Introduction to the OGC SensorThings API

Dr. Steve Liang
Associate Professor / AITF-Microsoft Industry Chair
steve.liang@ucalgary.ca

About Steve

- Associate Professor, Geomatics Engineering, Uni. Calgary
- AITF-Microsoft Industry Chair on Open Sensor Web
- Chair, OGC Sensor Web for Internet of Things Standards Working Group
- N.A. Chair, OGC University Domain Working Group
- Chair, ISPRS Working Group on Sensor Web and Internet of Things
- Rapporteur, ITU-T SG12/11 on Internet of Things Test Specifications

Background

Bell's Law for the Birth and Death of Computer Classes

"Coming rapidly are billions of cell phones for personal computing and the tens of billions of wireless sensor nets to unwire and interconnect everything. In 1951, a man could walk inside a computer and by 2010 ... computers are beginning to 'walk' inside of us"

Gordon Bell, 2008

Bell, G., 2008. Bell's Law for the Birth and Death of Computer Classes, Communications of the ACM, January 2008, volume 51(1), pp. 86-94

OGC®

“the number of Internet-connected devices will reach between 50 and 60 billion by the end of the decade”

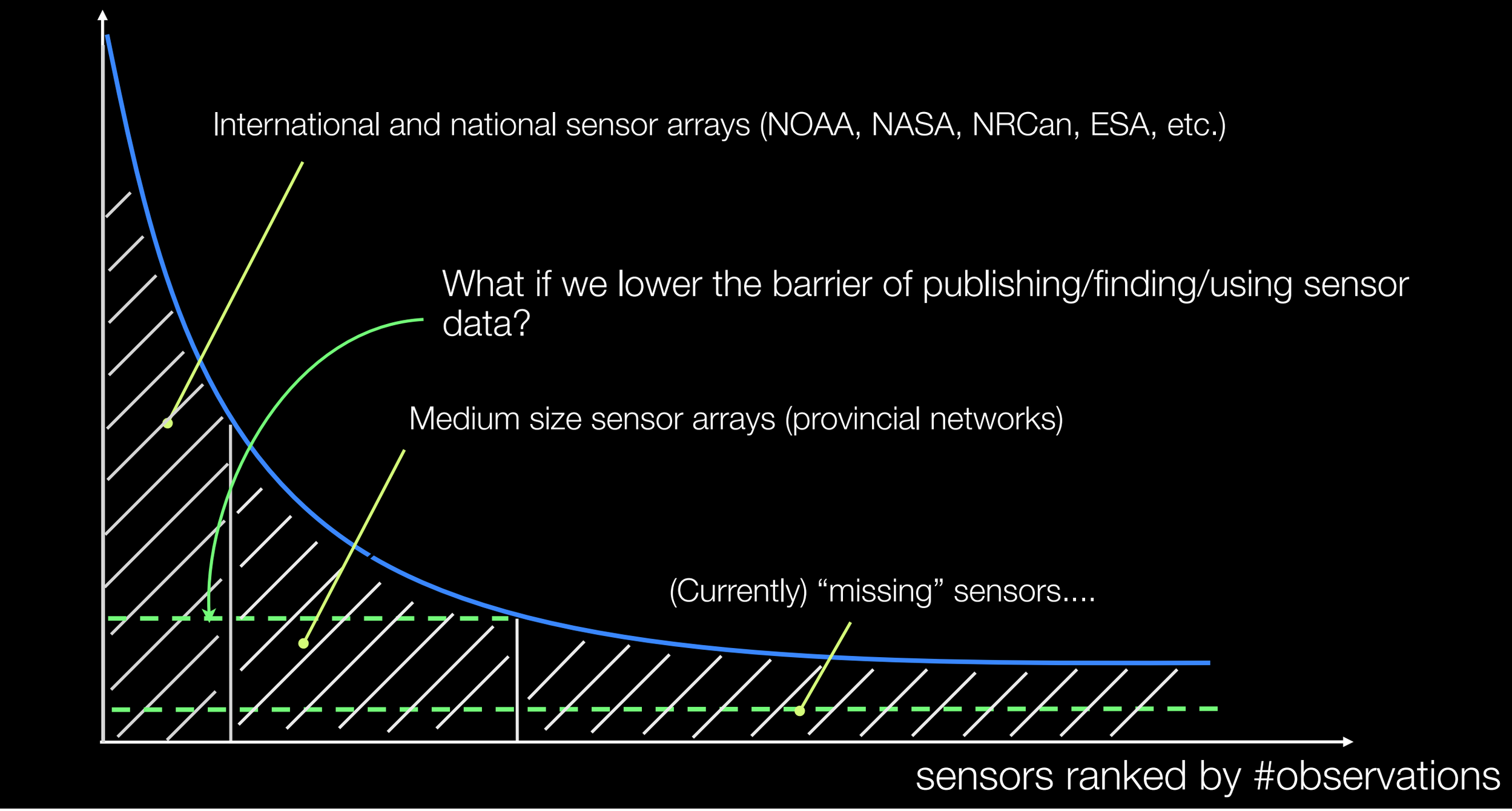
Verizon and Ericsson, 2011

Source: Wall Street Journal
http://online.wsj.com/article/SB10001424052702303544604576434013394780764.html

Long Tail of Big Sensor Data

(Liang, 2010)

# of missing sensors >> # of currently available sensors



3 Vs of BIG Data

- Volume
  - 5.6 billion mobile phones in use (2011)
  - 50~60 billion Internet-connected sensors by 2020
- Velocity
  - 30 billion pieces of content shared on Facebook every month
- Variety
  - Many kinds, fragmented, no sampling scheme, no metadata, no quality control

OGC Sensor Web Enablement Focus!!

Sensor Web Value

How to link different sensor networks?

air quality

traffic

Too Many formats, and everyone speaks their own languages

- Technical Interoperability
- Semantic Interoperability
- Institutional Interoperability

Sensor Networks to Sensor Web

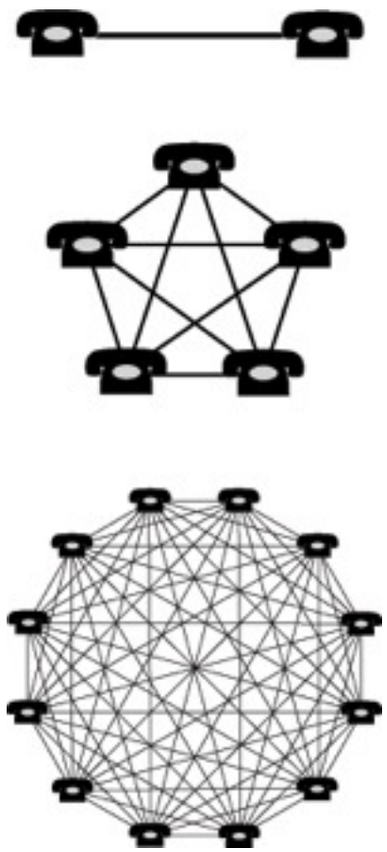
Sensor Networks → Sensor Web

- Single Use
- Single specific purpose and application
- Homogeneous platforms
- Sensor centric and system centric
- Proprietary and close architecture, centrally managed
- Specialized user interface
- Small and isolated

- Reusability and re-mixability
- Multiple un-expected purposes
- Heterogeneous platforms
- Observation centric, feature centric and event centric
- Network/Web-enabled, open, interoperable
- Many different user interfaces
- Large scale, network-effect

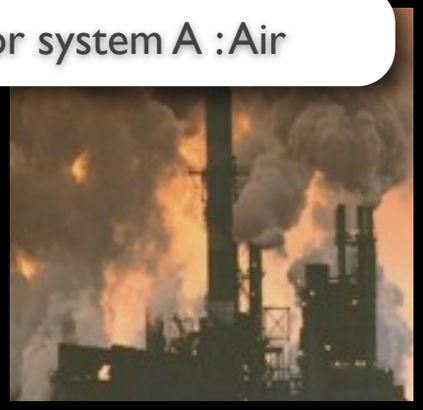
Re-mixability leads to Network Effect

- Innovation-in-assembly
- The value of a network is proportional to the square of the number of users of the system (n^2).
- How many unique links in a network?

