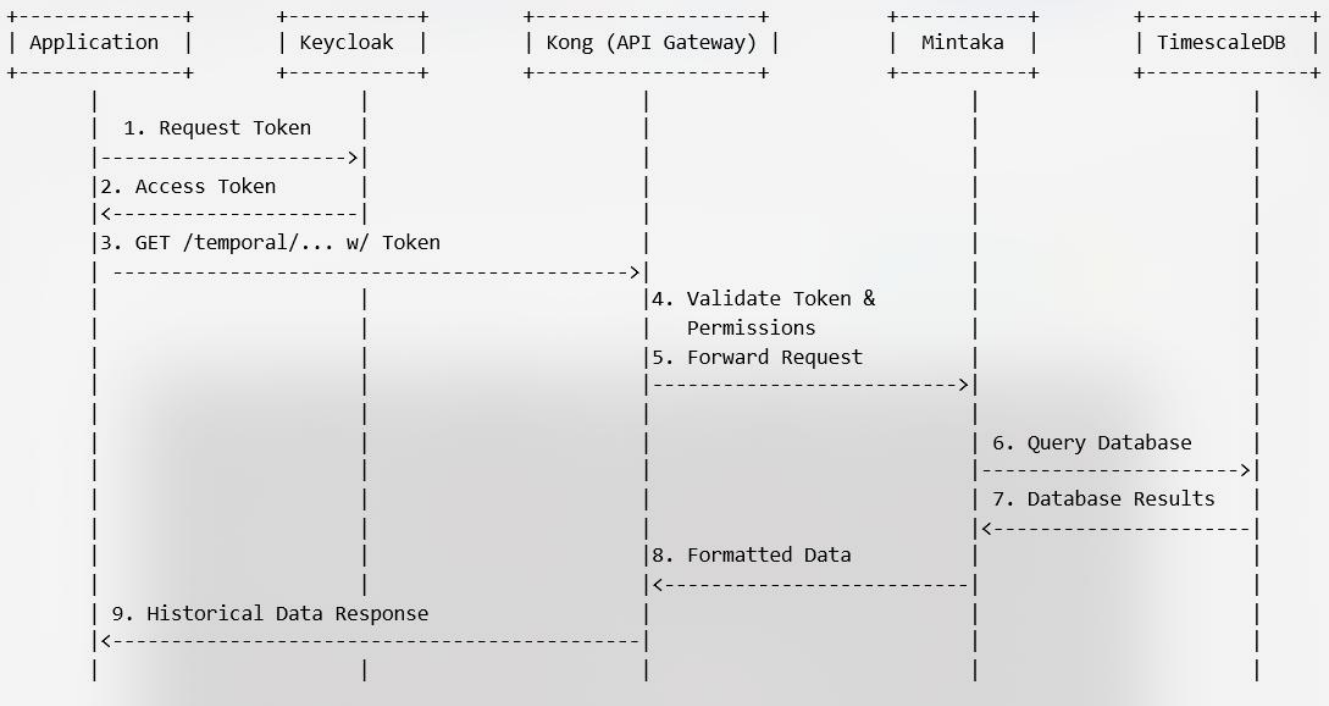

Figure 5 ASCII UML Sequence Diagram for Retrieval of historical information

Steps involved (Figure 5):

1. Request Token: The application first authenticates itself with Keycloak by providing its credentials (e.g., username/password).

2. Access Token: Keycloak validates the credentials and issues a signed JWT (JSON Web Token) access token back to the application.

3. Temporal Query: The application makes a *GET* request to retrieve historical data. Crucially, it sends this request to the Kong API Gateway's public endpoint for Mintaka and includes the access token in the *Authorization* header.

4. Validate Token & Permissions: Kong intercepts the request. Its security plugins validate the JWT access token to ensure it's authentic and not expired. It also checks that the userhas the necessary permissions to access the requested data. If the token is invalid, Kong will reject the request with a *401 Unauthorized error* .

5. Forward Request: If the token is valid, Kong forwards the request to the internal Mintaka service.

6. Query Database: Mintaka receives the request and translates the temporal query into a SQL query for the TimescaleDB database.

