

INTRODUCTION TO SENSOR WEB TECHNOLOGIES

Simon Jirka

Geospatial Sensing 2020

Virtual, 2020-08-30

MOTIVATION

- More and more sensors are deployed
- Human observations
- Lots of different phenomena
 - Weather
 - Pollution
 - Biodiversity
 - Hydrological Data
 - Traffic
 - ...

MOTIVATION



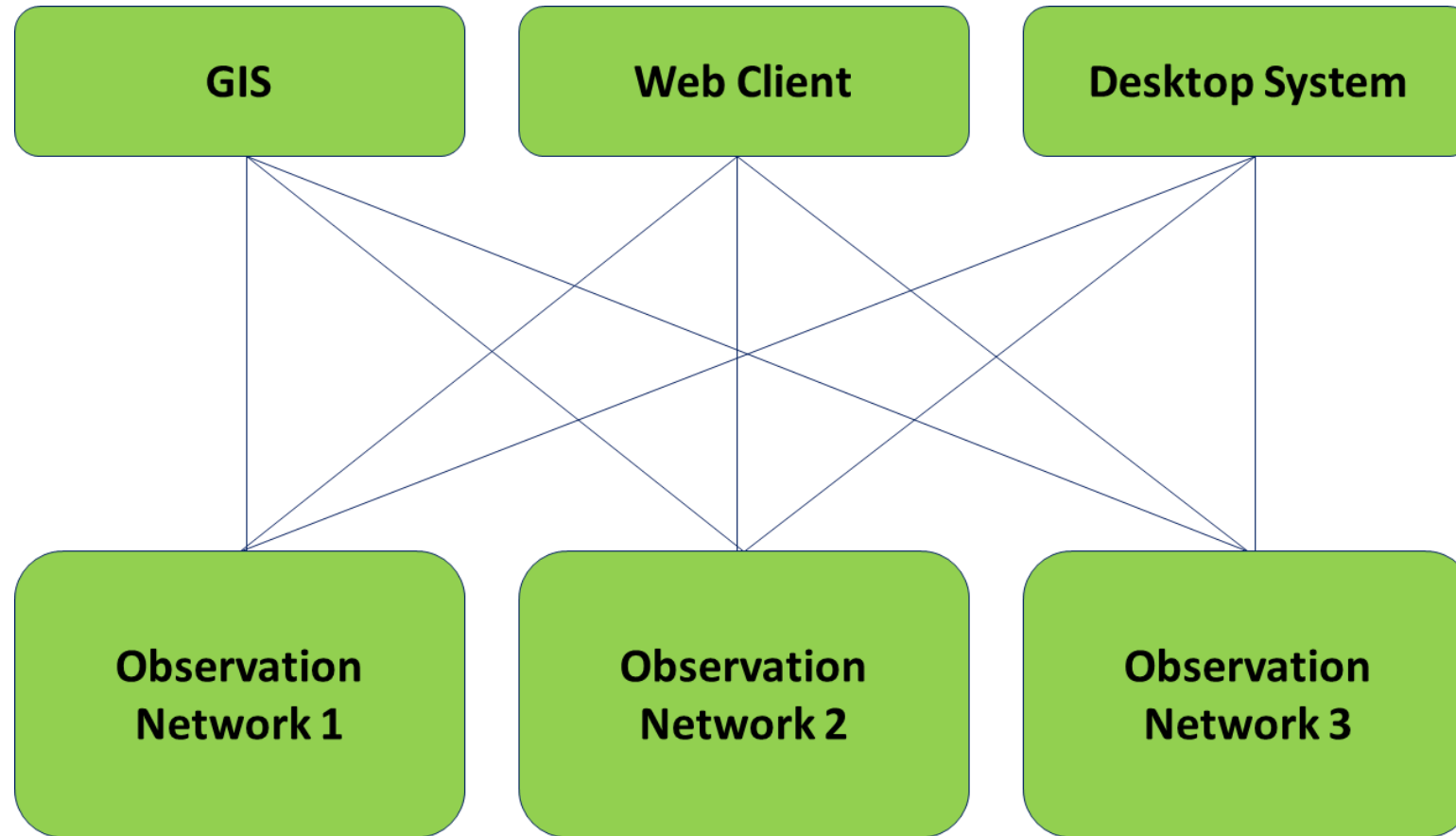
MOTIVATION

- Different structures and elements of observations
- Sensor interfaces are very heterogeneous
- Need for harmonisation → interoperability
- Provide a layer on top of observation databases and low level sensor network technology → integration into web applications

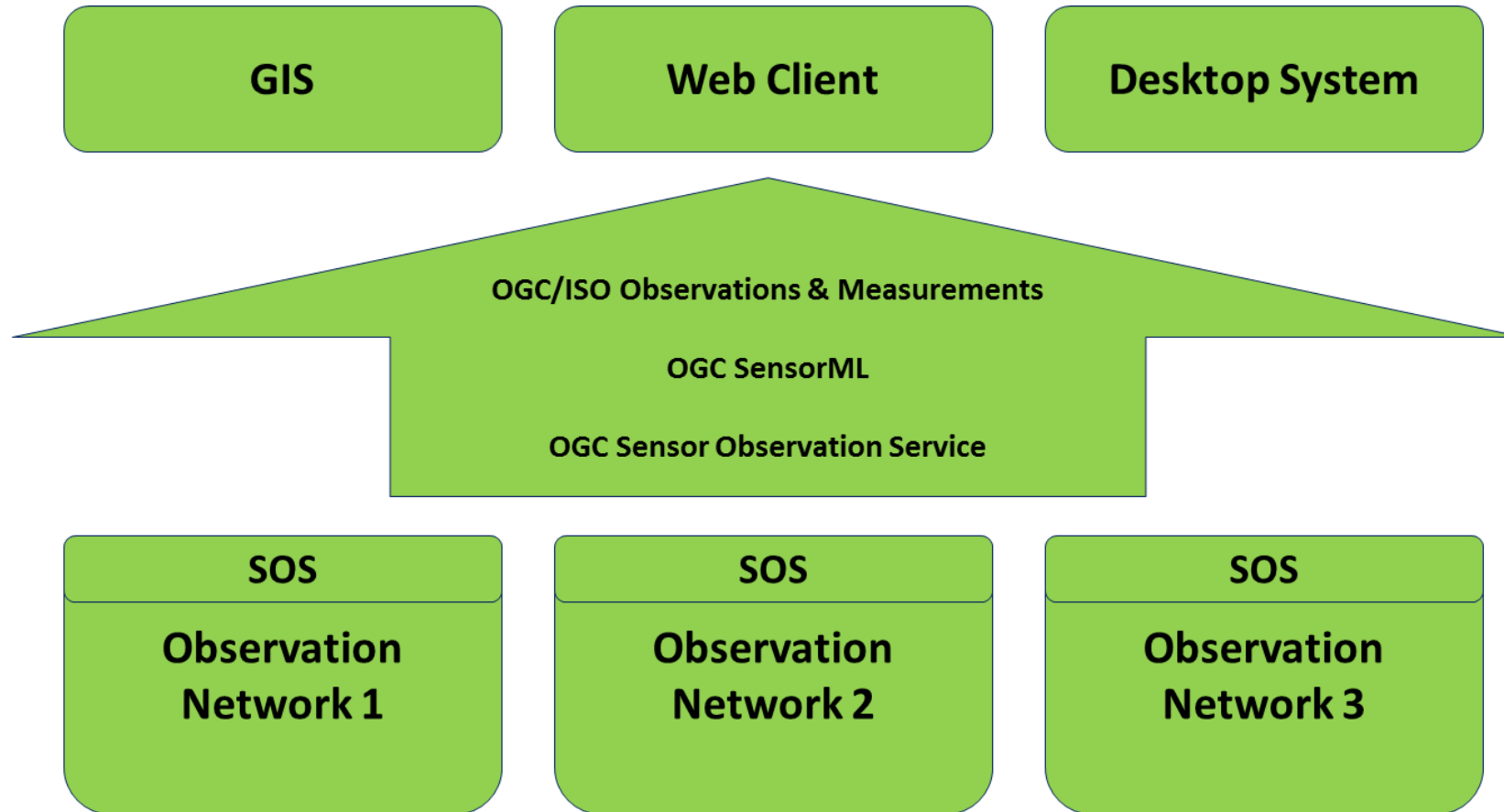
MOTIVATION

- Traditional services allow to
 - request maps (images)
 - Web Mapping Service
 - request (binary) raster data
 - Web Coverage Service
 - request vector data
 - Web Feature Service
- Lack of a generic framework for integration observation data into spatial data infrastructures

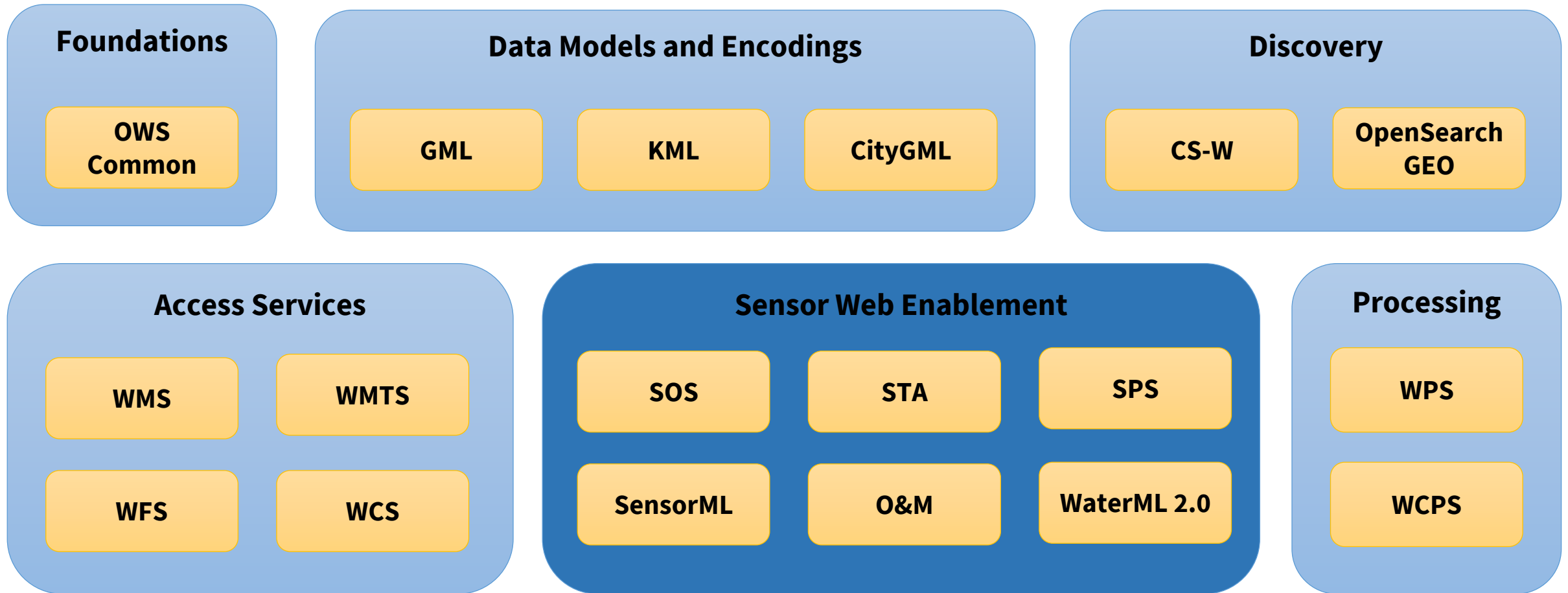
MOTIVATION



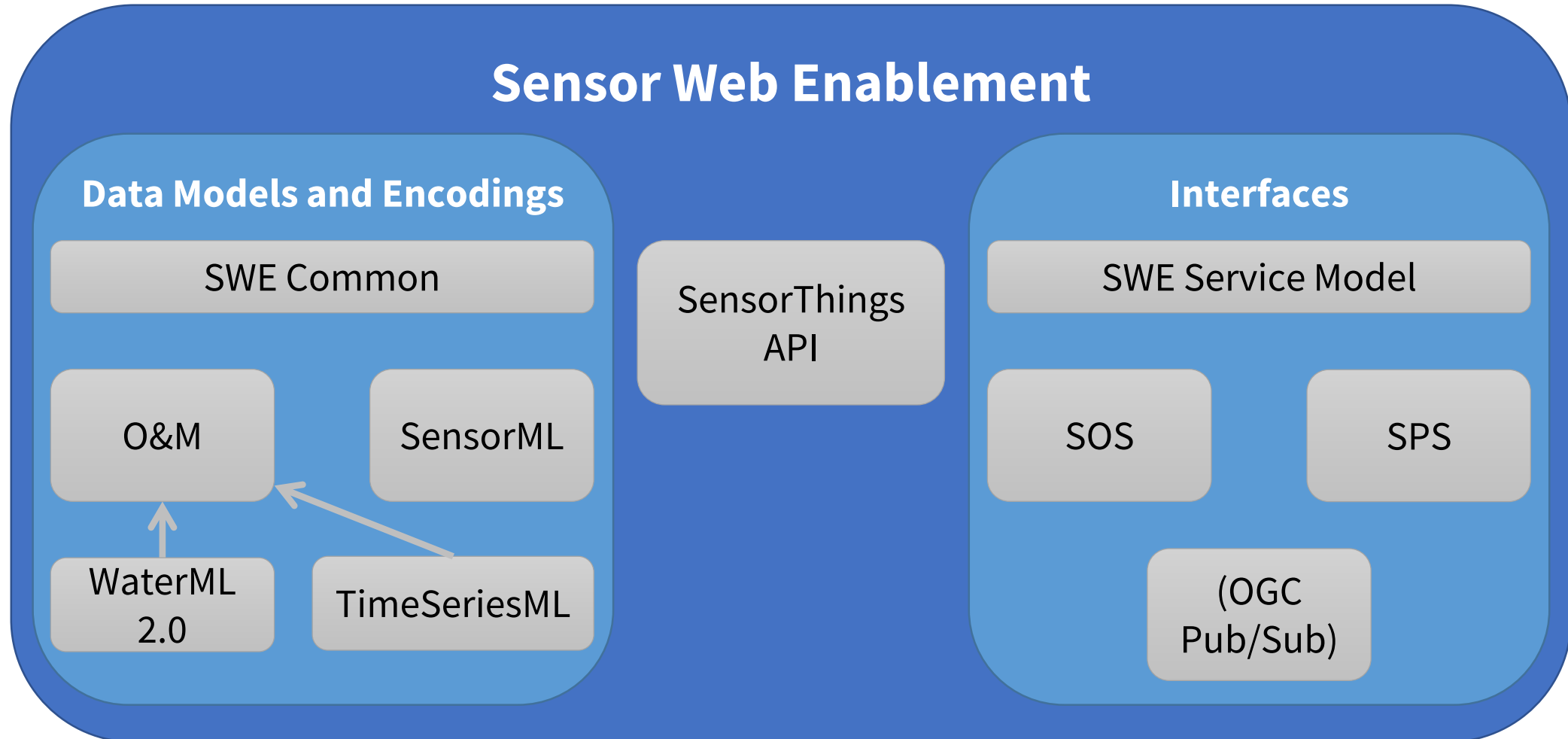
MOTIVATION



OGC ARCHITECTURE (SIMPLIFIED)

