UNIVERSITÀ DEGLI STUDI FIRENZE

DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

MOST CENTRO NAZIONALE PER LA MOBILITÀ SOSTENIBILE

SNAP4CITY

KM4CITY

Efficient and Scalable Semantic Data Ingestion for Smart City Digital Twin Platforms

Pierfrancesco Bellini, Enrico Collini, Marco Fanfani, Paolo Nesi, Christian Panconi

University of Florence, Florence, Italy
email: <name>.<surname>@unifi.it

DISIT lab, https://www.disit.org, https://www.snap4city.org

CISOSE 2025 (7/21-24 2025) Tucson, Arizona

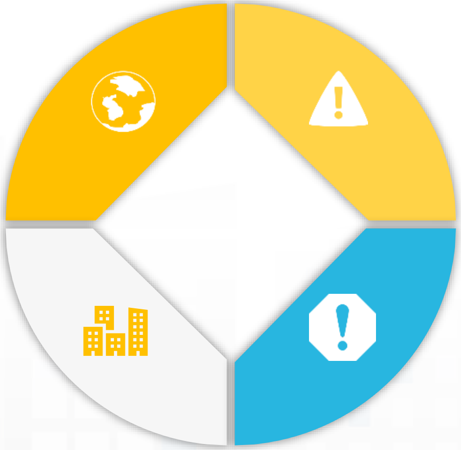
Semantic-Driven Smart City Digital Twin

GLOBAL CONTEXT

Modern cities require advanced tools for monitoring, simulation, and **decision-making**.

CHALLENGES

Real-time, heterogeneous IoT/IoE data demands **semantic modeling** and **efficient storage** solutions



DIGITAL TWINS

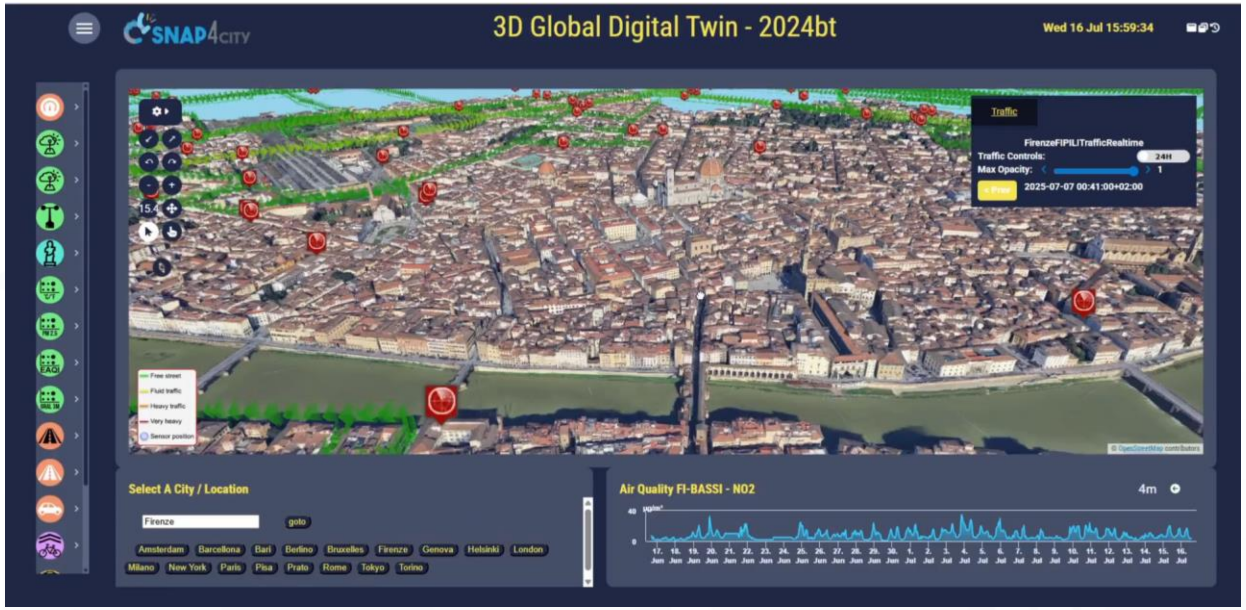
are emerging as key enablers for optimizing mobility, energy, environment, tourism.

PROBLEM

Traditional storage solutions **lack** semantic interoperability for advanced reasoning and consistency.

Snap4City SCDT Platform

Scalable and Semantic Digital Twin Infrastructure for Smart Cities



3D Global Digital Twin - 2024bt

Wed 16 Jul 15:59:34

FirenzeFIPILITrafficRealtime

Traffic Controls: 24H

Max Opacity: 1

2025-07-07 00:41:00+02:00

Select A City / Location

Firenze

Air Quality FI-BASSI - NO2

Snap4City enables the **Smart City Digital Twins** through a **hybrid data architecture**:

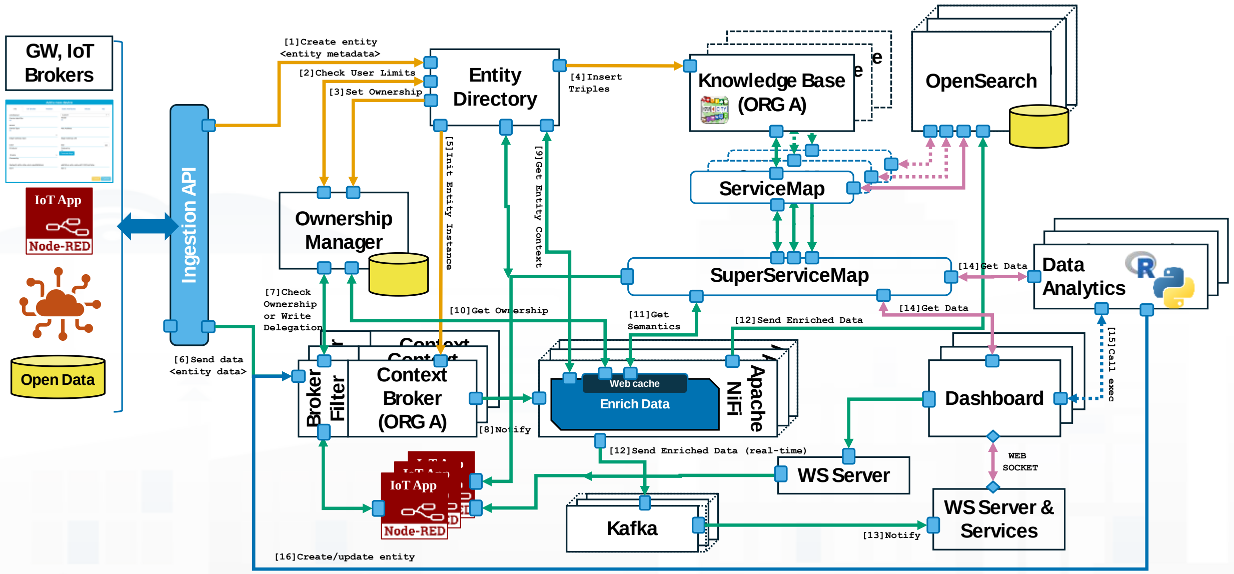
•**Knowledge Base (KB)** with **Km4City Ontology** for rich semantic modelling