7. Database Results: TimescaleDB executes the query and returns the historical data rows to Mintaka.

8. Formatted Data: Mintaka formats the database results into the standard NGSI-LD JSON format and sends the response back to Kong, maintaining timestamps (*observedAt, createdAt, modifiedAt*).

9. Historical Data Response: Kong passes the final response back to the application, which now has the historical data it requested.

4.3 Sale process

The sale process in the CIRCULOOS platform demonstrates how products — such as leather materials — are created, updated, and transferred between organizations within the circular manufacturing ecosystem. It provides a transparent and traceable digital representation of a "chain of transaction," where each change in ownership or status is recorded as an NGSI-LD entity in the platform's Orion-LD Context Broker.

In the current implementation, available in the open-source repository https://github.com/european-dynamics-rnd/circuloos-data-platform/tree/master/chain_of_transaction the process is implemented in a centralized configuration. All operations take place directly within the central Orion-LD Context Broker of the CIRCULOOS platform, with authentication handled by Keycloak and access secured through Kong.
This setup means that factories, buyers, and other applications interact directly with the shared platform through authenticated NGSI-LD REST API calls—no federation or local brokers are required.

The key JSON and Postman examples provided in the repository are:

- orion_create_leather.json : defines a new product (e.g., a leather material) including descriptive properties such as leather_type, kind_of_animal, and leather_type_tanned, and a relationship ownedBy referencing the seller company.
- orion_update_leather.json : updates an existing product's ownership (e.g., transferring the ownedBy relationship to another company to represent a sale).
- orion_create_transaction.json : records a transaction event between the involved parties (seller, buyer, and product).

These examples use standard NGSI-LD REST endpoints:

- *POST /ngsi-ld/v1/entityOperations/upsert* → create or update product and company entities.
- *POST /ngsi-ld/v1/entityOperations/update?options=replace* →update entity attributes, such as ownership transfer.
- *GET /ngsi-ld/v1/entities/{id}* →retrieve entity details.
- *GET /temporal/entities/{id}*→ retrieve entity history from Mintaka.

Steps Involved (Figure 6) :

1. Get Access Token: The factory or user first authenticates via Keycloak to obtain an access token for Orion-LD and Mintaka requests.
2. Create Company Entities: The seller (urn:ngsi-ld:Company:001) and buyer (urn:ngsi-ld:Company:002) are created using POST /entityOperations/upsert.
3. Create Product Entity: A new product is registered — for example, urn:ngsi-ld:leather:apm5zima95 — with attributes such as:

- leather_type: "animal"
- kind_of_animal: "pig"
- leather_type_tanned: "chrome"

- ownedBy: "urn:ngsi-ld:Company:001"

4. Retrieve Product Information: The current product data is retrieved using GET /ngsi-ld/v1/entities/{id} to verify ownership and metadata.
5. Transfer Ownership (Sale Event)

   Ownership is transferred by sending a request to: POST /ngsi-ld/v1/entityOperations/update?options=replace with an updated ownedBy relationship pointing to the buyer (Company:002). This action effectively represents the sale within the platform.

6. View Transaction History

   Using Mintaka, historical data are retrieved through GET /temporal/entities/{id} — allowing verification of the ownership change and timestamped audit trail.