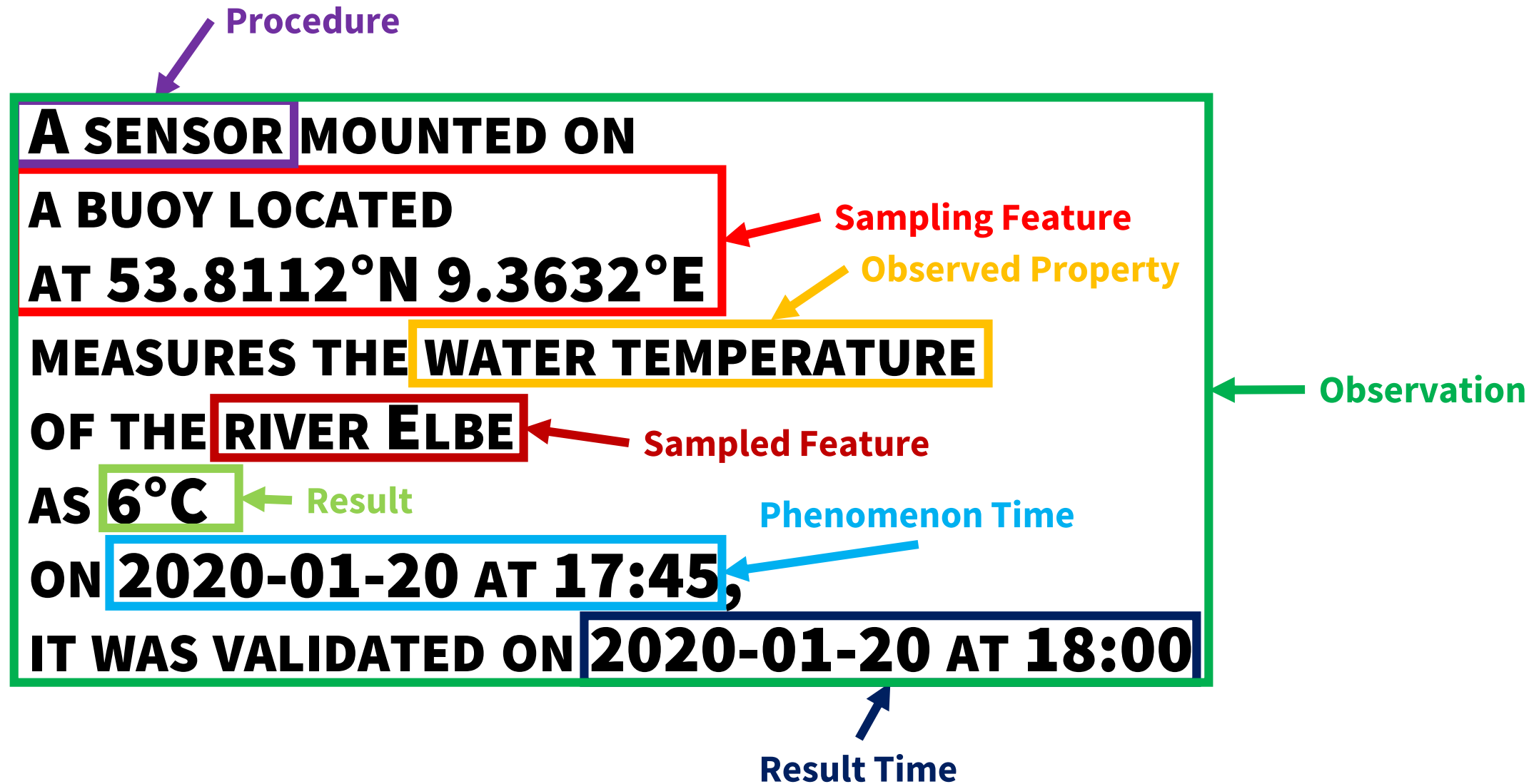


SWE BUILDING BLOCKS



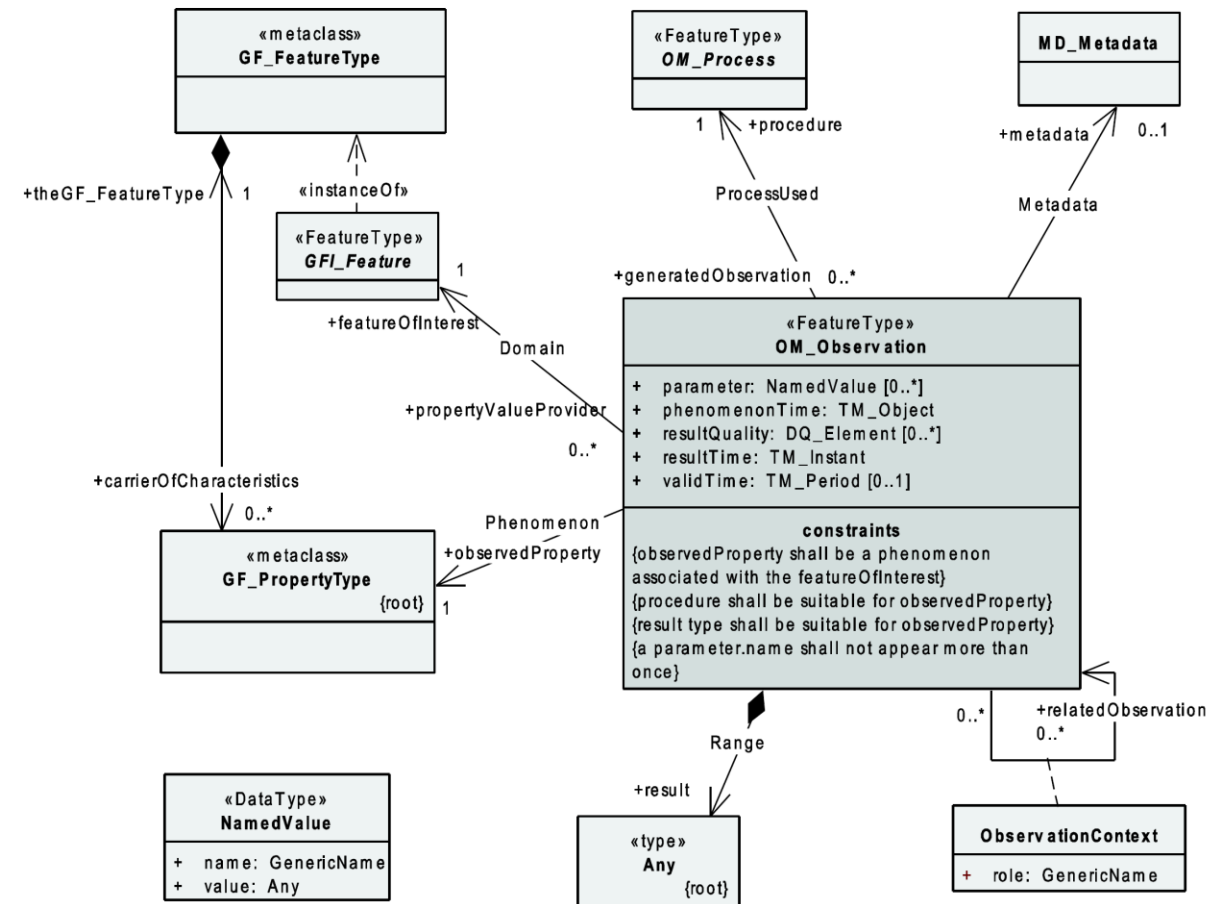
OBSERVATION & MEASUREMENTS

- Common ISO (ISO 19156:2011) and OGC (OGC 10-004r3) Standard
 - Geographic information — Observations and measurements
 - ISO 19156:2011
 - OGC 10-004r3
- Split into an abstract specification and an XML implementation
- Defines a conceptual model for observations and for features involved in sampling when making observations



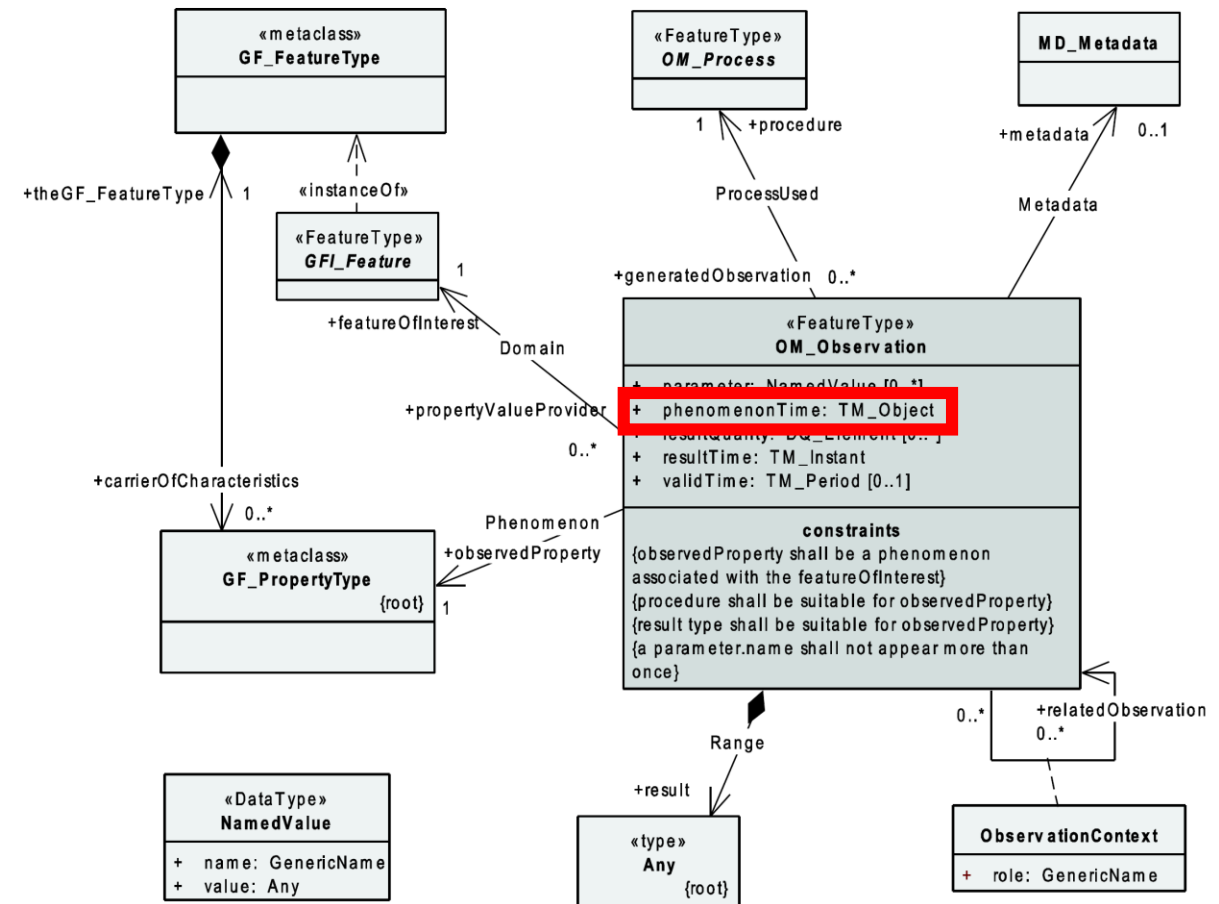
OBSERVATION

- At a specific time instant or period, a process assigns a value to a phenomenon that is a property of a feature.



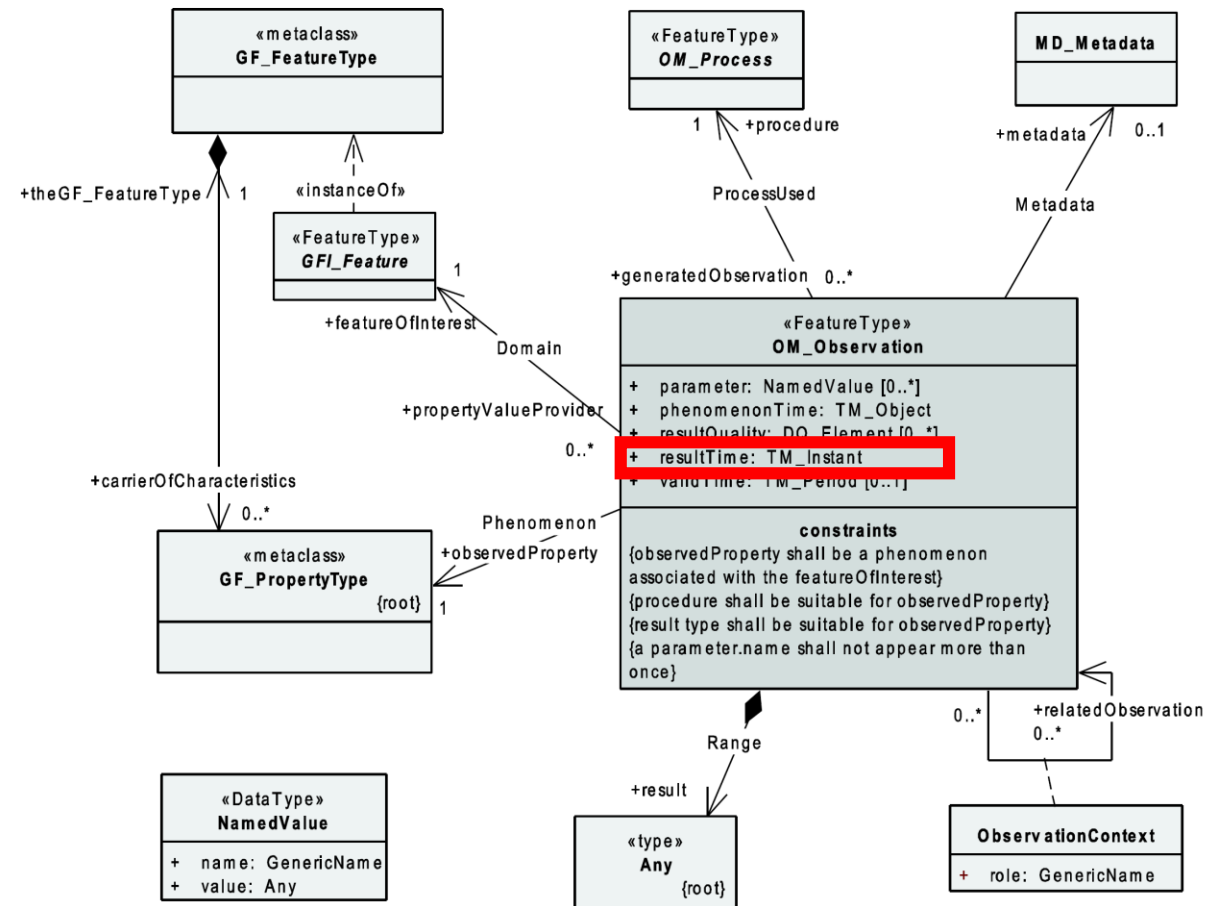
PHENOMENON TIME

- A time instant or time period that describes the time the result applies to the property of the feature of interest
- When analyzing data this is timestamp the you typically associate with the value



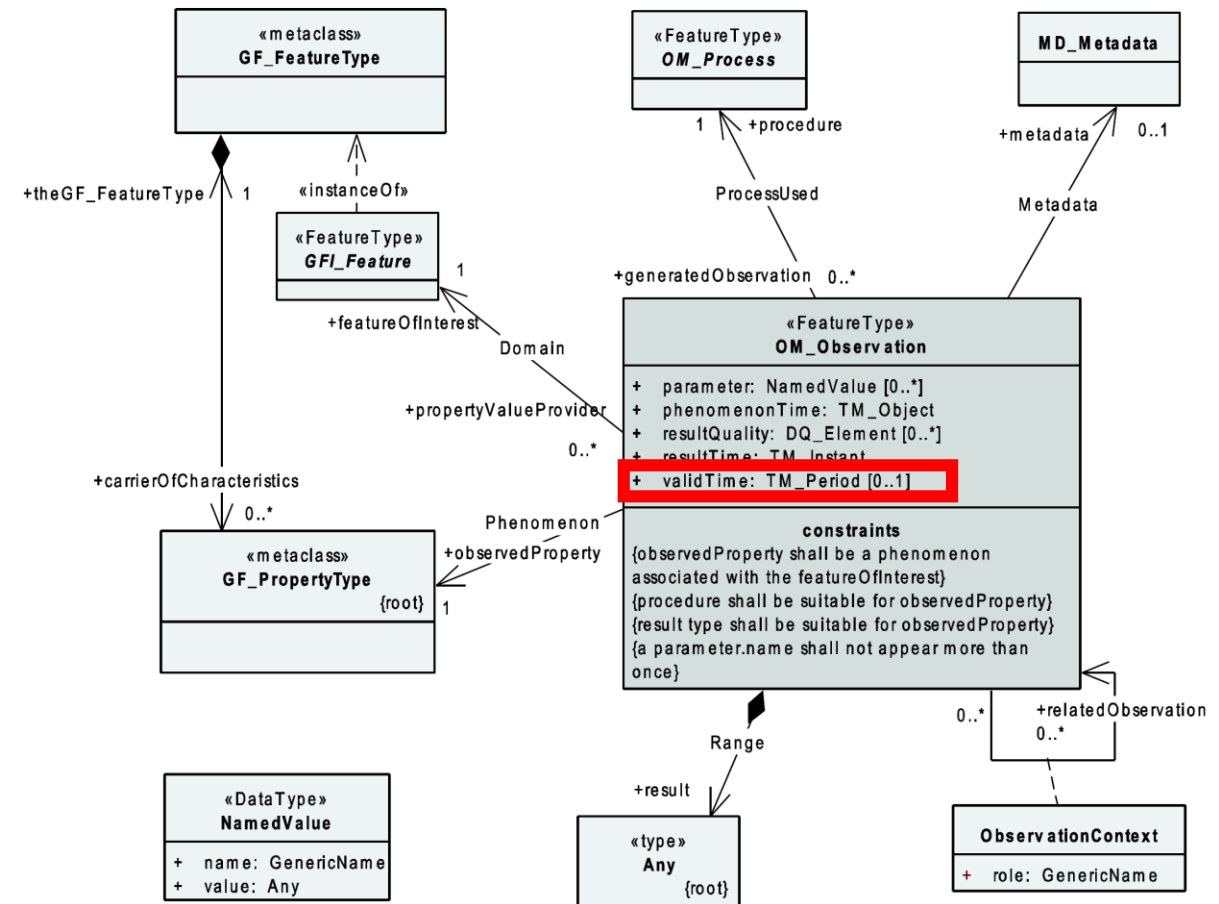
RESULT TIME

- A time instant that describes when the result became available
- Most of the time identical to the phenomenon time
- Typical use cases:
 - Post processing steps
 - Simulations
 - Specimen retrieval vs. analysis



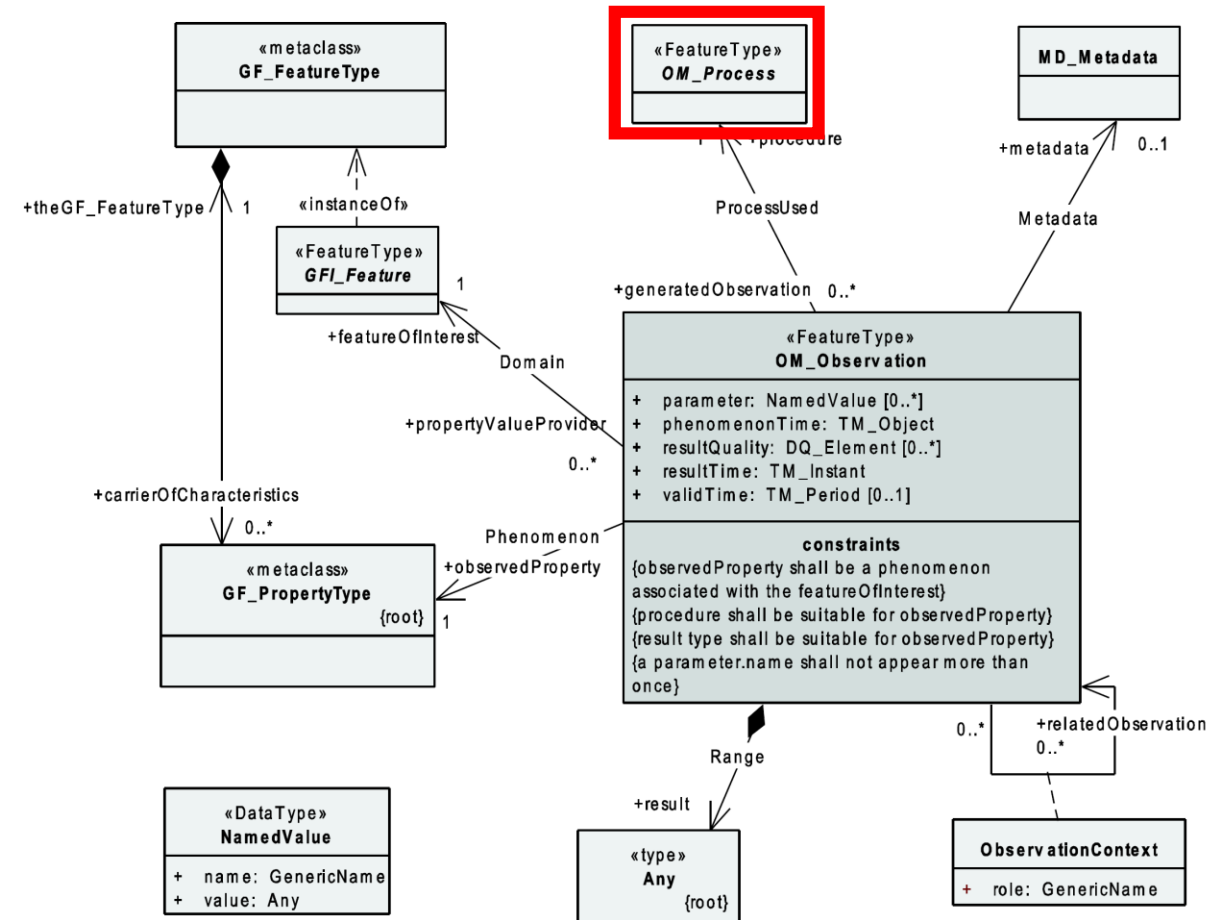
VALID TIME

- Describes the time period during which the result is intended to be used
- Typical use case:
 - The output of a forecasting model is only valid until the next model run
- Optional



