

istSOS⁴Things

FAIR & Open Source IoT platform for Open Science.

Massimiliano Cannata, Daniele Strigaro, Claudio Primerano

Dipartimento Ambiente Costruzione e Design, SUPSI

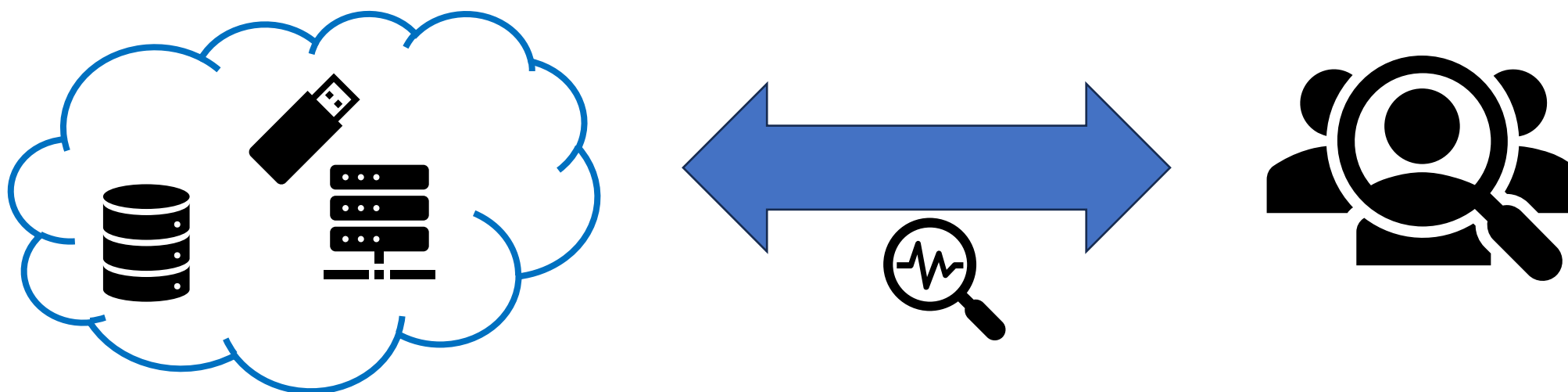


15-18 December 2024



This work is licensed under a
[Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

Interoperability and data service ?



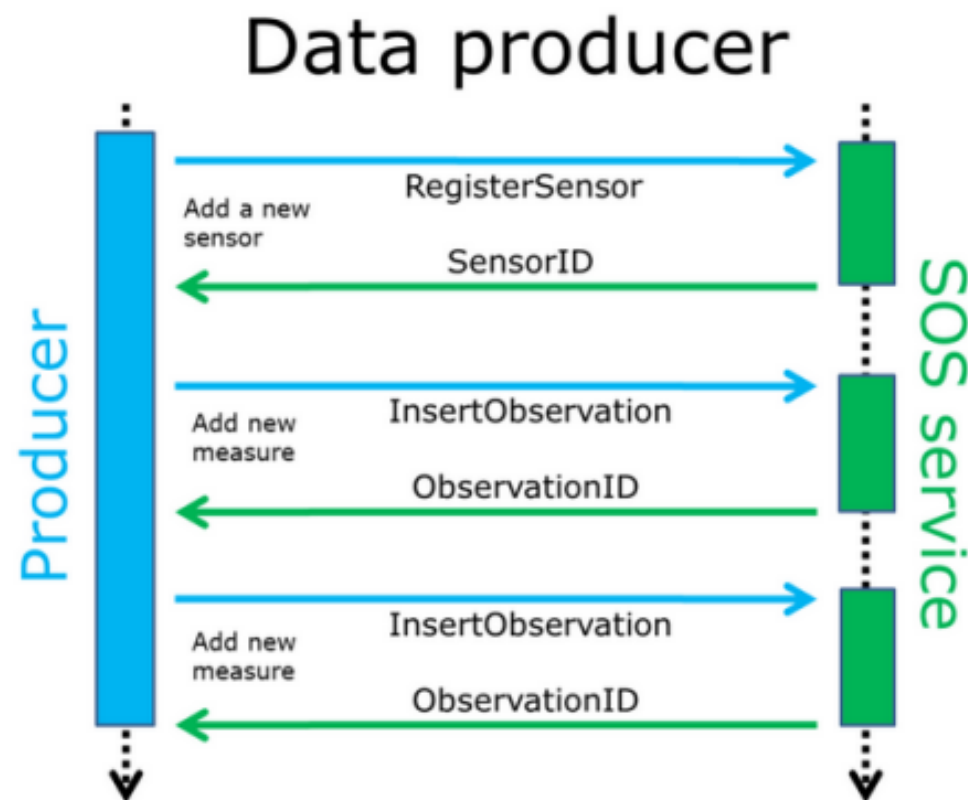
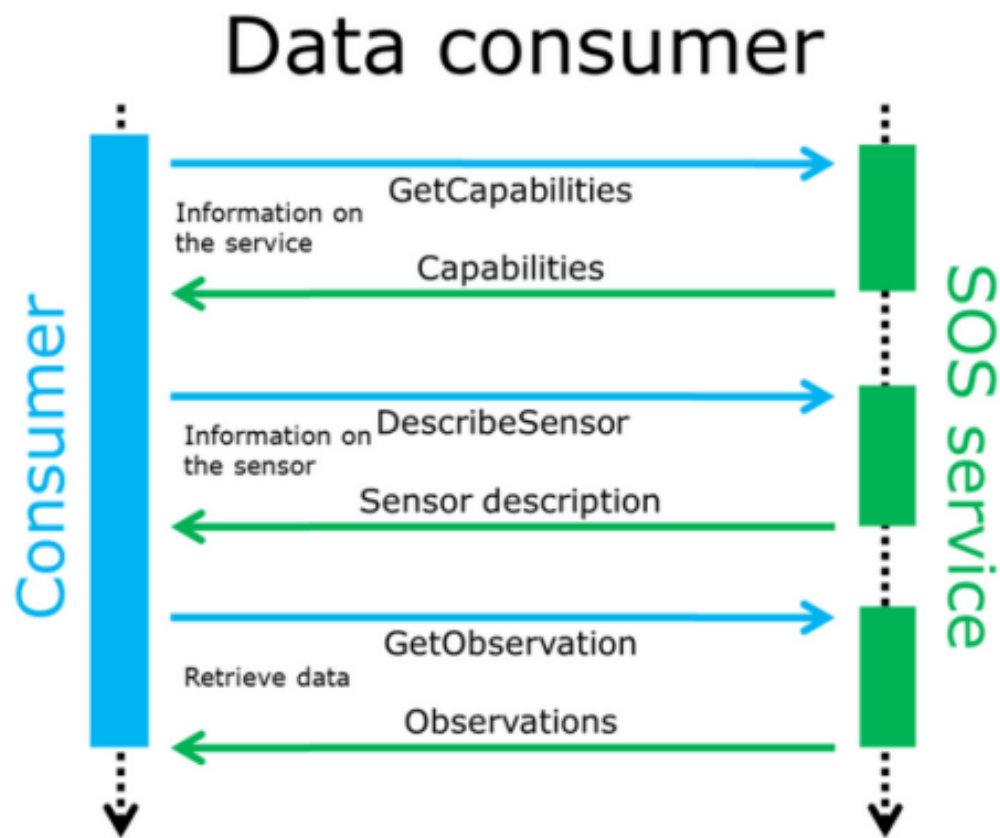
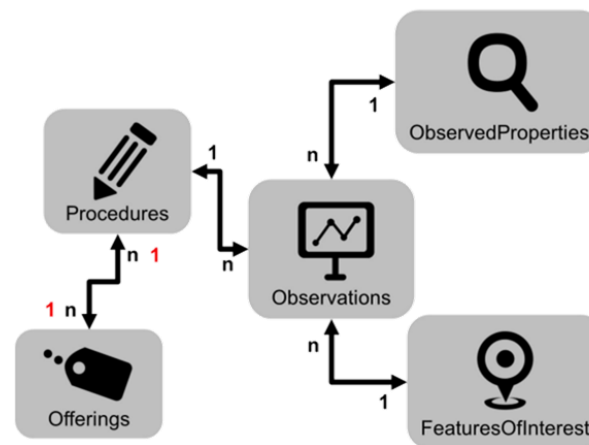
istSOS¹

In 2010 to provide a system to manage, alarm and protect the population and the economic assets of the Locarno region based on a hydrological model feed by data coming from 5 different monitoring networks belonging to two countries, we decided to implement an interoperable data service.

What we call now a digital twin?



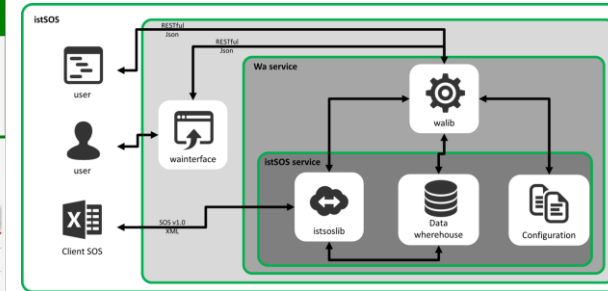
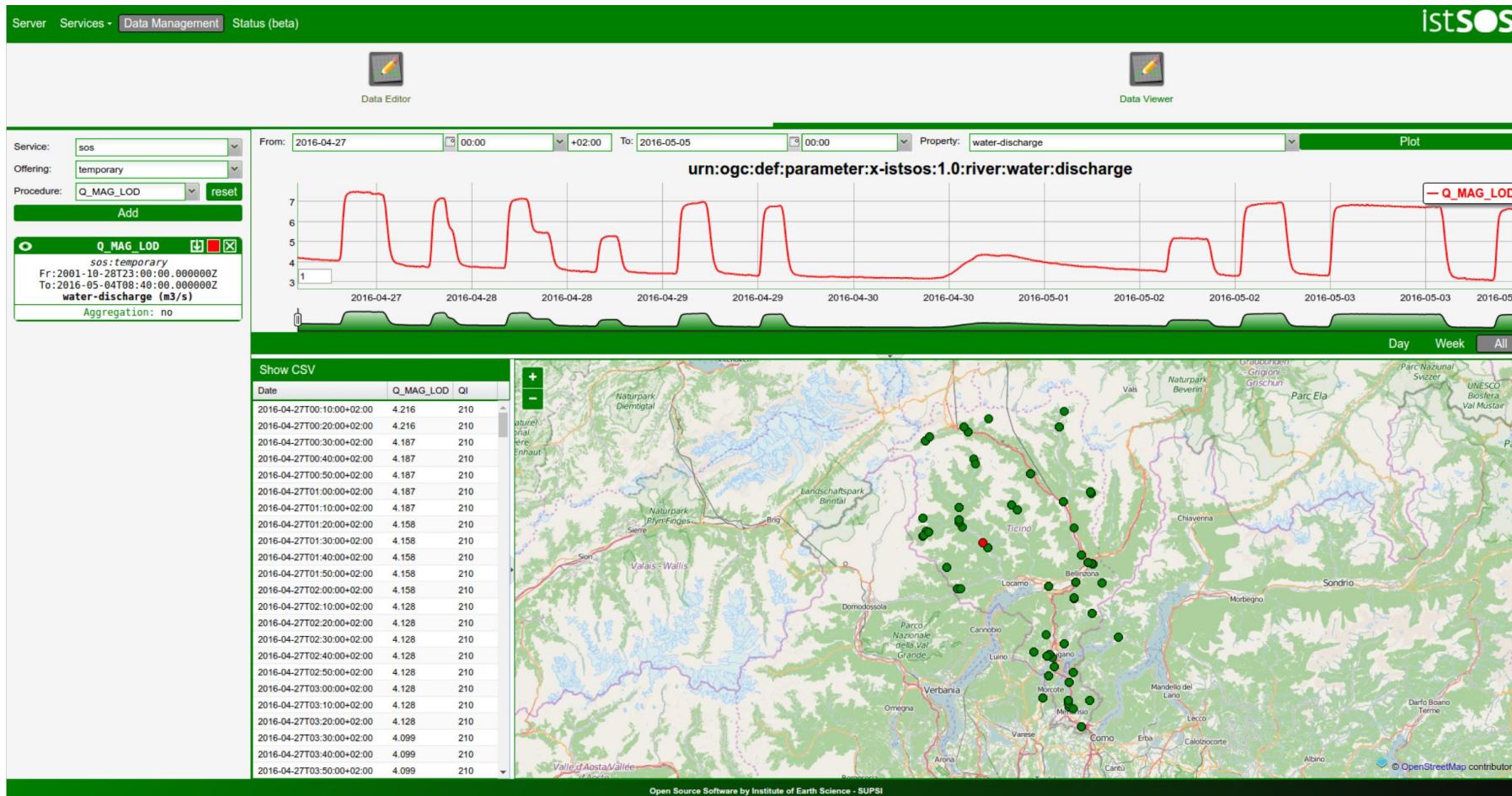
The OGC[®] Sensor Observation Service



istSOS²

Python implementation

A software compliant with the OGC® Sensor Observation Service with extended features and RESTful API to manage sensor data



istSOS 2.0 documentation

Welcome to istSOS-project

Free and Open Source Sensor Observation Service Data Management System

istSOS²

istSOS is an OGC SOS server implementation written in Python. istSOS allows for managing and dispatch observations from monitoring sensors according to the Sensor Observation Service standard.

The project provides also a Graphical user interface that allows for easing the daily operations and a RESTful Web api for automatizing administration procedures.