•**NoSQL Storage (OpenSearch)** for scalable, real-time data ingestion

Tight integration ensures high-throughput and semantic consistency

Snap4City SCDT Platform

Scalable and Semantic Digital Twin Infrastructure for Smart Cities



synergic interaction between KB, based on the **Km4City** ontology, and NoSQL database, implemented with **OpenSearch**

End-to-End Architecture

Entity instantiation, semantic enrichment, data ingestion, indexing, and retrieval

Enrich Data Process

Semantic mapping of raw data, entity structure creation, Event-driven data forwarding to dashboards and storage

Query & Retrieval Strategies

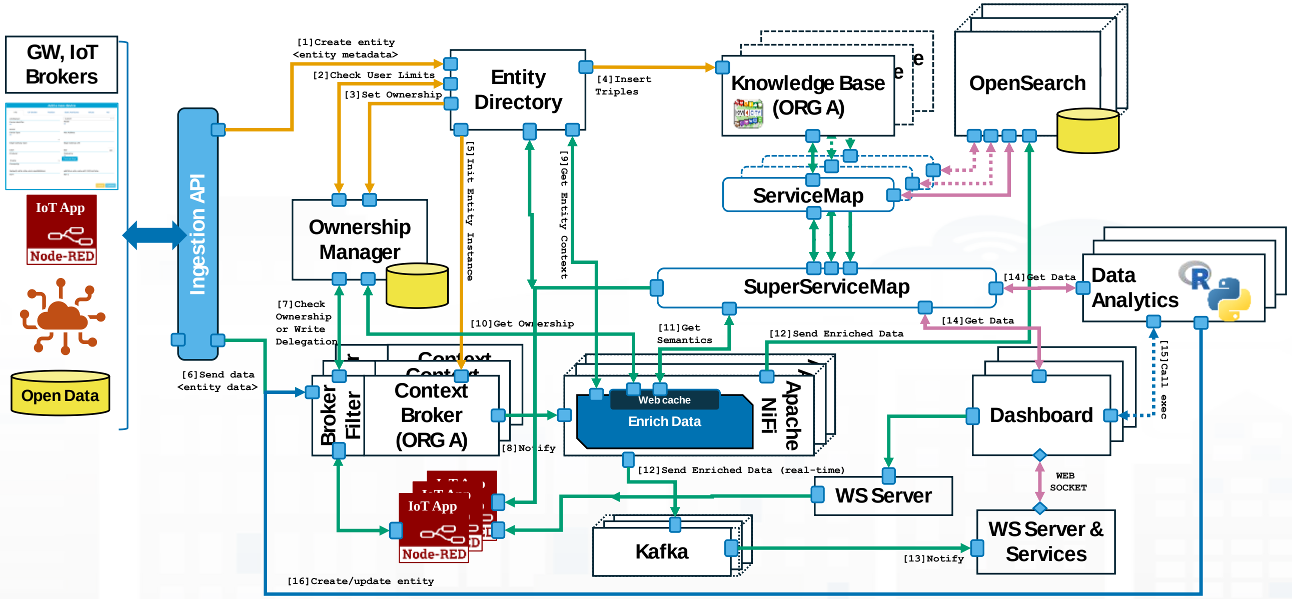
Persistent queries over NoSQL, Real-time updates via event-driven services, Integrated dashboard visualization

Scalability Validation

Ingestion latency and throughput evaluation. Real deployment volumes in large-scale smart city contexts

Architecture Overview

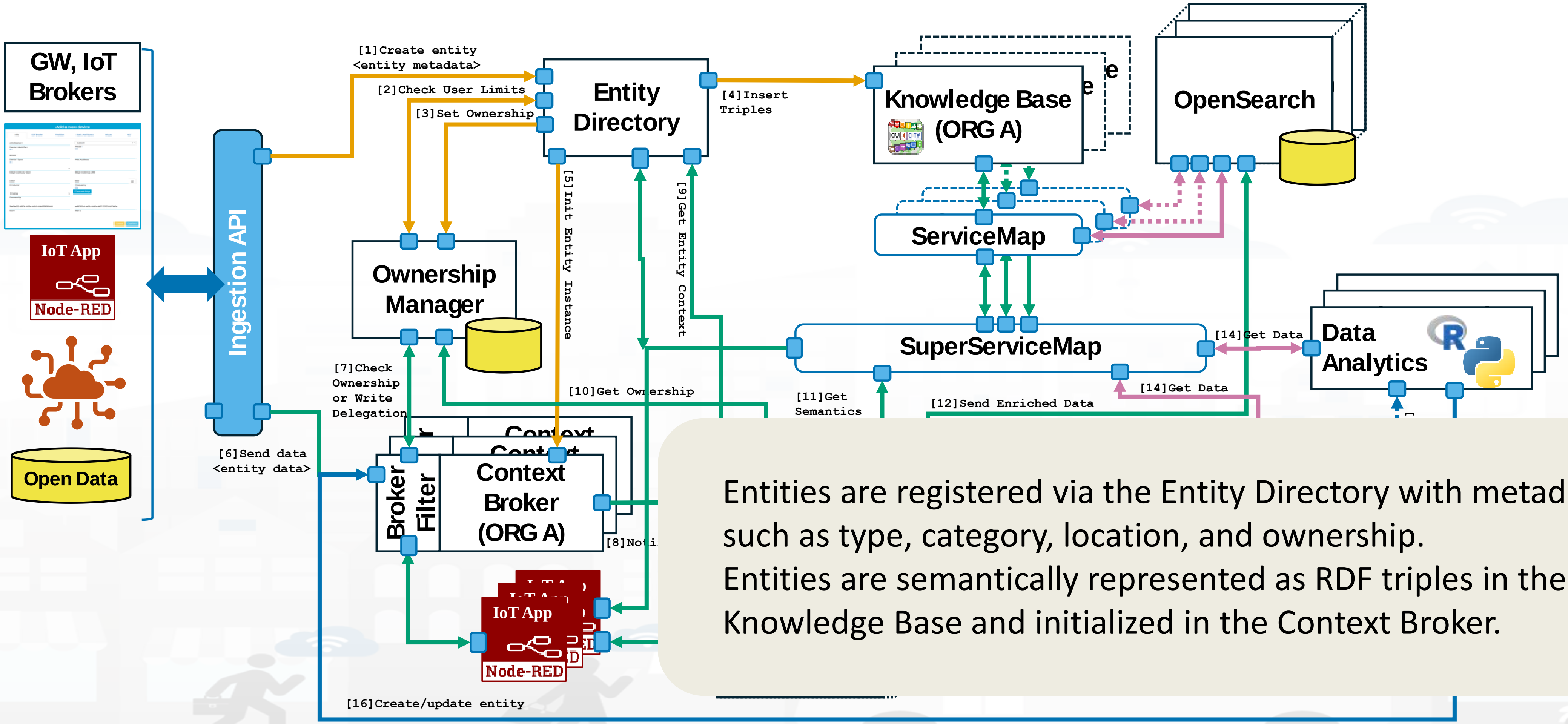
Snap4City is a modular, multi-tenant digital twin platform enabling semantic ingestion, real-time processing, and scalable storage of smart city data. Privacy, ownership, and interoperability are enforced across organizations and data sources.