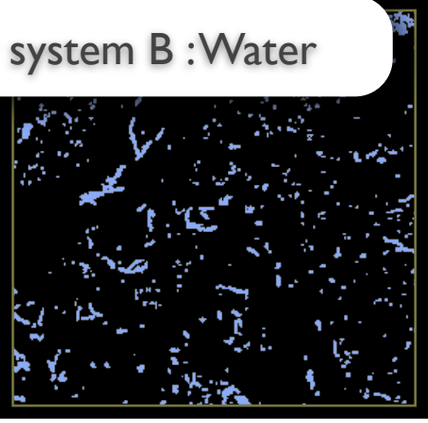


Vision: an word wide sensor web

sensor system A : Air

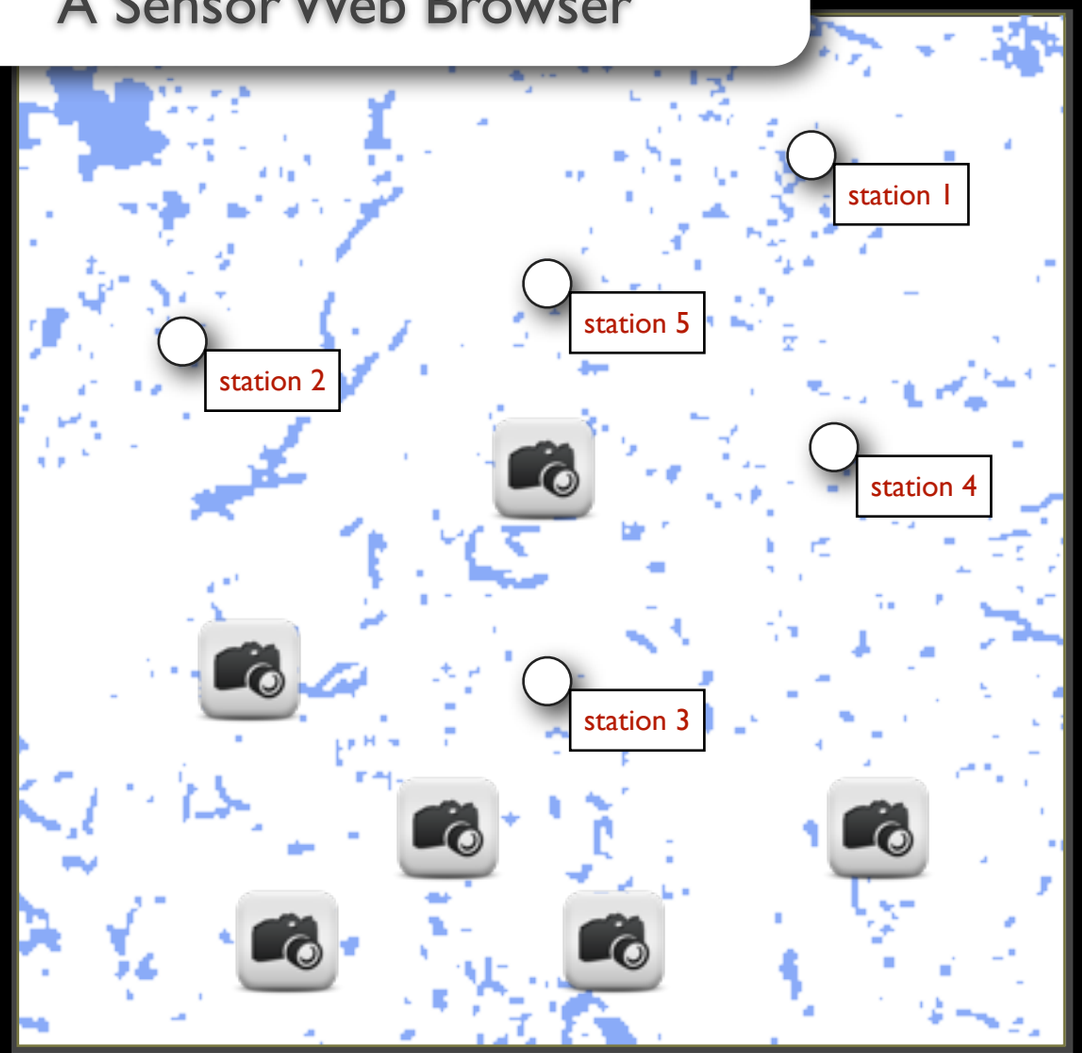


sensor system B : Water

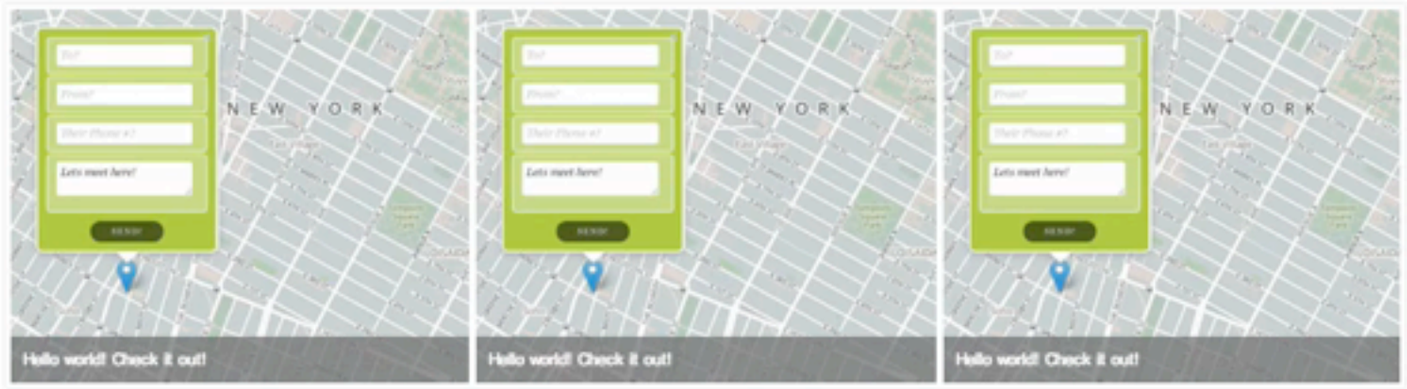


sensor system C : Smartphones

A Sensor Web Browser



Service Registry/
Search Engine



Welcome to Alberta Environment Service !

We provide our environment sensor data in Alberta.

Data Types

- All
- Temperature
- Water
- Weather
- Wind
- Ice

Providers

- All
- NDBC
- IFGI
- VITO
- Geocens
- IP3
- UC-Berkeley

NDBC Buoy Data
Temperature
23
by NDBC

IFGI WeatherSOService
LUMINANCE
28
by IFGI

VITO SOS
YSI Multi Water Parameter Sensor 6600ADV University Antwerp
11
by VITO

IFGI WeatherSOService
RAIN_GAUGE
8
by IFGI

VITO SOS
YSI Multi Water Parameter Sensor 6600ADV University Antwerp
2
by VITO

UC Berkeley Sensor DataBase
Temperature
14
by UC-Berkeley

IP3 Sensor Network
temperature
8
by IP3

Russian Ice Thickness
nsidc_ice_data
22
by Geocens

Canadian Ice Thickness
envirocan_canada_ice_thickness
4
by Geocens

IFGI WeatherSOService
Temperature
28

NDBC Buoy Data
wind
23
by NDBC

IP3 Sensor Network
wind
13


Unleash the Power of Sensors
Innovation in Assembly
A Web of Sensors
– Dr. Steve Liang – http://sensorweb.geomatics.ucalgary.ca –