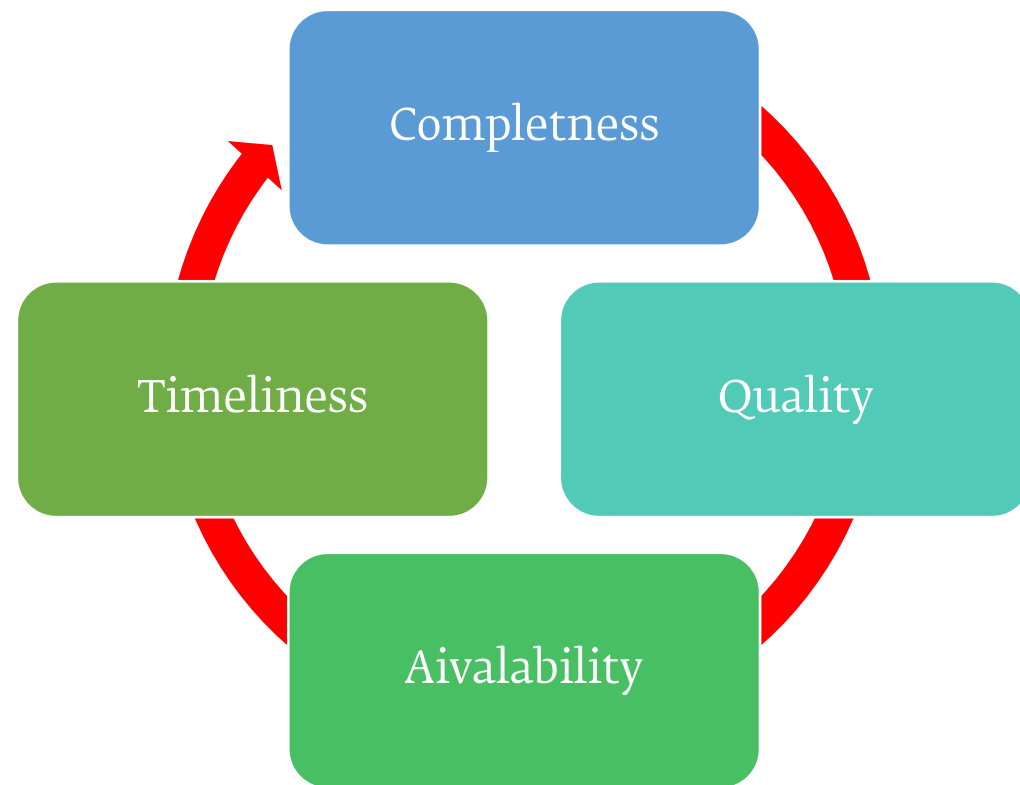
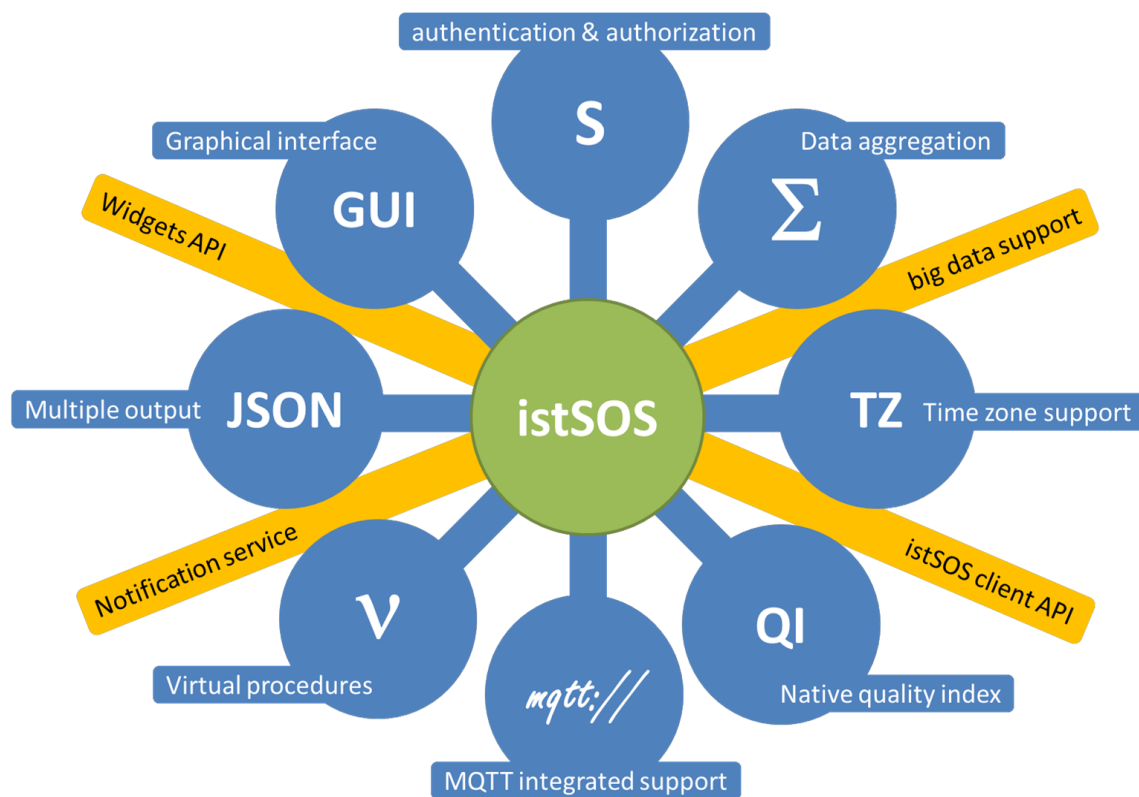
istSOS is released under the GPL License, and runs on all major platforms (Windows, Linux, Mac OS X), even though tests were conducted under a linux environment.

Web sections:

- Introduction to the standard
- istSOS
- Example of SOS requests
- Tutorial using istSOS
- istSOS Package

Data Science integration

Data curation and quality of service



Supporting disaster management

Lake Flooding Early Warning System

Based on a SOA (Service Oriented Architecture) and the Open Geospatial Consortium Standards

istSOS

FREEWAT

Open source water resource management platform

istSOS

The Canton Ticino HydroMet @ SUPSI

Configuration

669 registered sensors
15 observed properties
85 Mio served requests in 1Y
1 internal server error response (500)

49 years of data (1970-2019)
147 Mio registered observations
190 GB of database

99.902% Availability
9 hours Downtime

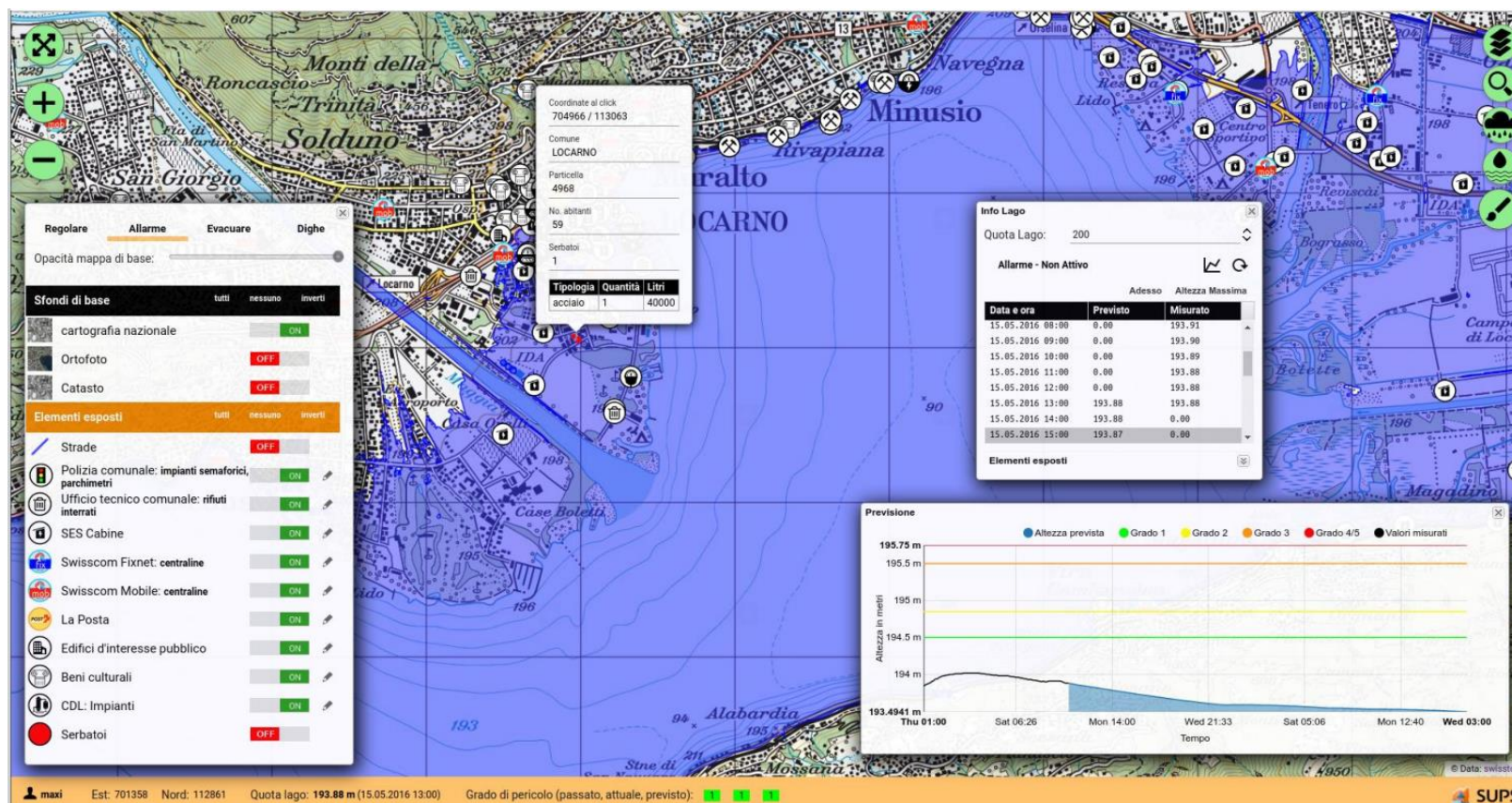
istSOS

SIMILE

Integrated Monitoring System of Sub Alpine Lakes quality

istSOS

interreg



MIARIA - Adaptive Hydrogeological Monitoring in support of the Alpine Integrated Risk plan

Dynamically identify impacts of impending scenarios and rise alerts

istSOS

interreg

ENORASIS: every drop counts

ENORASIS is a FP7-ENV project with the objective of develop an integrated decision support system for environmentally optimized and thus sustainable irrigation management for farmers and water management organizations based on advanced technologies and models.

istSOS

ALBIS

Tiger mosquito habitat monitoring

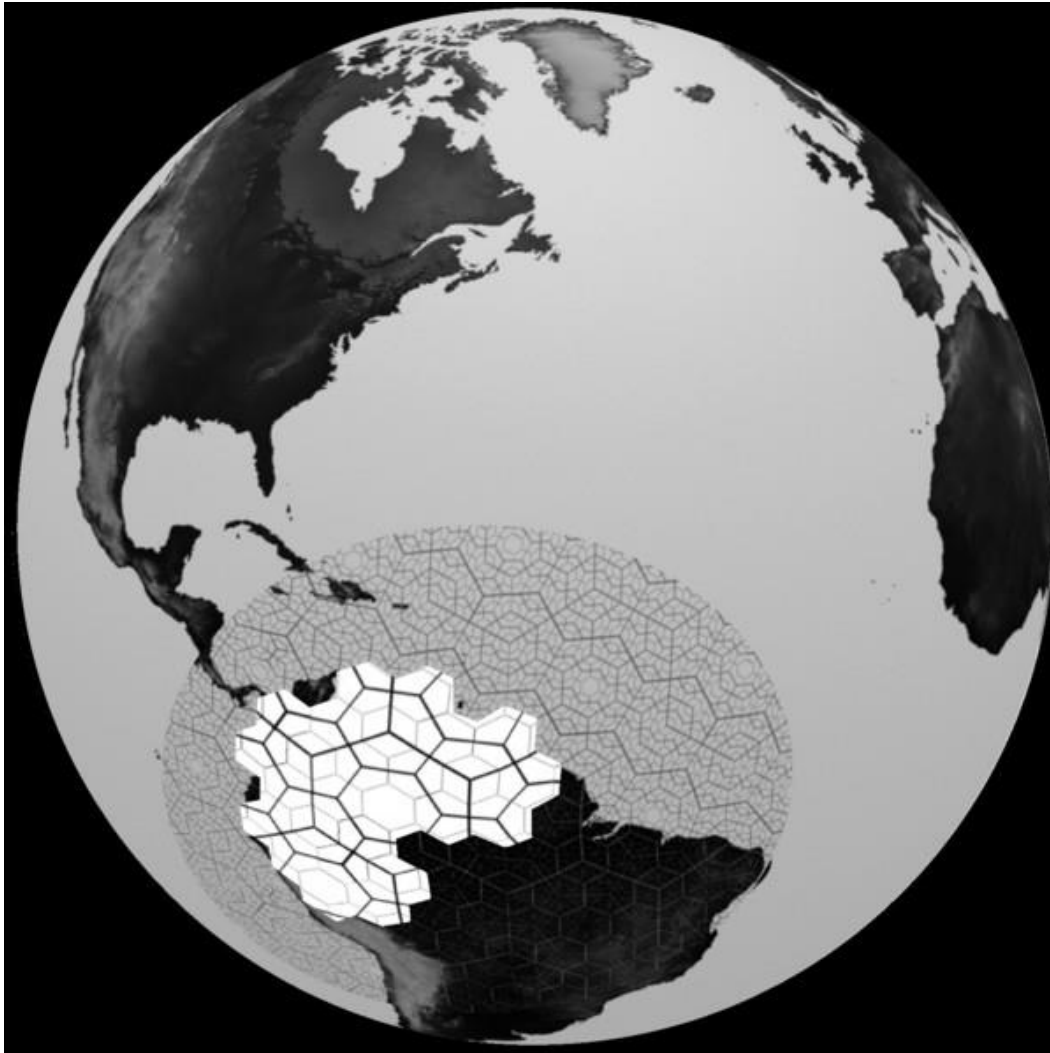
istSOS

4nse

Analysis of 4 times Open Non-conventional system for Serving the Environment. Empower environmental monitoring in developing countries using non-conventional and fully open solution.

istSOS

Digital Earth (Al Gore, 1998)



"Digital Earth... A multi-resolution, three-dimensional representation of the planet, into which we can embed vast quantities of geo-referenced data... Imagine a young child going to a Digital Earth... she sees Earth as it appears from space... she is interested in exploring... through a 3-D visualization of the terrain... land cover, distribution of plant and animal species, realtime weather, roads, political boundaries, and population. She can also visualize the environmental information that she and other students all over the world have collected... seamlessly fused... shared as part of a globe."

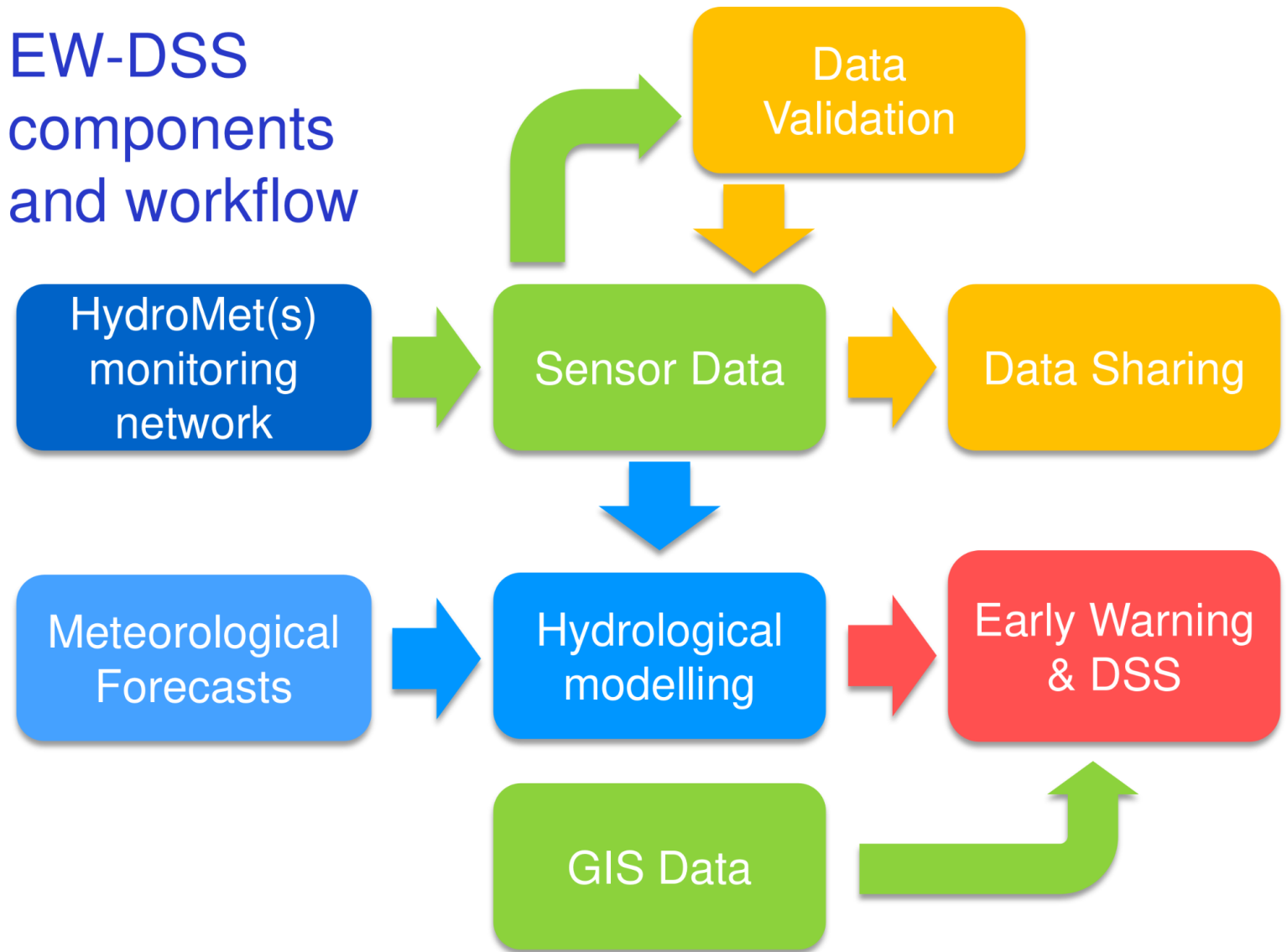
Sensor Data

Satellite Data

GIS Data (SDI)