Three Main Data Flows

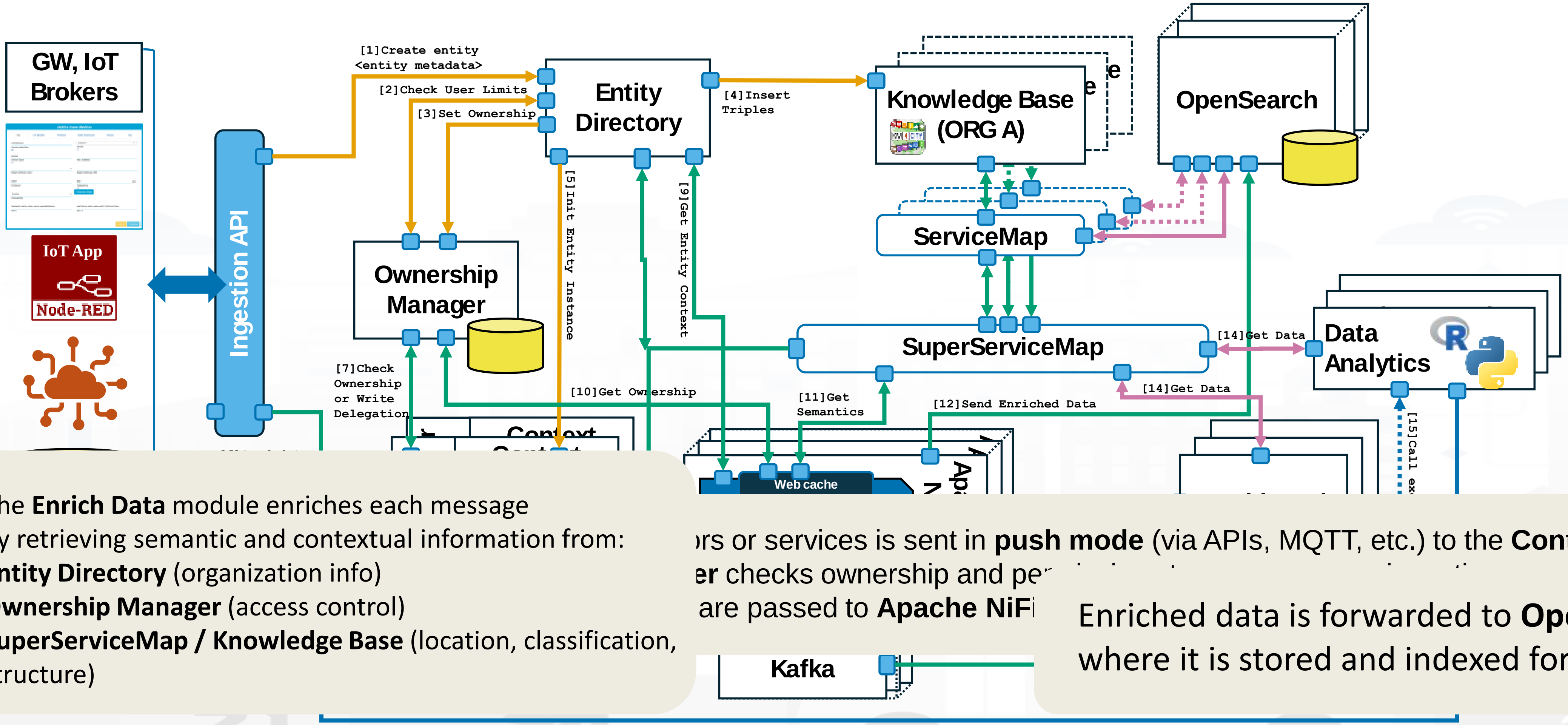
- **Entity Creation** → Entity metadata registered in Knowledge Base via Entity Directory and Context Broker (Orion).
- **Data Ingestion** → Messages pushed to Context Broker → Enriched (NiFi) with semantics & ownership info → Stored in OpenSearch.
- **Data Retrieval** → Dashboards/analytics retrieve data via pull (APIs) or push (Kafka + WebSocket for real-time updates).

ENTITY CREATION



Entities are registered via the Entity Directory with metadata such as type, category, location, and ownership.
Entities are semantically represented as RDF triples in the Knowledge Base and initialized in the Context Broker.

DATA INGESTION

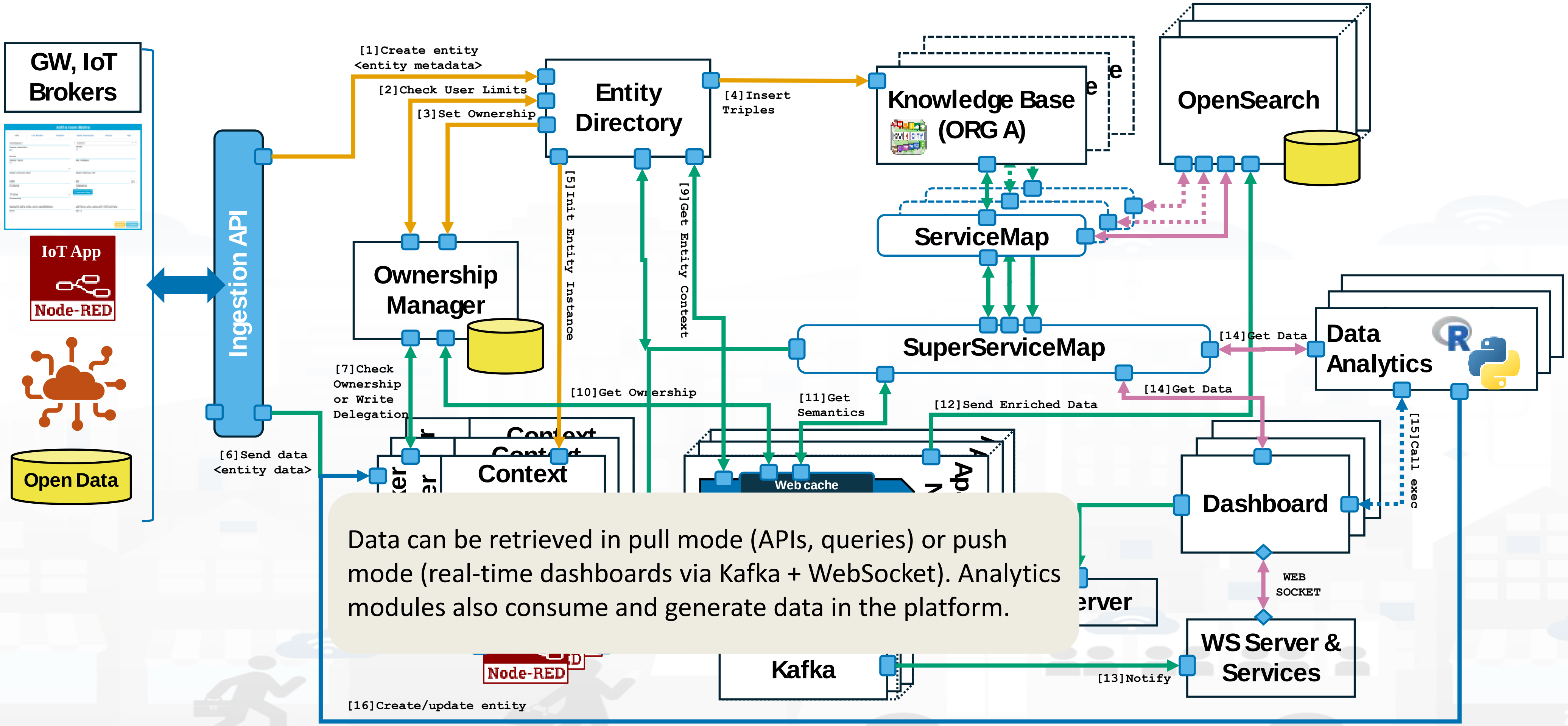


The **Enrich Data** module enriches each message by retrieving semantic and contextual information from:
Entity Directory (organization info)
Ownership Manager (access control)
SuperServiceMap / Knowledge Base (location, classification, structure)

ɪrs or services is sent in **push mode** (via APIs, MQTT, etc.) to the **Context Broker**.
er checks ownership and pe
are passed to **Apache NiFi**

Enriched data is forwarded to **OpenSearch**, where it is stored and indexed for retrieval.