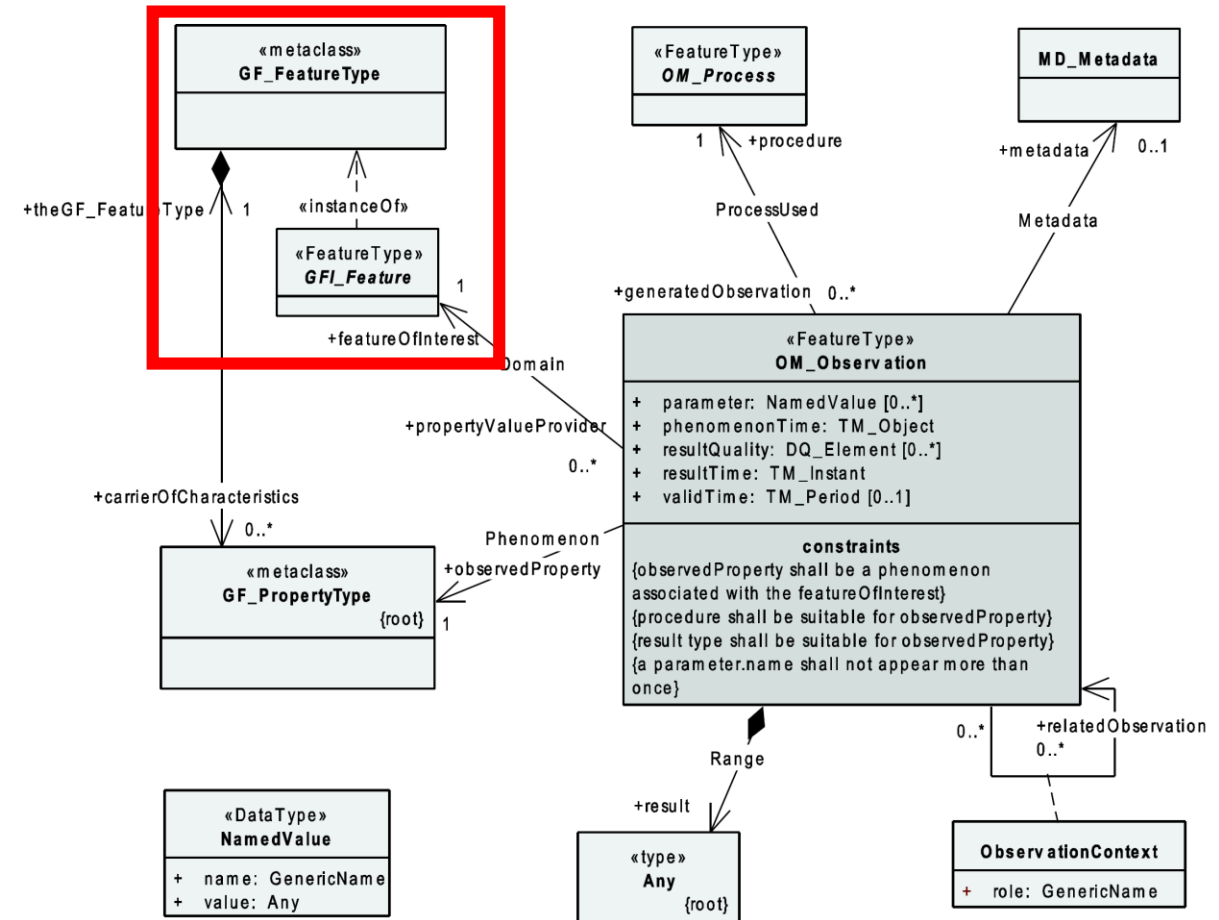
OBSERVATION PROCESS / PROCEDURE

- Abstract entity that represents the observation process and produces the observation result
- O&M doesn't make any assumptions or restrictions
- Can be anything:
 - Instrument or Sensor
 - Human Observer
 - Process or Algorithm
 - Simulation
- → SensorML



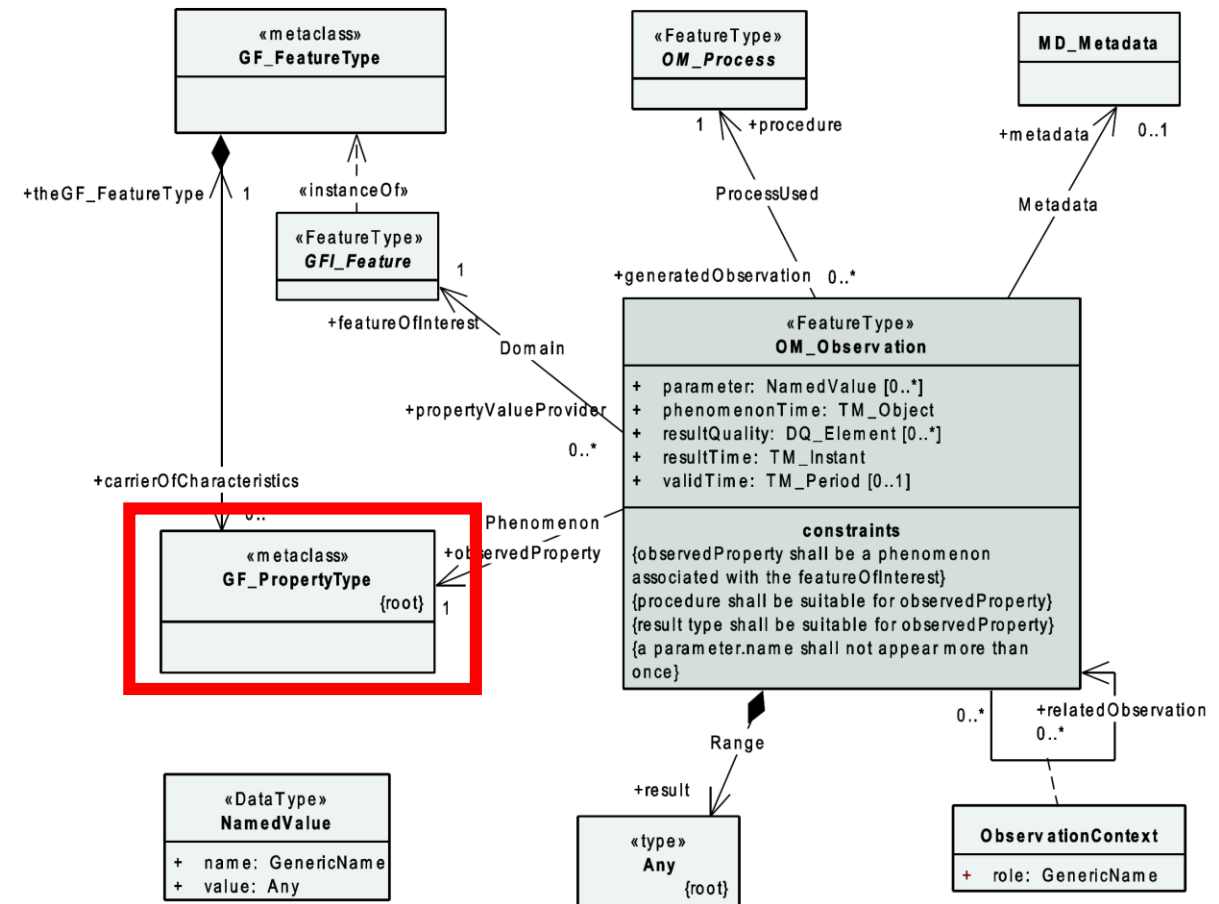
FEATURE OF INTEREST

- The feature of interest is the subject of the observation and carries the observed property
- The real-world object whose properties are under observation
- A feature intended to sample the real-world object
- Examples
 - Measurement station
 - River
 - River basin
 - ...



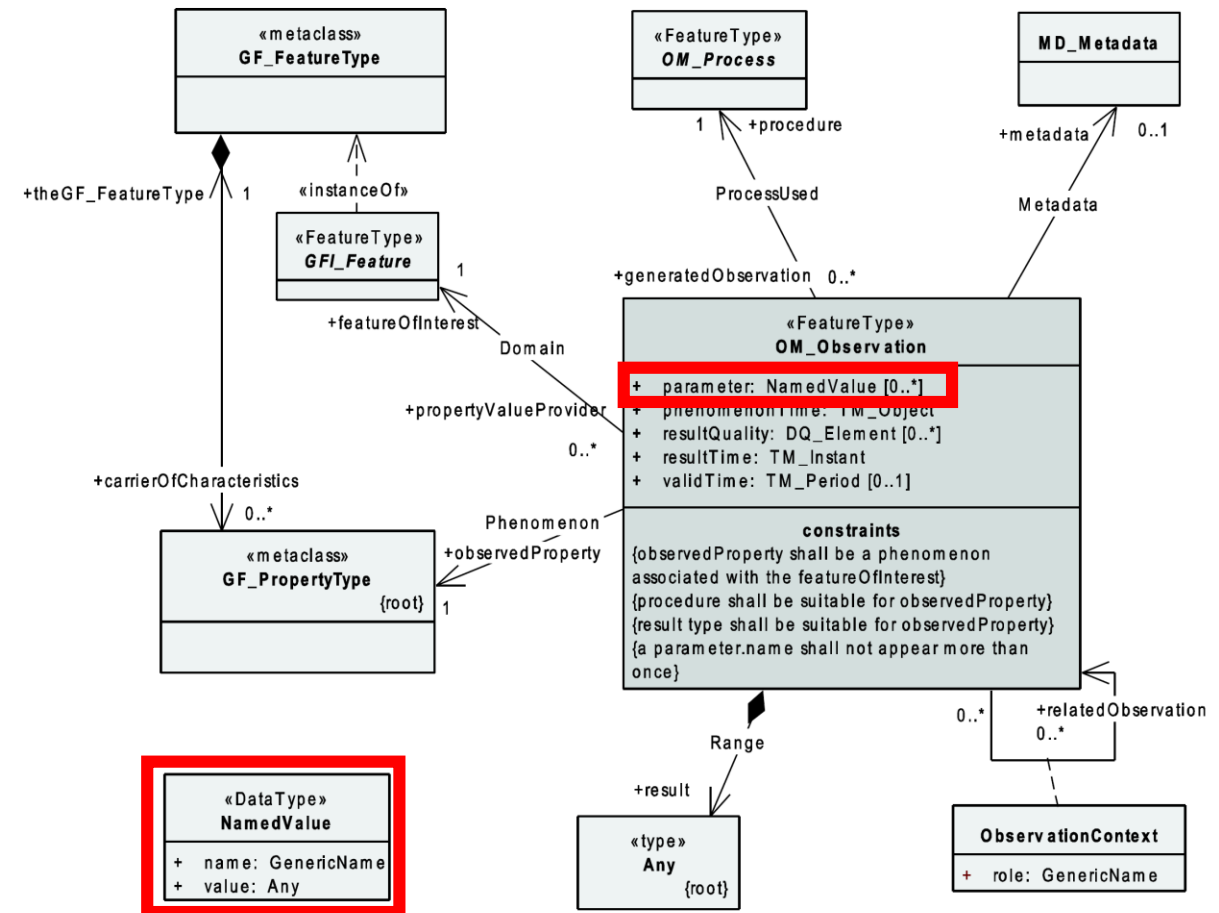
PHENOMENON / OBSERVED PROPERTY

- The phenomenon is a property of the observed feature for which the result of the observation provides an estimate
- May be organized into hierarchies or ontologies and managed in catalogues
- Should support semantica or thematic classification
- Examples:
 - “temperature”
 - <http://vocab.nerc.ac.uk/collection/P01/current/CATAVT10/>



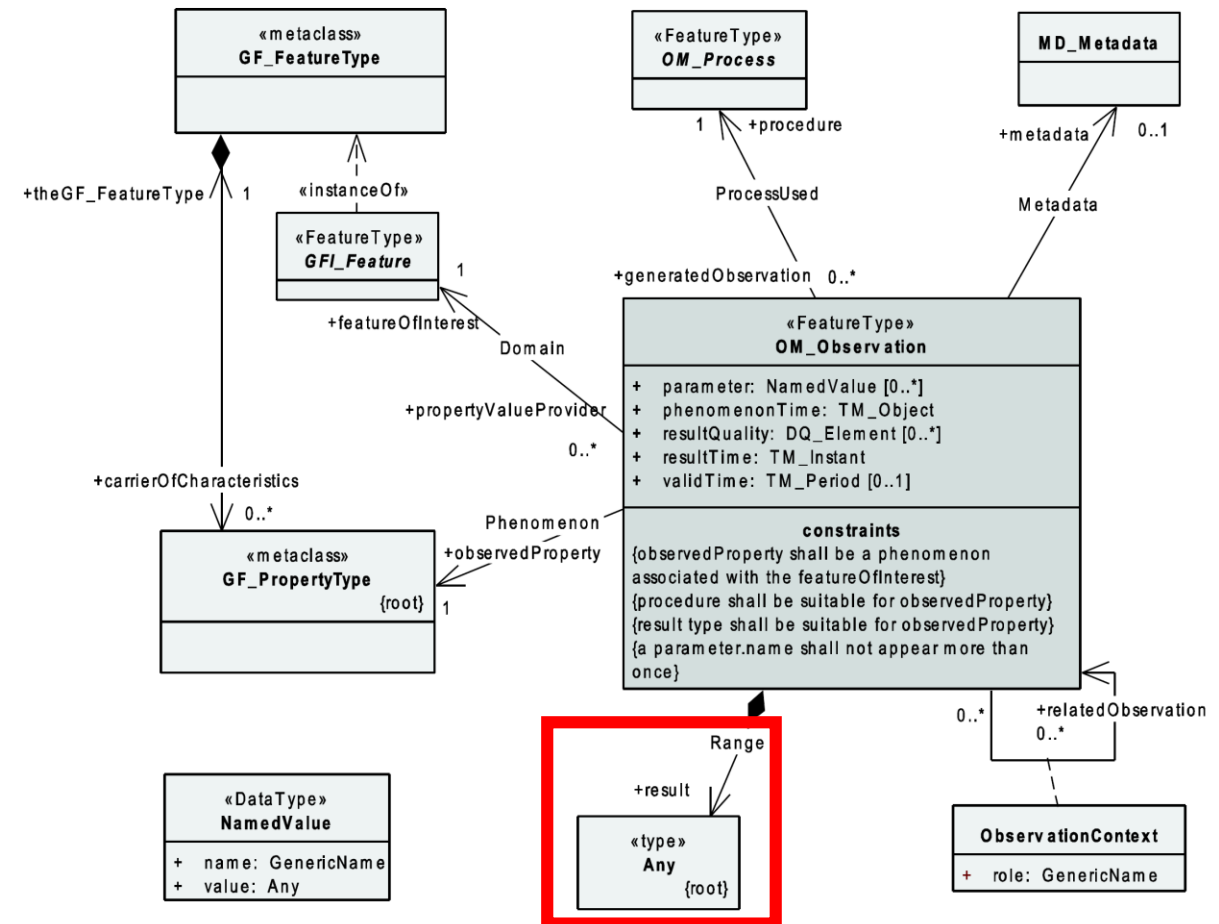
PARAMETER

- Can be used to embed arbitrary information into the observation that is not bound to the feature of interest or observation procedure
- Example use case:
 - Trajectories with the complete track as the feature of interest can describe the current position along the track
 - Measurements along a water column can describe the depth as a parameter



RESULT

- The value generated by the procedure and assigned to the observed property of the feature of interest
- O&M doesn't restrict the type the value
- But it defines some basic observation types



ENCODINGS

- Plain O&M 2.0
- OGC WaterML 2.0
- OGC TimeseriesML 1.0
- INSPIRE D2.9 Guidelines for the use of O&M

PLAIN O&M 2.0 – GENERAL XML

```
<om:OM_Observation>
  <gml:description>      <!-- description of the observation (optional) -->
  <gml:identifier>        <!-- identifier, used by GetObservationById (optional) -->
  <gml:name>              <!-- name pf the observation (optional) -->
  <om:type>               <!-- the type of the observation (optional) -->
  <om:metadata>           <!-- some metadata for the observation(optional) -->
  <om:relatedObservation> <!-- a related observation, e.g link to the raw values(optional) -->
  <om:phenomenonTime>     <!-- time when the measurement takes place -->
  <om:resultTime>         <!-- time when the observation became available-->
  <validTime>             <!-- time when the observation is valid, e.g. for forcast observations (optional) -->
  <om:procedure>          <!-- the procedure that creates the observation -->
  <om:parameter>          <!-- some addtional parameter(optional) -->
  <om:observedProperty>   <!-- the observed phenomenon-->
  <om:featureOfInterest>  <!-- the location where the measurment takes place-->
  <om:resultQuality>      <!-- information of the result quality, complex(optional) -->
  <om:result>             <!-- the result of the observation -->
</om:OM_Observation>
```

WATERML 2.0

- Data model format for exchanging hydrological data
- Based on XML
- Specialization of a more generic standard: ISO/OGC Observations and Measurement 2.0

WATERML 2.0

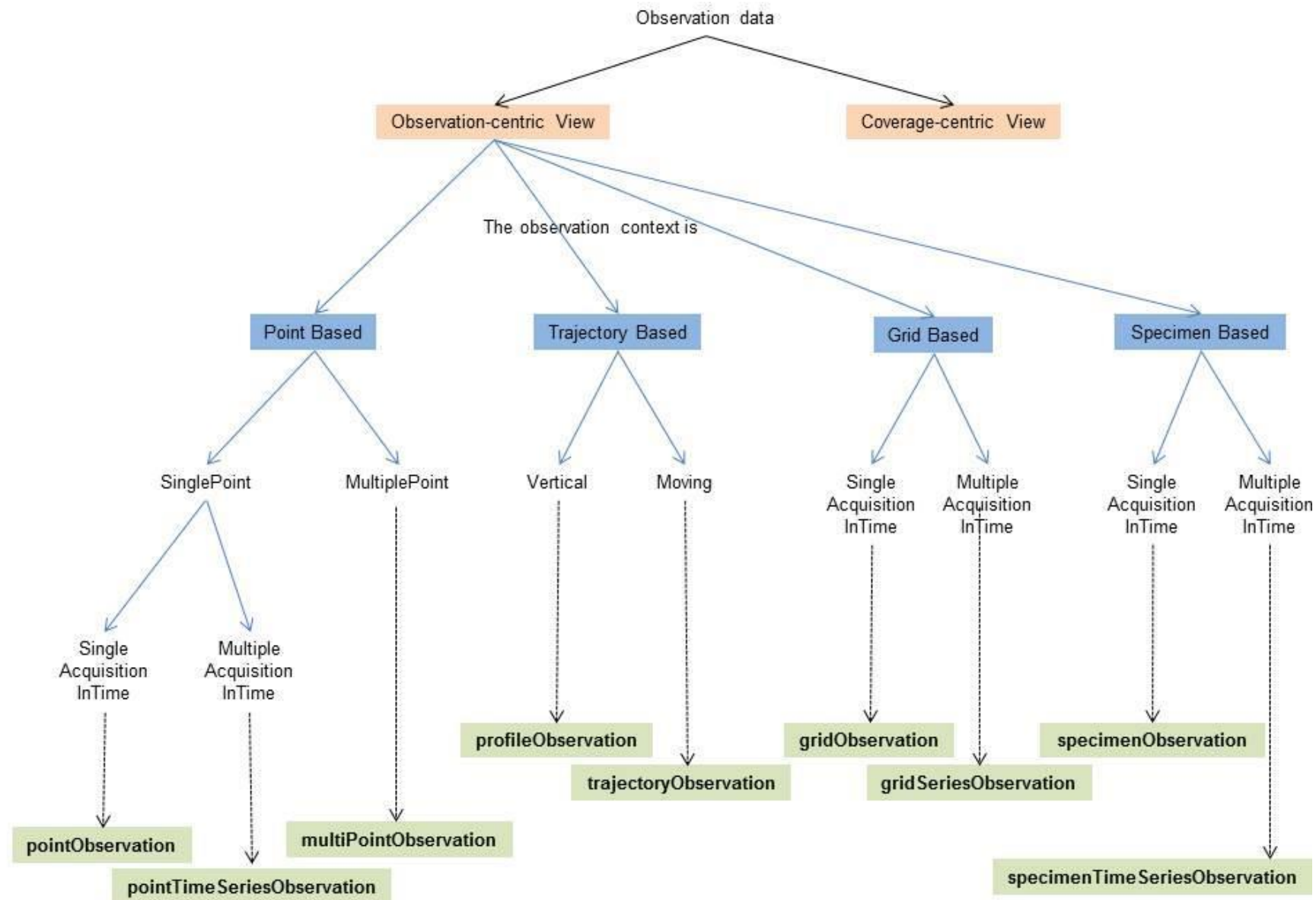
```
<om:result>
  <wml2:MeasurementTimeseries gml:id="Ki.Ts.1">
    <wml2:metadata>
      <wml2:MeasurementTimeseriesMetadata>
        <wml2:temporalExtent xlink:href="#Ki.Wml2.pTime.1"/>
        <wml2:cumulative>false</wml2:cumulative>
      </wml2:MeasurementTimeseriesMetadata>
    </wml2:metadata>
    <wml2:defaultPointMetadata>
      <wml2:DefaultTVPMeasurementMetadata>
        <wml2:quality xlink:href="http://www.opengis.net/def/WaterML/2.0/quality/good" xlink:title="Good"/>
        <wml2:uom code="m3/s"/>
        <wml2:interpolationType xlink:href="http://www.opengis.net/def/waterml/2.0/interpolationType/AveragePrec" xlink:title="Average In Preceeding Interval"/>
        <wml2:aggregationDuration>P1D</wml2:aggregationDuration>
      </wml2:DefaultTVPMeasurementMetadata>
    </wml2:defaultPointMetadata>
    <wml2:point>
      <wml2:MeasurementTVP>
        <wml2:time>2000-01-01T00:00:00.000Z</wml2:time>
        <wml2:value>266</wml2:value>
      </wml2:MeasurementTVP>
    </wml2:point>
    <wml2:point>
      <wml2:MeasurementTVP>
        <wml2:time>2000-01-02T00:00:00.000Z</wml2:time>
        <wml2:value>266</wml2:value>
      </wml2:MeasurementTVP>
    </wml2:point>
  </wml2:MeasurementTimeseries>
</om:result>
```