Modelling

EW-DSS
components
and workflow



Things has changed

Romagna maggio 2023, settembre 2024



Image: <https://teleromagna.it/it/attualita/2024/5/7/romagna-alluvione-un-anno-dopo-le-iniziative-in-ricordo-della-sciagura>



Image: <https://www.snpambiente.it/snpa/arpae-emilia-romagna/alluvione-19-settembre-in-emilia-romagna-unanalisi-preliminare/>

Valle Maggia 30.06.2024



Image: https://youtu.be/q7xezclKqLc?si=PcOIPE2cMU1_HSl



Image: <https://www.cdt.ch/news/ticino/violenti-temporali-in-vallemaggia-i-morti-sono-3-ce-ancora-un-disperso-356686>

Valencia 29.10.2024



Image: <https://english.elpais.com/climate/2024-10-30/victims-of-valencia-flood-i-thought-we-would-end-up-in-the-sea.html>

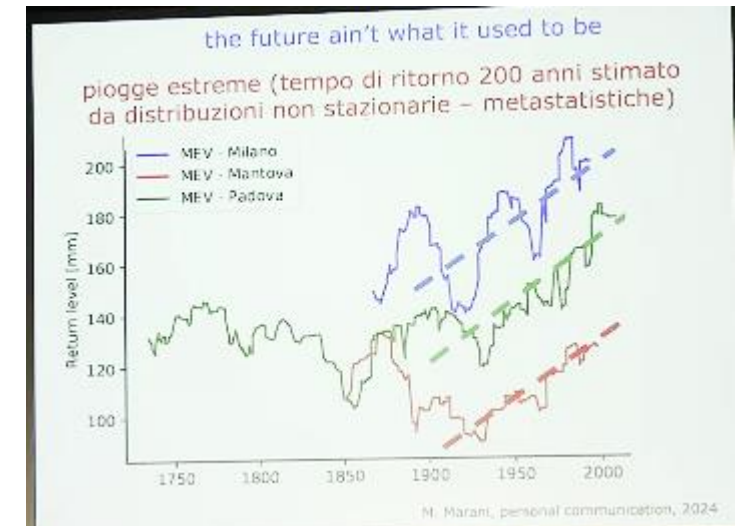
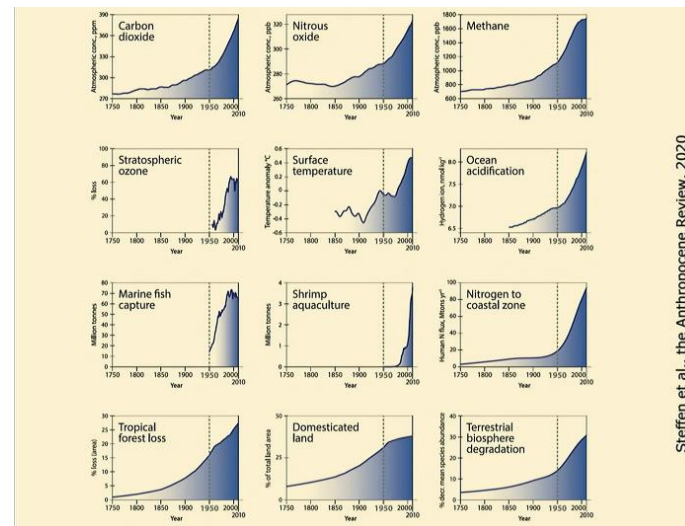
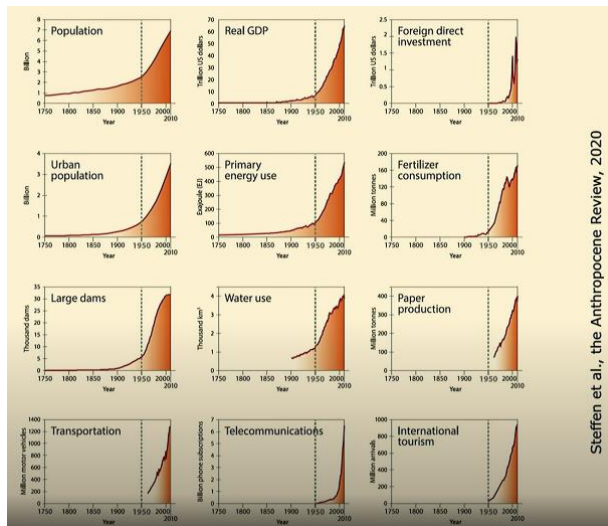


Image: <https://fortune.com/2024/11/02/spain-flooding-largest-peacetime-troop-deployment-relief-rescue/>

The future ain't what it used to be

Climate change and non-stationarity statistics

Human wellbeing and climate change indicators observed trends shows the same “hockey stick” behaviour denoting fast changing scenario with higher entropy: phenomena changed and observed statistics are no more valid !

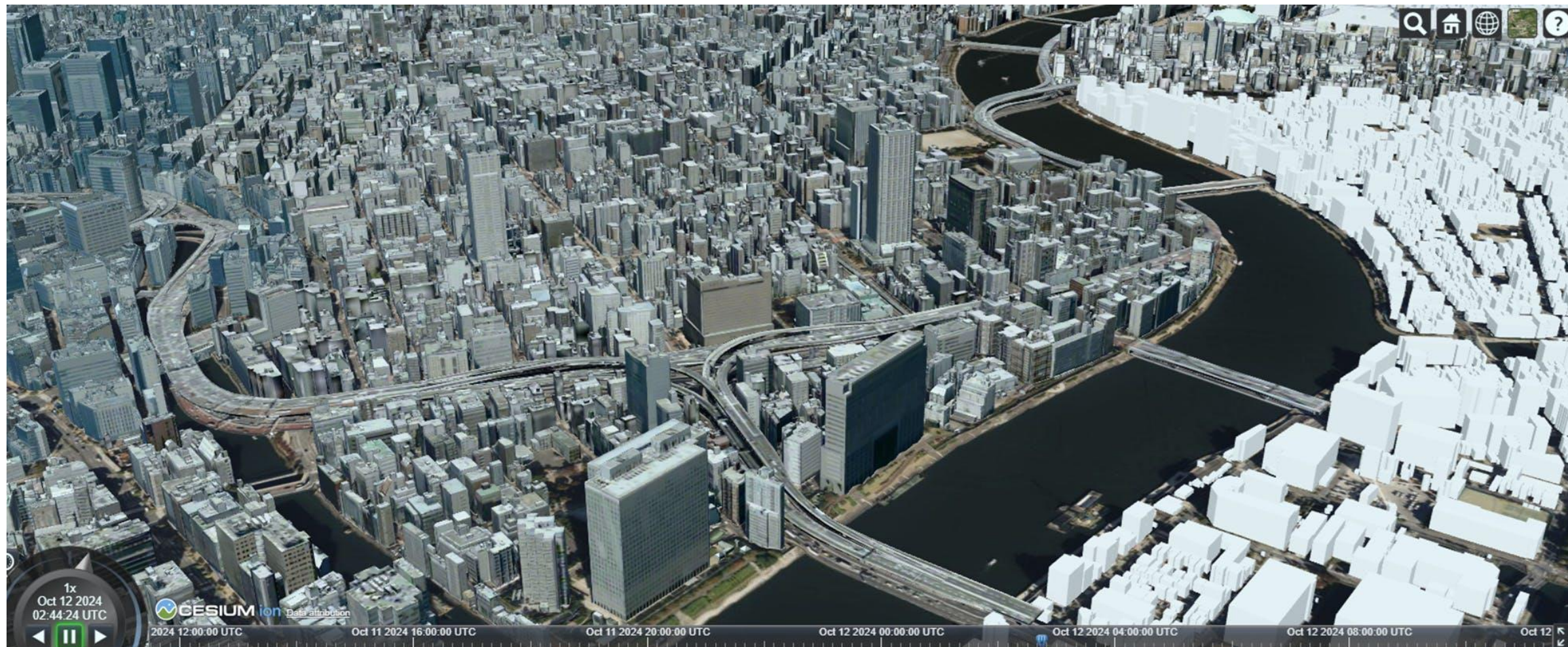


It is therefore urgent and critical to develop new and more sophisticated tools and methodologies to observe and possibly predict fundamental processes.

- We need more data...
- We need higher frequency...
- We need to mesh data...
- We need analyses ready data...
- We need real time data...
- We need geolocated data...



GeoData set the context



Satellite & In situ complement each other

- Spatially distributed
- High coverage
- Maintained by external
- Low temporal resolution
- Indirect measures
- Homogeneous formats
- External cost
- Point-wise measures
- Low spatial resolution
- Locally managed
- High temporal resolution
- High precision
- Heterogeneous formats
- Local cost

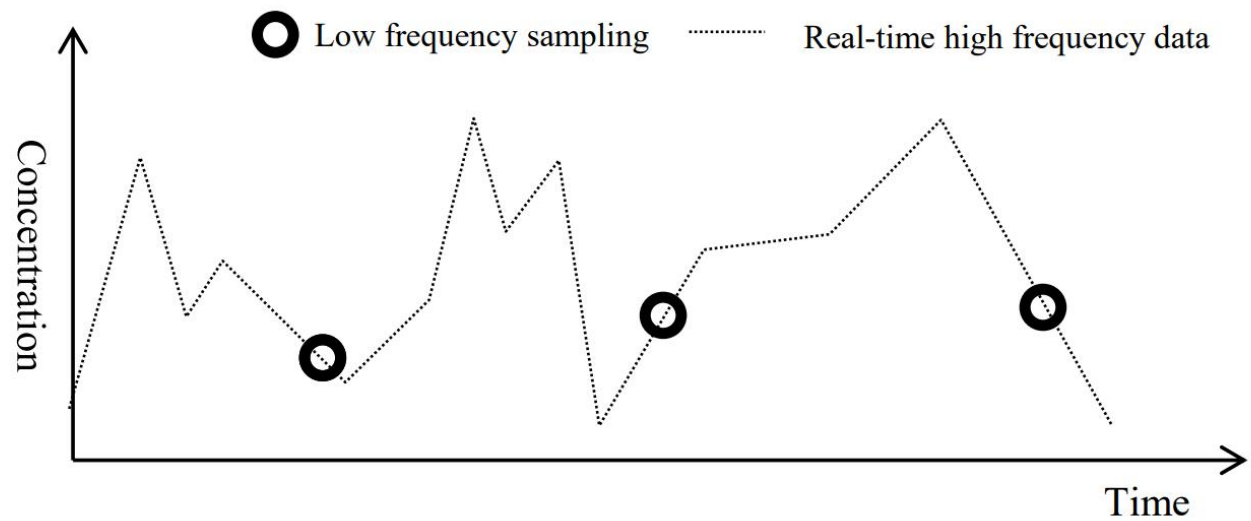
In situ: Fidelity, Resolution, consistency

Sensors permits of direct observing phenomena with high-frequency and with consistency in time.

FIDELITY, RESOLUTION, CONSISTENCY. Only in situ sensors, typically in close contact with the monitored medium, can measure a host of water-related quantity and quality parameters and processes (6) with reliable accuracy and sufficient frequency. Remote sensing provides indirect measurements normally limited to the near surface of the monitored object and affected by the media between the sensors and the monitored object. Remote-sensing ob-

Fekete, B.M.; Robarts, R.D.; Kumagai, M.; Nachtnebel, H.-P.; Odada, E.; Zhulidov, A.V. Time for in situ renaissance. *Science* 2015, 349, 685–686.

Local phenomena
High variability



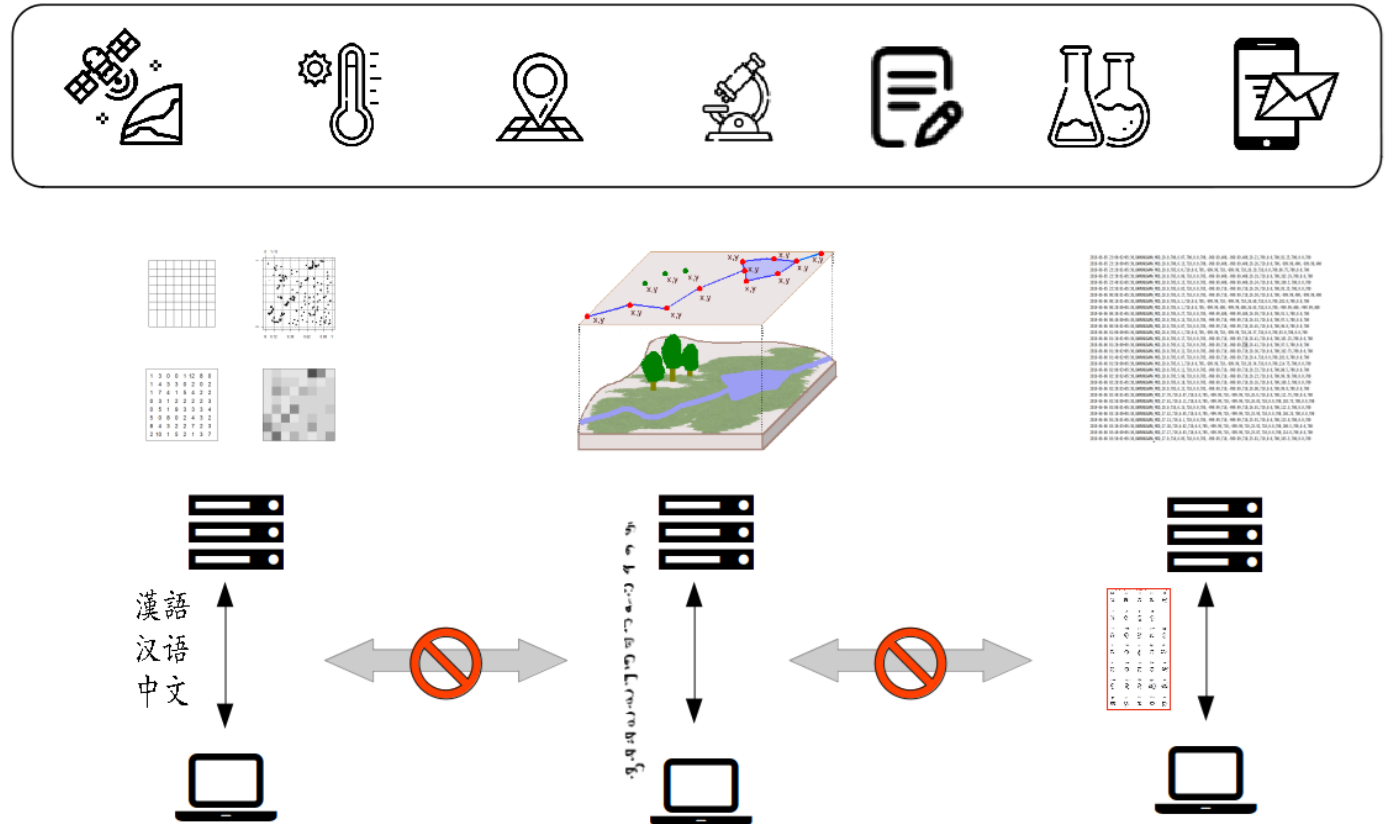
The interoperability issue persists

In-situ sensor data fusion & interoperability

In the IoT domain

1. specific business logic of sensor vendors
2. specific data package optimization approaches

led to a non-interoperable landscape lacking catalogues and metadata



after 15 years...



...how can we use new digital technologies to support adaptation?