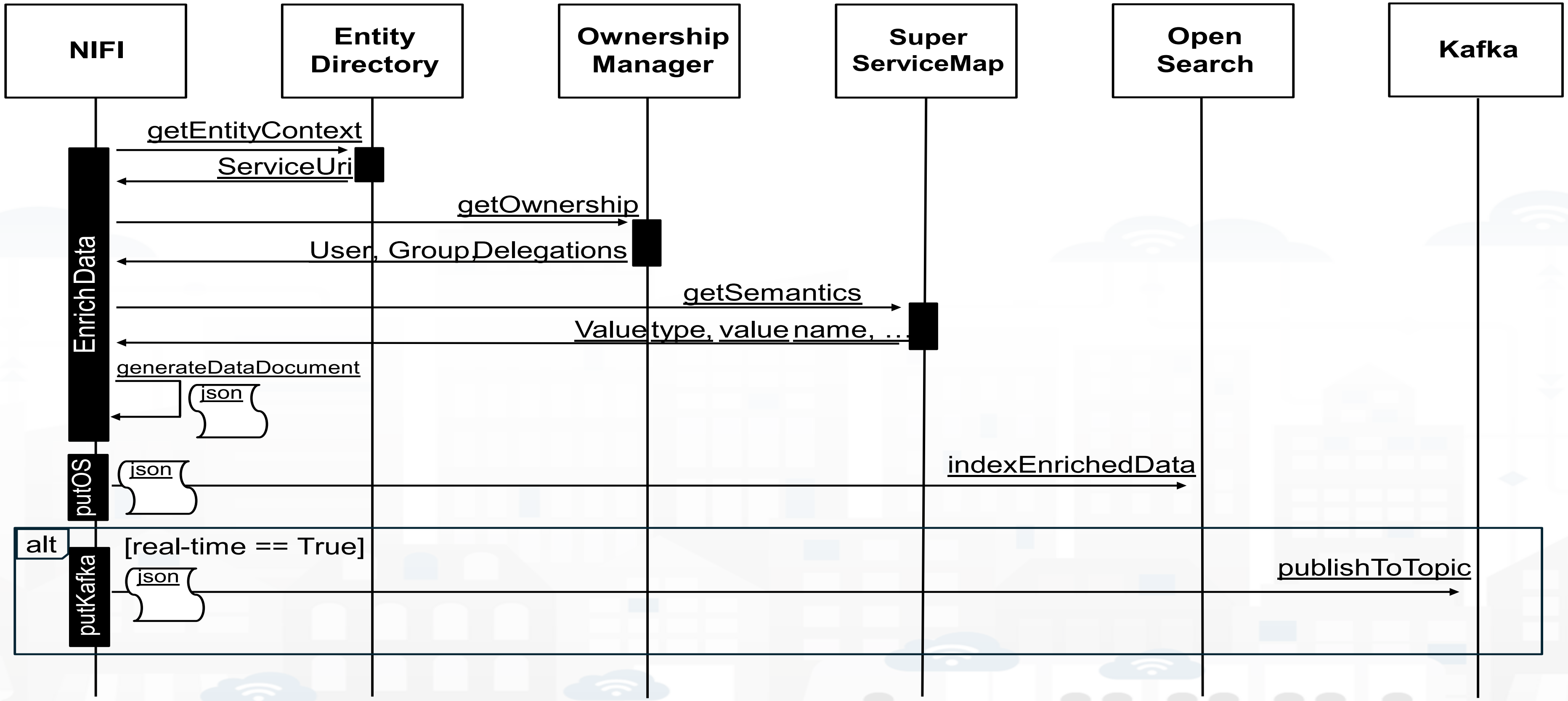
DATA RETRIEVAL



Data can be retrieved in pull mode (APIs, queries) or push mode (real-time dashboards via Kafka + WebSocket). Analytics modules also consume and generate data in the platform.