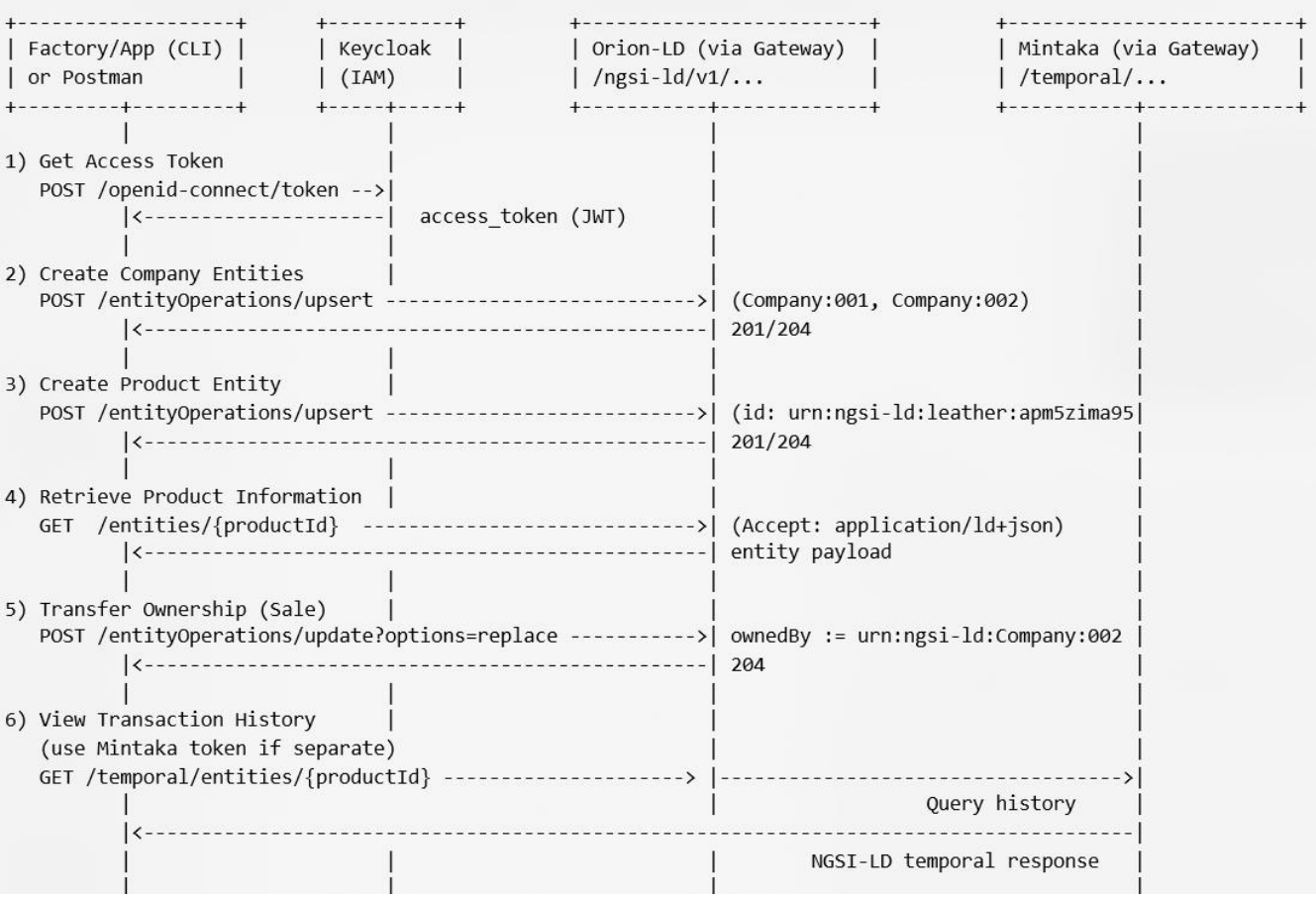

Figure 6 ASCII UML sequence diagram for the sale process

4.4 Digitised Material Extraction Module (Rectangular Leather Boards)

This module, included in the CIRCULOOS Data Platform GitHub repository, provides a tool for processing rectangular leather or fabric boards and generating 2D coordinate representations of the remaining material area after production cutting (Figure 7).

Repository reference: https://github.com/european-dynamics-rnd/circuloos-data-platform/tree/master?tab=readme-ov-file#rectangle-leather-board-covering-the-entire-image

It complements the "Irregular Leather Board Outline" Tool by supporting simpler rectangular cases in which the leather sheet covers the full camera frame. The goal is to automate the estimation of usable remaining surface area, which can then be transformed into NGSI-LD entities and sent to Orion-LD for digital tracking and circularity assessment. The service takes as input an image of a rectangular leather board with white background and regions removed (cut pieces).