Cloud Computing

Scalability
Containers

Redundancy
Availability

Some ingredients to
empower everyone
with early warning
system capacity and
tools

Open Science

Open
Hardware

Open
Standards

Open
Software

Open
Education

Low power

IoT

Real
Time

High
throughput

SUPSI

Open Source

... forever free usage for everyone with full open rights...

Interoperable

... adopting international open standards ...

Reliable

... applied and tested in production ...



Microservice

... specialized services that are designated to perform specific tasks ...

Reactive

... which is able to react in real-time when activated by specific pulses ...

Choreographed

... each activated service knows how to react without any external coordinator...



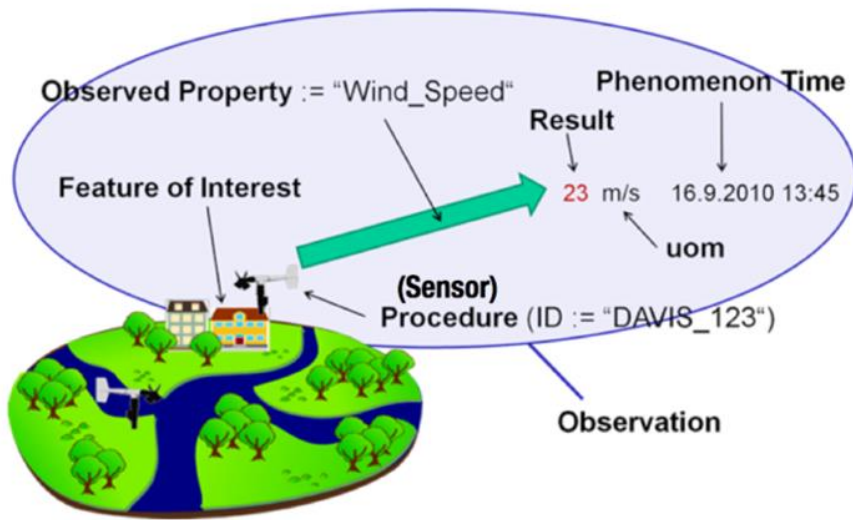
Advantages

- Scalability of microservices
(high loads scenario, big data)
- Reactivity of components
(real time reaction push)
- Atomicity of processes
(independence: best technology for the purpose)
- DevOps & Agile development approaches
(continuous integration)



OM:Observation

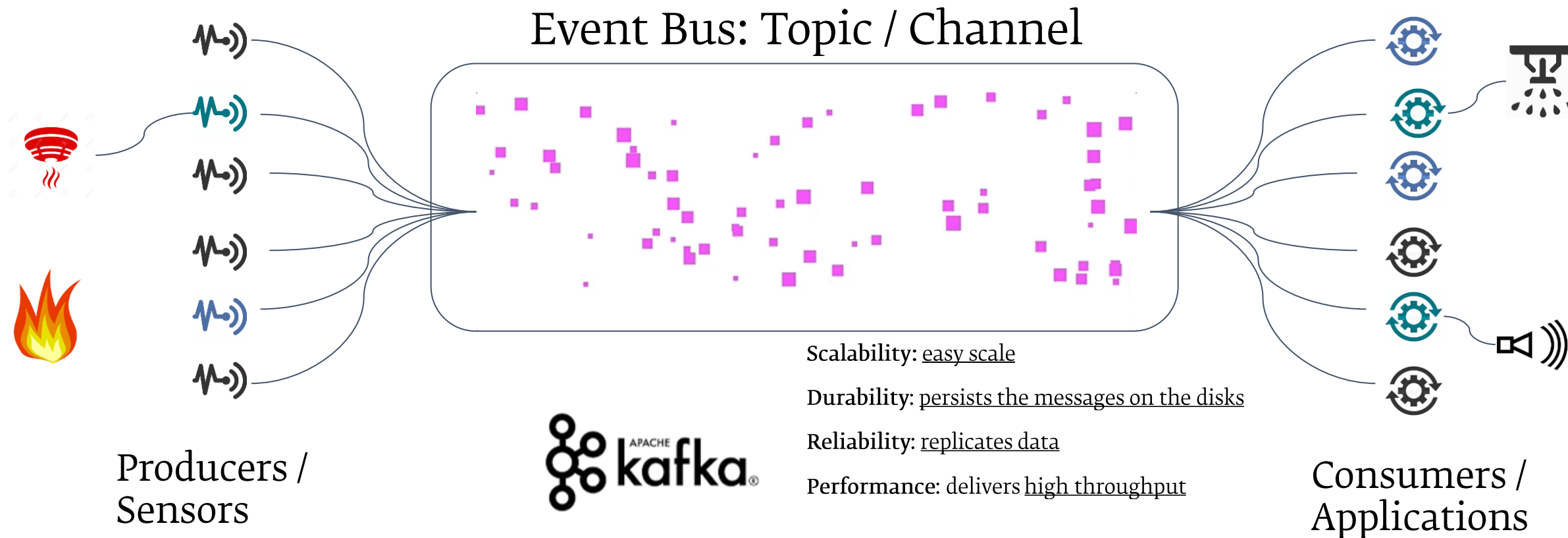
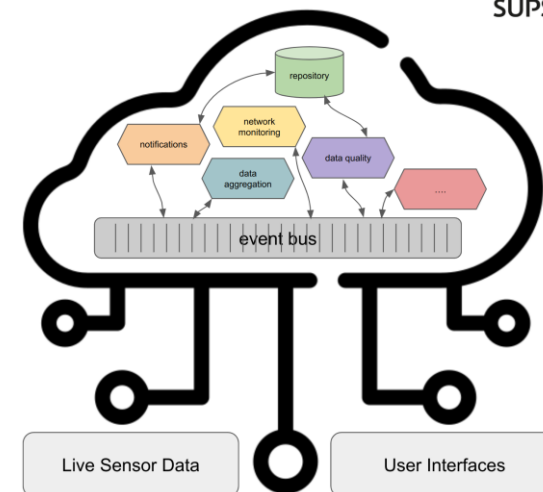
OGC Observations and Measurements as
core element of the ecosystem
(ISO 19156)



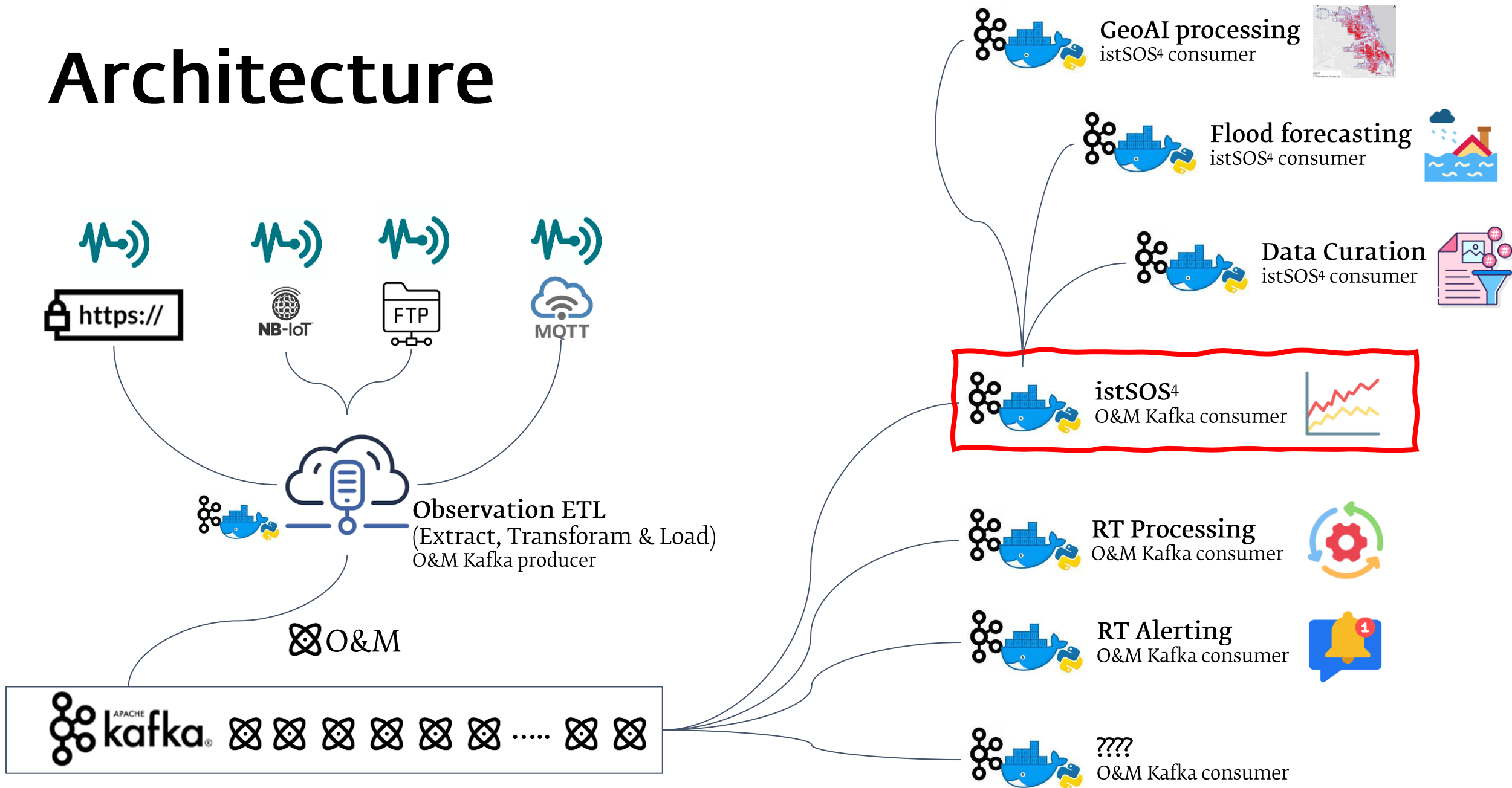
Observation

- + result — the value observed
- + phenomenonTime — the instant of the observation
- + resultTime — when the result was available
- + validTime — when the result is consummable
- + resultQuality — the quality of the result
- + observedProperty — the property observed
- + procedure — the way measure is obtained
- + featureOfInterest — the geolocated subject of the observation
- + uom — the unit of measure of the result

Event bus approach (reactive system)



Architecture





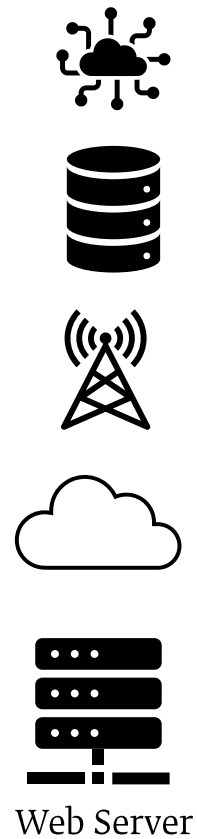
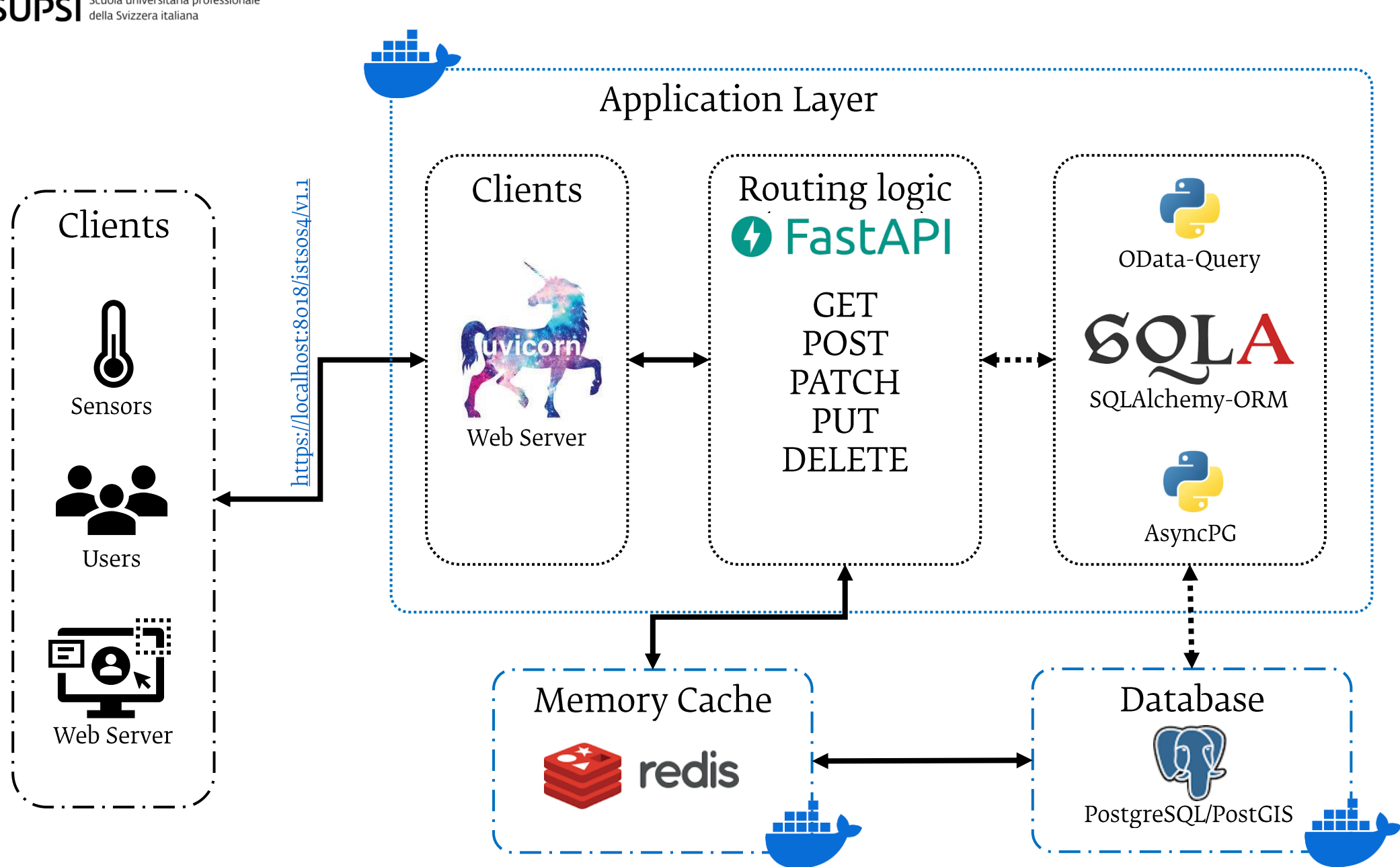
toward a modern and interoperable
IoT data management ecosystem