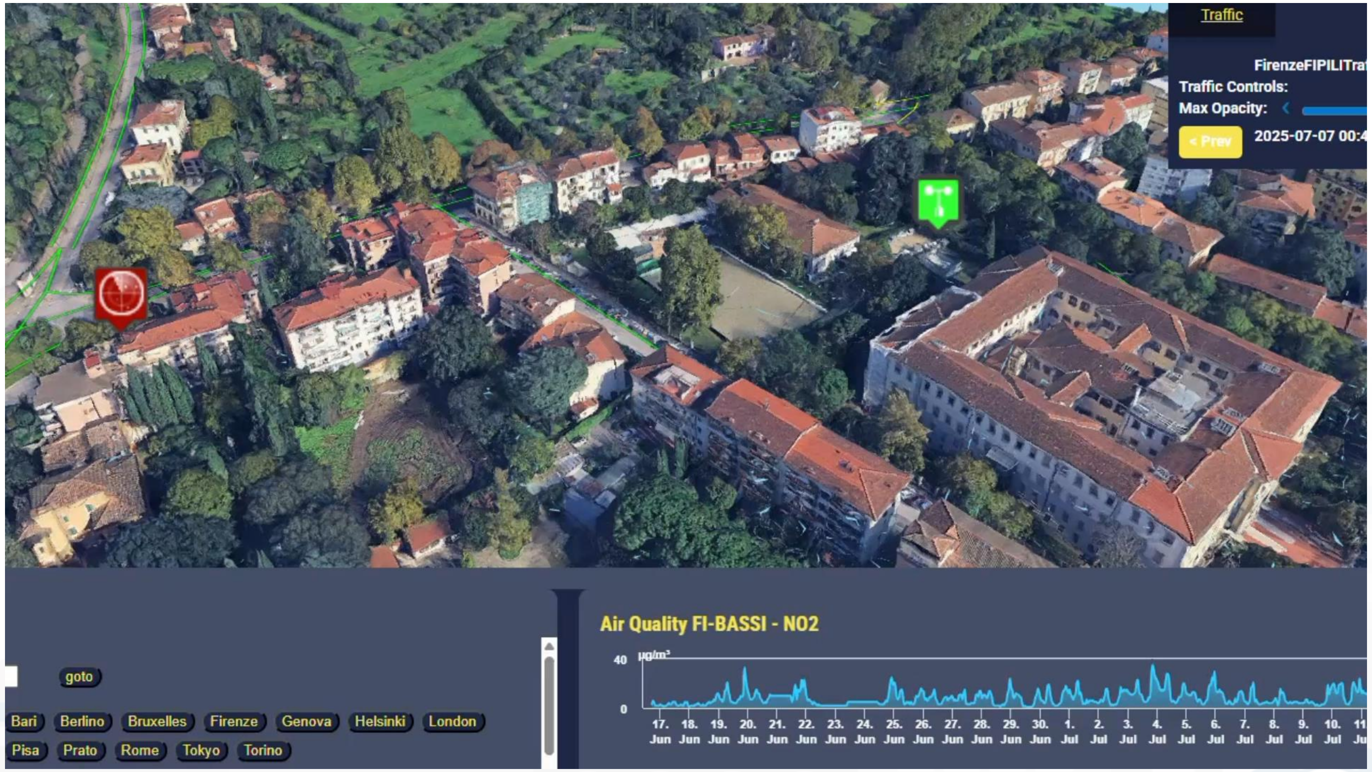
Ingestion Dataflow and Storage

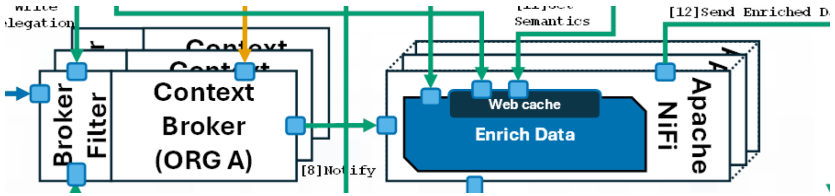
Upon receiving data, the Enrich Data processor enriches it with semantic and ownership metadata.
Data is then stored in OpenSearch (for time series) and optionally forwarded to Kafka for real-time updates.



Data ingestion example

A weather station sends values to Orion.
NiFi enriches the data with metadata and stores it in OpenSearch





A weather station measures **Benzene, NO2, SO2** sends updates to **Orion**.

Orion sends an **HTTP notification** to **Apache NiFi**.
The message includes:

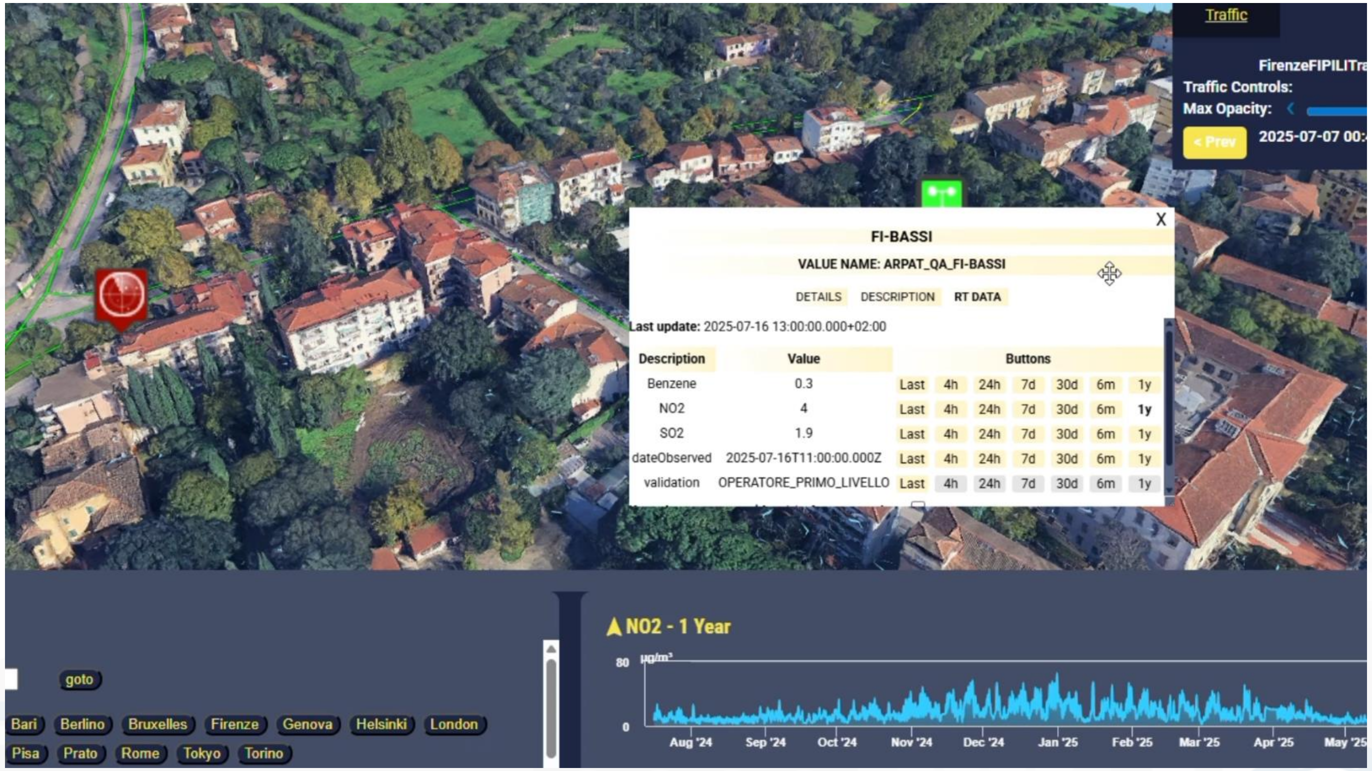
- subscriptionId (unique to Orion)
- data with variable values and device name

NiFi wraps this data into a **flow file**, which:

- Contains the payload (content)
- Uses flow file **attributes** to attach metadata

Data ingestion example

A weather station sends values to Orion.
NiFi enriches the data with metadata and stores it in OpenSearch



FI-BASSI

VALUE NAME: ARPAT_QA_FI-BASSI

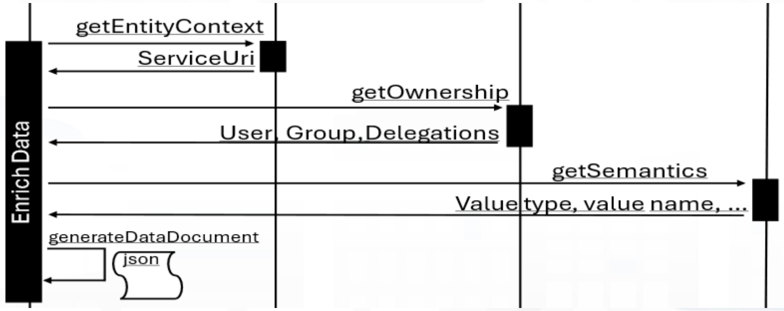
DETAILS | DESCRIPTION | RT DATA

Last update: 2025-07-16 13:00:00.000+02:00

Description	Value	Buttons						
Benzene	0.3	Last	4h	24h	7d	30d	6m	1y
NO2	4	Last	4h	24h	7d	30d	6m	1y
SO2	1.9	Last	4h	24h	7d	30d	6m	1y
dateObserved	2025-07-16T11:00:00.000Z	Last	4h	24h	7d	30d	6m	1y
validation	OPERATORE_PRIMO_LIVELLO	Last	4h	24h	7d	30d	6m	1y