Using OpenCV-based image segmentation, it detects the difference between the removed (white) and remaining areas, computes their relative proportions, and generates an output that includes:

- A processed visualization of the original, remaining, and removed material regions.
- Statistics such as total usable area, percentage of leftover material, and bounding box coordinates.
- An exportable file (.csv) containing coordinates of the remaining area that can be further processed by the Leather Board Outline Tool to generate NGSI-LD entities.

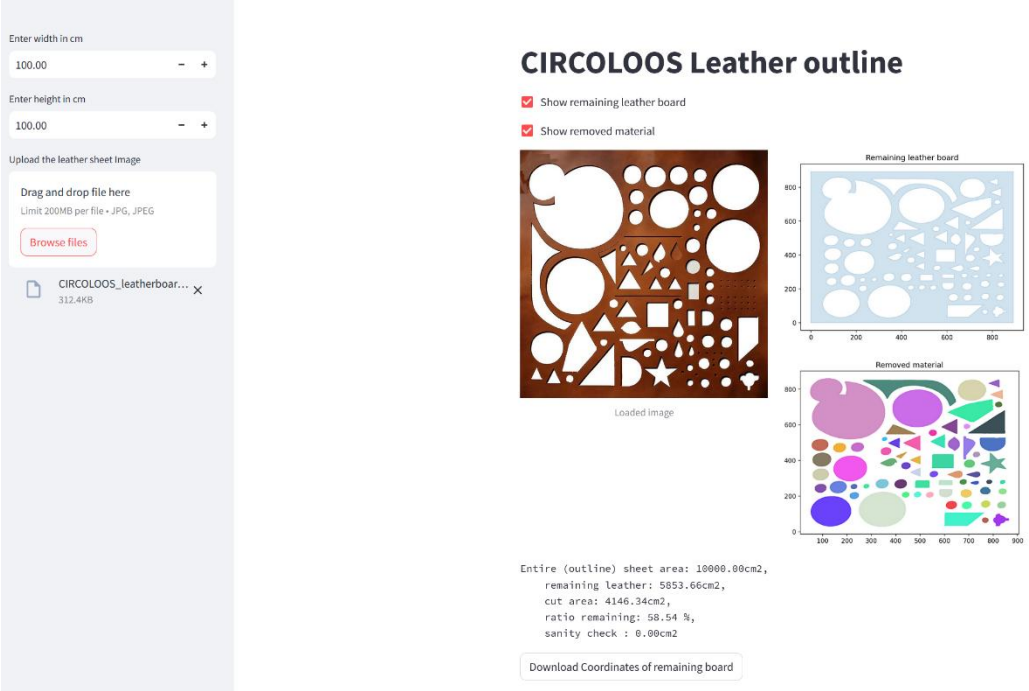

Figure 7 Leather Board tool

Steps (Figure 8):

1. Open http://localhost:8503.
2. Set the outside dimensions (width, height) of the leather board.
3. Upload a top-down image of the leather board. Removed/cut pieces must be white.

4. The tool shows the original image and highlights remaining vs. removed material with statistics.
5. Click “Download Coordinates of remaining board” to export a CSV with the coordinates (see leather_board_outline/leather_outline.csv).
6. The CSV can then be used with the previous tool to upload the data to Orion-LD.