AN OPEN STANDARD-BASED APPROACH

WWW and Sensor Web

WWW is enabled by open standards

Sensor Web Standard Stack

OGC

ISO

ETSI World Class Standards

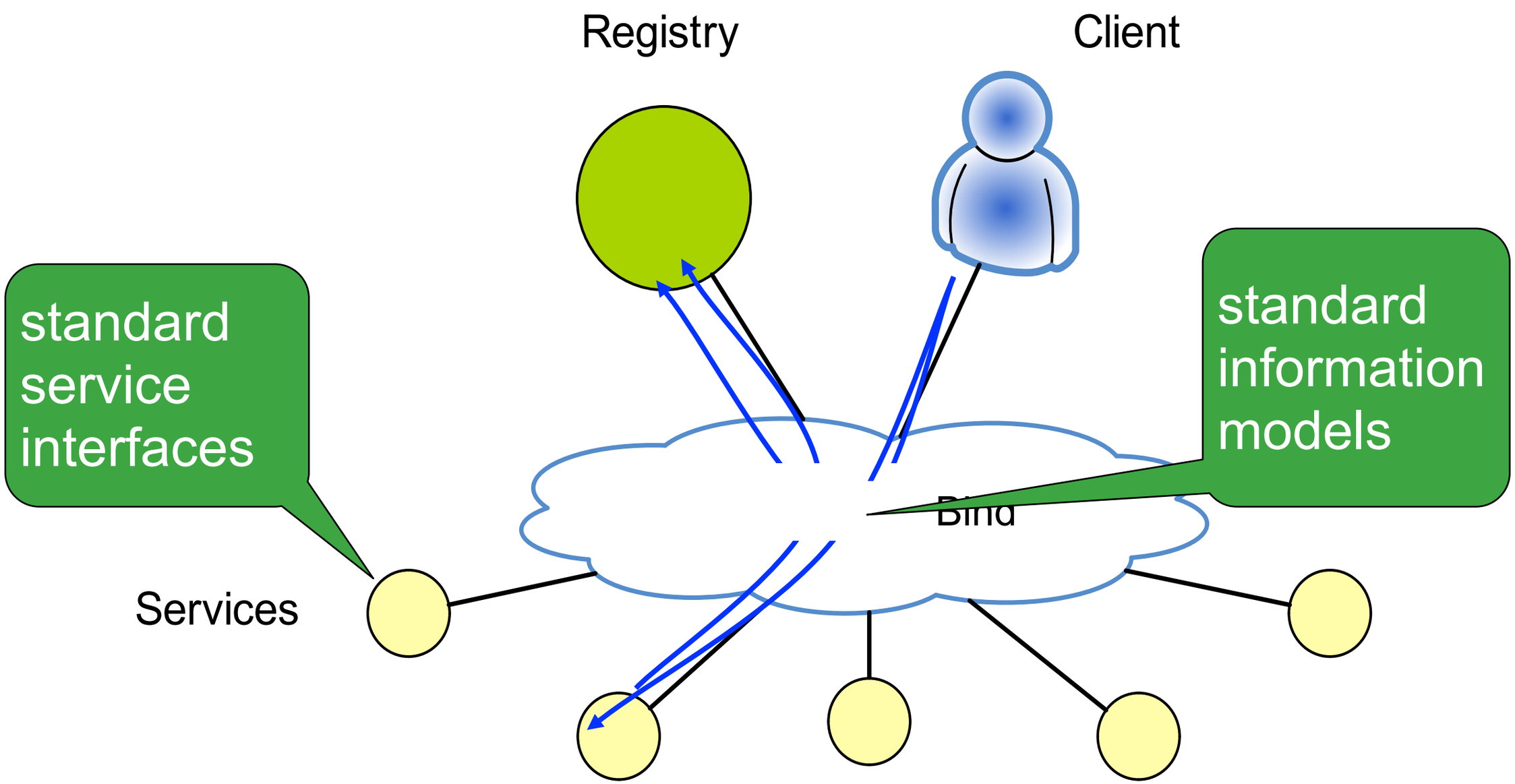
IEEE

OData

OASIS The Organization for the Advancement of Structured Information Standards

OGC Sensor Web Enablement

Anatomy of a Geospatial Interoperability Framework



Sensor Web Enablement (SWE) Specifications

- **Standard Information Models and Schema**
  - **Observations and Measurements (O&M)** – Core models and schema for observations
  - **Sensor Model Language (SensorML) for In-situ and Remote Sensors** - Core models and schema for observation processes: support for sensor components, georegistration, response models, post measurement processing
  - **TransducerML** – adds system integration and real-time streaming clusters of observations

Standard Web Service Interfaces

- **Sensor Observation Service** - Access Observations for a sensor or sensor constellation, and optionally, the associated sensor and platform data
- **Sensor Alert Service** – Subscribe to alerts based upon sensor observations
- **Sensor Planning Service** – Request collection feasibility and task sensor system for desired observations
- **Web Notification Service** –Manage message dialogue between client and Web service(s) for long duration (asynchronous) processes
- **Sensor Registries** – Discover sensors and sensor observations

OGC®

Helping the World to Communicate Geographically

SWE Information Models

Different Level of Models

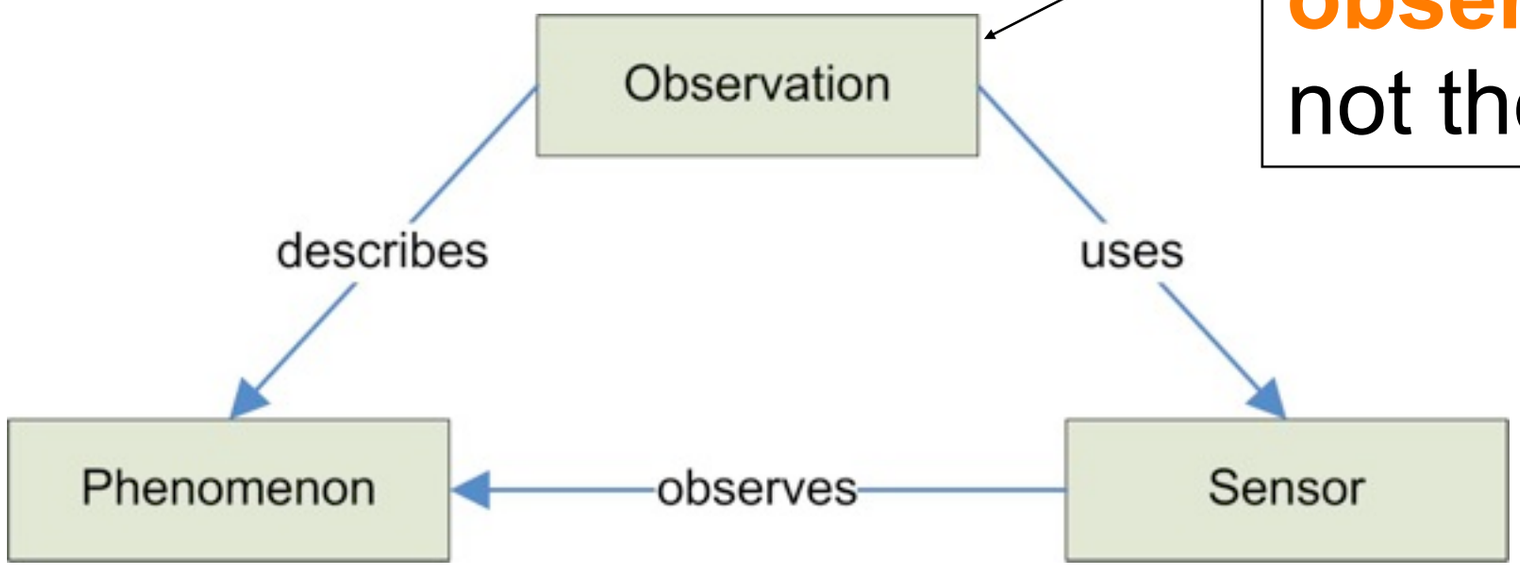
- Information Model
  - conceptual and logical model
    e.g., feature, coverage, observation

- Information Encoding
  - physical model
    e.g., TIFF, XML, JSON, etc.

Standard ways to model and encode sensing information

- the major entities that consists of the sensor web
- their relationship
- their properties

In fact, most users and applications are interested in **observations** but not the **sensors**.