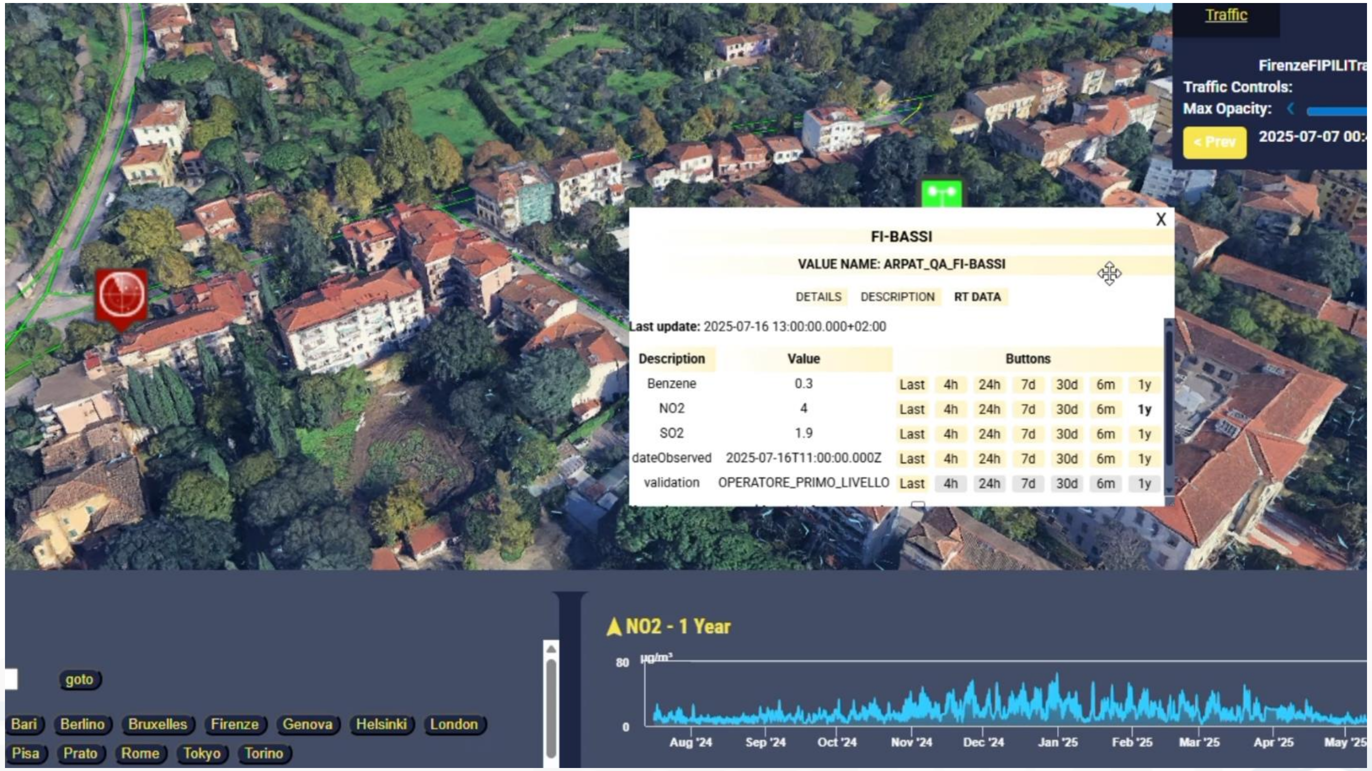
- **Enrich Data** processor

Retrieves serviceUri from the Entity Directory
looks up ownership and semantic metadata
Merges everything using configured strategies
(e.g., Celsius → temperature, linked to location, etc.)



Data ingestion example

A weather station sends values to Orion.
NiFi enriches the data with metadata and stores it in OpenSearch



FI-BASSI

VALUE NAME: ARPAT_QA_FI-BASSI

Last update: 2025-07-16 13:00:00.000+02:00

Description	Value	Buttons						
Benzene	0.3	Last	4h	24h	7d	30d	6m	1y
NO2	4	Last	4h	24h	7d	30d	6m	1y
SO2	1.9	Last	4h	24h	7d	30d	6m	1y
dateObserved	2025-07-16T11:00:00.000Z	Last	4h	24h	7d	30d	6m	1y
validation	OPERATORE_PRIMO_LIVELLO	Last	4h	24h	7d	30d	6m	1y

Final outputs:
One JSON with all values
(e.g., Benzene + NO2+ SO2) → full state at time t.

Three separate JSONs, one per variable
→ optimized for querying time series of single measures

These outputs are stored in **OpenSearch** and, if needed, sent to **Kafka topics** for event-driven dashboards.