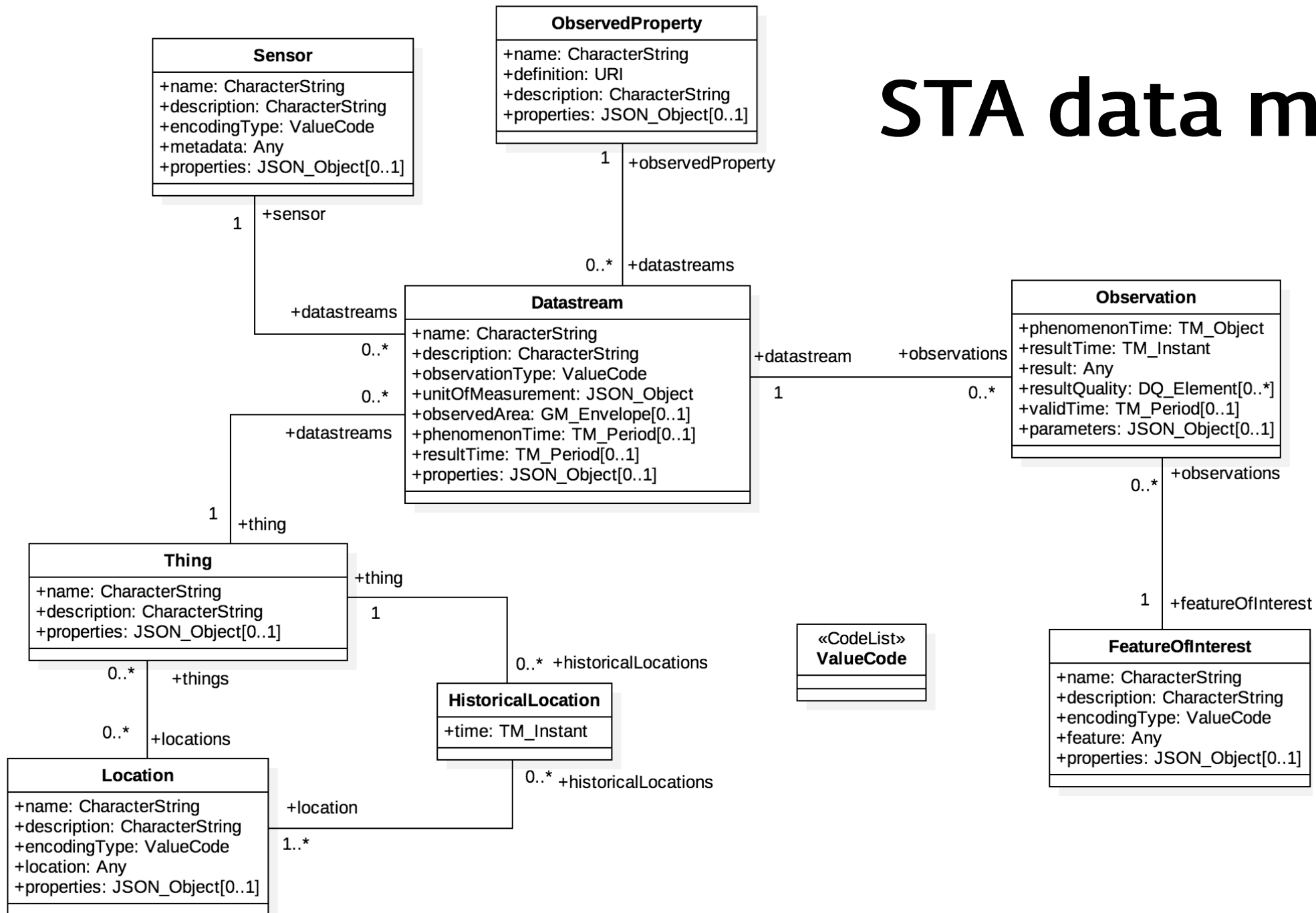
istSOS4

Is a IoT platform based on
the OGC® SensorThings
API (STA)
open standard
which is:

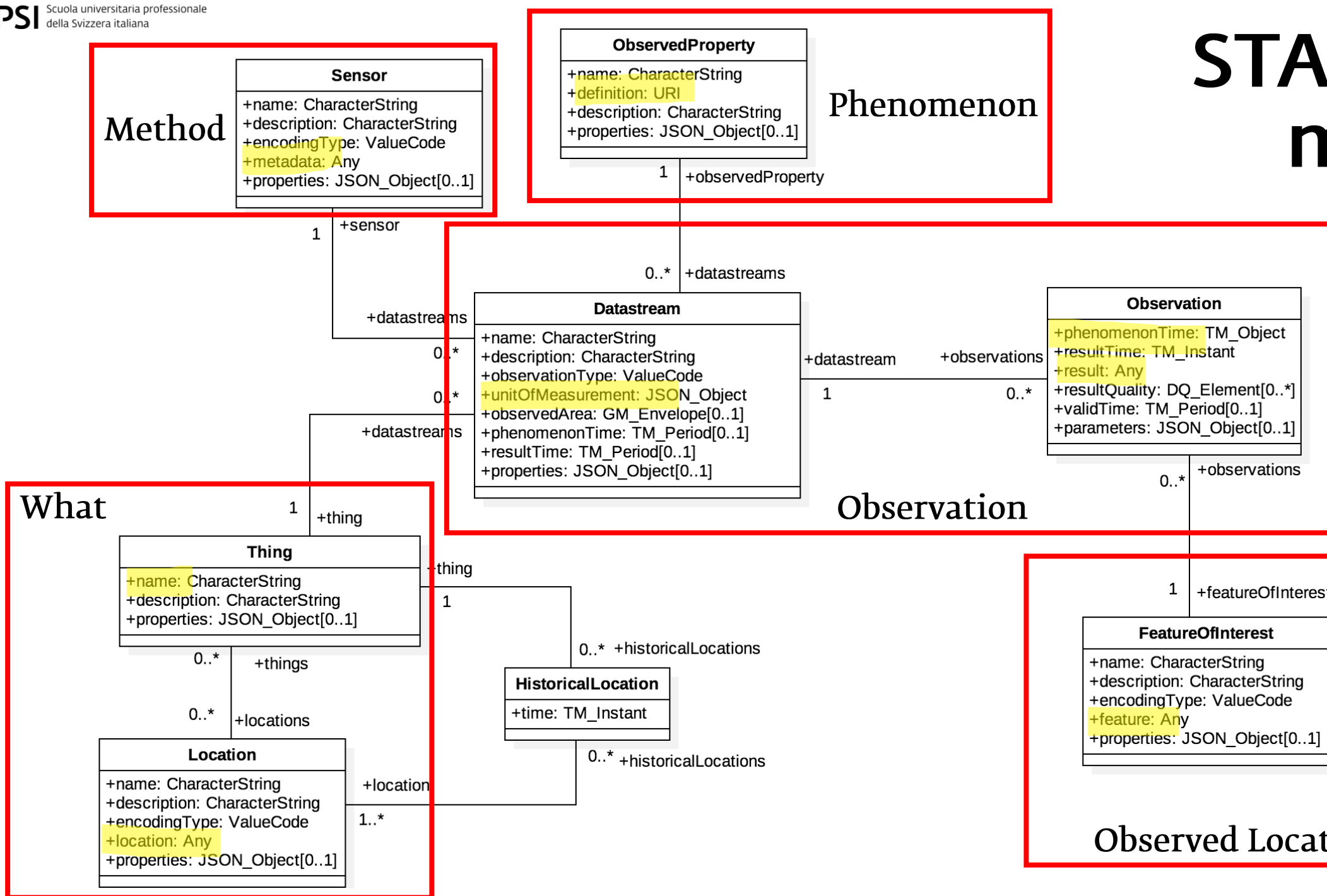
- Python
- Geolocated
- RESTful
- JSON encoded
- OData Query