Feature, Coverage, and Observation

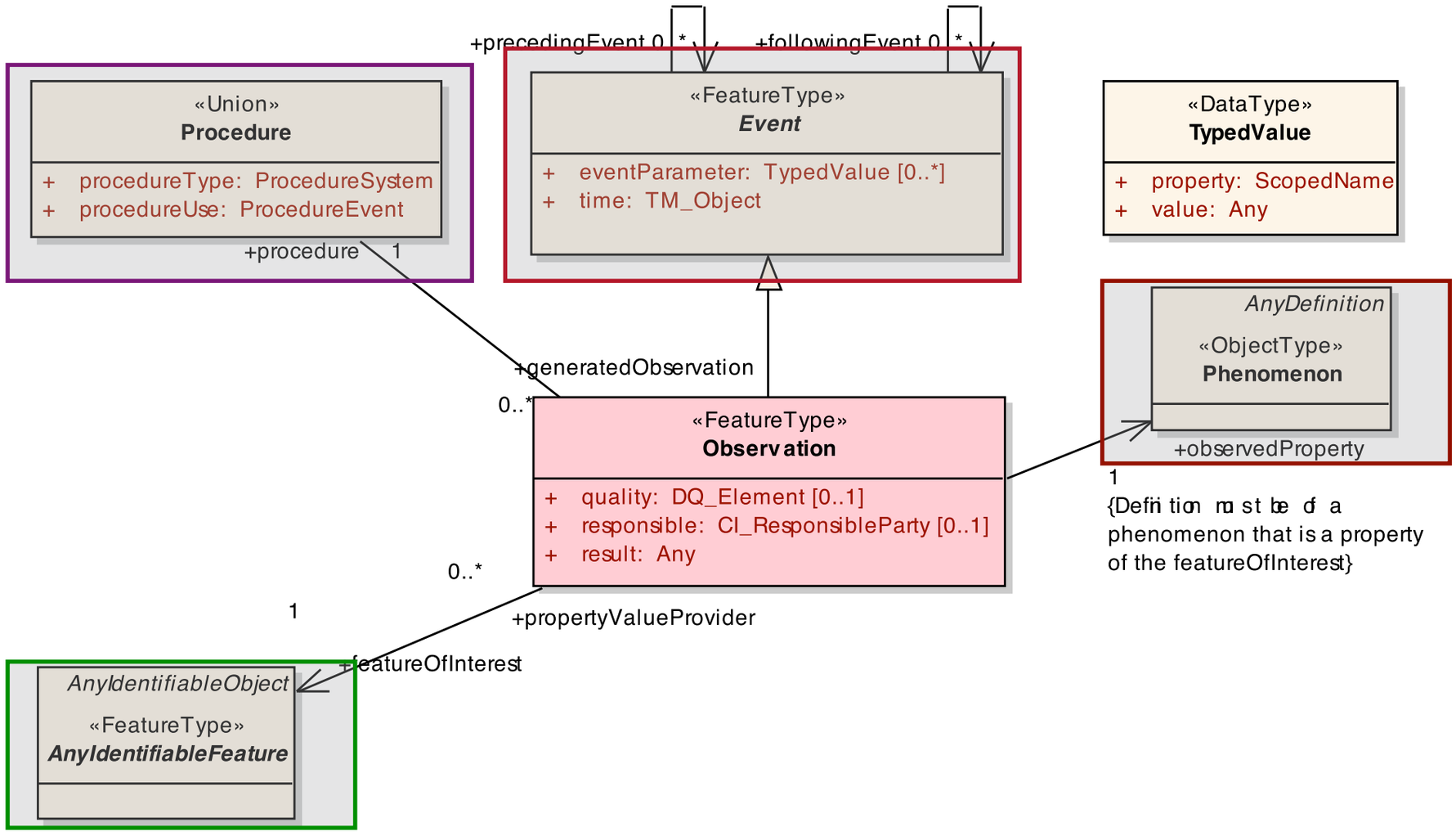
at t_1

Location	Properties			
	Property 1	Property 2	...	Property m
(x_1, y_2)	$Value_1^1$	$Value_1^2$	...	$Value_1^m$
(x_2, y_2)	$Value_2^1$	$Value_2^2$	...	$Value_2^m$
(x_3, y_3)	$Value_3^1$	$Value_3^2$	...	$Value_3^m$
(x_n, y_n)	$Value_n^1$	$Value_n^2$	...	$Value_n^m$

Feature 3

Coverage 2

A common pattern: the observation model



An **Observation** is an **Event** whose **result** is an **estimate** of the **value** of some **Property** of the **Feature-of-interest**, obtained using a specified **Procedure**

The **Feature-of-interest** concept reconciles remote and in-situ observations

OGC

Helping the World to Communicate Geographically

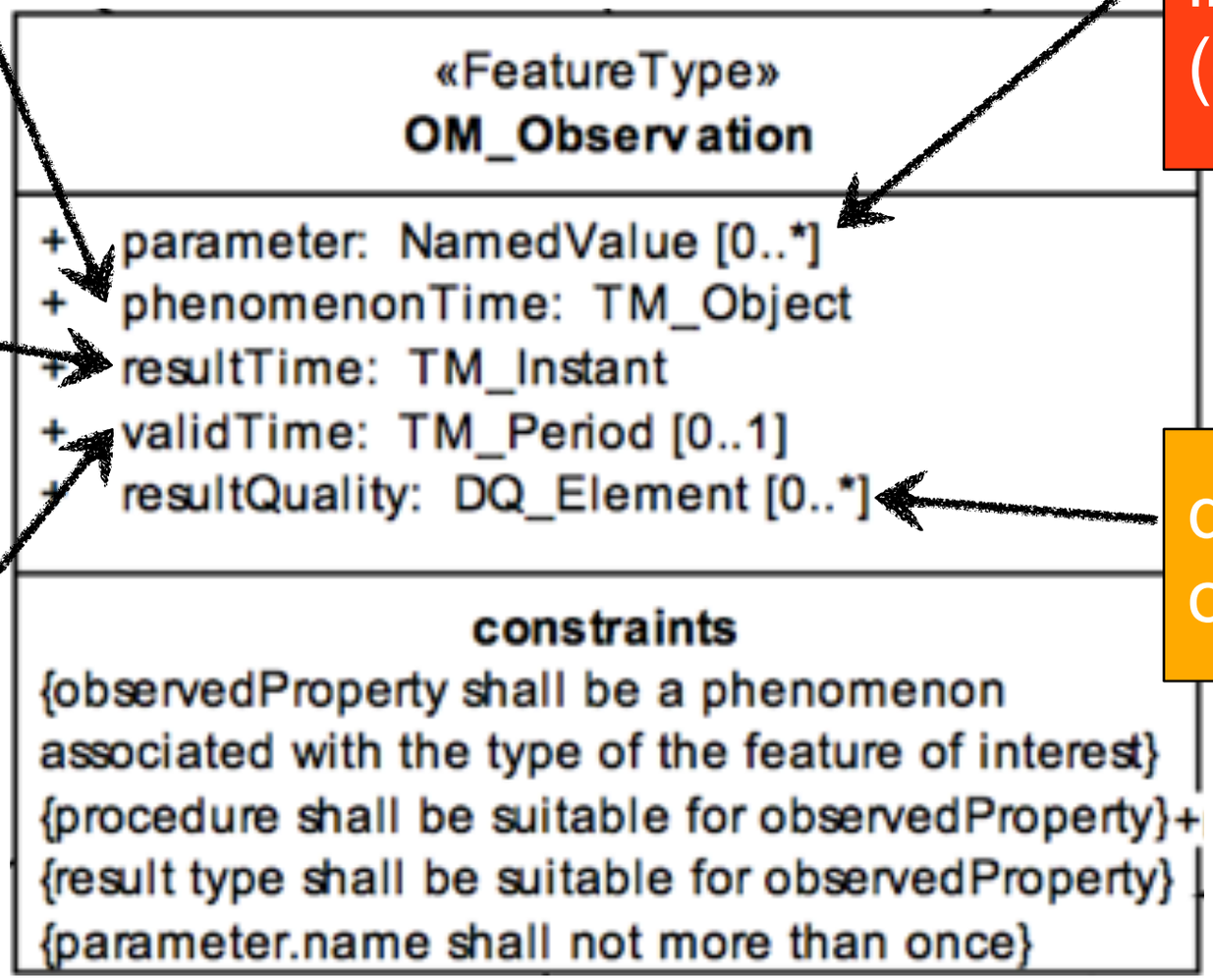
time of the result applied to the property of the FoI

the time when the result becomes available

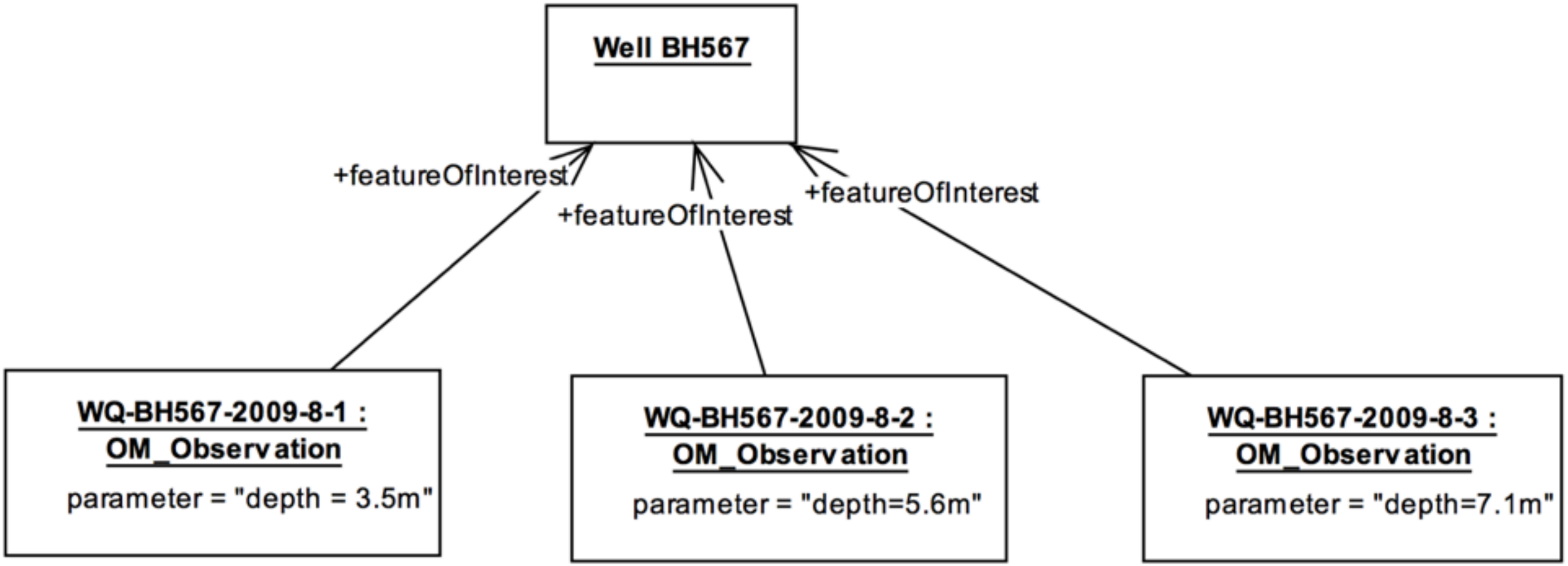
the time period during which the result is intended to be used

an event specific parameter, e.g., instrument setting (depth = 3.5m)