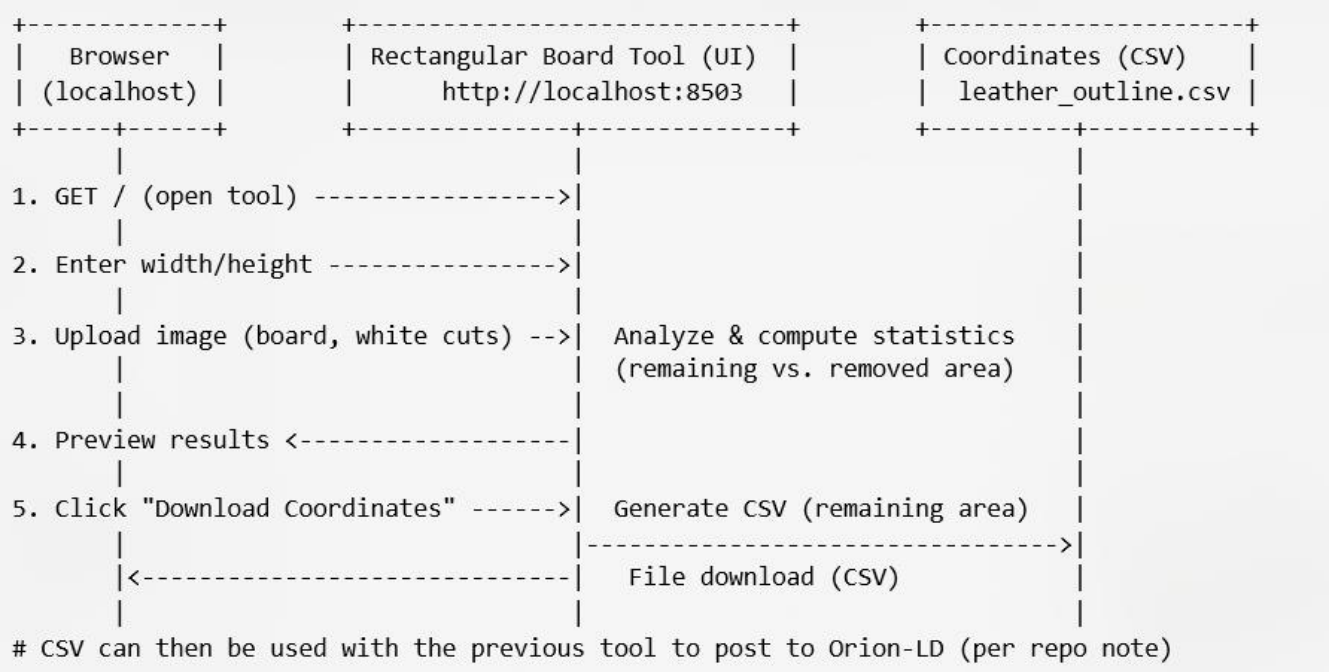

Figure 8 ASCII UML – From Image to Coordinates CSV

5 Conclusion

The first version of the CIRCULOOS Integrated Platform successfully delivers a fully functional, standards-based environment for managing, exchanging, and analyzing circular manufacturing data. By adopting NGSI-LD as the core communication and data modeling standard, and by deploying a complete suite of FIWARE components, the platform ensures interoperability, openness, and scalability across multiple domains and use cases. Through the development of the CIRCULOOS CDM, the platform provides a shared semantic foundation that unifies data from factories, processes, and digital tools.

This version demonstrates seamless integration of core functionalities, including data ingestion, real-time and historical data access, secure authentication and authorization, and the digital sale process, where transactions are represented as linked data entities. The current release implements a centralized deployment architecture, where all data interactions occur directly on the central Orion-LD broker. This configuration has been essential to validate system interoperability, streamline integration across pilots, and ensure reliable testing of the core features.

In the next iteration (D4.5 Final version of Integrated Platform), the focus will shift to extending the architecture with federation capabilities, enabling each manufacturing partner to operate its own local instance of the platform and securely share selected contextual information with the central system. This distributed model will reinforce data sovereignty, support cross-factory collaboration, and enable scalable, privacy-preserving integration across all pilot sites. The continuous evolution of the CIRCULOOS Platform will ensure its long-term sustainability, alignment with European data space principles, and readiness for future exploitation beyond the project's lifetime.

Appendix A Example of data exchange between tools (via NGSI-LD)

This appendix provides an example of data exchange between CIRCULOOS components through the NGSI-LD standard interfaces.
It demonstrates how information flows between different tools and services within the CIRCULOOS platform using the shared Common Data Model (CDM), the FIWARE context broker (Orion-LD), and supporting services such as Mintaka, Keycloak, and Kong.

The data exchange mechanisms described here are based on the implementations available in the CIRCULOOS Data Platform GitHub repository.