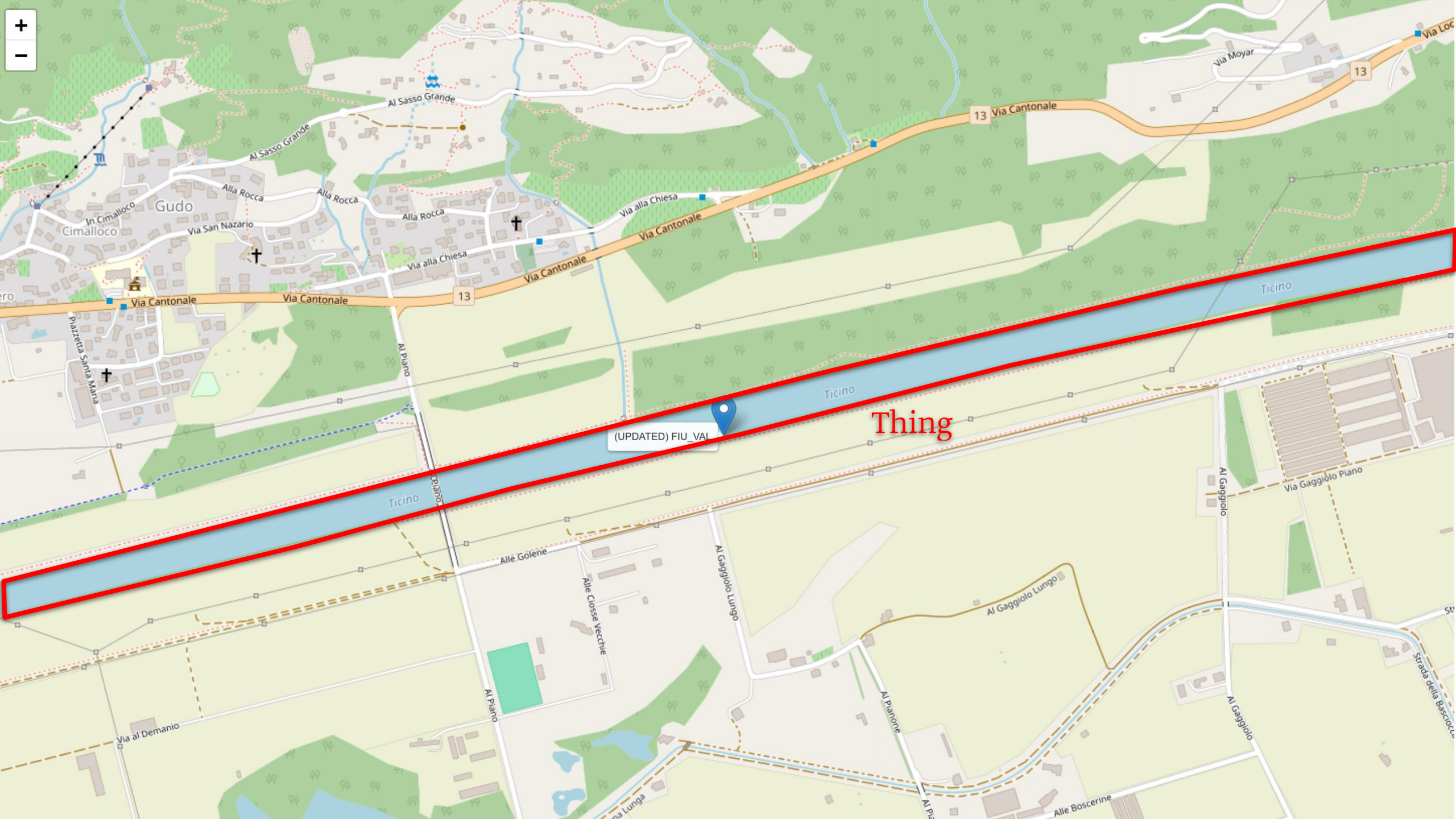






STA data model

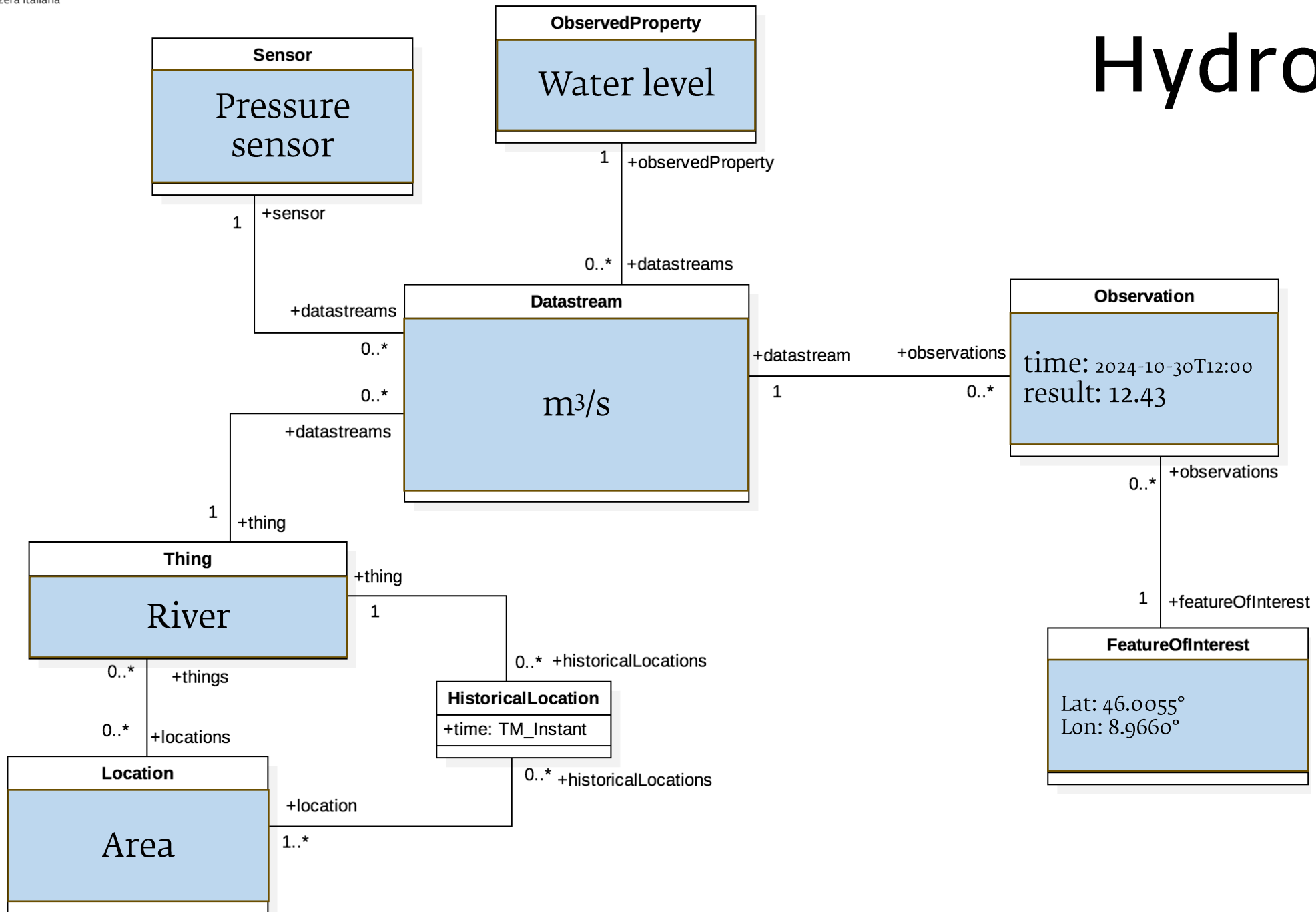




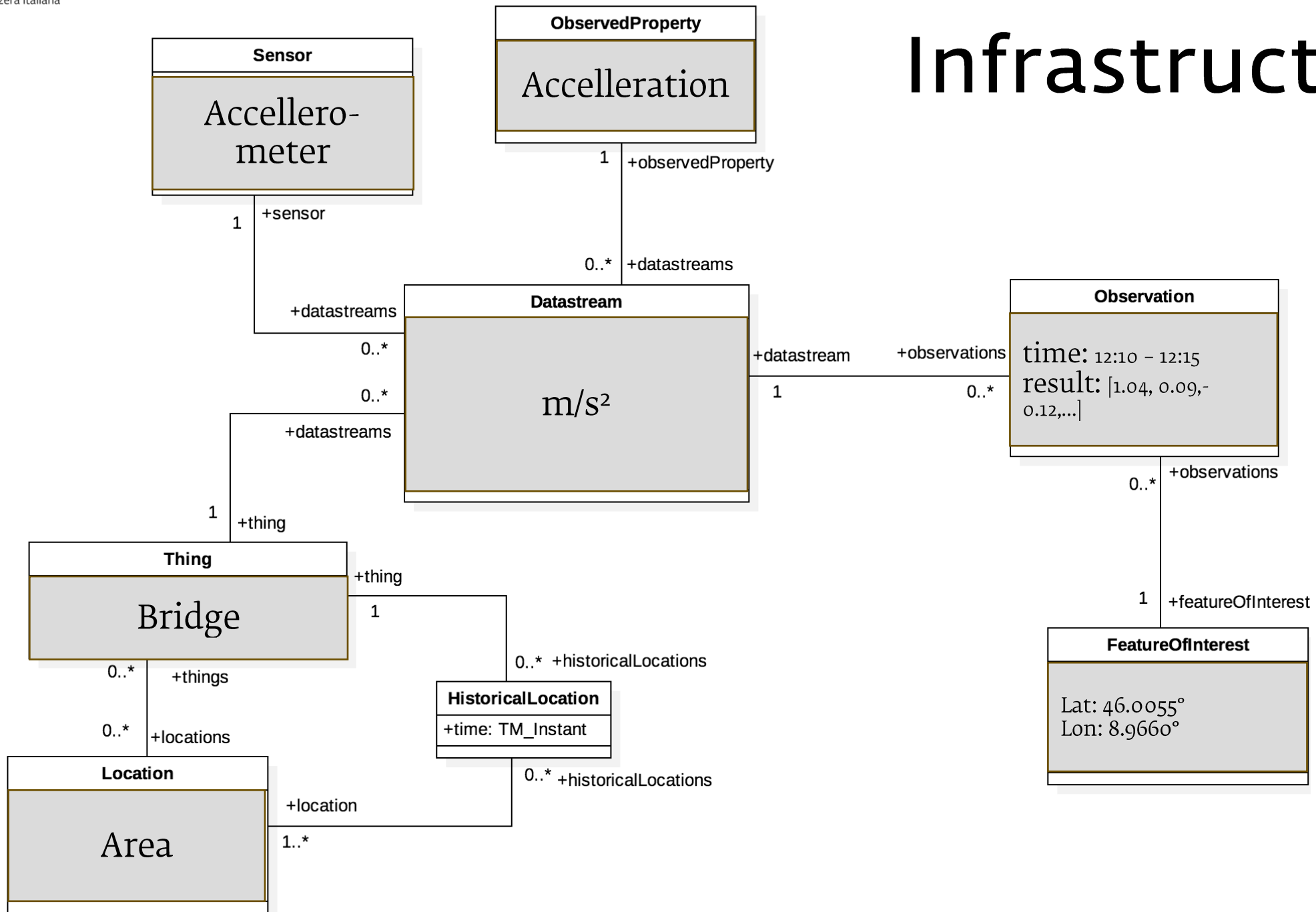
Thing

(UPDATED) FIU_VAL

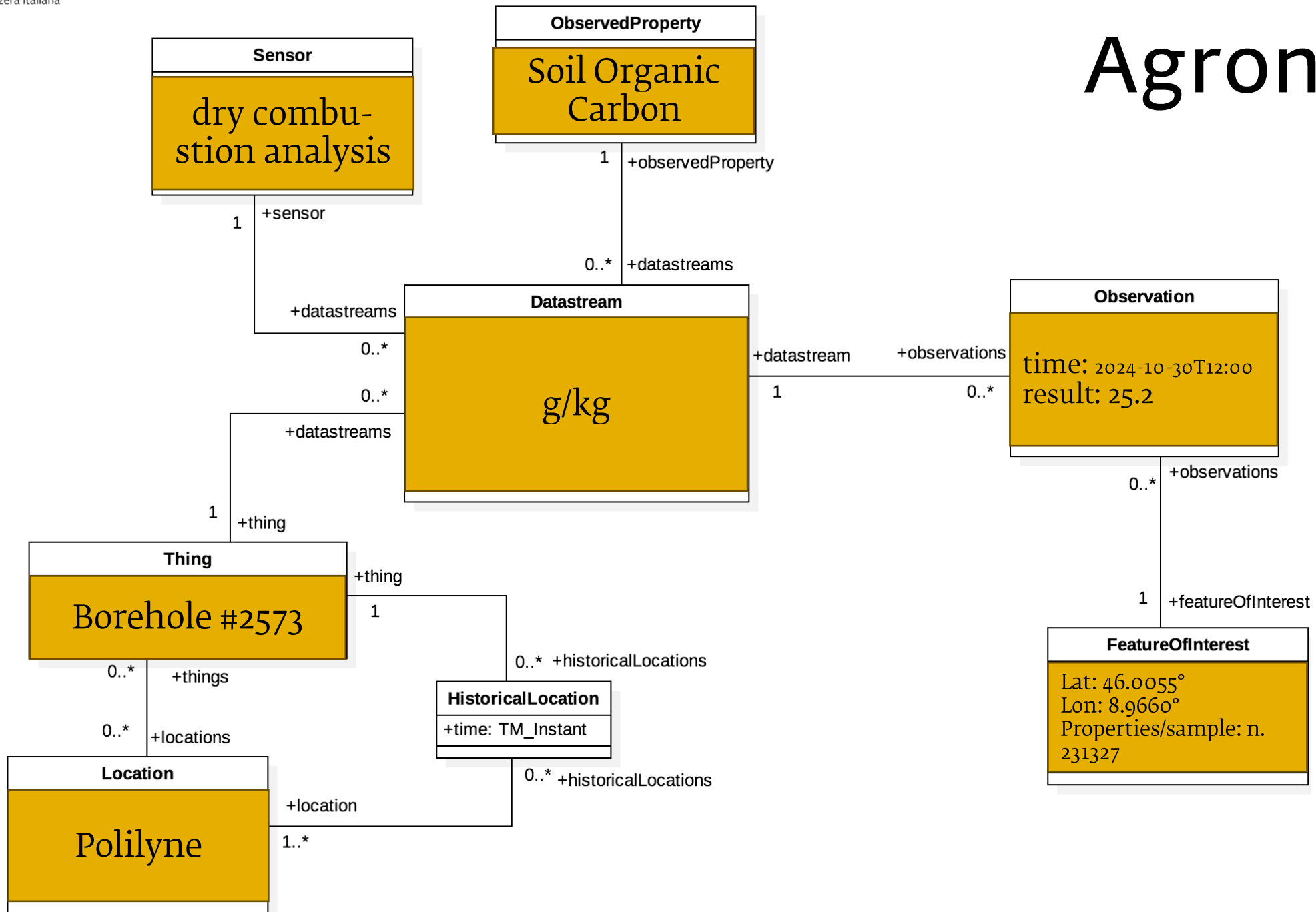
Hydrology



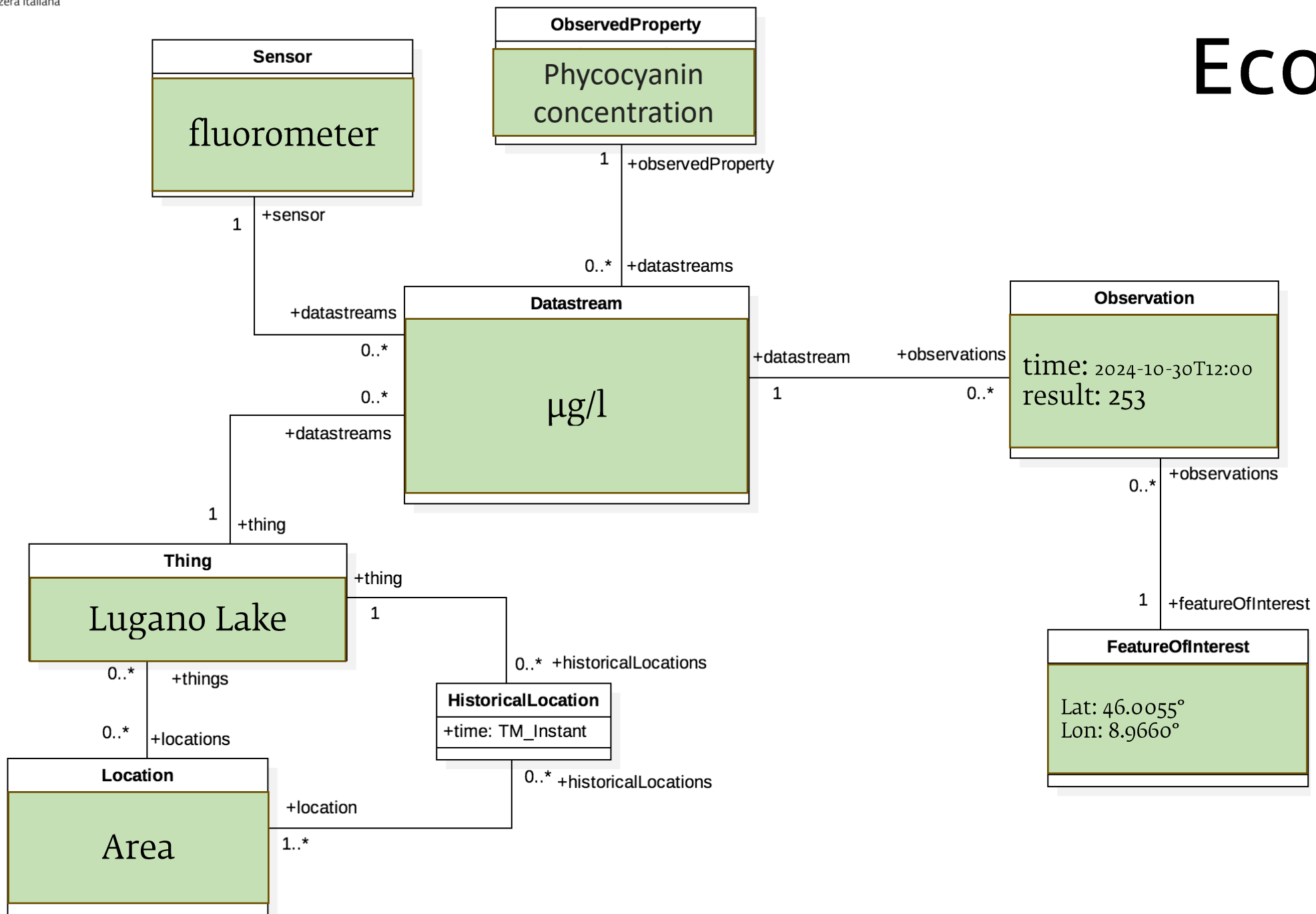
Infrastructures



Agronomy



Ecology



OData

Service URL: <http://api:5000/istsos4/v1.1/>
returns the service «capabilities»

entities

Things

Locations

Datastreams

Sensors

ObservedProperties

Observations

FeatureOfInterest

| Create | Read | Update | Delete |
|--|---|--|---|
| POST | GET | PATCH, PUT | DELETE |
| <ul style="list-style-type: none">- v1.1/entity <p>Insert a new item</p> | <ul style="list-style-type: none">- v1.1/entity- v1.1/entity(id) <p>Read a set of items or a single item</p> | <ul style="list-style-type: none">- v1.1/entity(id) <p>Update (PATCH) or replace (PUT) a single item</p> | <ul style="list-style-type: none">- v1.1/entity(id) <p>Delete a single item</p> |

GET

<http://api:5000/istsos4/v1.1>

```
{
  - value: [
    - {
      name: "Datastreams",
      url: "http://api:5000/istsos4/v1.1/Datastreams"
    },
    - {
      name: "FeaturesOfInterest",
      url: "http://api:5000/istsos4/v1.1/FeaturesOfInterest"
    },
    - {
      name: "HistoricalLocations",
      url: "http://api:5000/istsos4/v1.1/HistoricalLocations"
    },
    - {
      name: "Locations",
      url: "http://api:5000/istsos4/v1.1/Locations"
    },
    - {
      name: "Observations",
      url: "http://api:5000/istsos4/v1.1/Observations"
    },
    - {
      name: "ObservedProperties",
      url: "http://api:5000/istsos4/v1.1/ObservedProperties"
    },
    - {
      name: "Sensors",
      url: "http://api:5000/istsos4/v1.1/Sensors"
    },
    - {
      name: "Things",
      url: "http://api:5000/istsos4/v1.1/Things"
    }
  ],
  - serverSettings: {
    - conformance: [
      "http://www.opengis.net/spec/iot_sensing/1.1/req/datamodel/thing",
      "http://www.opengis.net/spec/iot_sensing/1.1/req/datamodel/location",
    ]
  }
}
```

GET - items

<http://api:5000/istsos4/v1.1/Sensors>

```
{
  - value: [
    - {
      @iot.id: 1,
      @iot.selfLink: "http://api:5000/istsos4/v1.1/Sensors(1)",
      Datastreams@iot.navigationLink: "http://api:5000/istsos4/v1.1/Sensors(1)/Datastreams",
      name: "sensor name 1",
      description: "sensor 1",
      encodingType: "application/pdf",
      metadata: "Temperature sensor",
      properties: null
    },
    - {
      @iot.id: 2,
      @iot.selfLink: "http://api:5000/istsos4/v1.1/Sensors(2)",
      Datastreams@iot.navigationLink: "http://api:5000/istsos4/v1.1/Sensors(2)/Datastreams",
      name: "sensor name 2",
      description: "sensor 2",
      encodingType: "application/pdf",
      metadata: "Motion sensor",
      properties: null
    },
    - {
      @iot.id: 3,
      @iot.selfLink: "http://api:5000/istsos4/v1.1/Sensors(3)",
      Datastreams@iot.navigationLink: "http://api:5000/istsos4/v1.1/Sensors(3)/Datastreams",
      name: "sensor name 3",
      description: "sensor 3",
      encodingType: "application/pdf",
      metadata: "Humidity sensor",
      properties: null
    },
    - {
      @iot.id: 4,
      @iot.selfLink: "http://api:5000/istsos4/v1.1/Sensors(4)",
      Datastreams@iot.navigationLink: "http://api:5000/istsos4/v1.1/Sensors(4)/Datastreams",
      name: "sensor name 4",
      description: "sensor 4",
      encodingType: "application/pdf",
      metadata: "Humidity sensor",
      properties: null
    }
  ]
}
```

GET - a single item

[http://api:5000/istsos4/v1.1/Sensors\(4\)](http://api:5000/istsos4/v1.1/Sensors(4))

```
{
  @iot.id: 4,
  @iot.selfLink: "http://api:5000/istsos4/v1.1/Sensors\(4\)",
  Datastreams@iot.navigationLink: "http://api:5000/istsos4/v1.1/Sensors\(4\)/Datastreams",
  name: "sensor name 4",
  description: "sensor 4",
  encodingType: "application/pdf",
  metadata: "Humidity sensor",
  properties: null
}
```

OData advanced

- Sort results (\$orderby=...)
[http://api:5000/istsos4/v1.1/Things?\\$orderby=name](http://api:5000/istsos4/v1.1/Things?$orderby=name)
- Paginate results (\$top=..., \$skip=..., \$count=...)
[http://api:5000/istsos4/v1.1/Things?\\$top=10&\\$skip=20&\\$count=true](http://api:5000/istsos4/v1.1/Things?$top=10&$skip=20&$count=true)
- Select fields (\$select=...)
[http://api:5000/istsos4/v1.1/Things?\\$select=name,description](http://api:5000/istsos4/v1.1/Things?$select=name,description)
- Nest related objects (\$expand=...)
[http://api:5000/istsos4/v1.1/Things?\\$expand=Locations](http://api:5000/istsos4/v1.1/Things?$expand=Locations)
- Filter based on conditions (\$filter=...)
[http://api:5000/istsos4/v1.1/Observations?\\$filter=result gt 91](http://api:5000/istsos4/v1.1/Observations?$filter=result gt 91)