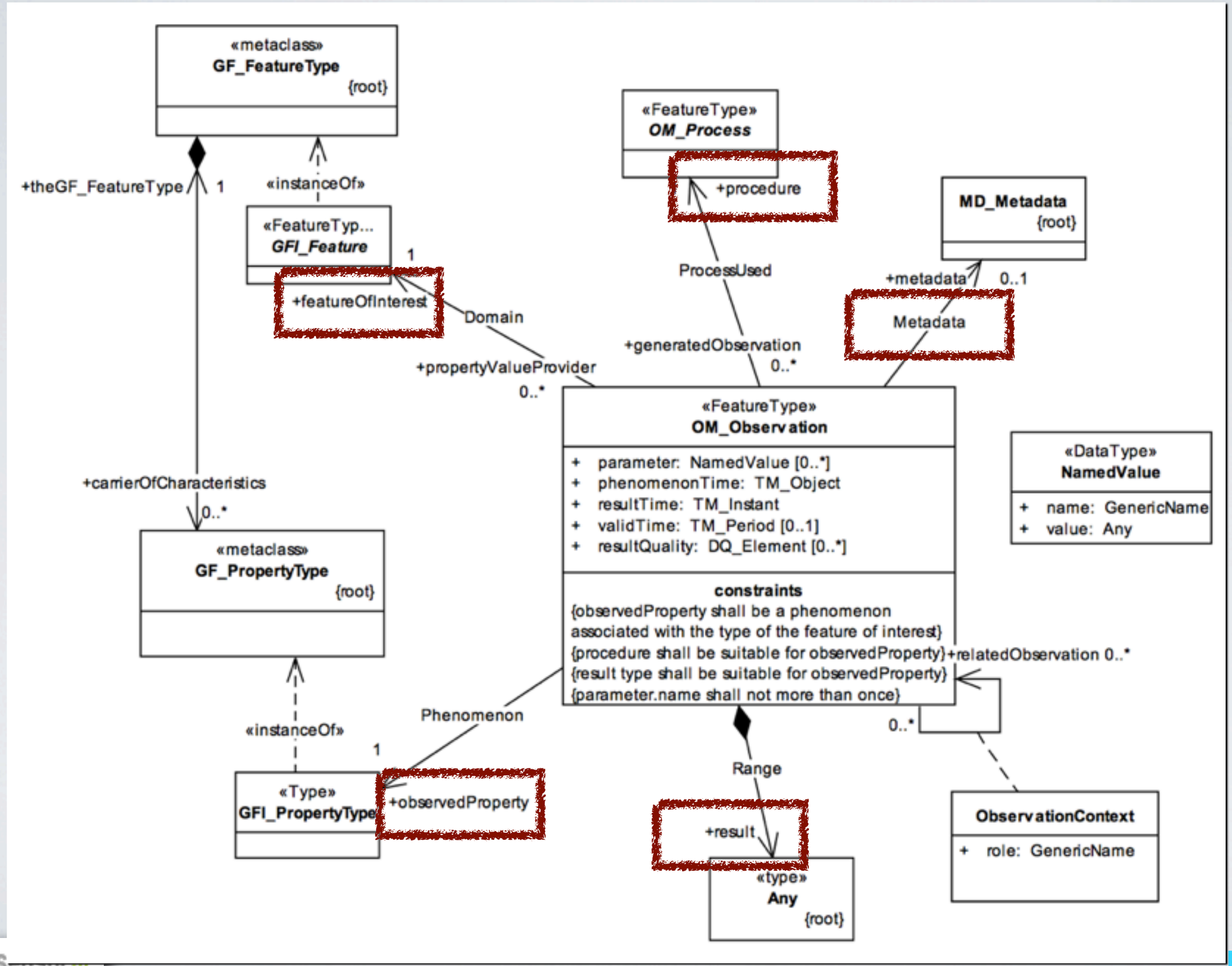
describes the quality of the result



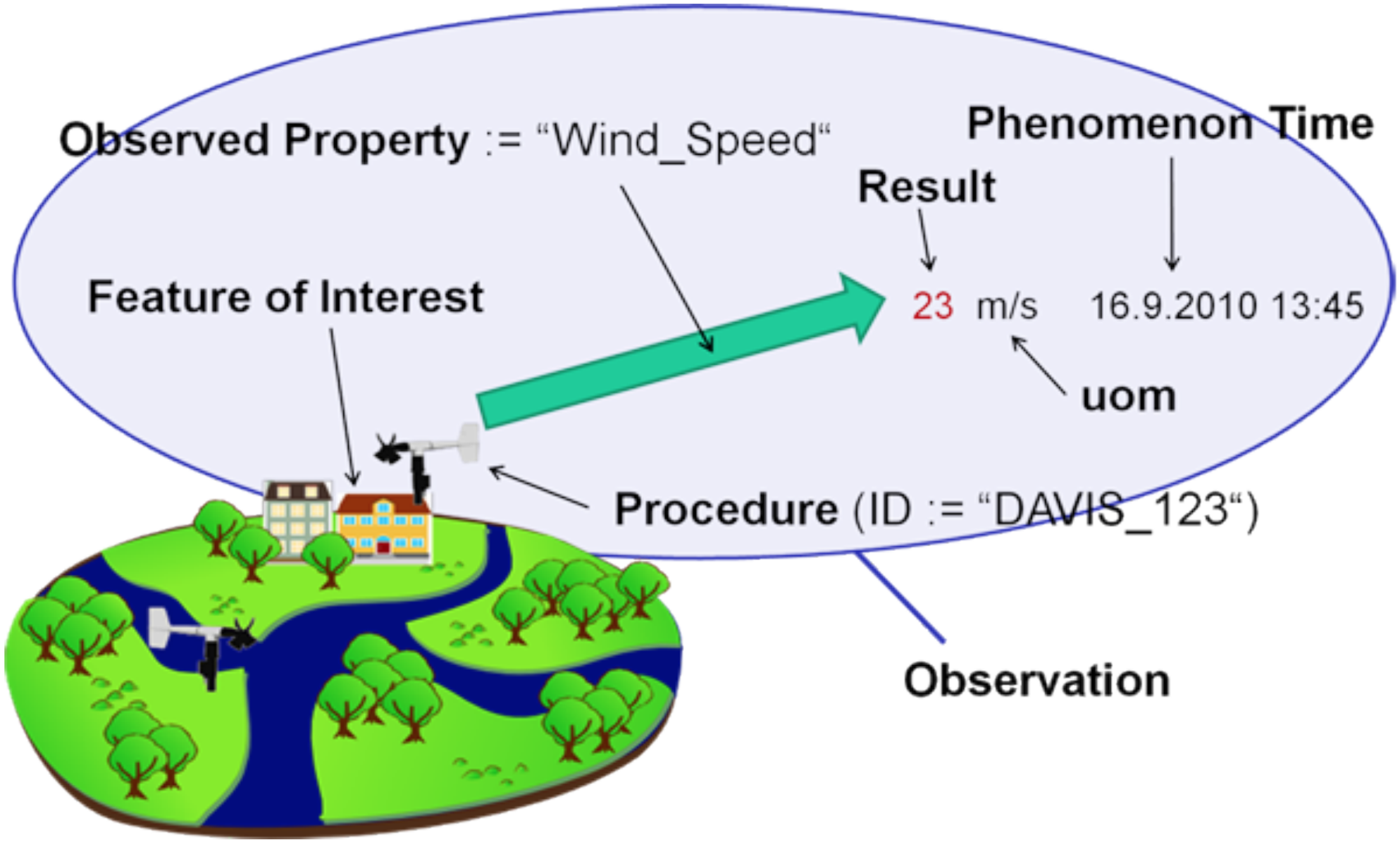
Observation Parameter Example



OBSERVATION AND MEASUREMENT (O&M)