Overview

All tools integrated in the platform (e.g., Orchestrator, GRETA, SCDT, and other WP3 components) interact with the central Orion-LD Context Broker through standardized NGSI-LD REST APIs.
Each tool acts either as a data provider (creating or updating entities) or a data consumer (querying entities or subscribing to updates).

Data exchange is secured and managed through the Kong API Gateway, which validates authentication tokens issued by Keycloak.
Temporal data and time-series histories are accessed through Mintaka, which interfaces with TimescaleDB to store and retrieve temporal NGSI-LD data.

Example: Tool-to-Tool Data Exchange (via Orion-LD)

A simple example that exists in the repository is the exchange of product and transaction data — implemented under
/chain_of_transaction/ and /csv_NGSILD_Agent/.

Example Flow:

1. Tool A (CSV Agent) ⇢ converts tabular data (e.g., material or product records) into NGSI-LD entities using *csv_NGSILD_Agent/load_csv_ngsild.py.*

- Data is read from a CSV file (e.g., leatherProducts.csv).
- The agent automatically generates a valid NGSI-LD JSON payload, including attributes such as:

{

"id": "urn:ngsi-ld:leather:apm5zima95",

"type": "leather",

"leather_type": {"type": "Property", "value": "animal"},

"kind_of_animal": {"type": "Property", "value": "pig"},

"leather_type_tanned": {"type": "Property", "value": "chrome"},

"ownedBy": {"type": "Relationship", "object": "urn:ngsi-ld:Company:001"}

}

2. Tool A sends data to the central Orion-LD, via the NGSI-LD REST API: P*OST /ngsi-ld/v1/entityOperations/upsert*

The request passes through Kong, which verifies the JWT token issued by Keycloak, before Orion-LD accepts and stores the entity.

3. Tool B (Orchestrator or GRETA)retrieves the same entity for further processing or visualization by querying: *GET /ngsi-ld/v1/entities/{entityId}*

The response includes all relevant attributes and relationships, allowing the consuming tool to access up-to-date contextual data.

4. Temporal Data Retrieval (Optional) when a tool requires historical information (e.g., ownership or quality changes), it queries Mintaka through Kong: *GET /temporal/entities/urn:ngsi-ld:leather:apm5zima95*

Mintaka accesses TimescaleDB and returns all entity changes with timestamps.

Example Data Flow

Below is an ASCII UML sequence diagram illustrating this tool-to-tool exchange:

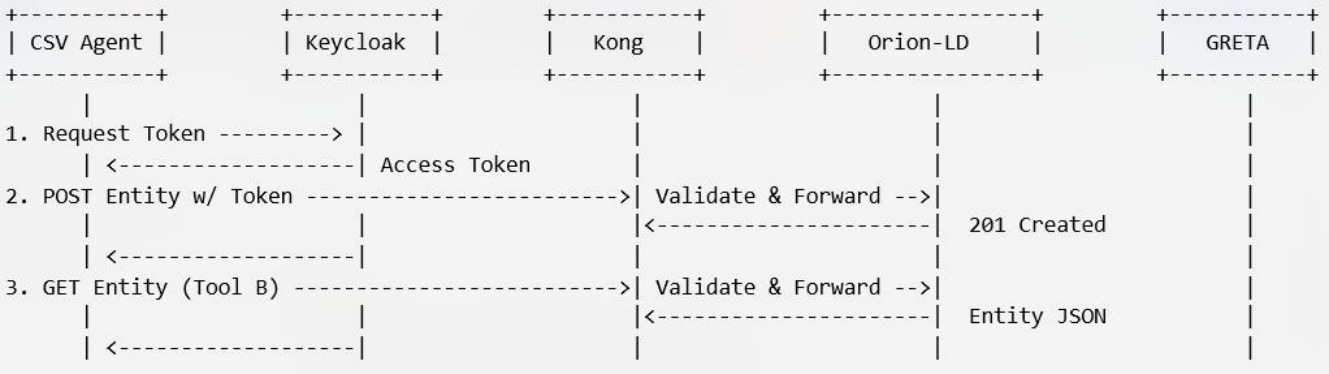

Figure 9 Sequence Diagram for the illustration of tool-tool exchange

Key Features Observed in the Repository

- Authentication and Authorization: Managed by Keycloak and enforced through Kong in all HTTP requests.
- Interoperability: All tools use the same NGSI-LD data model, ensuring seamless data exchange.
- Context Management: Orion-LD maintains real-time context for all entities, accessible through REST APIs.
- Temporal Storage: Historical data changes are captured by Mintaka and stored in TimescaleDB.
- Reusability: Each module (e.g., CSV Agent, chain_of_transaction) is containerized and deployable independently, allowing consistent use of the same APIs and CDM structure across pilots.

References (as in GitHub)

- csv_NGSILD_Agent/load_csv_ngsild.py
- csv_NGSILD_Agent/circuloos-csv-ngsild-agent.yml
- chain_of_transaction/orion_create_leather.json
- chain_of_transaction/orion_update_leather.json
- chain_of_transaction/orion_create_transaction.json

Appendix B Example of CSV + Orion Agent operation

The CSV-to-Orion-LD Agent is a lightweight custom service developed within CIRCULOOS to automate the conversion of tabular (CSV) datasets into fully compliant NGSI-LD entities, and to upload them to the Orion-LD Context Broker.
This agent is one of the core custom services described in the repository under */csv_NGSILD_Agent/*, and is responsible for connecting existing factory or pilot data to the CIRCULOOS platform without the need for custom integration scripts.

Overview

The CSV Agent reads a .csv file containing raw data, transforms each row into an NGSI-LD JSON object, and publishes it to the Orion-LD broker using authenticated REST calls.
This workflow supports manual or automated data ingestion for different domains (e.g., material batches, sensor readings, or production data).

The service can run locally through Docker Compose, as defined in the configuration file *circuloos-csv-ngsild-agent.yml.*

Core Components

Table 1 Core components of the Docker Compose file

File	Function
load_csv_ngsild.py	Main Python script handling CSV upload, transformation, and entity posting to Orion-LD.
csv_ngsild_agent_utils.py	Utility module that validates data structure, builds NGSI-LD JSON payloads, and manages timestamps.
circuloos-csv-ngsild-agent.yml	Docker Compose configuration defining service deployment, credentials, and environment variables.
leatherProducts.csv	Example input CSV file containing sample leather entities used in the demonstration.

CSV File Requirements

- Each input CSV must follow the column rules defined in load_csv_ngsild.py:
- The first two columns must be: ***id,type***
- All additional columns represent NGSI-LD attributes (Properties or Relationships).
- Optional column *observedat* allows inclusion of timestamps in ISO 8601 format (e.g., 2024-01-31T12:03:02Z). If omitted, the timestamp is automatically set to the current system time.

Example (*leatherProducts.csv*):

id	type	leather_type	kind_of_animal	observedat
urn:ngsi-ld:leather:001	leather	animal	pig	2024-08-02T09:26:35Z

Operation Workflow

Once the service is running (either via Docker or Python), the workflow is as follows:

1. Start the service: Launch locally using

docker compose -f circuloos-csv-ngsild-agent.yml up

or run manually:

python3 load_csv_ngsild.py

2. Access the Web UI:Open http://localhost:5000.
   The interface provides buttons for selecting, uploading, generating, and posting CSV data.
3. Select and upload a CSV file: Click "Browse…" → choose the dataset (e.g., *leatherProducts.csv)* → Upload.
4. Generate NGSI-LD entities: Press "Generate NGSI-LD entities." The agent converts the CSV into JSON-LD entities and displays them on screen.
5. Post entities to Orion-LD: Press "Post NGSI-LD entities to Orion-LD." The system sends authenticated POST requests (using credentials defined in *circuloos-csv-ngsild-agent.yml*) to the endpoint

http://<orion-host>:1026/ngsi-ld/v1/entityOperations/upsert

6. View confirmation: The UI displays the IDs of successfully posted entities and confirmation messages returned from Orion-LD.
7. Inspect uploaded data: Verify the entities on Orion-LD using the helper command included in the repo:

./getDataOrionSensors.sh leather

Authentication and Configuration

To send data to the official CIRCULOOS instance, update credentials in *circuloos-csv-ngsild-agent.yml*:

environment:

- USERNAME=<partner_username>

- PASSWORD=<partner_password>

Before redeploying, stop any existing containers:

./service.sh stop

Then relaunch the service:

```
docker compose -f circuloos-csv-ngsild-agent.yml up
```

Example of Generated NGSI-LD Output

Generated JSON entity (from leatherProducts.csv):

```
{
"id": "urn:ngsi-ld:leather:001",
"type": "leather",
"leather_type": {
"type": "Property",
"value": "animal"},
"kind_of_animal": {
"type": "Property",
"value": "pig"},
"observedAt": "2024-08-02T09:26:35Z"
}
```

This JSON is posted to the central Orion-LD Context Broker and becomes part of the shared contextual graph accessible to other platform tools (e.g., GRETA, SCDT, Orchestrator).

ASCII UML Sequence Diagram -CSV Agent Operation

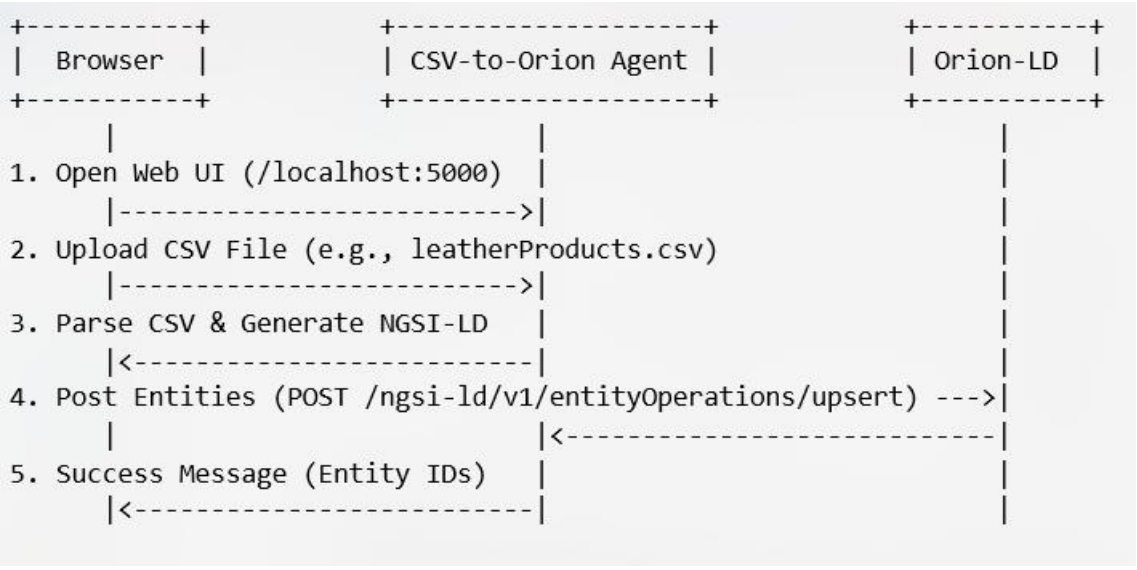

Figure 10 Sequence Diagram for uploading of csv files to Orion LD

Integration and Reuse

The generated CSV Agent output can be reused by other tools in the platform:

- Uploaded JSON entities become accessible via Orion-LD to other services (e.g., GRETA sustainability analysis or SCOPT optimization).
- Entities are compatible with the CIRCULOOS Common Data Model and use standardized contexts *(circuloos-context.jsonld)*.

References (from GitHub)

- csv_NGSILD_Agent/load_csv_ngsild.py
- csv_NGSILD_Agent/csv_ngsild_agent_utils.py
- csv_NGSILD_Agent/circuloos-csv-ngsild-agent.yml
- csv_NGSILD_Agent/leatherProducts.csv

CIRCULOOS

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101092295. The herewith information reflects only the author's view. The European Commission is not responsible for any use that may be made of the information herewith included.