Operations & Functions

| Operator | Description | Example | String Functions | |
|----------------------|-----------------------|--|---|--|
| Comparison Operators | | | bool substringof(string p0, string p1) | substringof('Sensor Things',description) |
| eq | Equal | /Datastreams?\${filter}=unitOfMeasurement/ni | bool endswith(string p0, string p1) | endswith(description,'Things') |
| ne | Not equal | /Datastreams?\${filter}=unitOfMeasurement/ni | bool startswith(string p0, string p1) | startswith(description,'Sensor') |
| gt | Greater than | /Observations?\${filter}=result gt 20.0 | int length(string p0) | length(description) eq 13 |
| ge | Greater than or equal | /Observations?\${filter}=result ge 20.0 | int indexof(string p0, string p1) | indexof(description,'Sensor') eq 1 |
| lt | Less than | /Observations?\${filter}=result lt 100 | string substring(string p0, int p1) | substring(description,1) eq 'ensor Things' substring(description,2,4) eq 'nsor' |
| le | Less than or equal | /Observations?\${filter}=result le 100 | string substring(string p0, int p1, int p2) | |
| Logical Operators | | | string tolower(string p0) | tolower(description) eq 'sensor things' |
| and | Logical and | /Observations?\${filter}=result le 3.5 and F | string toupper(string p0) | toupper(description) eq 'SENSOR THINGS' |
| or | Logical or | /Observations?\${filter}=result gt 20 or resul | string trim(string p0) | trim(description) eq 'Sensor Things' |
| not | Logical negation | /Things?\${filter}=not startswith(description | string concat(string p0, string p1) | concat(concat(unitOfMeasurement/symbol,', '), unitOfMeasurement/name) eq 'degree, Celsius' |
| | | | Date Functions | |
| | | | int year | year(resultTime) eq 2015 |
| | | | int month | month(resultTime) eq 12 |
| | | | int day | day(resultTime) eq 8 |
| | | | int hour | hour(resultTime) eq 1 |
| | | | int minute | minute(resultTime) eq 0 |
| | | | int second | second(resultTime) eq 0 |
| | | | int fractionalseconds | second(resultTime) eq 0 |
| | | | int date | date(resultTime) ne date(validTime) |
| | | | time | time(resultTime) le validTime |
| | | | int totaloffsetminutes | totaloffsetminutes(resultTime) eq 60 |
| Arithmetic Operators | | | now | resultTime ge now() |
| add | Addition | /Observations?\${filter}=result add 5 gt | mindatetime | resultTime eq mindatetime() |
| sub | Subtraction | /Observations?\${filter}=result sub 5 gt | maxdatetime | resultTime eq maxdatetime() |
| mul | Multiplication | /Observations?\${filter}=result mul 2 gt | | |
| div | Division | /Observations?\${filter}=result div 2 gt | | |
| mod | Modulo | /Observations?\${filter}=result mod 2 eq | | |
| Grouping Operators | | | | |
| () | Precedence grouping | /Observations?\${filter}=(result sub 5) ; | | |

| Spatial Relationship Functions | |
|--------------------------------|---|
| bool st_equals | st_equals(location, geography'POINT (30 10)') |
| bool st_disjoint | st_disjoint(location, geography'POLYGON ((30 10, 10 20, 20 40, 40 40, 30 10))') |
| bool st_touches | st_touches(location, geography'LINESTRING (30 10, 10 30, 40 40)') |
| bool st_within | st_within(location, geography'POLYGON ((30 10, 10 20, 20 40, 40 40, 30 10))') |
| bool st_overlaps | st_overlaps(location, geography'POLYGON ((30 10, 10 20, 20 40, 40 40, 30 10))') |
| bool st_crosses | st_crosses(location, geography'LINESTRING (30 10, 10 30, 40 40)') |
| bool st_intersects | st_intersects(location, geography'LINESTRING (30 10, 10 30, 40 40)') |
| bool st_contains | st_contains(location, geography'POINT (30 10)') |

| Math Functions | |
|----------------|-----------------------|
| round | round(result) eq 32 |
| floor | floor(result) eq 32 |
| ceiling | ceiling(result) eq 33 |

| Geospatial Functions | |
|---|--|
| double geo.distance(Point p0, Point p1) | geo.distance(location, geogr 10)') |
| double geo.length(LineString p0) | geo.length(geography'LINESTR 40 40)') |
| bool geo.intersects(Point p0, Polygon p1) | geo.intersects(location, geo ((30 10, 10 20, 20 40, 40 40, |

Transaction: create

POST /v1.1/Things HTTP/1.1

Host: <http://api:5000/istsos4>

Content-Type: application/json

This create a new element with
the provided fields !

RESPONSE HEADER

HTTP/1.1 201 Created

Content-Type: application/json

Location: [http://localhost:8018/istsos4/v1.1/Things\(1\)](http://localhost:8018/istsos4/v1.1/Things(1))

```
{
  "name": "Lugano Lake",
  "description": "The Apline Lake located in  
Southern Switzerland",
  "properties": {
    "Catchment area": "565.6 km2",
    "Surface Area": "38.7 km2",
    "Avg. Depth": "124 m",
    "Max depth": "288 m",
    "Water Volume": "6.5 km3"
    "Surface elevation": "271 m"
    "Primary inflows": ["Vedeggio",
                        "Cassarate", "Cuccio", "Laveggio",
                        "Magliasina", "Bolletta",
                        "Scairolo"]
    "Primary outflows": "Tresa"
  }
}
```

Transaction: update

PATCH /v1.1/Things(1) HTTP/1.1

Host: <http://api:5000/istsos4>

Content-Type: application/json

This updates only the provided fields, **Primary inflows** field will still be the same !

RESPONSE HEADER

HTTP/1.1 204 No Content

Content-Type: application/json

Location: [http://localhost:8018/istsos4/v1.1/Things\(1\)](http://localhost:8018/istsos4/v1.1/Things(1))

```
{  
  "properties": {  
    "Surface Area": "48.7 km²",  
    "Avg. Depth": "134"  
  }  
}
```

Transaction: replace

PUT /v1.1/Things(1) HTTP/1.1
Host: <http://api:5000/istsos4>
Content-Type: application/json

This replace the entire object and
Primary inflows filed will not be
present anymore !

RESPONSE HEADER

HTTP/1.1 201 Created
Content-Type: application/json
Location: [http://localhost:8018/istsos4/v1.1/Things\(1\)](http://localhost:8018/istsos4/v1.1/Things(1))

```
{  
  "name": "Lugano Lake",  
  "description": "The Apline Lake located in  
    Southern Switzerland",  
  "properties": {  
    "Catchment area": "565.6 km2",  
    "Surface Area": "48.7 km2",  
    "Avg. Depth": "134 m",  
    "Max depth": "288 m",  
    "Water Volume": "6.5 km3",  
    "Surface elevation": "271 m"  
  }  
}
```

Transaction: delete

DELETE /v1.1/Things(1) HTTP/1.1

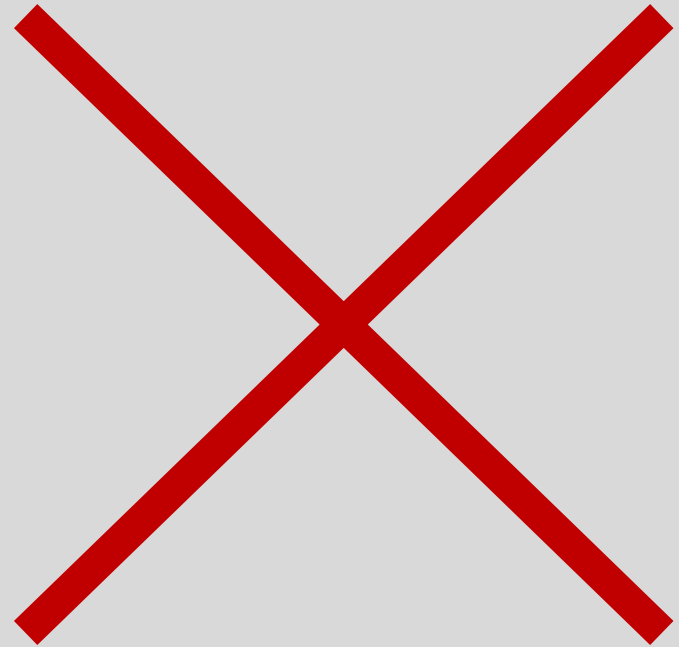
Host: <http://api:5000/istsos4>

This delete the entire object !

RESPONSE HEADER

HTTP/1.1 204 No Content

Content-Type: application/json



Create nested

```
"name": "Lugano Lake",
"description": "The Alpine Lake located in Southern Switzerland",
"properties": {
  "Catchment area": "565.6 km²",
  "Surface Area": "48.7 km²",
  "Avg. Depth": "134 m",
  "Max depth": "288 m",
  "Water Volume": "6.5 km³",
  "Surface elevation": "271 m",
  "Primary inflows": ["Veleggio", "Cassarate", "Cuccio", "Laveggio",
    "Magliasina", "Bolletta", "Scairolo"],
  "Primary outflows": "Tresa"
},
"Locations": [
{
  "name": "Measurement Location - Lugano Lake",
  "description": "Specific location within the lake where
    water quality is measured.",
  "location": {
    "type": "Point",
    "coordinates": [8.9667, 45.9833]
  }
},
],
"Datastreams": [
{
  "name": "pH Level",
```

```
"description": "Datastream for measuring pH
  levels of water in Lugano Lake",
  "unitOfMeasurement": {
    "name": "pH",
    "symbol": "pH",
    "definition": "http://www.qudt.org/qudt/owl/1.0.o/
      unit/Instances.html#pH"
  },
  "observationType": "http://www.opengis.net/def/
    observationType/OGC-OM/2.0/OM_Measurement",
  "ObservedProperty": {
    "name": "pH Level",
    "description": "The pH level of the lake water.",
    "definition": "http://www.opengis.net/def/property/OGC/o/pH"
  },
  "Sensor": {
    "name": "pH Sensor",
    "description": "A sensor that measures pH levels of the lake water.",
    "encodingType": "application/pdf",
    "metadata": "https://sensor.com/ph-sensor-manual.pdf"
  },
  "Observations": [
    {
      "result": 7.5,
      "phenomenonTime": "2024-10-01T08:00:00Z"
    },
    {
      "result": 7.6,
      "phenomenonTime": "2024-10-02T08:00:00Z"
    }
  ]
}
]
```

RESPONSE HEADER
HTTP/1.1 201 Created
Content-Type: application/json
Location: [http://localhost:8018/istsos4/v1.1/Things\(1\)](http://localhost:8018/istsos4/v1.1/Things(1))

istSOS 4 -Load test (Preliminary results)

2000 concurrent users

| Type | Name | Request Count | Failure Count | Median Response Time (ms) | Average Response Time | Min Response Time | Requests/s |
|------|-------------------------|---------------|---------------|---------------------------|-----------------------|-------------------|------------|
| GET | Datastream() | 110371 | 0 | 17000 | 17900.29 | 36.09 | 5.89 |
| GET | GetCapabilities | 22039 | 0 | 74 | 473.08 | 2.04 | 1.18 |
| GET | GetDatastreams | 45648 | 0 | 18000 | 19225.35 | 164.00 | 2.44 |
| GET | GetObservations 1 day | 550251 | 0 | 17000 | 17822.92 | 29.20 | 29.35 |
| GET | GetObservations 1 month | 44420 | 0 | 18000 | 18130.47 | 151.31 | 2.37 |
| GET | GetObservations 1 week | 1102127 | 0 | 17000 | 17894.92 | 47.75 | 58.79 |
| GET | GetThings | 45512 | 0 | 17000 | 18003.86 | 22.71 | 2.43 |
| POST | InsertObservations | 28600 | 0 | 20000 | 20666.64 | 547.28 | 1.53 |
| GET | Thing() | 110286 | 0 | 17000 | 17808.32 | 14.69 | 5.88 |
| | Aggregated | 2059254 | 0 | 17000 | 17760.35 | 2.04 | 109.85 |



OSIReS

Open **S**ervices for **I**nteroperable and **R**eproducible research
based on **S**patiotemporal varying data

Massimiliano Cannata, Daniele Strigaro, Claudio Primerano (SUPSI)

Gregory Giuliani (UNIGE)

Jens Ingensand, Olivier Ertz, Maxime Collombin (HEIG-VD)



This work is licensed under a
[Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

Open Science programme - **swissuniversities**

Can I using OGC standards to perform:

- Auditing and Compliance (**reproducibility**):
Retrieve data states for audits or regulatory requirements.
- Data Recovery:
Recover previous versions in case of accidental modifications or deletions.
- Analysis Over Time:
Compare data states across different periods for trend analysis or insights.



UNIVERSITÉ
DE GENÈVE

Reproducibility for satellite data services



HAUTE ÉCOLE
D'INGÉNIERIE
ET DE GESTION
DU CANTON
DE VAUD

Reproducibility for vector data services

SUPSI

Reproducibility for IoT data services

Preprints.org

Instructions for Authors

About

FAQ

Blog and News

Home > Computer Science and Mathematics > Information Systems > DOI:10.20944/preprints202312.2316.v1

Preprint

Review

The Challenges of Reproducibility for Research Based on Geodata Web Services

Massimiliano Cannata , Maxime Collombin , Olivier Ertz , Gregory Giuliani , Jens Ingensand , Claudio Primerano , Daniele Strigaro

This version is not peer-reviewed

Abstract

Modern research applies the Open Science approach that fosters the production and sharing of Open Data according to the FAIR (Findable, Accessible, Interoperable, Reusable) principles.



Cannata, M.; Collombin, M.; Ertz, O.; Giuliani, G.; Ingensand, J.; Primerano, C.; Strigaro, D. The Challenges of Reproducibility for Research Based on Geodata Web Services. *Preprints* **2023**, 2023122316. <https://doi.org/10.20944/preprints202312.2316.v1>

istSOS⁴_{Things}

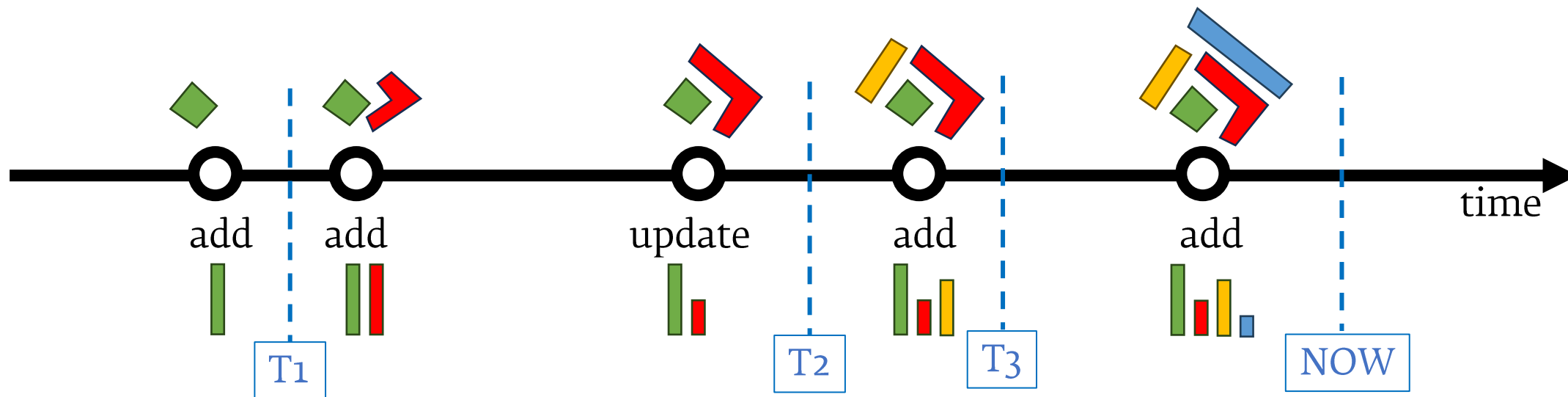
...a service for reproducible research
and data curation

Traveltime extension

The ability to
access and query data
as it was at a specific point in the past.

This feature allows users to "rewind" the state of the data to view previous data versions without impacting the current state.

STA – traveltime extension



- Get as it was (\$as_of=...)

[http://localhost:8018/istsos4/v1.1/Things?\\$as_of=2002-03-01T13:00:00Z](http://localhost:8018/istsos4/v1.1/Things?$as_of=2002-03-01T13:00:00Z)

- Get how it changes (\$from_to=...)