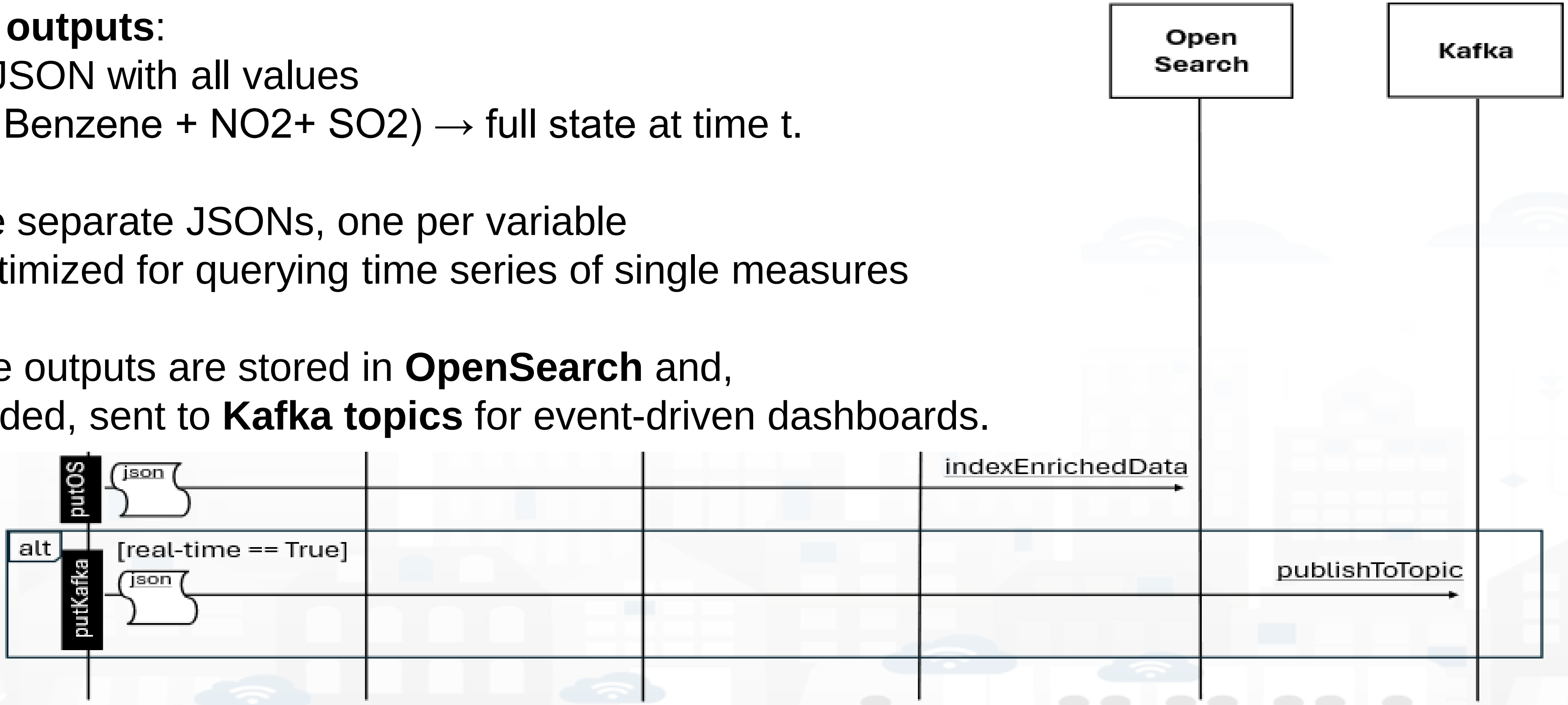
Data ingestion example

A weather station sends values to Orion.
NiFi enriches the data with metadata and stores it in OpenSearch

Final outputs:
One JSON with all values
(e.g., Benzene + NO2+ SO2) → full state at time t.

Three separate JSONs, one per variable
→ optimized for querying time series of single measures

These outputs are stored in **OpenSearch** and,
if needed, sent to **Kafka topics** for event-driven dashboards.

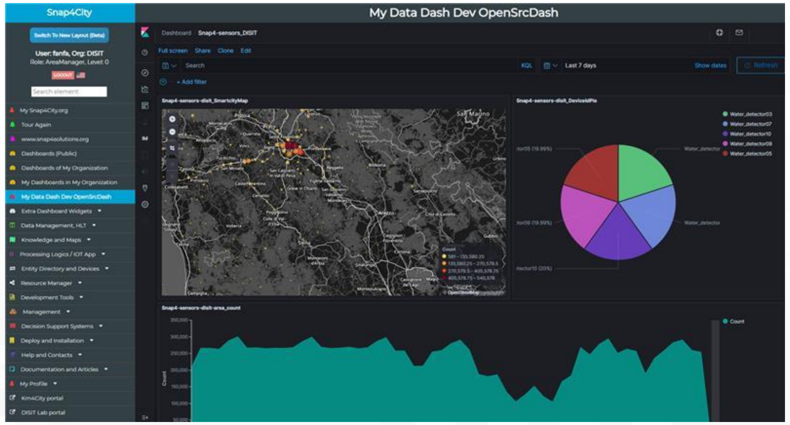


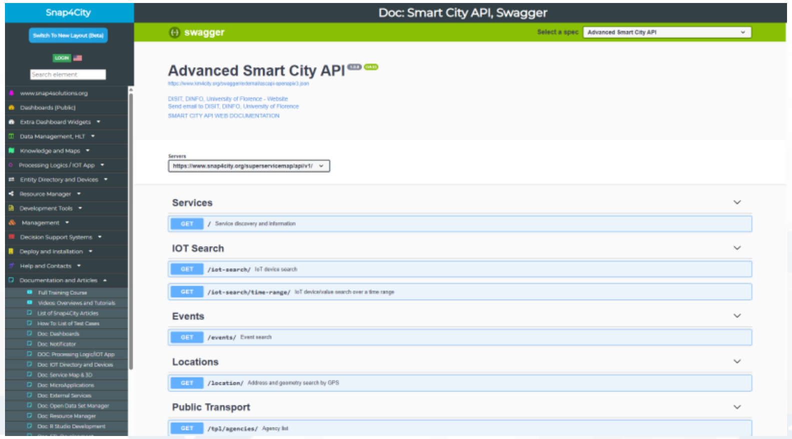
Data Retrieval

Two main methods are available for retrieving stored data:

OpenSearch Dashboards
(visualization)

REST APIs via SuperServiceMap
(semantic queries)





Doc: Smart City API, Swagger

Advanced Smart City API

https://www.km4city.org/swagger/external/ascapi-openapi3.json

DISIT, DINFO, University of Florence - Website
Send email to DISIT, DINFO, University of Florence
SMART CITY API WEB DOCUMENTATION

Servers
https://www.snap4city.org/superservicemap/api/v1/

Services
GET / Service discovery and information

IOT Search
GET /iot-search/ IoT device search
GET /iot-search/time-range/ IoT device/value search over a time range

Events
GET /events/ Event search

Locations
GET /location/ Address and geometry search by GPS

Public Transport
GET /tpl/agencies/ Agency list

SCAPI

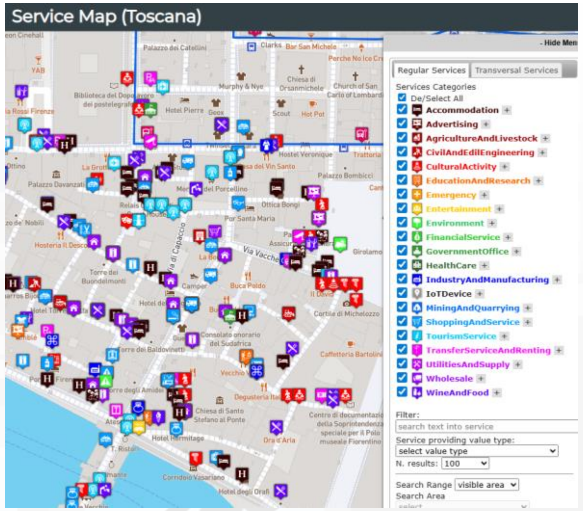
SuperServiceMap API allows to perform semantic, relational, temporal and geographical queries

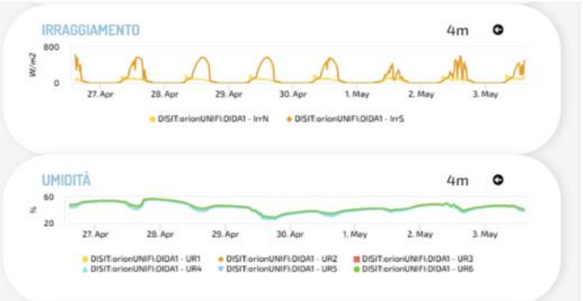
A resource can be retrieved using its unique resource identifier (**Service URI**), allowing for direct access to the specific resource and its semantic information

Entities can be also searched by considering **geographical positions**, and by specifying entity **categories/subcategories** or data models

Additionally, entity data can be obtained using specific **time ranges** (e.g., fromTime/toTime)

This allows for more specific queries, such as retrieving data for a certain period, in specific areas, and by categories.