O&M example



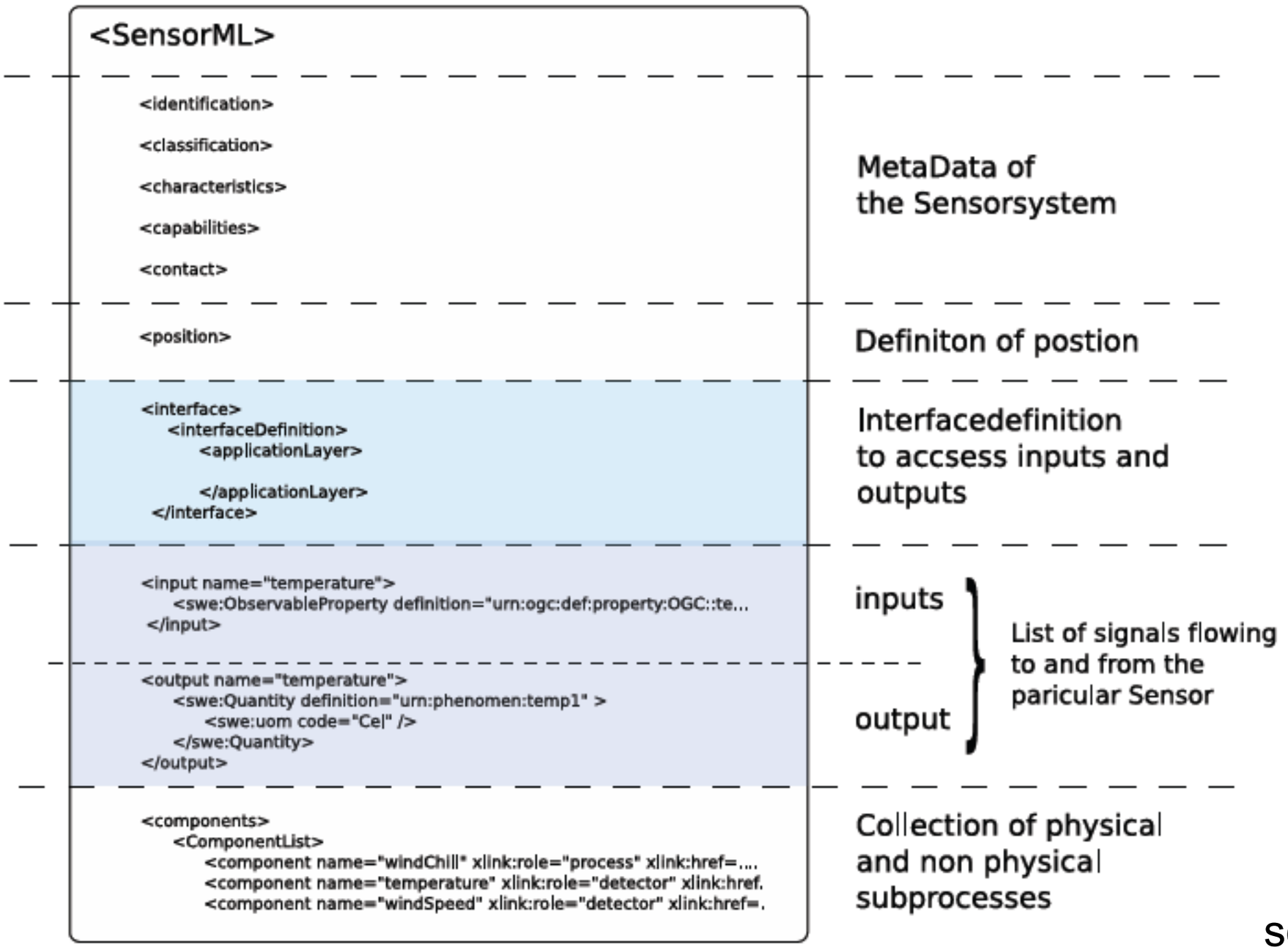
source: OGC Network

```
<om:OM_Observation
    gml:id="obsTest1"
    xmlns:om="http://www.opengis.net/om/2.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:gml="http://www.opengis.net/gml/3.2"
    xsi:schemaLocation="http://www.opengis.net/om/2.0 http://schemas.opengis.net/om/2.0/observation.xsd">

    <gml:description>Observation test instance: fruit mass</gml:description>
    <gml:name>Observation test 1</gml:name>
    <om:type xlink:href="http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement"/>
    <om:phenomenonTime>
        <gml:TimeInstant
            gml:id="ot1t">
            <gml:timePosition>2005-01-11T16:22:25.00</gml:timePosition>
        </gml:TimeInstant>
    </om:phenomenonTime>
    <om:resultTime xlink:href="#ot1t"/>
    <!-- a notional URL identifying a  procedure ... -->
    <om:procedure
        xlink:href="http://www.example.org/register/process/scales34.xml"/>
<!-- environmental conditions during measurement -->
<om:parameter>
    <om:NamedValue>
        <om:name xlink:href="http://sweet.jpl.nasa.gov/ontology/property.owl#Temperature"/>
        <om:value xsi:type="gml:MeasureType" uom="Cel">22.3</om:value>
    </om:NamedValue>
</om:parameter>
<!-- a notional URN identifying the observed property -->
<om:observedProperty
    xlink:href="http://sweet.jpl.nasa.gov/2.0/phys.owl#Mass"/>
<!-- a notional WFS call identifying the object regarding which the observation was made -->
<om:featureOfInterest
    xlink:href="http://wfs.example.org?request=getFeature&amp;featureid=fruit37f "/>

<om:result
    xsi:type="gml:MeasureType"
    uom="kg">0.28</om:result>
<!-- The XML Schema type of the result is indicated using the value of the xsi:type attribute -->
```

SensorML



source: OGC Network

SWE Service Interfaces

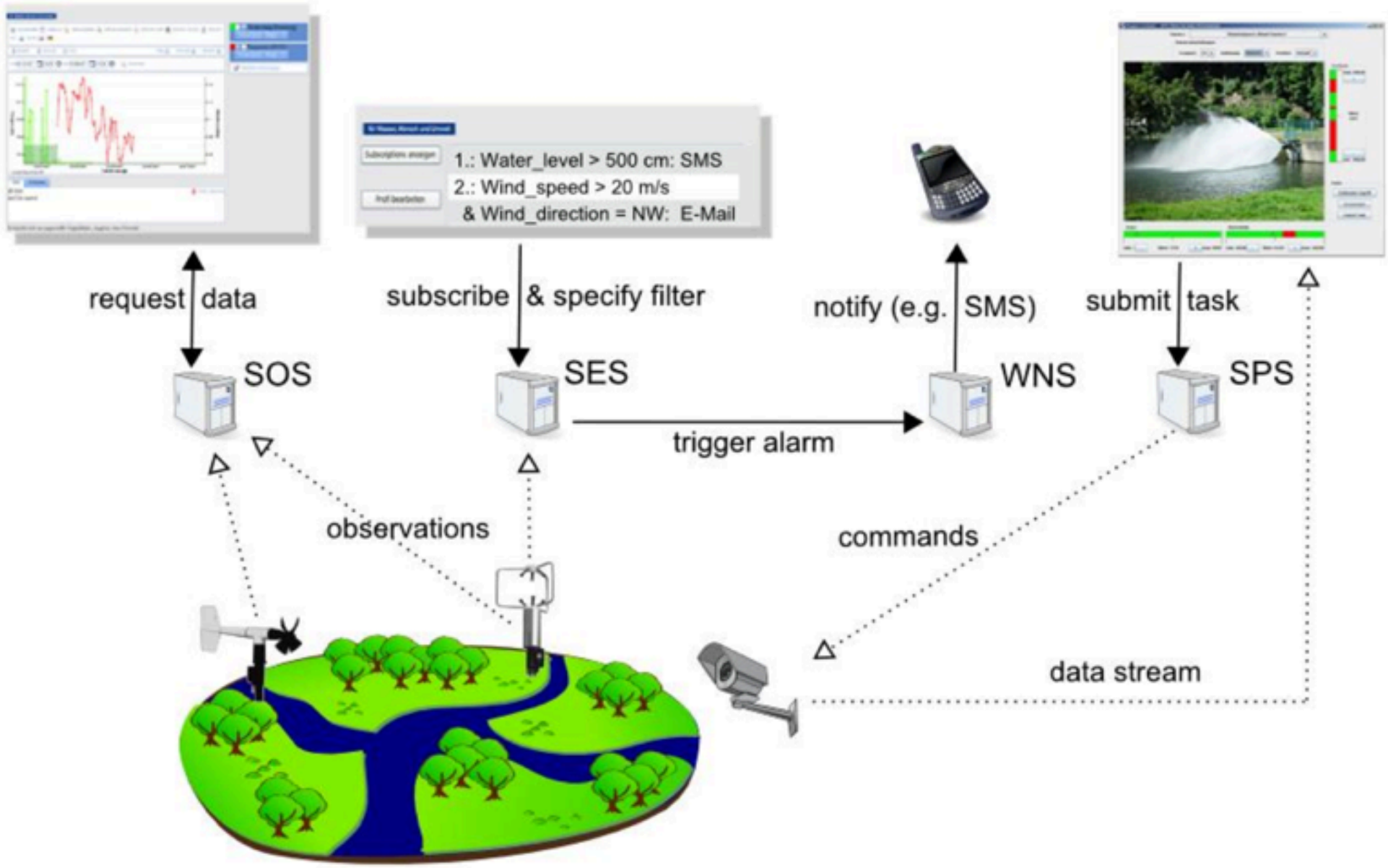
SWE service interfaces

- **Every component exposes a http interface**
  - **Supports service chain composition**