TIMESERIESML 1.0

- TimeseriesML → standard derived from WaterML 2.0
- Data model format for exchanging timeseries data
- Based on XML

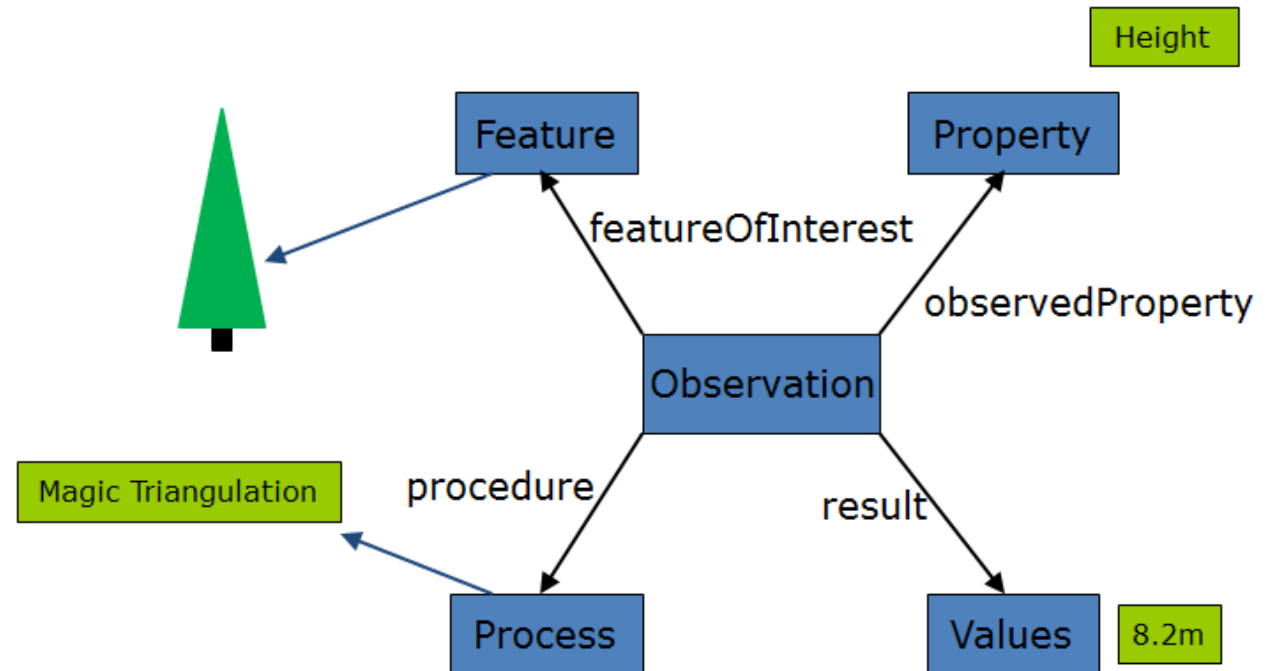
INSPIRE - D2.9 GUIDELINES FOR THE USE OF O&M

- INSPIRE – Spatial Data Infrastructure of the European Commission
- Several themes with measured, modelled and simulated data
- Technical Guidelines for data interoperability
 - Services (e.g. SOS as Download service)
 - Data formats
- D2.9 - Guideline for observation data
 - Based on OGC O&M



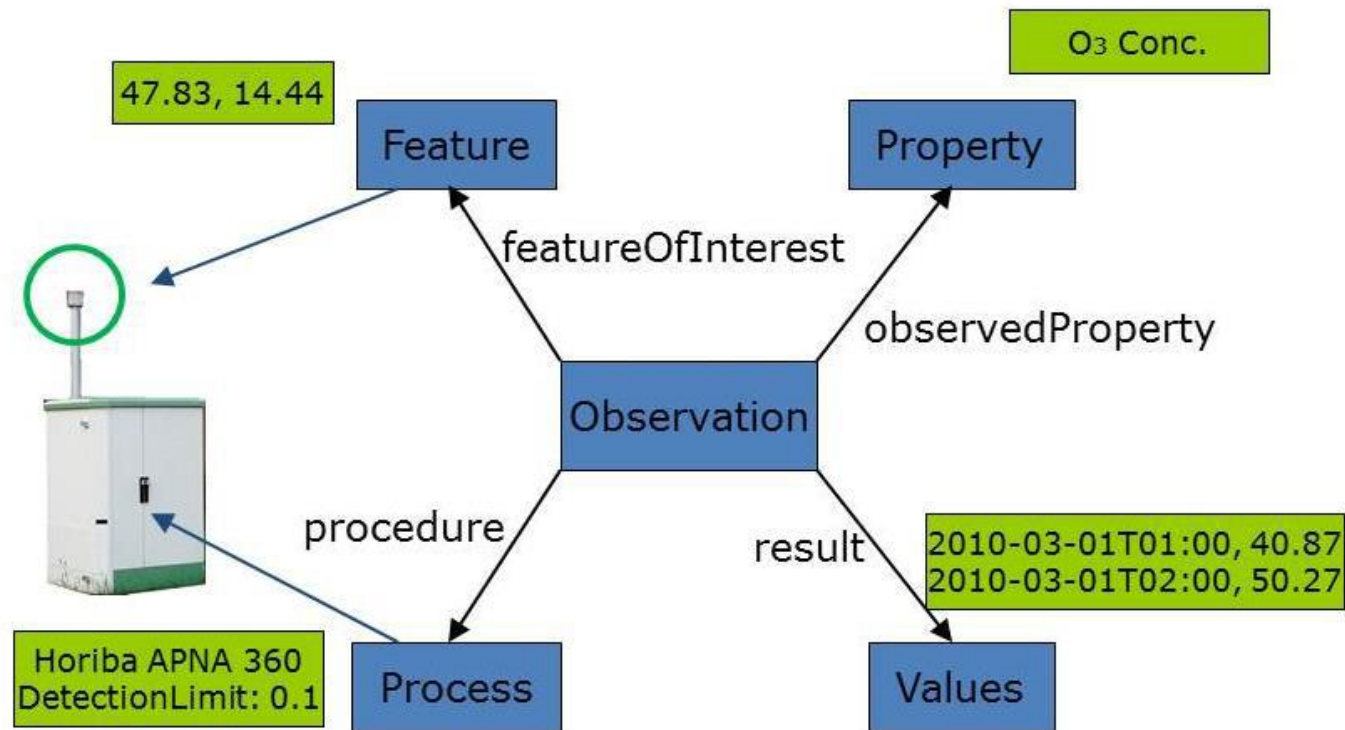
POINTOBSERVATION

- Similar to a simple O&M observation (e.g. measurement)
- Result additionally contains a point (location)



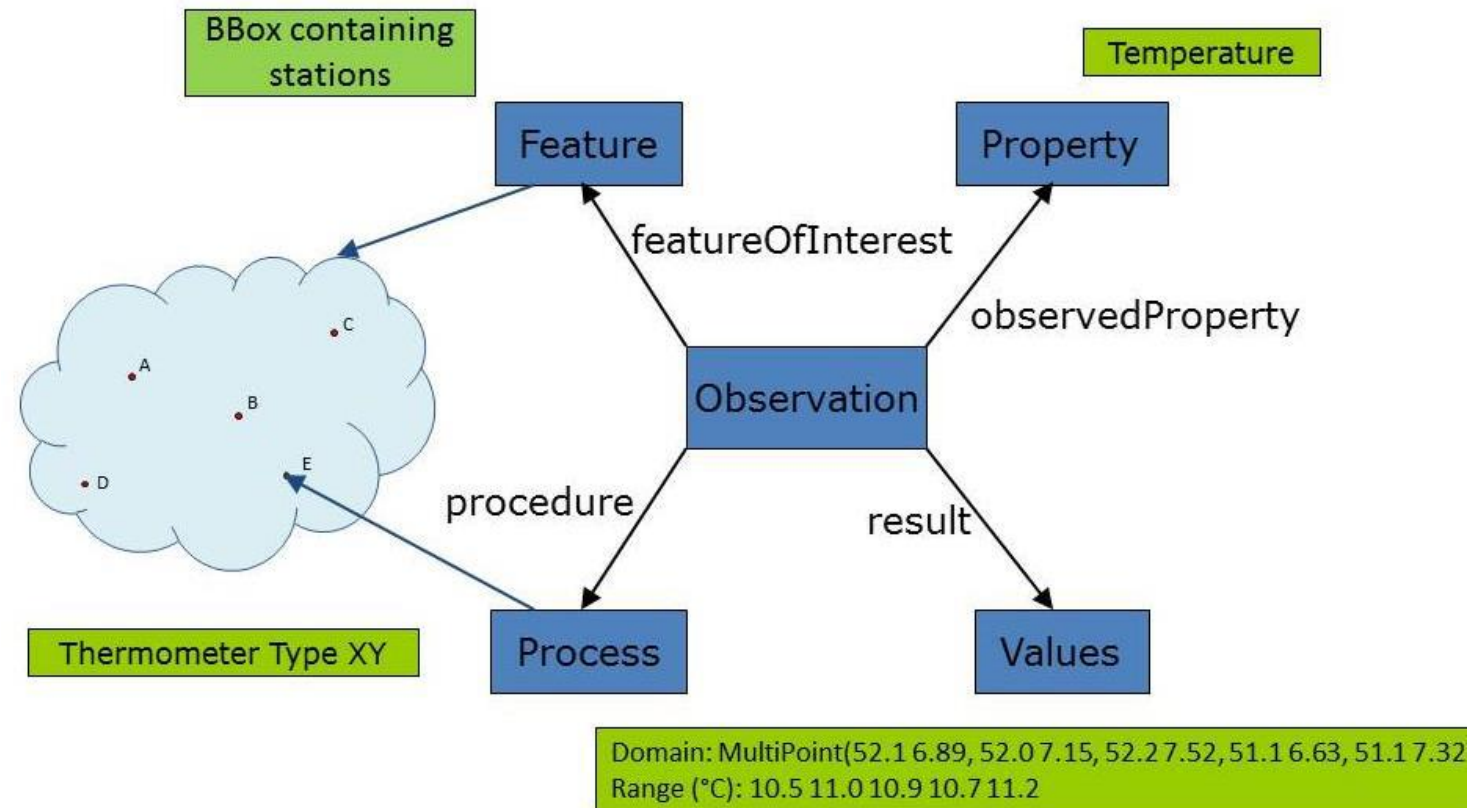
POINTTIMESERIESOBSERVATION

- Result is a WaterML 2.0 Timeseries



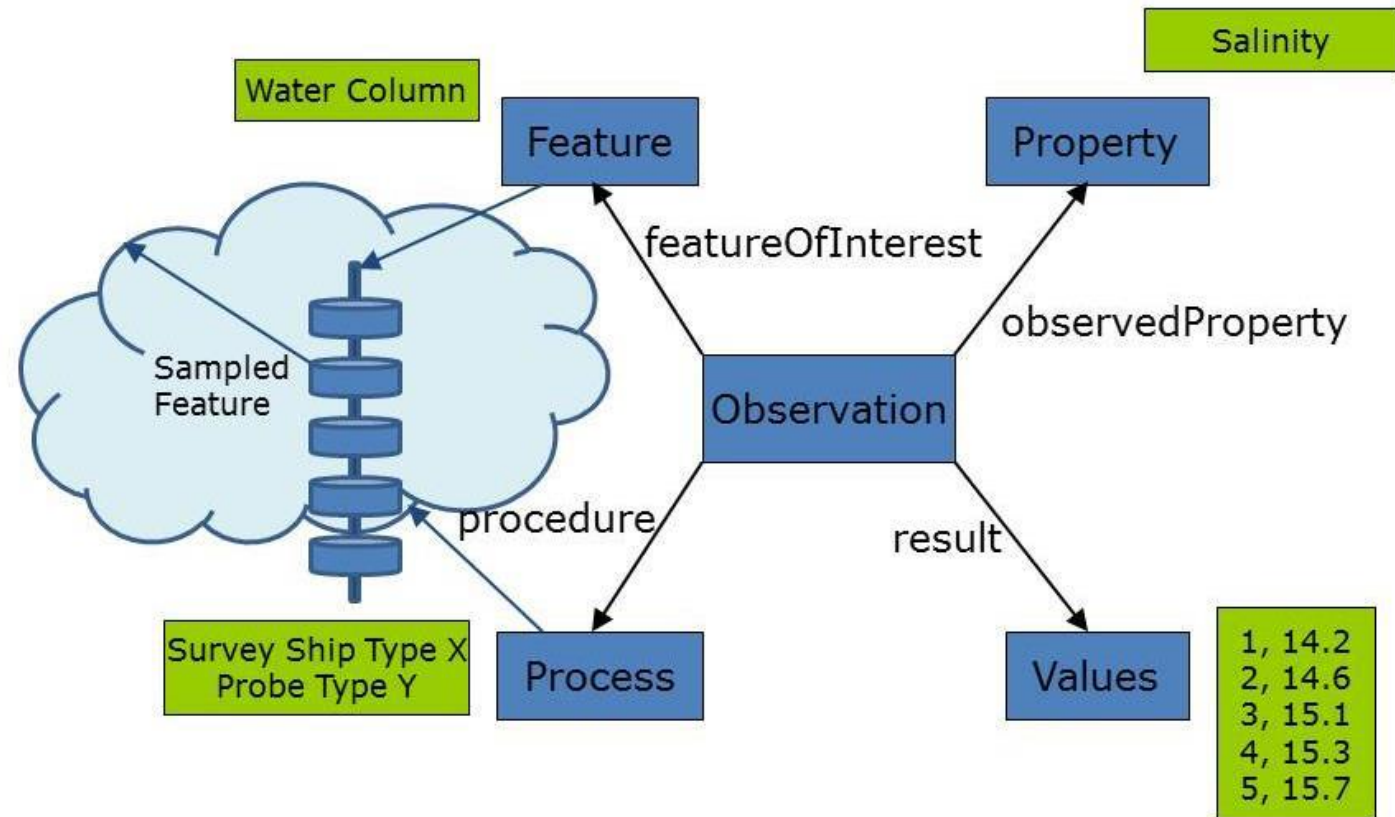
MULTIPOINTOBSERVATION

- Measuring the same parameter at the same time at different location



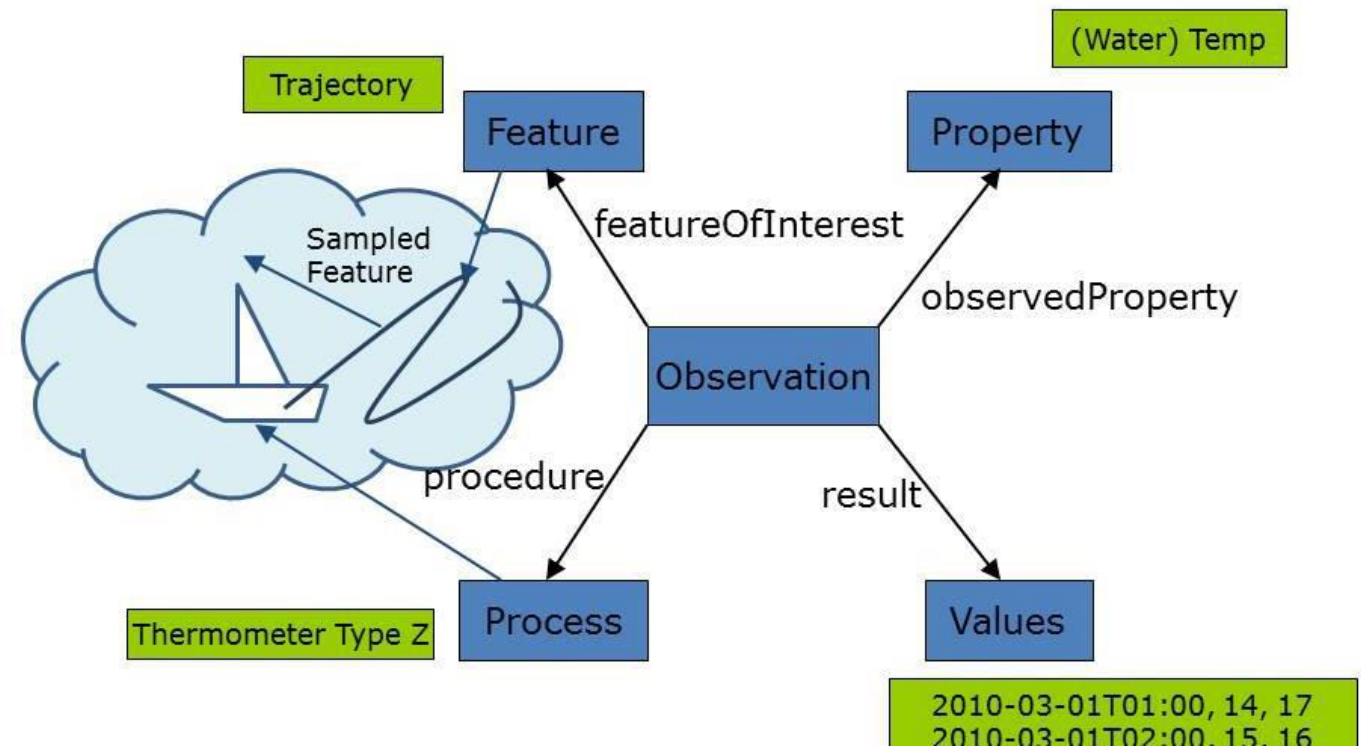
PROFILE OBSERVATION

- Measuring values at varying depths along a water column.