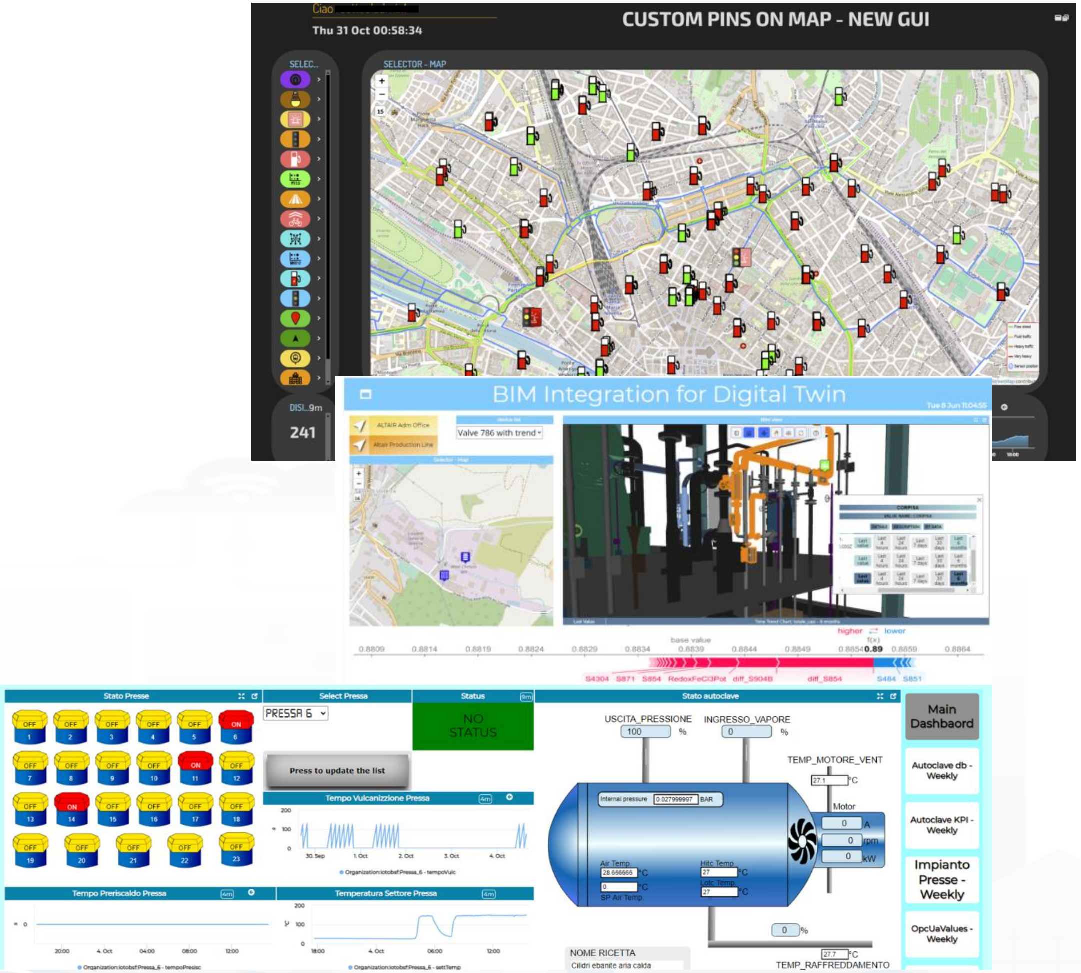
Real-time Notification

APACHE kafka Synoptics

Kafka enables real-time updates of dashboards via WebSockets.
Used for SVG synoptics, live status, and sensor value displays.

↓

When data changes or new events are detected, Kafka sends a message to the subscribed synoptic based on the specific topic, which then automatically updates the new information in the dashboard exploiting web sockets for fast transmission.



Critical monitoring (e.g., mobility, safety, environment)
User dashboards requiring immediate feedback
Interactive visualizations on maps, control panels, and alert systems

Data Ingestion Performance Evaluation

The main Snap4City deployment at the DISIT Lab (University of Florence

Handles 58,000+ entities across 22 organizations

Avg. rate: 200 events/min (peaks: 500/min)

Ingestion latency: avg. 900ms

Including the enrichment by the **Enrich Data** processor

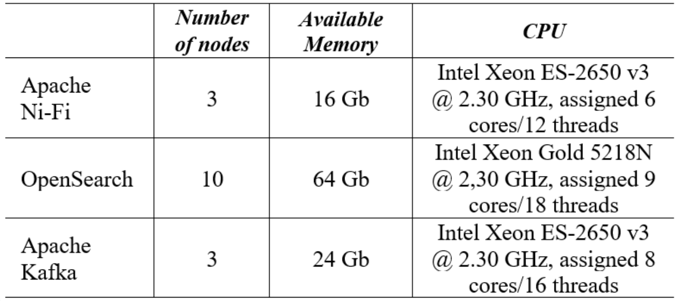
- Access control and semantic metadata

The indexing rate of the OpenSearch cluster has also been measured.

Avg. indexing rate: 70 documents per second, (peaks 400/s)

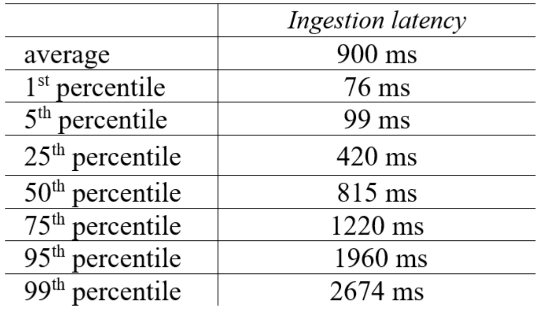
The significantly higher number of indexed documents compared to events is attributed to the storage of multiple representations per event but significantly enhances **query efficiency.**

TABLE I. DISIT LAB SNAP4CITY DEPLOYMENT SPECIFICATIONS



	Number of nodes	*Available Memory*	*CPU*
Apache Ni-Fi	3	16 Gb	Intel Xeon ES-2650 v3 @ 2.30 GHz, assigned 6 cores/12 threads
OpenSearch	10	64 Gb	Intel Xeon Gold 5218N @ 2,30 GHz, assigned 9 cores/18 threads
Apache Kafka	3	24 Gb	Intel Xeon ES-2650 v3 @ 2.30 GHz, assigned 8 cores/16 threads

TABLE II. MEASURED INGESTION LATENCY DISTRIBUTION



	Ingestion latency
average	900 ms
1st percentile	76 ms
5th percentile	99 ms
25th percentile	420 ms
50th percentile	815 ms
75th percentile	1220 ms
95th percentile	1960 ms
99th percentile	2674 ms

Conclusions

- In this paper, we introduced a **scalable and efficient semantic data ingestion architecture** tailored for Smart City Digital Twin platforms, specifically within the Snap4City framework.
- The key innovation is the **Ni-Fi-based Enrich Data process**, which augments incoming data with semantic, contextual, and access-control metadata. This hybrid architecture bridges **semantic expressiveness** (ontologies + KBs) with **NoSQL performance** (OpenSearch, Kafka), supporting both **real-time visualization** and **historical querying**.
- Our large-scale deployment handles:
- **58,000+ entities**
- **300,000+ variables**
- **Billions of messages** over time
  with sustained high ingestion rates and low latency.
- The architecture has been **validated in production** through major initiatives like **CN MOST** and **OPTIFaaS**, proving its robustness across domains like urban mobility, energy, and environment.

Efficient and Scalable Semantic Data Ingestion for Smart City Digital Twin Platforms

Thanks for your attention

Pierfrancesco Bellini ,Enrico Collini, Marco Fanfani, Paolo Nesi, Christian Panconi

University of Florence, Florence, Italy
email: <name>.<surname>@unifi.it

DISIT lab, https://www.disit.org, https://www.snap4city.org