  - **Sensor Observation Service (SOS)**
    - Fine-grained access to sensor observation results
    - GetObservation( ), GetFeatureOfInterest( ), GetResult( ), DescribeSensor(id), …
    - XML document response
  - **Sensor Event Service (SES)**
    - Subscription to events
  - **Sensor Planning Service (SPS)**
    - Observation feasibility and tasking
  - **Sensor registries**
    - Catalogue service profiles for sensors and services

- **Hide the low-level protocols– e.g. IEEE 1451**

An Example Deployment



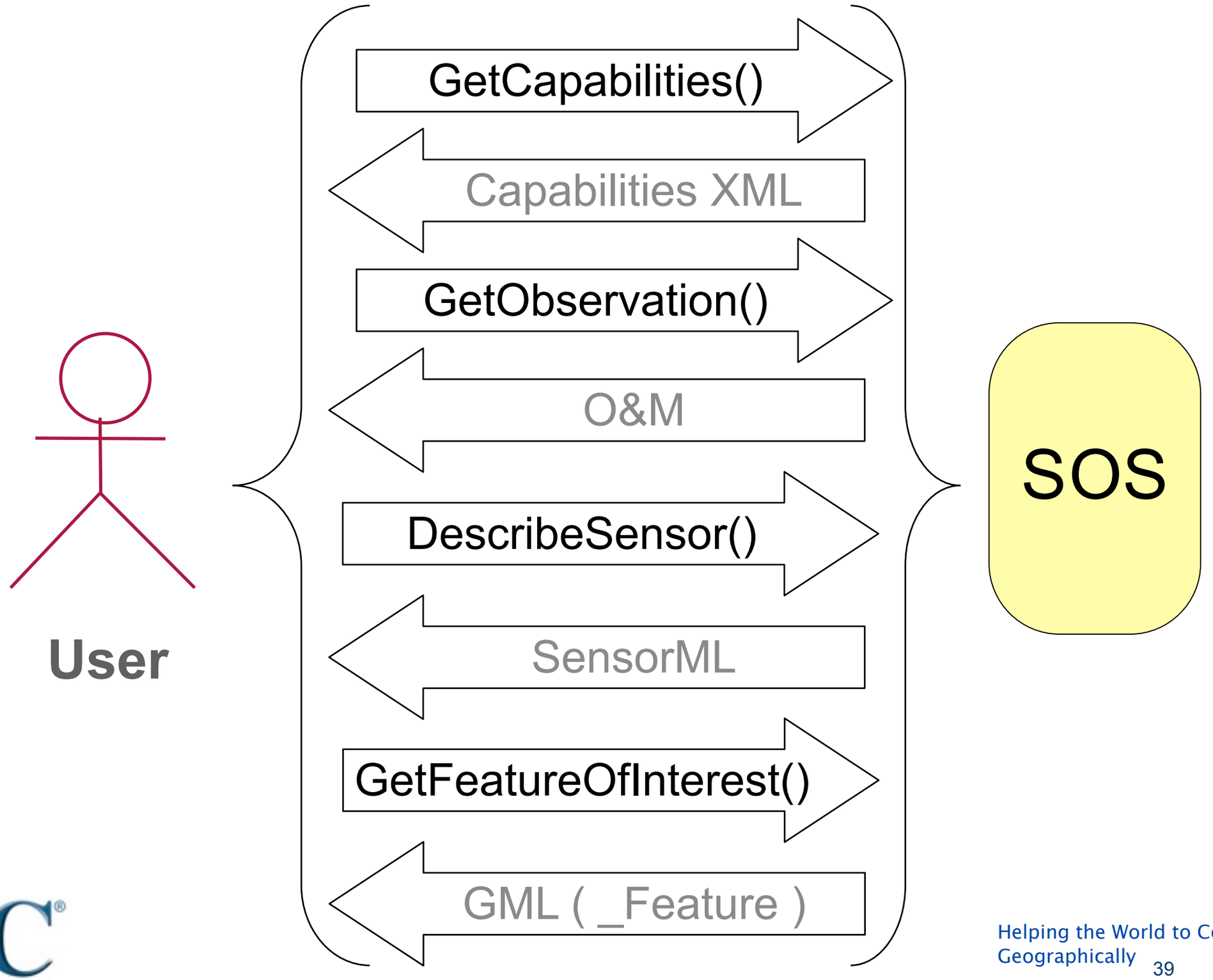
Sensor Observation Services (SOS)

Sensor Observation Service (SOS)

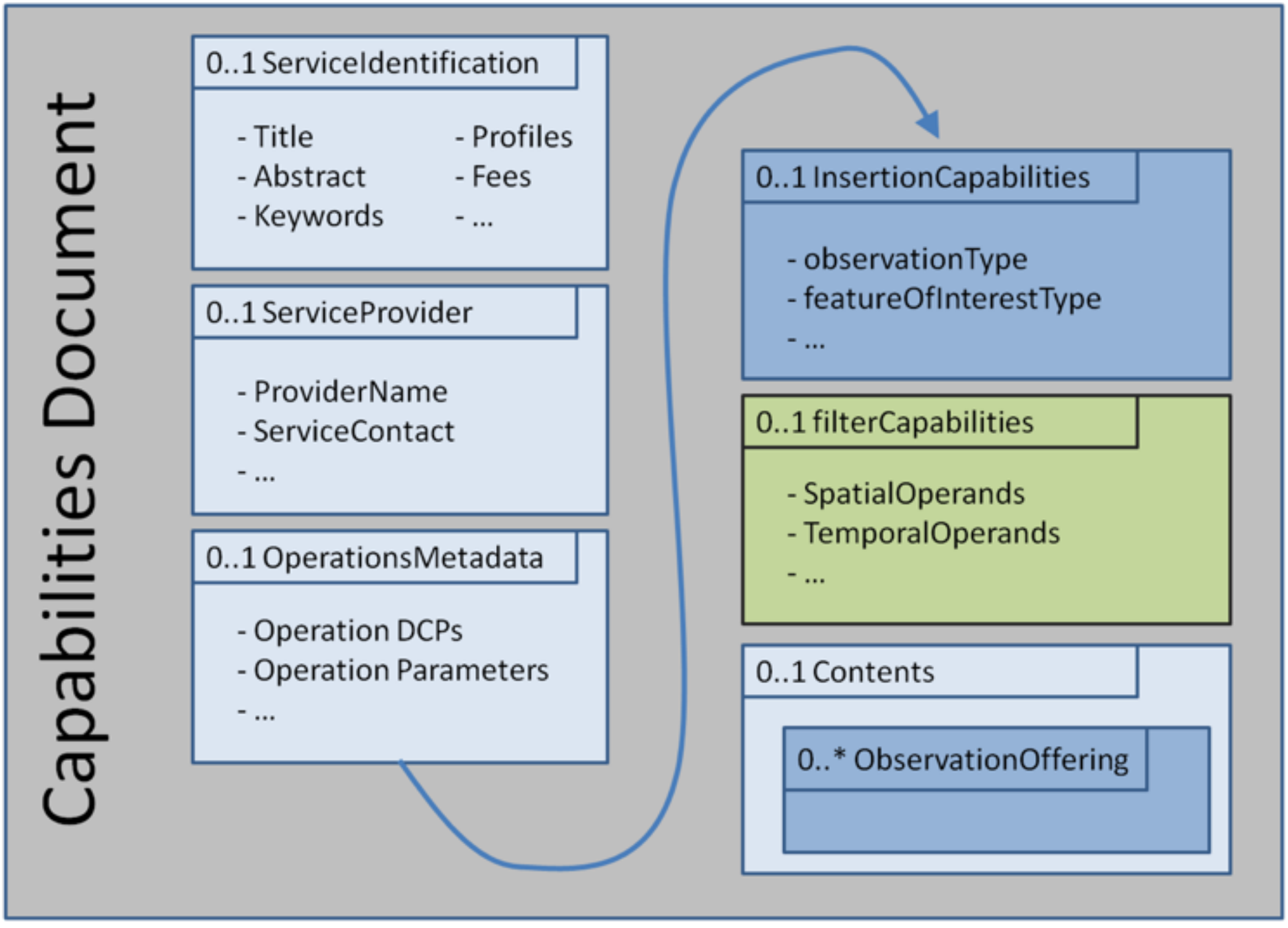
- ***premises:***
- **O&M is the high-level information model**
- **SOS is the primary information-access interface**