TRAJECTORYOBSERVATION

- Result is a extension of WaterML 2.0 Timeseries with a TimeLocationValuePair



SWE SERVICES

- OGC Sensor Observation Service (SOS)
- OGC SensorThings API (STA) Part 1: Sensing

OGC SENSOR OBSERVATION SERVICE (SOS)

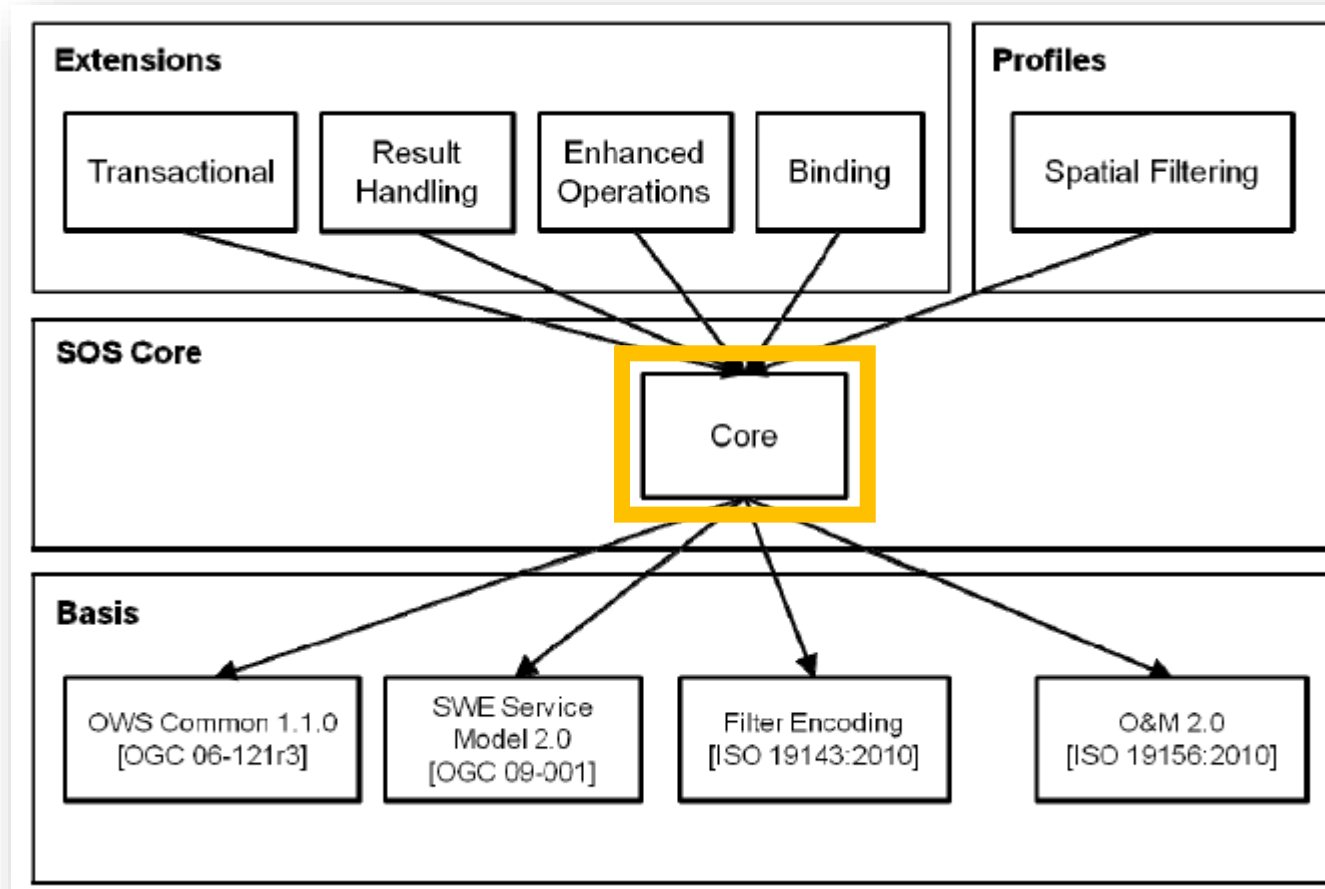
Sos OVERVIEW

- Access to sensor data
- Consistent interface and data format for all kinds of sensors
- Returns O&M
 - Contrast to WFS: no generic schemas
 - Interoperability
 - A priori-knowledge
- SOS 1.0
- SOS 2.0 → latest version of the standard

SOS INTRODUCTION

- Pull-based access to observations
- Mediator between:
 - client ⇔ data archive / simulation / real-time sensor system
- Hides the heterogeneous structure of proprietary sensor data formats and protocols
- Data formats: O&M and SensorML
 - Versions: 1.0 and 2.0

SOS 2.0 MODEL



SOS - CORE

- GetCapabilities operation
 - Returns a metadata document describing the SOS instance, e.g.
 - Supported operations
 - Spatial, temporal and thematic content → offerings
 - Mandatory for every OGC Web Service
 - Structure and syntax from OWS Web Services Common specification

GETCAPABILITIES

- Examples based on XML implementation:
 - Request: <http://schemas.opengis.net/sos/2.0/examples/core/GetCapabilities1.xml>
 - Response: http://schemas.opengis.net/sos/2.0/examples/core/GetCapabilities1_response.xml
- Example request encoded as KVP:
 - `http://hostname:port/path?
service=SOS&
request=GetCapabilities&
AcceptVersions=2.0.0`

GETOBSERVATION

- GetObservation operation
 - Returns sensor data
 - Response format is usually a O&M-document
 - Supports comprehensive filter criteria
 - Feature of interest
 - Temporal filters
 - Spatial filters
 - Observed property
 - Sensor/procedure
 - Offerings

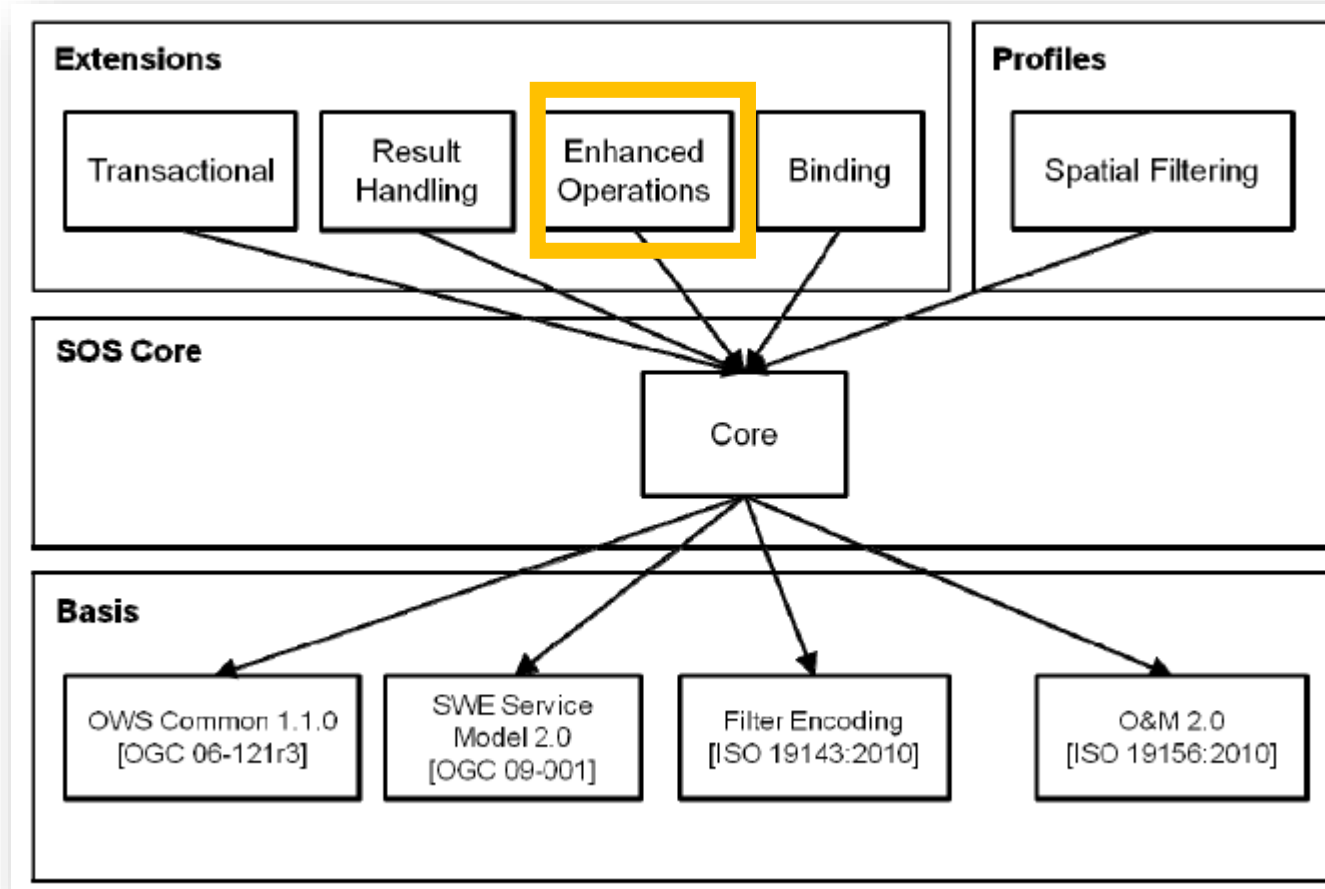
DESCRIBESENSOR

- DescribeSensor operation
 - Returns detailed metadata of sensors and measurement processes
 - Response format is usually SensorML
 - Query parameters
 - Sensor/procedure id
 - Temporal filter (optional)

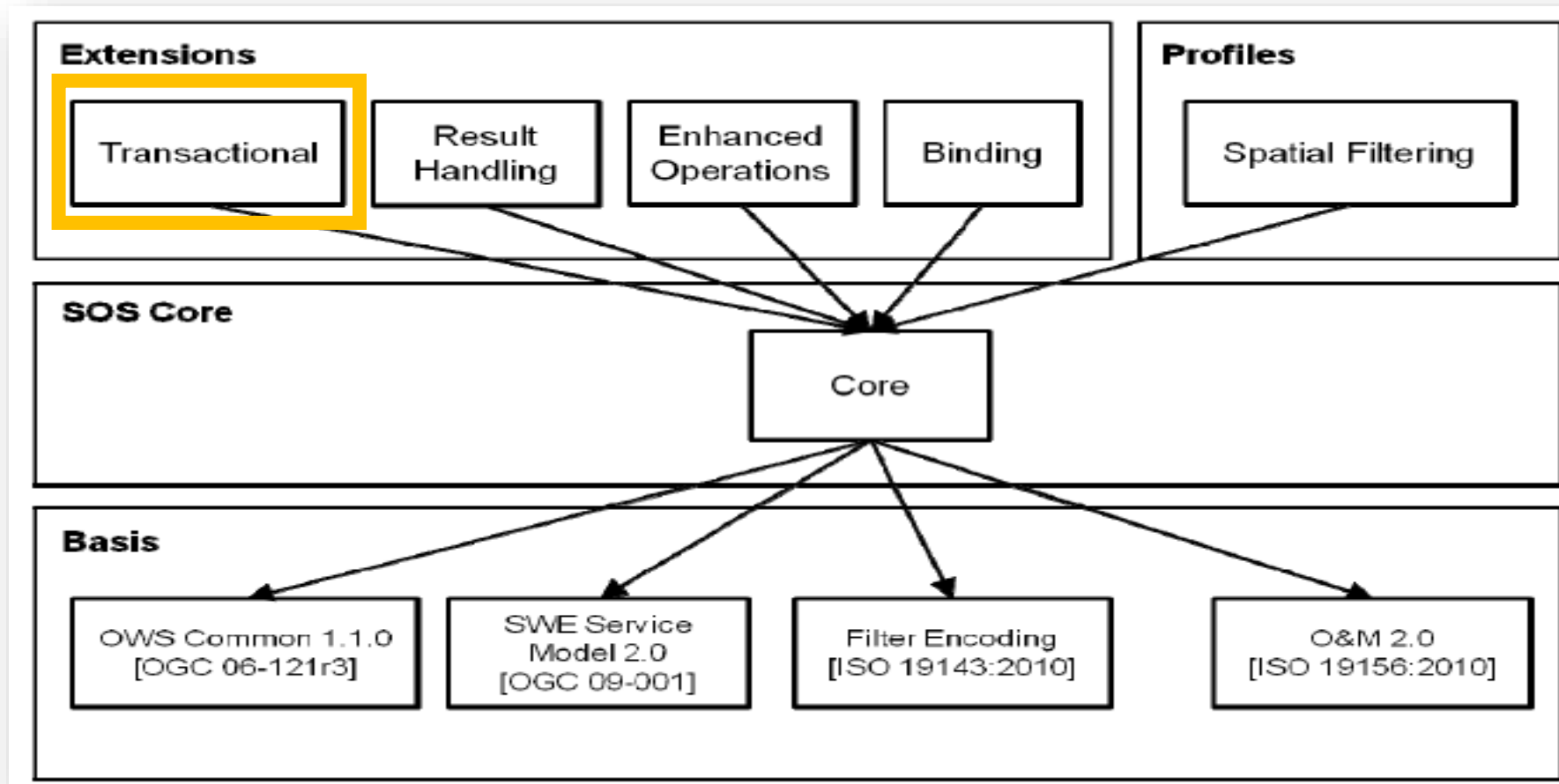
DESCRIBESSENSOR

- Example based on XML implementation:
 - Request: http://schemas.opengis.net/sos/2.0/examples/SOAP/DescribeSensor1_SOAP.xml
- Example request encoded as KVP:
 - `http://www.myserver.org:port/path?
service=SOS&version=2.0.0&
request=DescribeSensor&
procedure=urn:ogc:object:Sensor:
MyOrg:thermometer1&
procedureDescriptionFormat=
http://www.opengis.net/sensorML/1.0.1&
validTime=2010-01-01T18:31:42Z`

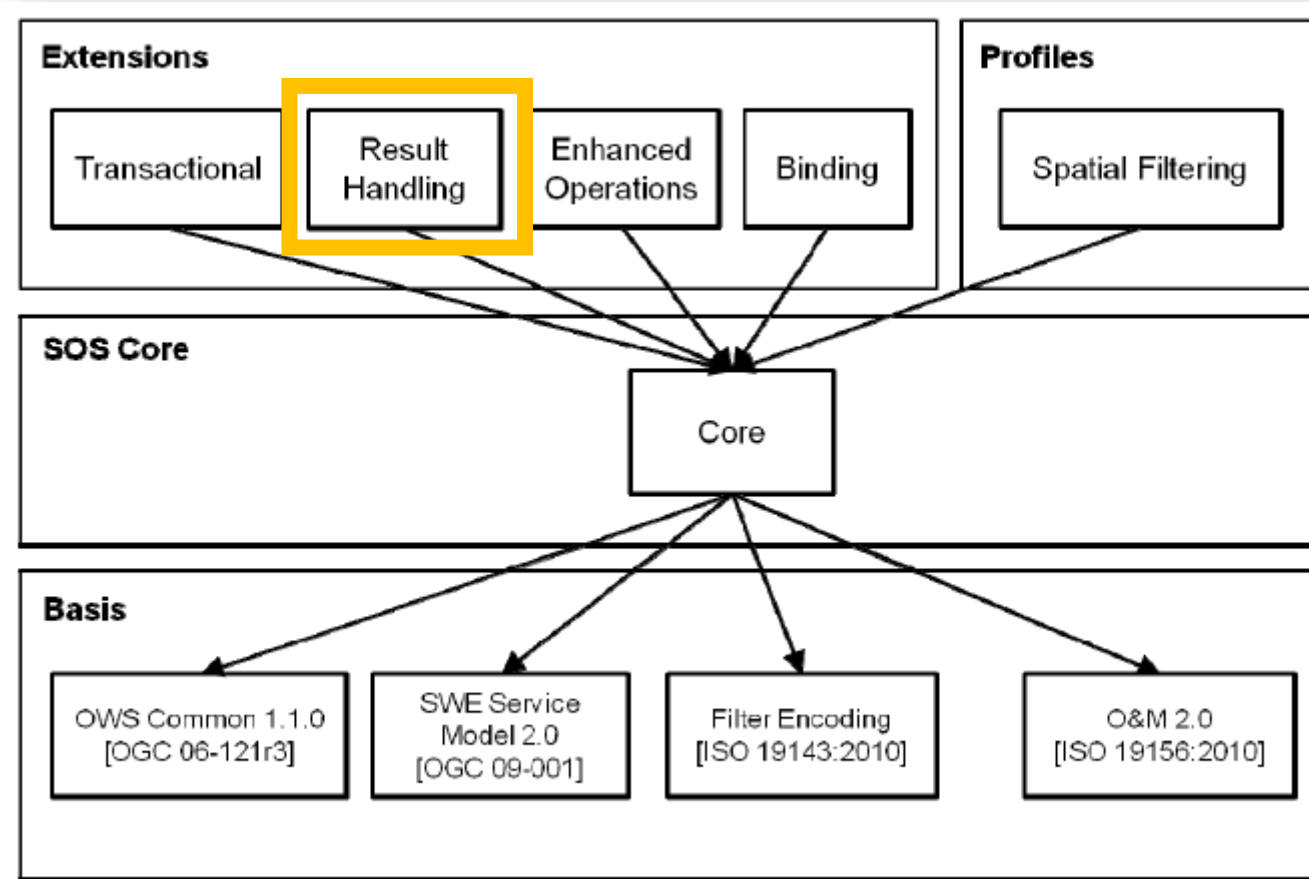
SOS 2.0 MODEL



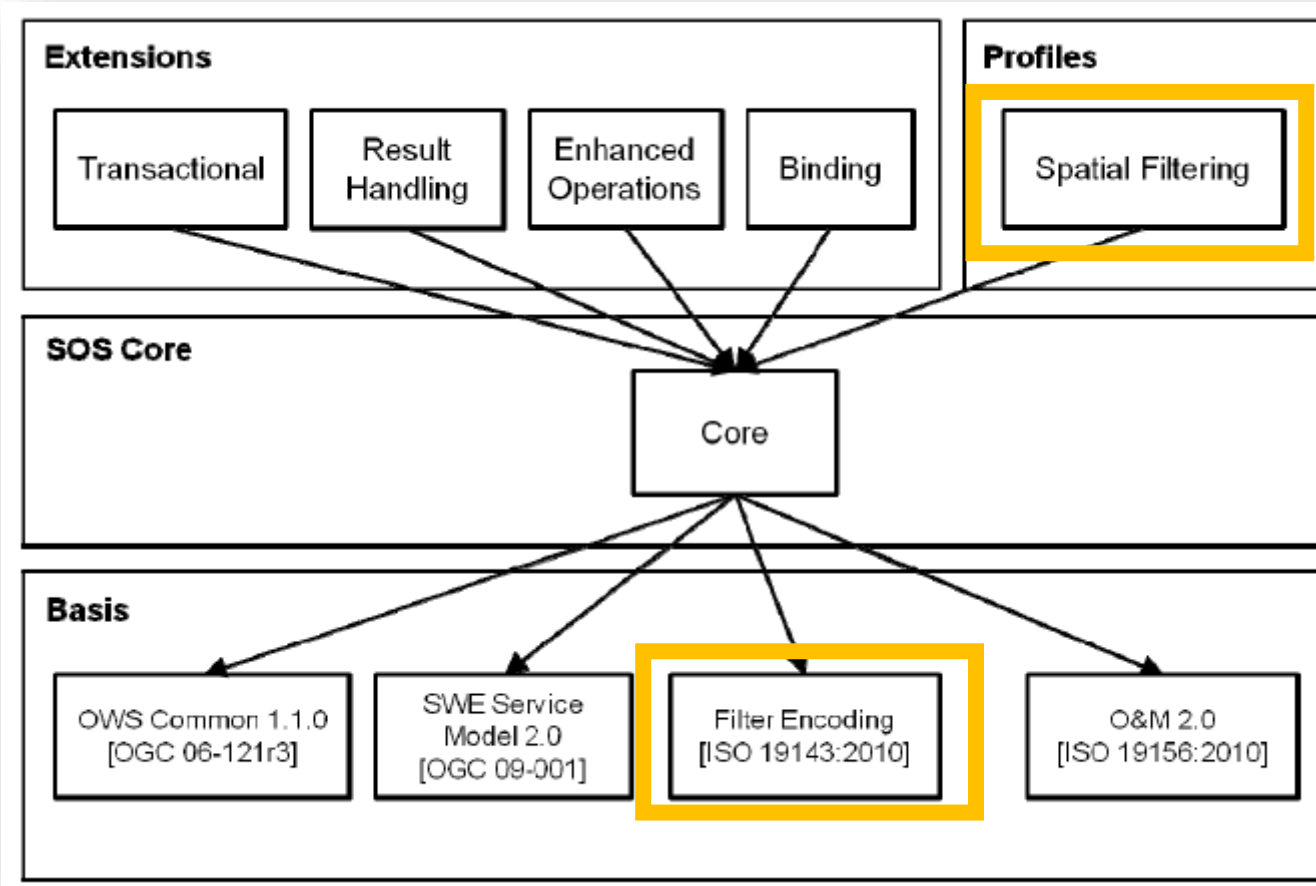
SOS 2.0 MODEL



SOS 2.0 MODEL



SOS 2.0 MODEL



SPATIAL FILTERING PROFILE

- O&M XML – SpatialObservation
 - om:parameter with sampling location

```
<om:parameter>
  <om:NamedValue>
    <om:name xlink:href="http://www.opengis.net/def/param-name/OGCOM/2.0/samplingGeometry"/>
    <om:value>
      <gml:Point gml:id="SamplingPoint">
        <gml:pos srsName="urn:ogc:def:crs:EPSG::4326">52.9 7.52</gml:pos>
      </gml:Point>
    </om:value>
  </om:NamedValue>
</om:parameter>
```