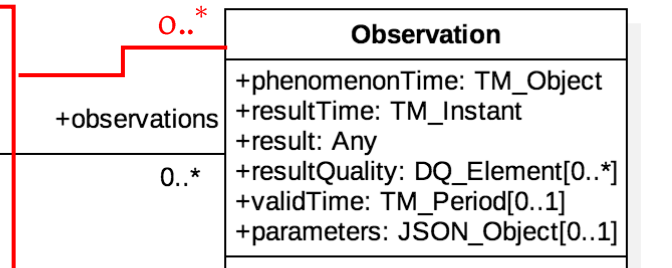
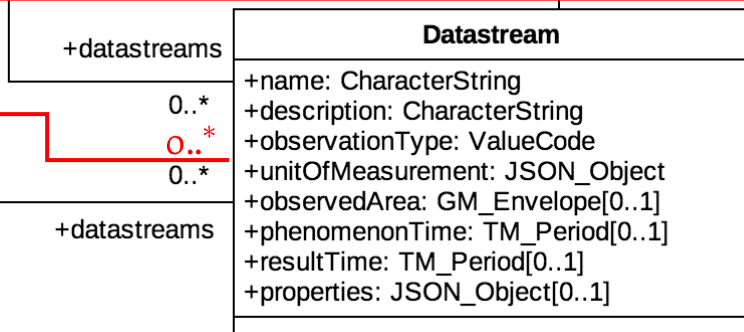
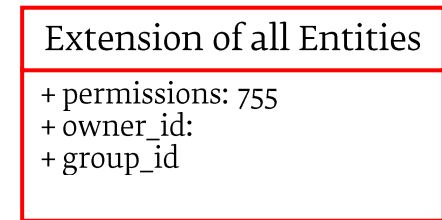
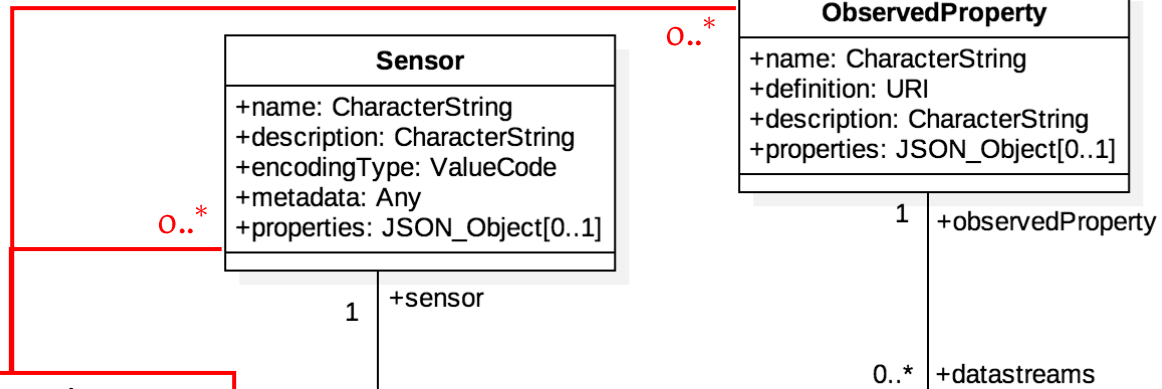
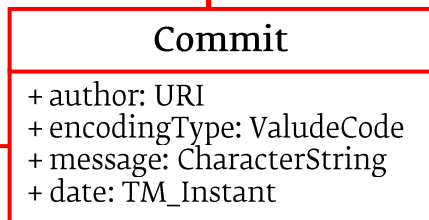
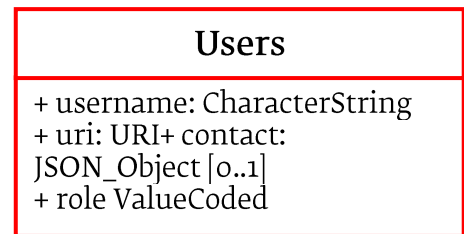
[http://localhost:8018/istsos4/v1.1/Things?\\$from_to=2002-03-01T13:00:00Z/2003-05-11T15:30:00Z](http://localhost:8018/istsos4/v1.1/Things?$from_to=2002-03-01T13:00:00Z/2003-05-11T15:30:00Z)

Commit - immutable entity (1:n)

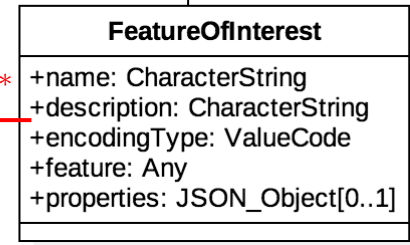
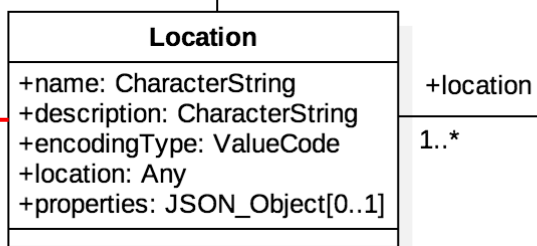
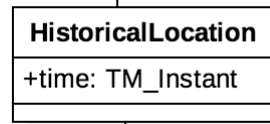
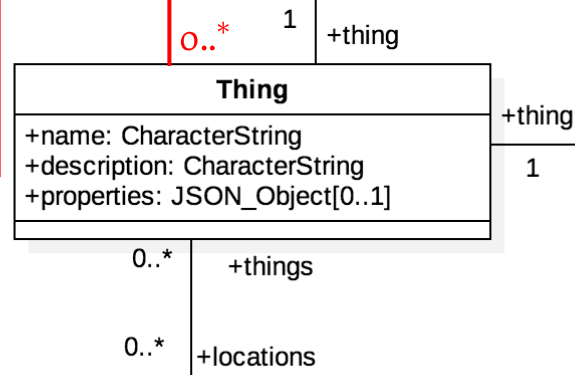
| Properties | Type | Multiplicity and use | Description |
|---------------------|--------------------|----------------------|--|
| <i>author</i> | string(128) | One (system managed) | Personal user identifier (Authority, like ORCID, or User Profile Link) |
| <i>actionType</i> | ValudeCode | One (system managed) | The action used when the message was generated (CREATE, UPDATE, DELETE) |
| <i>encodingType</i> | string | One (optional) | The encoding type of the message (default is text/plain). |
| <i>message</i> | string(256) | One (mandatory) | Commit message detailing the scope, motivation, and method of the transaction. |
| <i>date</i> | ISO 8601 date-time | One (system managed) | A date-time that specifies the exact moment when the commit was executed. |

User entity

| Properties | Type | Multiplicity and use | Description |
|-----------------|-------------|----------------------|--|
| <i>username</i> | string(128) | One (mandatory) | Postgresql user with its own password, that is created by admin at user creation |
| <i>uri</i> | URI | One (optional) | Personal user identifier (e.g. ORCID) |
| <i>contact</i> | JSON_Object | One (optional) | ContactInformation element in JSON representation (see OGC API reference) |
| <i>role</i> | ValueCoded | One (mandatory) | One of: Admin, Editor, Viewer, Obs_manager, Sensor |



STA data model



User

POST [/v1.1/users](#) HTTP/1.1

Host: <http://api:5000/istsos4>

```
{
  username: "utente1"
  password: "*****",
  uri: "https://orcid.org/0000-0003-2527-1416",
  role: "admin"
  contact: {
    "ContactPersonPrimary": {
      "ContactPerson": "Mario Rossi",
      "ContactOrganization": "Comune di Milano"
    },
    "ContactPosition": "Responsabile GIS",
    "ContactAddress": {
      "AddressType": "Posta",
      "Address": "Via Roma, 1",
      "City": "Milano",
      "StateOrProvince": "MI",
      "PostCode": 20121,
      "Country": "Italia"
    },
    "ContactVoiceTelephone": "+39 02 123 4567",
    "ContactFacsimileTelephone": "+39 02 765 4321",
    "ContactElectronicMailAddress": "mario.rossi@comune.milano.it"
  }
}
```

Contact information object

Oauth login

POST [/v1.1/login](#) HTTP/1.1

Host: <http://api:5000/istsos4>

Body

Grant-type=
password&username=utente1&password=*****

Existing Roles

Admin: full rights on schema

Editor: CRUD on schema (no drop!)

Viewer: SELECT rights on schema

Obs_manager: CRUD on Observations

Sensor: INSERT on Observations (no commit!)

POST, PATCH, DELETE:

mandatory commit in header (except for sensor role)

POST /v1.1/Things HTTP/1.1

Host: <http://api:5000/istsos4>

Content-Type: application/json

Commit-message: Corrected max depth information

This update the Lugano Lake element with the provided fields!

```
{
  "name": "Lugano Lake",
  "description": "The Apline Lake located in Southern Switzerland",
  "properties": {
    "Catchment area": "565.6 km²",
    "Surface Area": "38.7 km²",
    "Avg. Depth": "124 m",
    "Max depth": "288 m",
    "Water Volume": "6.5 km³",
    "Surface elevation": "271 m",
    "Primary inflows": ["Vedeggio", "Cassarate", "Cuccio", "Laveggio", "Magliasina", "Bolletta", "Scairolo"],
    "Primary outflows": "Tresa"
  }
}
```

GET – items as it was

[http://localhost/istsos4/v1.1/Things\(1\)?\\$as_of=2024-09-02T15:22:05.04Z](http://localhost/istsos4/v1.1/Things(1)?$as_of=2024-09-02T15:22:05.04Z)

```
{
  "@iot.id": 1,
  "@iot.selfLink": "http://localhost:8018/istsos-miu/v1.1/Things(1)?$as_of=2024-09-02T15:22:05.04Z",
  "Locations@iot.navigationLink": "http://localhost:8018/istsos-miu/v1.1/Things(1)/Locations?$as_of=2024-09-02T15:22:05.04Z",
  "HistoricalLocations@iot.navigationLink": "http://localhost:8018/istsos-miu/v1.1/Things(1)/HistoricalLocations?$as_of=2024-09-02T15:22:05.04Z",
  "Datastreams@iot.navigationLink": "http://localhost:8018/istsos-miu/v1.1/Things(1)/Datastreams?$as_of=2024-09-02T15:22:05.04Z",
  "Commit@iot.navigationLink": "http://localhost:8018/istsos-miu/v1.1/Things(1)/Commit(1)",
  "name": "Lake Lugano",
  "description": "The Apline Lake located in Southern Switzerland",
  "properties": {
    "max_depth": "288 m"
  }
}
```

GET –item as it was with commit

[http://localhost/istsos4/v1.1/Things\(1\)?\\$as_of=2024-09-02T15:22:05.04Z\\$expand=Commit](http://localhost/istsos4/v1.1/Things(1)?$as_of=2024-09-02T15:22:05.04Z$expand=Commit)

```
{
  "@iot.id": 1,
  "@iot.selfLink": "http://localhost:8018/istsos-miu/v1.1/Things(1)?$as_of=2024-09-02T15:22:05.04Z",
  "Locations@iot.navigationLink": "http://localhost:8018/istsos-miu/v1.1/Things(1)/Locations?$as_of=2024-09-02T15:22:05.04Z",
  "HistoricalLocations@iot.navigationLink": "http://localhost:8018/istsos-miu/v1.1/Things(1)/HistoricalLocations?$as_of=2024-09-02T15:22:05.04Z",
  "Datastreams@iot.navigationLink": "http://localhost:8018/istsos-miu/v1.1/Things(1)/Datastreams?$as_of=2024-09-02T15:22:05.04Z",
  "Commit@iot.navigationLink": "http://localhost:8018/istsos-miu/v1.1/Things(1)/Commit(1)",
  "name": "Lake Lugano",
  "description": "The Apline Lake located in Southern Switzerland",
  "properties": {
    "max_depth": "288 m"
  },
  "Commit": {
    "@iot.id": 1,
    "date": "2024-09-02T15:22:03.044308+00:00Z",
    "author": "https://orcid.org/0000-0003-2527-1416",
    "message": "Corrected max depth information",
    "encodingType": "text/plain",
    "actionType": "CREATE",
    "@iot.selfLink": "http://localhost:8018/istsos-miu/v1.1/Commits(1)"
  }
}
```

GET – items as they has been

[http://api:5000/istos4/v1.1/Observations?
\\$filter=id eq 2108168&
\\$from to=2024-11-09T08:00:00/2024-11-
09T12:00:00](http://api:5000/istos4/v1.1/Observations?<u>$filter=id eq 2108168&$from to=2024-11-09T08:00:00/2024-11-09T12:00:00</u>)

```
"value": [  
  {  
    "@iot.id": 2108168,  
    "@iot.selfLink": "http://api:5000/istos4/v1.1/Observations(2108168)",  
    "FeatureOfInterest@iot.navigationLink": "http://api:5000/istos4/v1.1/Observations(2108168)/FeatureOfInterest",  
    "Datastream@iot.navigationLink": "http://api:5000/istos4/v1.1/Observations(2108168)/Datastream",  
    "phenomenonTime": "2024-09-27T18:00:00+00:00",  
    "resultTime": null,  
    "result": 32.8,  
    "resultQuality": 313,  
    "validTime": null,  
    "parameters": null,  
    "systemTimeValidity": "2024-11-09 08:45:51.792071+00/2024-11-10 08:49:43.484626+00",  
    "Commit@iot.navigationLink": "http://api:5000/istos4/v1.1/Observations(2108168)/Commit(1)"  
  },  
  {  
    "@iot.id": 2108168,  
    "@iot.selfLink": "http://api:5000/istos4/v1.1/Observations(2108168)",  
    "FeatureOfInterest@iot.navigationLink": "http://api:5000/istos4/v1.1/Observations(2108168)/FeatureOfInterest",  
    "Datastream@iot.navigationLink": "http://api:5000/istos4/v1.1/Observations(2108168)/Datastream",  
    "phenomenonTime": "2024-09-27T18:00:00+00:00",  
    "resultTime": null,  
    "result": 23.75,  
    "resultQuality": 100,  
    "validTime": null,  
    "parameters": null,  
    "systemTimeValidity": "2024-11-10 08:49:43.484626+00/2024-11-15 08:49:56.901067+00",  
    "Commit@iot.navigationLink": "http://api:5000/istos4/v1.1/Observations(2108168)/Commit(2)"  
  },  
  {  
    "@iot.id": 2108168,  
    "@iot.selfLink": "http://api:5000/istos4/v1.1/Observations(2108168)",  
    "FeatureOfInterest@iot.navigationLink": "http://api:5000/istos4/v1.1/Observations(2108168)/FeatureOfInterest",  
    "Datastream@iot.navigationLink": "http://api:5000/istos4/v1.1/Observations(2108168)/Datastream",  
    "phenomenonTime": "2024-09-27T18:00:00+00:00",  
    "resultTime": null,  
    "result": 23.75,  
    "resultQuality": 100,  
    "validTime": null,  
    "parameters": null,  
    "systemTimeValidity": "2024-11-15 08:49:56.901067+00/infinity",  
    "Commit@iot.navigationLink": "http://api:5000/istos4/v1.1/Observations(2108168)/Commit(3)"  
  }  
]
```

Quickstart (branch edu_traveltime)

Download & startup service

```
» git clone -b edu_traveltime https://github.com/istSOS/istSOS4.git  
» cd edu_traveltime  
» docker-compose -f edu_docker-compose.yml up -d
```

Activate versioning

```
» docker-compose -f edu_docker-compose.yml down --volumes --rmi all  
» sed -i 's/^VERSIONING=.* /VERSIONING=true/' .env  
» docker-compose -f edu_docker-compose.yml up -d
```

Near Future planned works

- Support the STAplus extension
(data license & citizen science support)
- Integrate istSOS in «Green Deal Data Spaces»

Conclusions

- A new OGC STA service in python
- Performance of the service are better than the reference implementation
- Innovative features has been implemented:
 - A traveltime extension has been proposed and implemented to support reproducibility (persistent identifier)
 - A user authentication and authorization has been implemented to control data access (sovereignty)
 - A commit system to support (data lineage)

Thanks

Join the community as User or Developer
<https://github.com/istSOS/istSOS4/discussions>

Aknowledgements:

This research has been partially funded by SWISSUNIVERSITIES under the project OSIReS **swissuniversities**

Some content of this presentation has been elaborated after: Sylvain Grellet – BRGM ENSG – mastère GDM – 2023-01-20