- Filtering with valueReference value:
 - <http://www.opengis.net/req/omxml/2.0/data/samplingGeometry>

BINDINGS

- Official bindings
 - KVP
 - SOAP
- 52N SOS additionally supported bindings
 - POX (official only for SOS 1.0.0)
 - JSON (52N SOS, not yet specified)

OGC SENSORTHINGS API (STA)

PART 1: SENSING

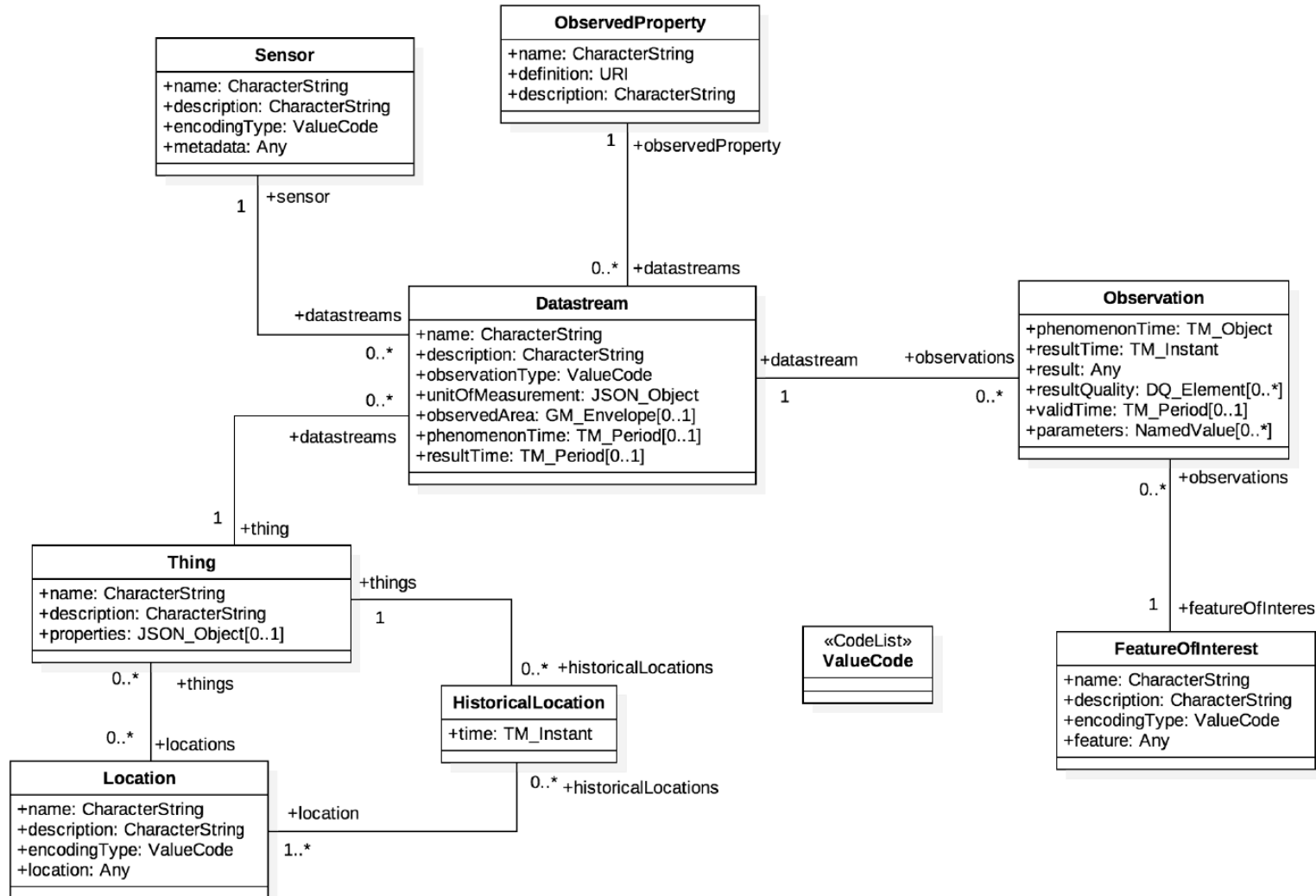
STA - OVERVIEW

- OGC SensorThings API
- Complementary specification to enhance the OGC SWE framework for Internet of Things applications
- Simplified approach
- Based on REST and JSON
 - ~ REST binding for SOS functionalities
 - ~ JSON binding for the O&M model
- Two parts
 - Data access (Part 1: Sensing)
 - Sensor tasking
- CRUD

STA - ENTITIES

SensorThings API Entities	O&M 2.0 Concepts
Thing (and Locations, HistoricalLocations)	-
Datastream	-
Sensor	Procedure
Observation	Observation
Observed Property	Observed Property
FeaureOfInterest	FeaureOfInterest

STA - ENTITIES



STA - ENTITIES

- Thing
 - IoT device
- Location
 - The location where the Thing is currently placed
- HistoricalLocation
 - Locations where the Thing was placed before
- Datastream
 - Groups observations for the same Thing, ObservedProperty and Sensor
 - Provides some metadata (unit, phenomenonTime, observedArea, ...)

STA - ENTITIES

- Sensor
 - The instrument that observes a phenomenon
- ObservedProperty
 - The phenomenon of the observation
- FeatureOfInterest
 - The observed location
 - Can be the Thing-Location
 - A room, a geographical area, ...

STA - ENTITIES

- Observation
 - The measurement
 - Values
 - quantity, count, text, category, boolean
- MultiDatastream Extension
 - Similar to O&M ComplexObservation
- DataArray Extension
 - Similar to O&M SweDataArrayObservation

STA - QUERYING

- <http://example.org/v1.0/Things>
- [http://example.org/v1.0/Things\(1\)](http://example.org/v1.0/Things(1))
- [http://example.org/v1.0/Things\(1\)/Locations](http://example.org/v1.0/Things(1)/Locations)
- [http://example.org/v1.0/Things\(1\)/description](http://example.org/v1.0/Things(1)/description)
- [http://example.org/v1.0/Things\(1\)/description/\\$value](http://example.org/v1.0/Things(1)/description/$value)

STA - FILTERING

- \$skip
 - number for the queries items that shall be excluded from the result
- \$top
 - limit on the number of items returned
- \$expand
 - Represent entity inline
- \$select
 - Return only the selected properties

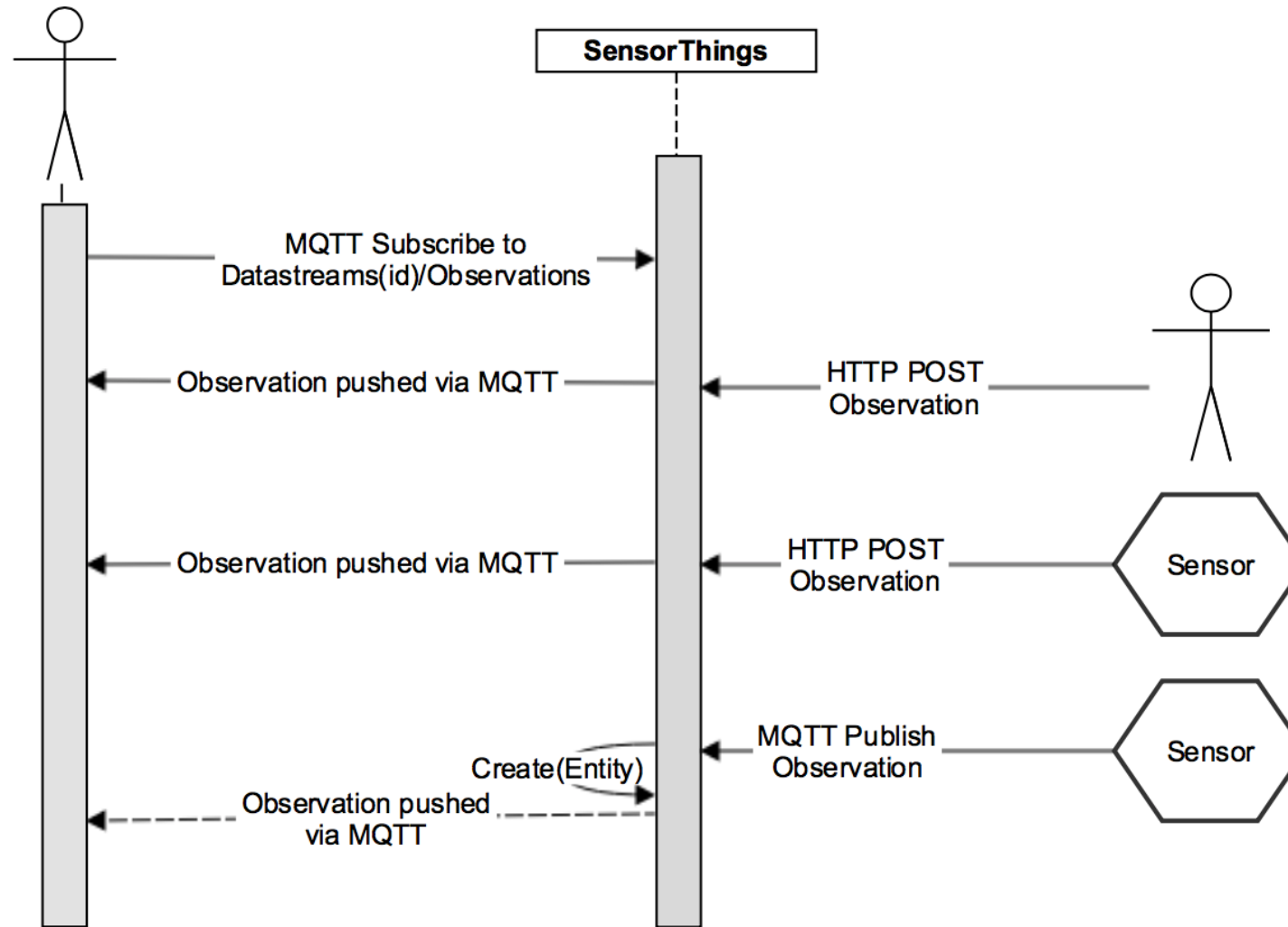
STA - FILTERING

- \$count
 - total count of items within a collection matching the request
- \$orderby
 - The order in which the items should be returned
- \$filter
 - Filter the resulting items
 - Comparison, Logical and Arithmetic Operators
 - String, Date, Math and Geospatial functions

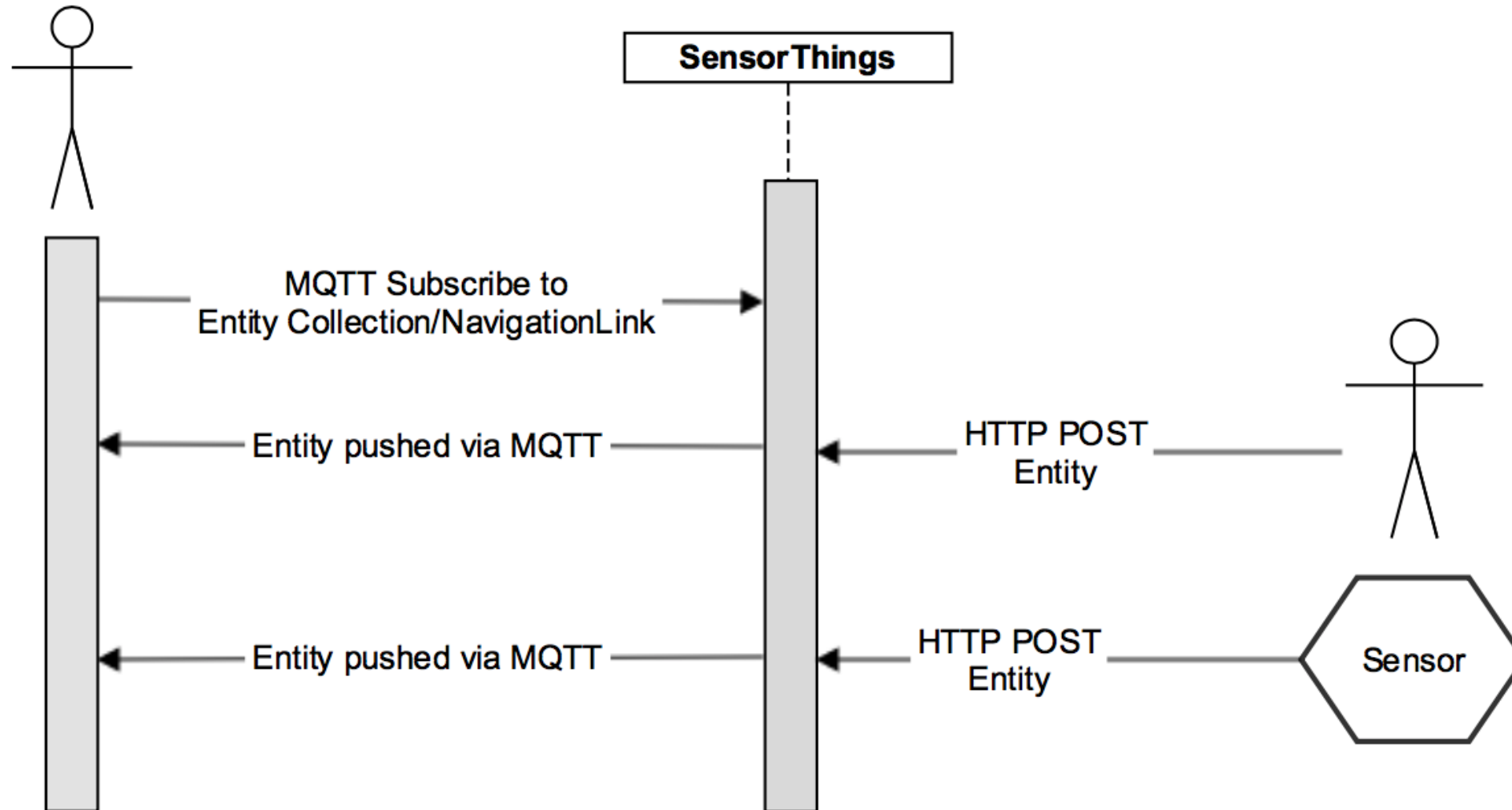
STA - MQTT

- MQTT - Message Queuing Telemetry Transport
- Publish
 - Observations
- Subscribe
 - Receive notification for updated entities
 - Examples
 - Datastreams(1)/Observations (new observation added)
 - Datastreams(1) (property changed)
 - Datastreams(1)/observedArea (value of property changed)

STA - MQTT PUBLISH



STA - MQTT SUBSCRIBE



SENSORTHINGS API

- Demo Servers:
 - <https://cos4cloud.demo.52north.org/sta/>
 - <https://aqsens.52north.org/data/reference/sta/>

SENSORTHINGS API - ROOT

- /
 - Root query
- Click the URL of the entity you are interested in.

```
{
  "value": [
    {
      "name": "Datastreams",
      "url": "https://aqsens.52north.org/data/reference/sta/Datastreams"
    },
    {
      "name": "Observations",
      "url": "https://aqsens.52north.org/data/reference/sta/Observations"
    },
    {
      "name": "Things",
      "url": "https://aqsens.52north.org/data/reference/sta/Things"
    },
    {
      "name": "Locations",
      "url": "https://aqsens.52north.org/data/reference/sta/Locations"
    },
    {
      "name": "HistoricalLocations",
      "url": "https://aqsens.52north.org/data/reference/sta/HistoricalLocations"
    },
    {
      "name": "Sensors",
      "url": "https://aqsens.52north.org/data/reference/sta/Sensors"
    },
    {
      "name": "ObservedProperties",
      "url": "https://aqsens.52north.org/data/reference/sta/ObservedProperties"
    },
    {
      "name": "FeaturesOfInterest",
      "url": "https://aqsens.52north.org/data/reference/sta/FeaturesOfInterest"
    }
  ]
}
```

SENSORTHINGS API- THING

- /Things
 - Get all Things
- /Things(1)
 - Get the Thing with ID „1“
- Example
 - [https://aqsens.52north.org/data/reference/sta/Things\(ID_163\)](https://aqsens.52north.org/data/reference/sta/Things(ID_163))

```
{
  "@iot.id": "ID_163",
  "@iot.selfLink": "https://aqsens.52north.org/data/reference/sta/Things(ID_163)",
  "name": "Rv 4, Aker sykehus",
  "description": null,
  "properties": null,
  "Datastreams@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Things(ID_163)/Datastreams",
  "Locations@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Things(ID_163)/Locations",
  "HistoricalLocations@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Things(ID_163)/HistoricalLocations"
}
```


SENSORTHINGS API - LOCATION

- /Locations
 - Get all Locations
- /Locations(1)
 - Get the Location with ID „1“

```
{
  "@iot.count": 1,
  "value": [
    {
      "@iot.id": "ID_163",
      "@iot.selfLink": "https://aqsens.52north.org/data/reference/sta/Locations(ID_163)",
      "name": "Rv 4, Aker sykehus",
      "description": "Rv 4, Aker sykehus",
      "encodingType": "application/vnd.geo+json",
      "location": {
        "type": "Point",
        "coordinates": [
          10.79803,
          59.94103
        ],
        "crs": {
          "type": "name",
          "properties": {
            "name": "EPSG:4326"
          }
        }
      },
      "Things@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Locations(ID_163)/Things",
      "HistoricalLocations@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Locations(ID_163)/HistoricalLocations"
    }
  ]
}
```

- Example
 - [https://aqsens.52north.org/data/reference/sta/Things\(ID_163\)/Locations](https://aqsens.52north.org/data/reference/sta/Things(ID_163)/Locations)

SENSORTHINGS API - HISTORICALLOCATION

- / HistoricalLocations
 - Get all HistoricalLocations
- / HistoricalLocations(1)
 - Get the HistoricalLocations with ID „1“

```
{
  "@iot.id": "02098019-e8c2-494e-a25f-5dbb10b39747",
  "@iot.selfLink": "https://aqsens.52north.org/data/reference/sta/HistoricalLocations(02098019-e8c2-494e-a25f-5dbb10b39747)",
  "time": "2020-06-05T16:44:36.537Z",
  "Locations@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/HistoricalLocations(02098019-e8c2-494e-a25f-5dbb10b39747)/Locations",
  "Thing@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/HistoricalLocations(02098019-e8c2-494e-a25f-5dbb10b39747)/Thing"
}
```

SENSORTHINGS API - DATASTREAM

- / Datastreams
 - Get all Datastreams
- / Datastreams(1)
 - Get the Datastreams with ID „1“
- Example
 - [https://aqsens.52north.org/data/reference/sta/Datastreams\(019722ba-89cf-4bf8-88d8-47614d43b34b\)](https://aqsens.52north.org/data/reference/sta/Datastreams(019722ba-89cf-4bf8-88d8-47614d43b34b))

SENSORTHINGS API - DATASTREAM

- Links to Thing, ObservedProperty, Sensor and Observations

```
{
  "@iot.id": "3d2db226-1580-4e6b-a372-a8bf4fe5796a",
  "@iot.selfLink": "https://aqsens.52north.org/data/reference/sta/Datastreams(3d2db226-1580-4e6b-a372-a8bf4fe5796a)",
  "name": "Rv 4, Aker sykehus_Station Rv 4, Aker sykehus (Stor-Oslo), PM2.5_PM2.5",
  "description": "Datastream for Thing 'Rv 4, Aker sykehus' and Sensor 'Station Rv 4, Aker sykehus (Stor-Oslo), PM2.5' and ObservedProperty 'PM2.5'.",
  "observationType": "http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement",
  "unitOfMeasurement": {
    "name": null,
    "symbol": "µg/m³",
    "definition": null
  },
  "observedArea": null,
  "resultTime": "2020-05-01T02:00:00.000Z/2020-06-08T13:00:00.000Z",
  "phenomenonTime": "2020-05-01T02:00:00.000Z/2020-06-08T13:00:00.000Z",
  "ObservedProperty@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Datastreams(3d2db226-1580-4e6b-a372-a8bf4fe5796a)/ObservedProperty",
  "Observations@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Datastreams(3d2db226-1580-4e6b-a372-a8bf4fe5796a)/Observations",
  "Thing@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Datastreams(3d2db226-1580-4e6b-a372-a8bf4fe5796a)/Thing",
  "Sensor@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Datastreams(3d2db226-1580-4e6b-a372-a8bf4fe5796a)/Sensor"
}
```

Raw

Parsed

SENSORTHINGS API - SENSOR

- / Sensors
 - Get all Sensors
- / Sensors(1)
 - Get the Sensors with ID „1“
- Example
 - [https://aqsens.52north.org/data/reference/sta/Sensors\(station.stor-oslo.rv_4_aker_sykehus.pm2.5\)](https://aqsens.52north.org/data/reference/sta/Sensors(station.stor-oslo.rv_4_aker_sykehus.pm2.5))

SENSORTHINGS API - SENSOR

```
{
  "@iot.id": "station.stor-oslo.rv_4_aker_sykehus.pm2.5",
  "@iot.selfLink": "https://aqsens.52north.org/data/reference/sta/Sensors(station.stor-oslo.rv_4_aker_sykehus.pm2.5)",
  "name": "Station Rv 4, Aker sykehus (Stor-Oslo), PM2.5",
  "description": null,
  "encodingType": "http://www.opengis.net/doc/IS/SensorML/2.0",
  "metadata": "<sml:PhysicalComponent xmlns:sml=\"http://www.opengis.net/sensorml/2.0\" xmlns:gml=\"http://www.opengis.net/gml/3.2\"
xmlns:swe=\"http://www.opengis.net/swe/2.0\" xmlns:xlink=\"http://www.w3.org/1999/xlink\" gml:id=\"gmlid_1\">\n  <gml:identifier codeSpace=\"uniqueID\">station.stor-
oslo.rv_4_aker_sykehus.pm2.5</gml:identifier>\n  <sml:identification>\n    <sml:IdentifierList>\n      <sml:identifier>\n        <sml:Term
definition=\"urn:ogc:def:identifier:OGC:1.0:longName\">\n          <sml:label>longName</sml:label>\n          <sml:value>Station Rv 4, Aker sykehus (Stor-Oslo),
PM2.5</sml:value>\n        </sml:Term>\n      </sml:identifier>\n      <sml:identifier>\n        <sml:Term definition=\"urn:ogc:def:identifier:OGC:1.0:shortName\">\n
<sml:label>shortName</sml:label>\n        <sml:value>Station Rv 4, Aker sykehus (Stor-Oslo), PM2.5</sml:value>\n        </sml:Term>\n      </sml:identifier>\n    </sml:IdentifierList>\n    </sml:identification>\n    <sml:capabilities name=\"metadata\">\n      <sml:CapabilityList>\n        <sml:capability name=\"insitu\">\n
<swe:Boolean definition=\"insitu\">\n          <swe:value>true</swe:value>\n        </swe:Boolean>\n        </sml:capability>\n        <sml:capability name=\"mobile\">\n
<swe:Boolean definition=\"mobile\">\n          <swe:value>false</swe:value>\n        </swe:Boolean>\n        </sml:capability>\n      </sml:CapabilityList>\n    </sml:capabilities>\n    <sml:outputs>\n      <sml:OutputList>\n        <sml:output name=\"PM2.5\">\n          <swe:Quantity definition=\"PM2.5\">\n            <swe:uom
code=\"µg/m³\">\n          </swe:Quantity>\n        </sml:output>\n      </sml:OutputList>\n    </sml:outputs>\n    <sml:attachedTo xlink:href=\"station.stor-
oslo.rv_4_aker_sykehus\" xlink:title=\"station.stor-oslo.rv_4_aker_sykehus\">\n  </sml:PhysicalComponent>\",
  "Datastreams@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Sensors(station.stor-oslo.rv_4_aker_sykehus.pm2.5)/Datastreams"
}
```

SENSORTHINGS API - OBSERVEDPROPERTY

- / ObservedProperties
 - Get all ObservedProperties
- / ObservedProperties(1)
 - Get the ObservedProperties with ID „1“
- Example
 - [https://aqsens.52north.org/data/reference/sta/ObservedProperties\(PM2.5\)](https://aqsens.52north.org/data/reference/sta/ObservedProperties(PM2.5))

```
{
  "@iot.id": "PM2.5",
  "@iot.selfLink": "https://aqsens.52north.org/data/reference/sta/ObservedProperties(PM2.5)",
  "name": "PM2.5",
  "description": null,
  "definition": "PM2.5",
  "Datastreams@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/ObservedProperties(PM2.5)/Datastreams"
}
```

SENSORTHINGS API - OBSERVATION

- / Observations
 - Get all Observations
- / Observations(1)
 - Get the Observations with ID „1“
- Example
 - [https://aqsens.52north.org/data/reference/sta/Observations\(0038bb98-d608-4f76-ab54-1efc86dec89a\)](https://aqsens.52north.org/data/reference/sta/Observations(0038bb98-d608-4f76-ab54-1efc86dec89a))

```
{
  "@iot.id": "42772",
  "@iot.selfLink": "https://aqsens.52north.org/data/reference/sta/Observations(42772)",
  "result": "1.9769960000",
  "resultTime": null,
  "phenomenonTime": "2020-05-01T02:00:00.000Z",
  "resultQuality": null,
  "validTime": null,
  "parameters": [],
  "Datastream@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Observations(42772)/Datastream",
  "FeatureOfInterest@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Observations(42772)/FeatureOfInterest"
}
```


SENSORTHINGS API – OBSERVATION FOR DATASTREAM

- [https://aqsens.52north.org/data/reference/sta/Datastreams\(019722ba-89cf-4bf8-88d8-47614d43b34b\)/Observations?\\$top=2](https://aqsens.52north.org/data/reference/sta/Datastreams(019722ba-89cf-4bf8-88d8-47614d43b34b)/Observations?$top=2)

```
{
  "@iot.count": 886,
  "@iot.nextLink": "https://aqsens.52north.org/data/reference/sta/Datastreams(3d2db226-1580-4e6b-a372-a8bf4fe5796a)/Observations?$skip=2&$top=2",
  "value": [
    {
      "@iot.id": "42772",
      "@iot.selfLink": "https://aqsens.52north.org/data/reference/sta/Observations(42772)",
      "result": "1.9769960000",
      "resultTime": null,
      "phenomenonTime": "2020-05-01T02:00:00.000Z",
      "resultQuality": null,
      "validTime": null,
      "parameters": [],
      "Datastream@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Observations(42772)/Datastream",
      "FeatureOfInterest@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Observations(42772)/FeatureOfInterest"
    },
    {
      "@iot.id": "42773",
      "@iot.selfLink": "https://aqsens.52north.org/data/reference/sta/Observations(42773)",
      "result": "-3.0547050000",
      "resultTime": null,
      "phenomenonTime": "2020-05-01T03:00:00.000Z",
      "resultQuality": null,
      "validTime": null,
      "parameters": [],
      "Datastream@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Observations(42773)/Datastream",
      "FeatureOfInterest@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/Observations(42773)/FeatureOfInterest"
    }
  ]
}
```

SENSOR THINGS API - FEATURE OF INTEREST

- / FeaturesOfInterest
 - Get all FeaturesOfInterest
- / FeatureOfInterest(1)
 - Get the FeaturesOfInterest with ID „1“

```
{
  "@iot.id": "ID_163",
  "@iot.selfLink": "https://aqsens.52north.org/data/reference/sta/FeaturesOfInterest(ID_163)",
  "name": "Rv 4, Aker sykehus",
  "description": null,
  "encodingType": "application/vnd.geo+json",
  "feature": {
    "type": "Point",
    "coordinates": [
      10.79803,
      59.94103
    ],
    "crs": {
      "type": "name",
      "properties": {
        "name": "EPSG:4326"
      }
    }
  },
  "Observations@iot.navigationLink": "https://aqsens.52north.org/data/reference/sta/FeaturesOfInterest(ID_163)/Observations"
}
```

- Example
 - [https://aqsens.52north.org/data/reference/sta/FeaturesOfInterest\(ID_163\)](https://aqsens.52north.org/data/reference/sta/FeaturesOfInterest(ID_163))

SENSORTHINGS API - FILTER

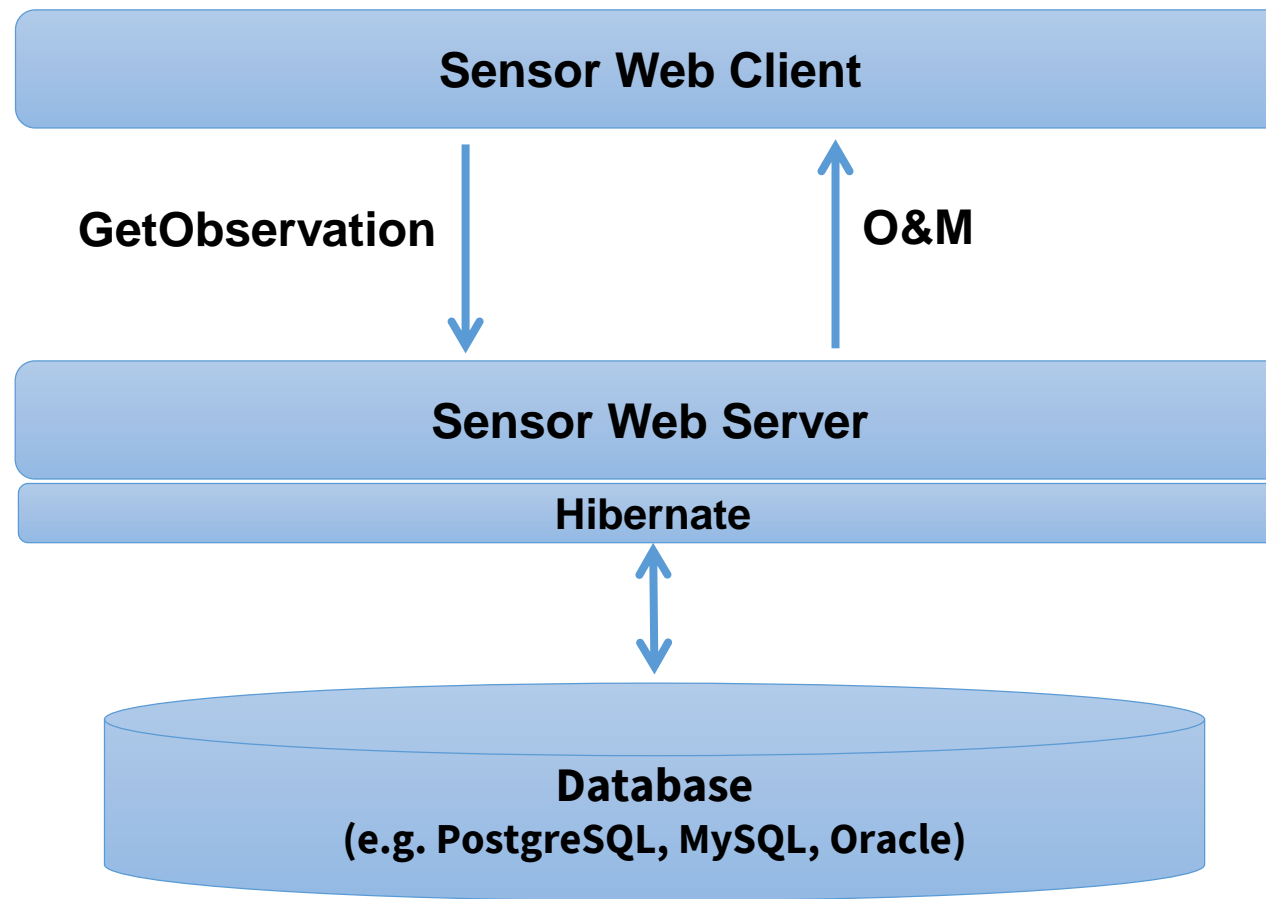
- \$top
 - [https://aqsens.52north.org/data/reference/sta/FeaturesOfInterest?\\$top=3](https://aqsens.52north.org/data/reference/sta/FeaturesOfInterest?$top=3)
- \$skip
 - [https://aqsens.52north.org/data/reference/sta/FeaturesOfInterest?\\$skip=3](https://aqsens.52north.org/data/reference/sta/FeaturesOfInterest?$skip=3)
- \$top and \$skip
 - [https://aqsens.52north.org/data/reference/sta/FeaturesOfInterest?\\$skip=3&\\$top=3](https://aqsens.52north.org/data/reference/sta/FeaturesOfInterest?$skip=3&$top=3)

SENSORTHINGS API - FILTER

- \$orderby
 - [https://aqsens.52north.org/data/reference/sta/ObservedProperties?\\$orderby=name](https://aqsens.52north.org/data/reference/sta/ObservedProperties?$orderby=name)
 - [https://aqsens.52north.org/data/reference/sta/ObservedProperties?\\$orderby=name%20desc](https://aqsens.52north.org/data/reference/sta/ObservedProperties?$orderby=name%20desc)
- \$select
 - Select only the name property
 - [https://aqsens.52north.org/data/reference/sta/ObservedProperties?\\$select=name](https://aqsens.52north.org/data/reference/sta/ObservedProperties?$select=name)
 - Name and description
 - [https://aqsens.52north.org/data/reference/sta/ObservedProperties?\\$select=name,description](https://aqsens.52north.org/data/reference/sta/ObservedProperties?$select=name,description)

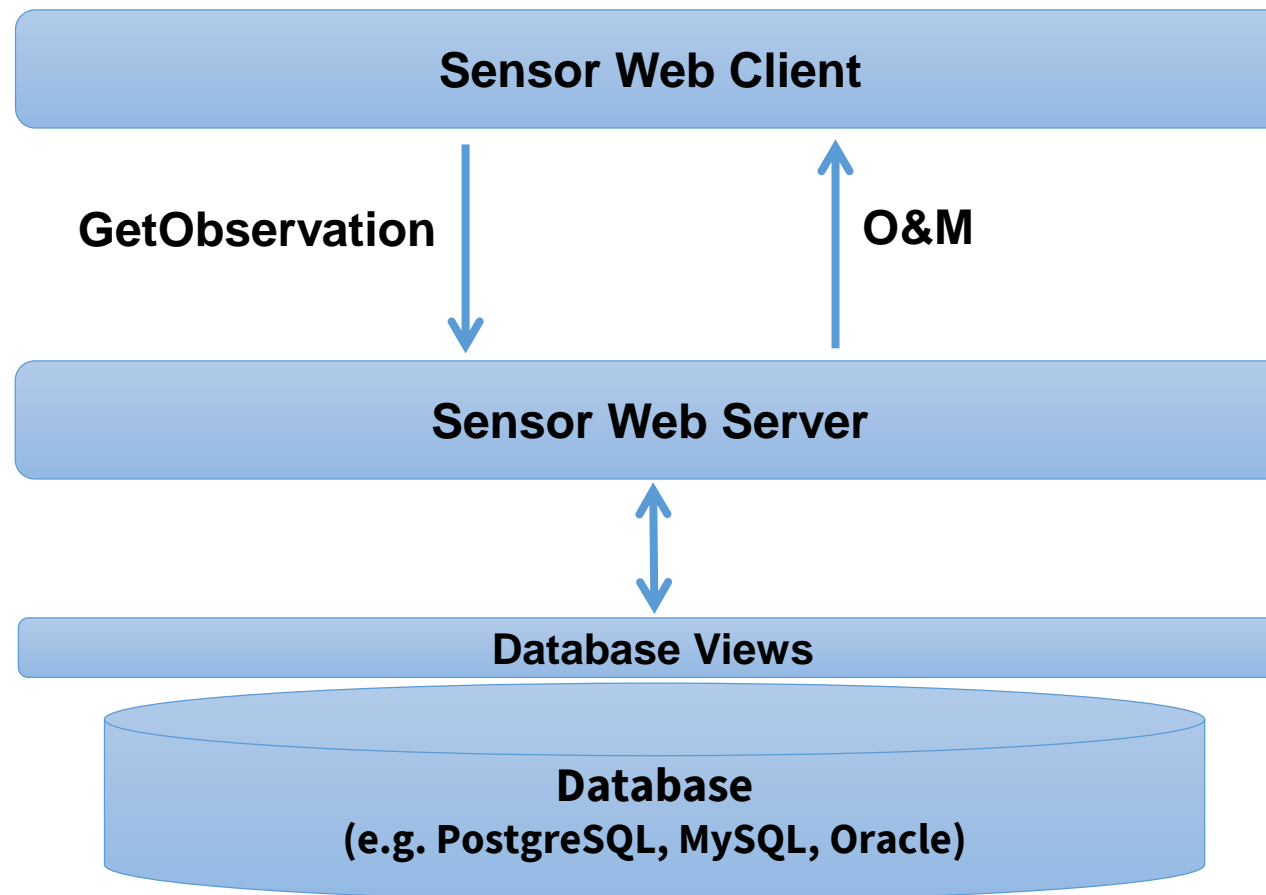
TYPICAL SET UPS

Existing database via Hibernate



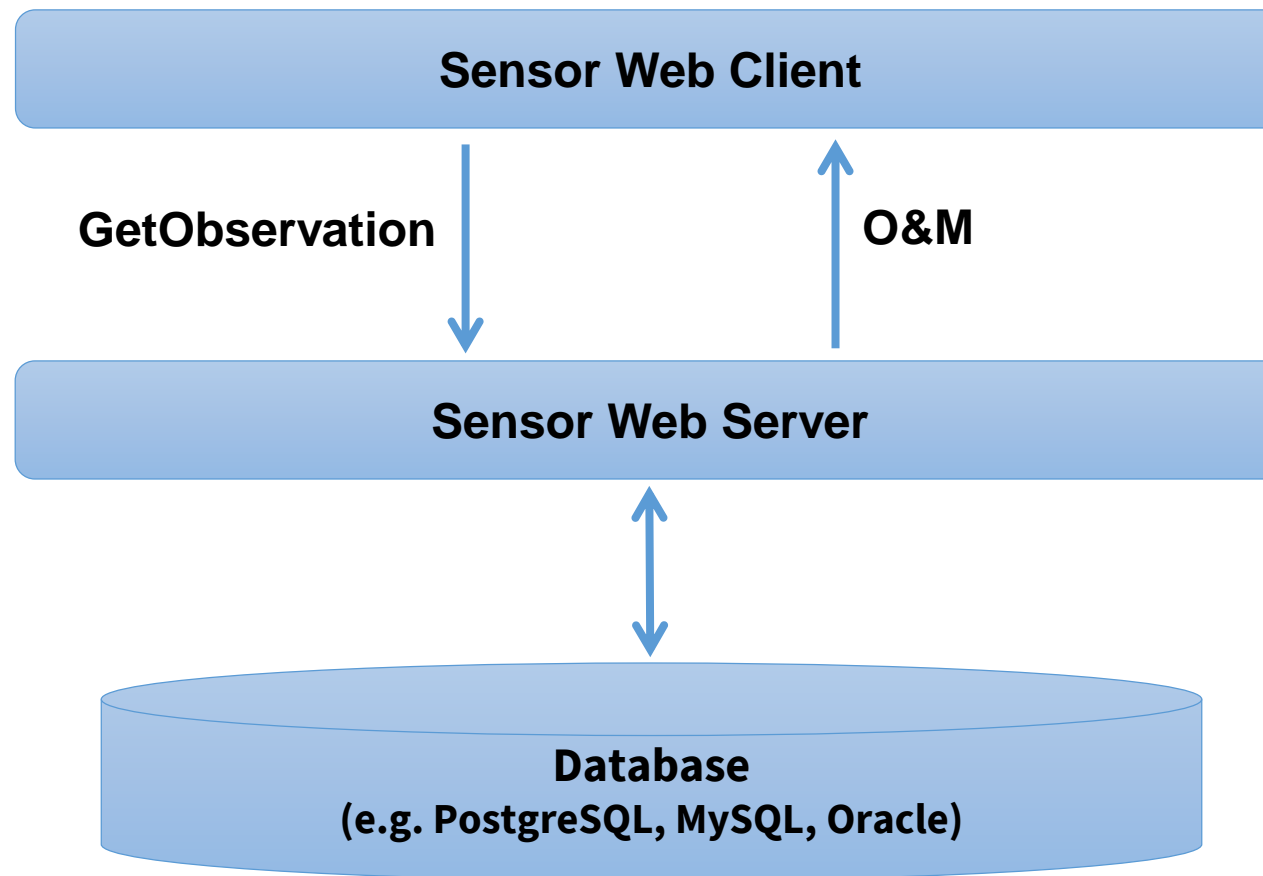
TYPICAL SET UPS

Existing database via Database Views



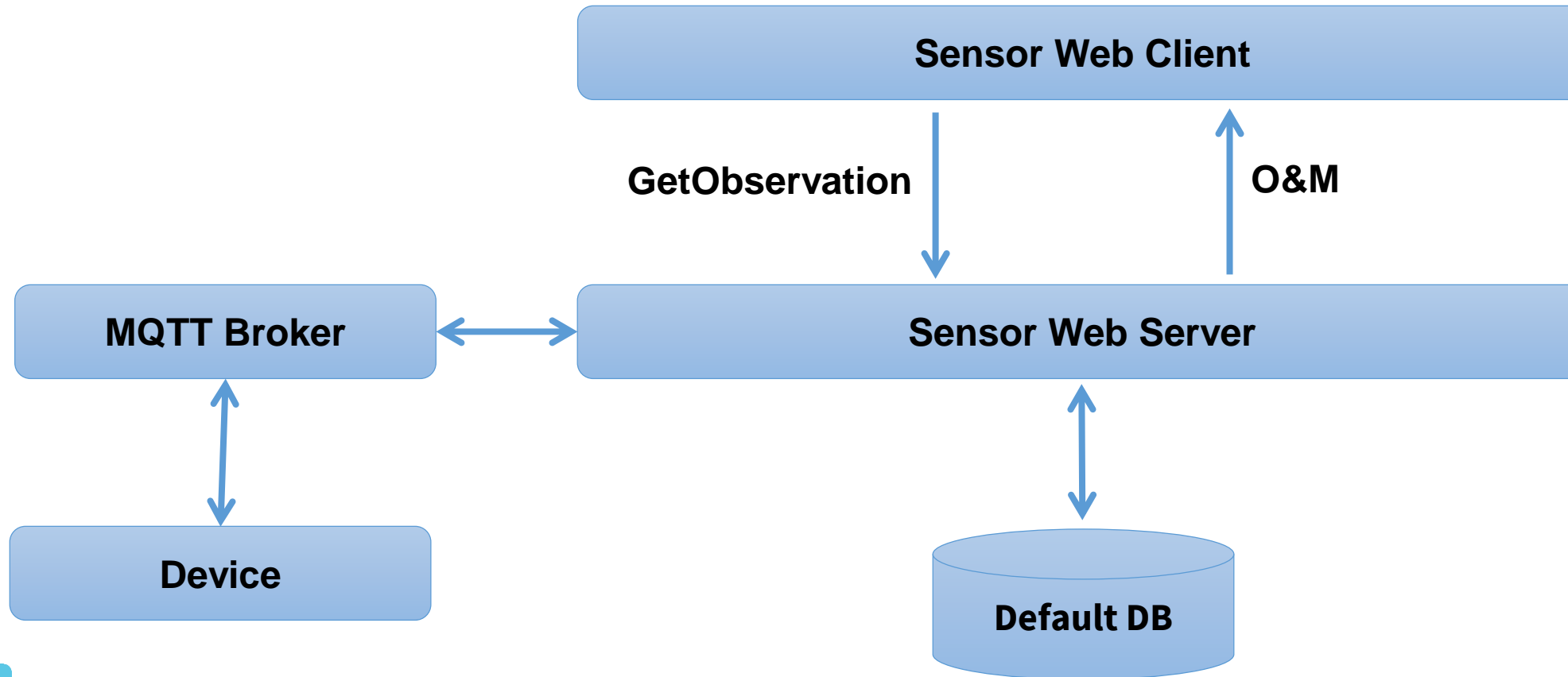
TYPICAL SET UPS

Existing database with customised SOS



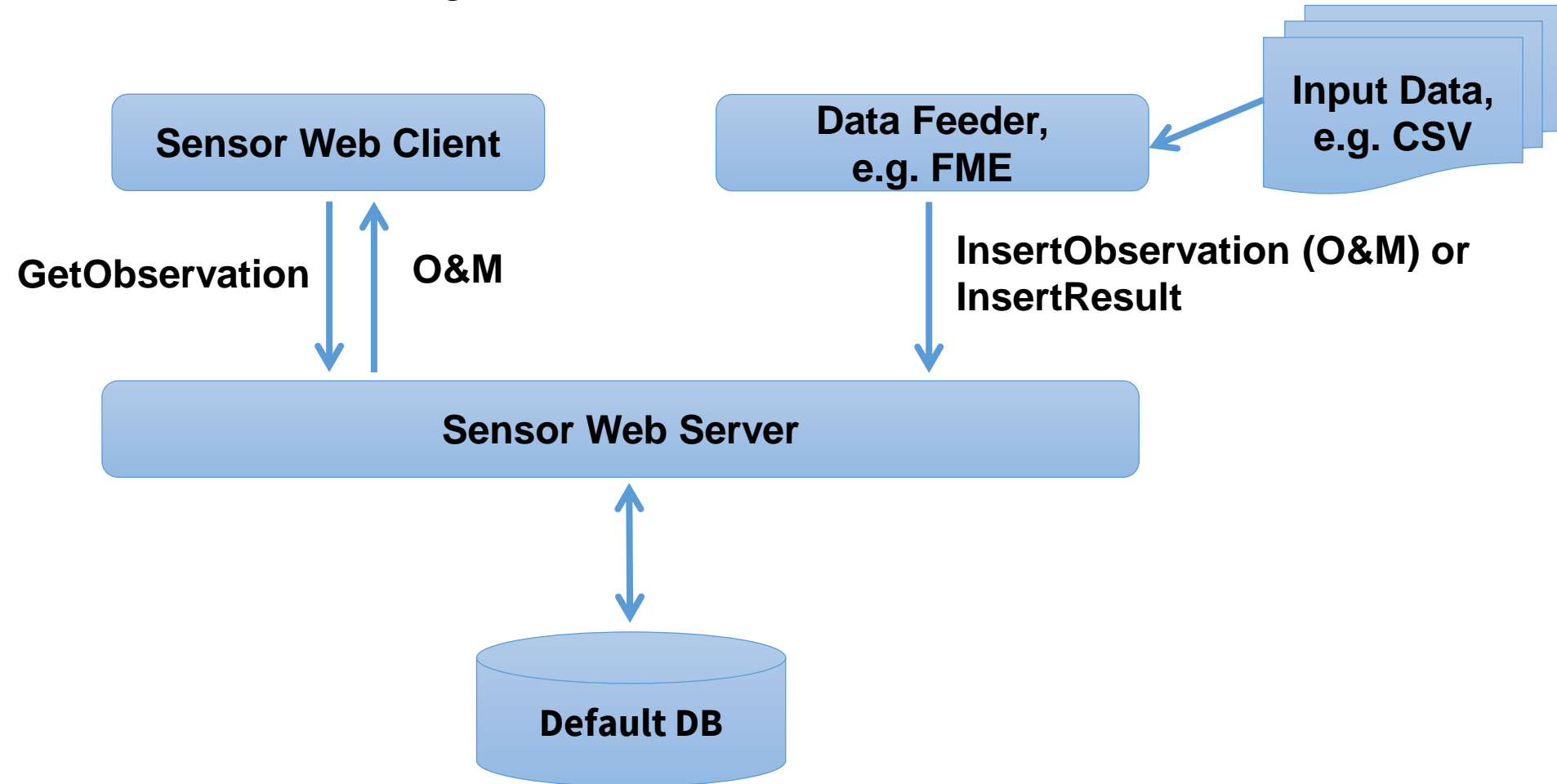
TYPICAL SET UPS

MQTT Feeding



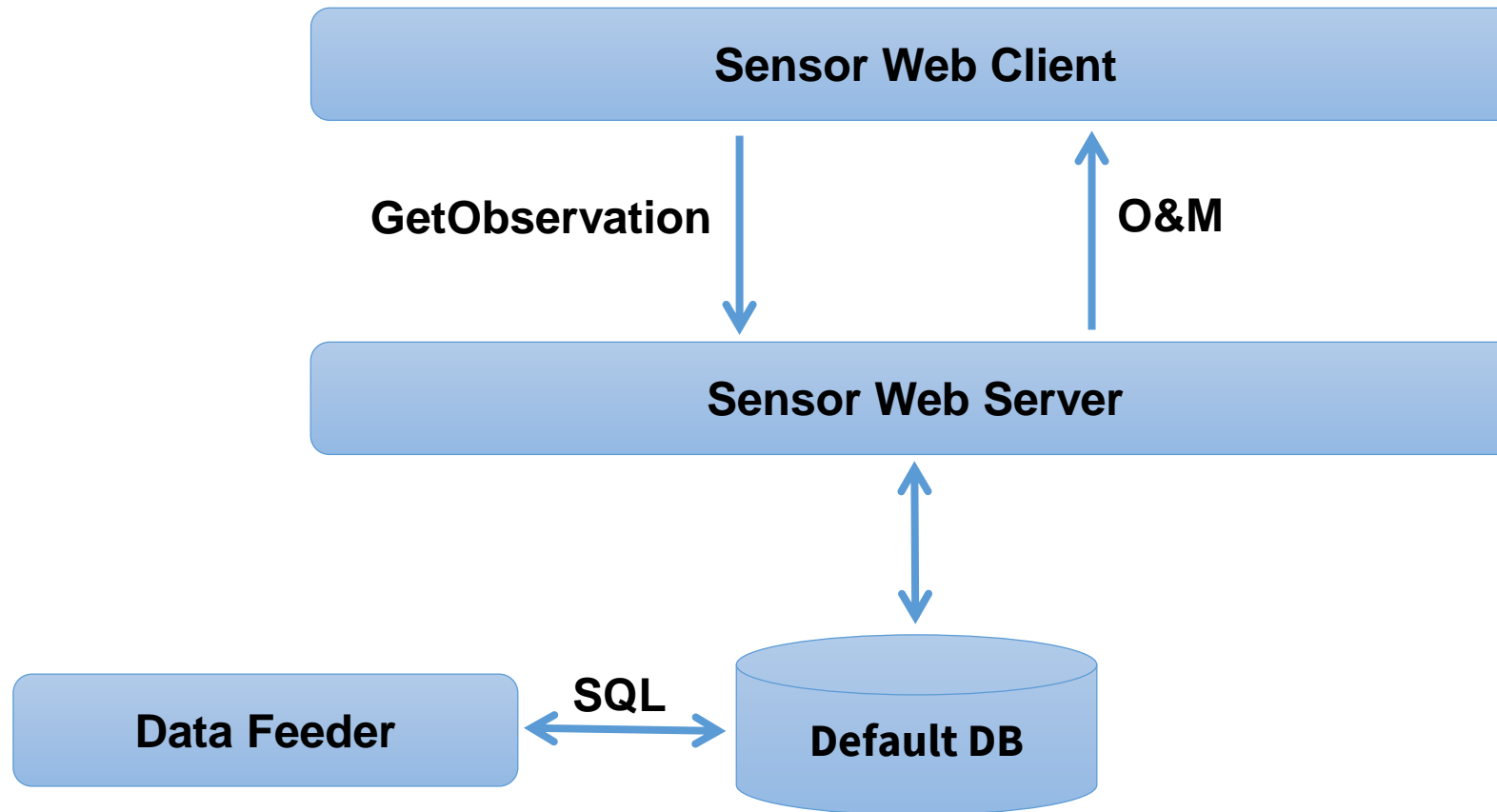
TYPICAL SET UPS

Default SOS database, standardised feeding (transactional operations)



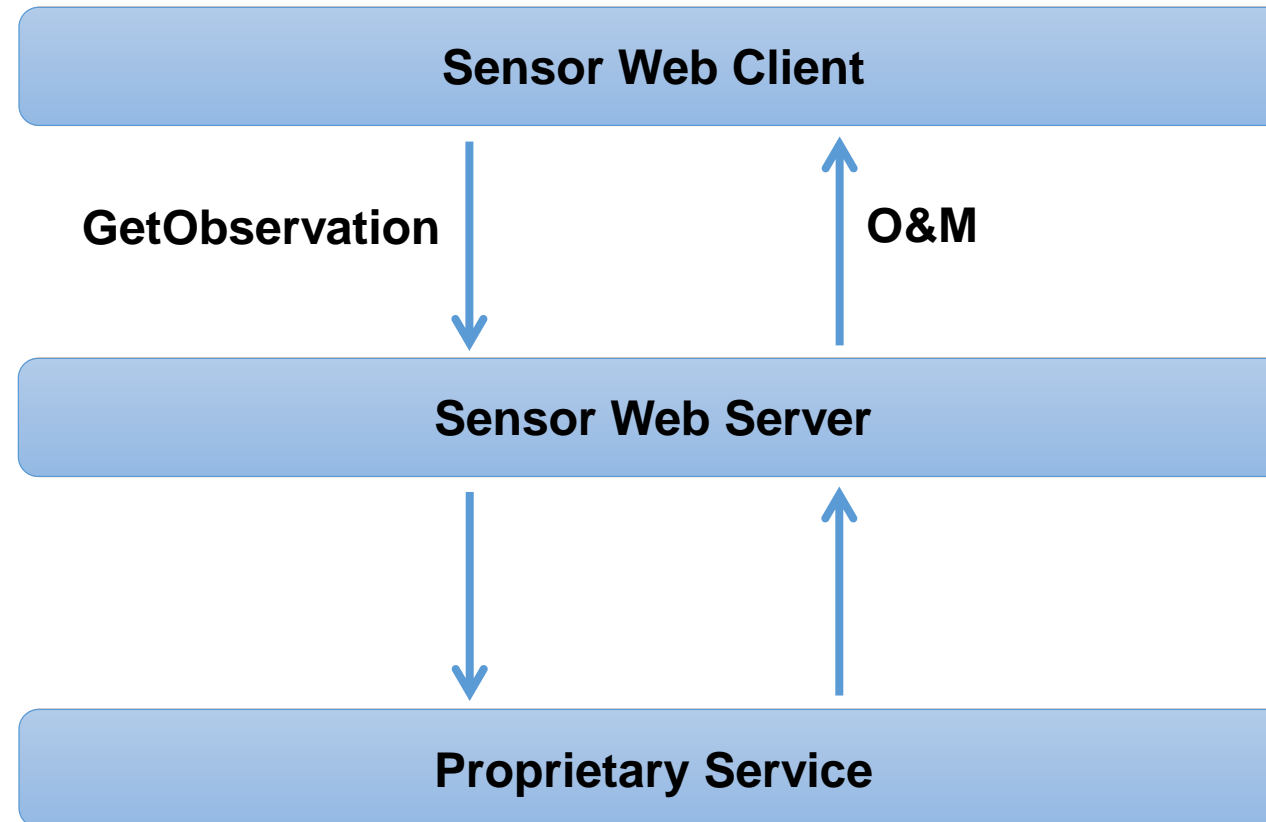
TYPICAL SET UPS

Default SOS database, data feeding by SQL script



TYPICAL SET UPS

SOS as proxy for proprietary data access service



QUESTIONS!