- **SOS can serve:**
  - **an *Observation* (O&M)**
    - getObservation()
  - **a *feature of interest* (Feature)**
    - getFeatureOfInterest()
  - **a *sensor* (SensorML document)**
    - describeSensor()
  - **or *Observation/result* (often a time-series)**
    - getResult()

Example – Sensor Observation Service

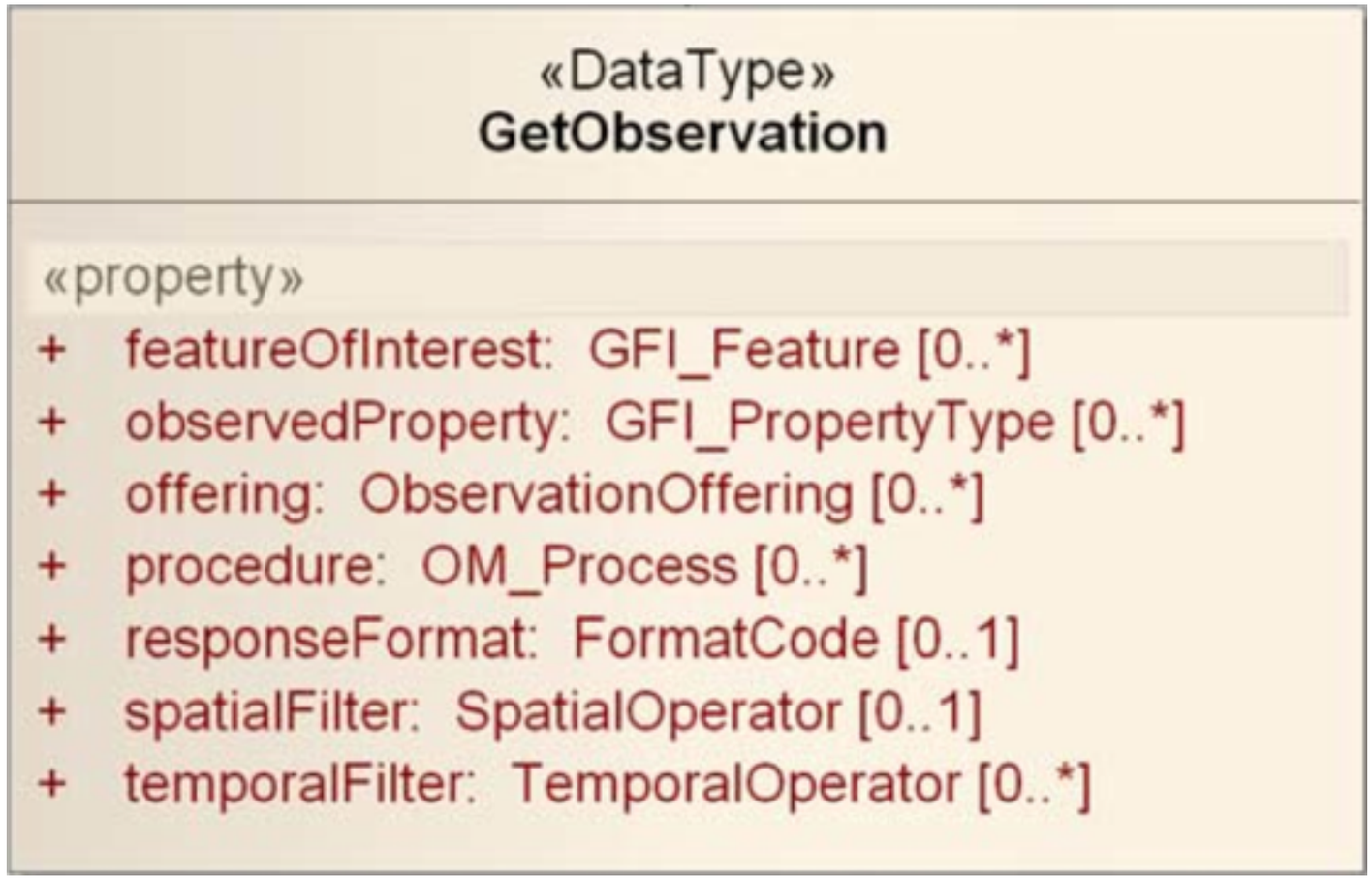


GetCapabilities response

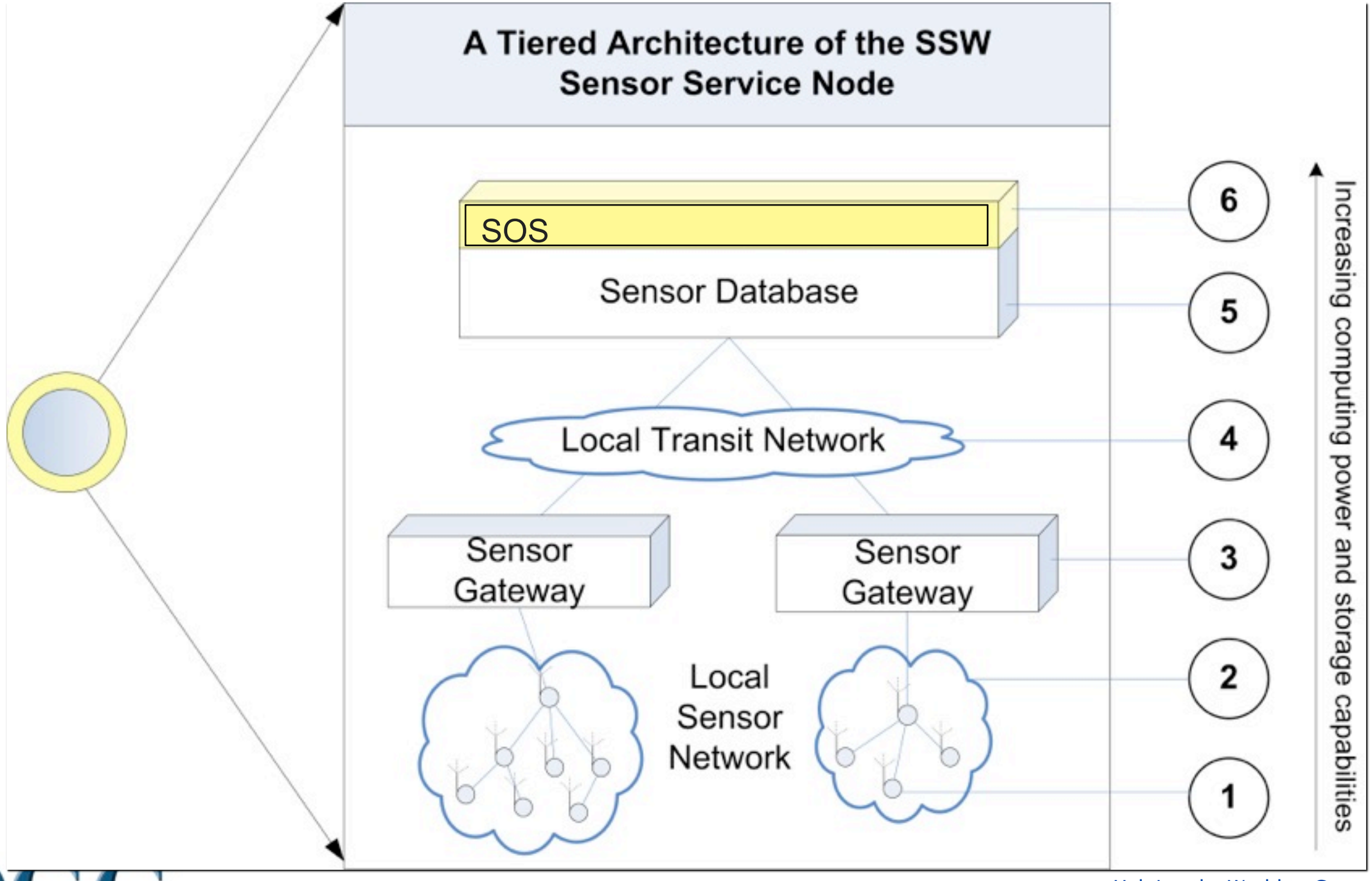


source: OGC Network

GetObservation()



Interested in implementing your own SOS?
A Layering Diagram of an SOS



SOS Evaluation

- For a healthy SWE developer and user community, it's very important to evaluate ourselves regularly.
- We compared three open source SOS implementations, namely
  - 52North
  - MapServer
  - Deegree

Poorazizi, M. E., A. Hunter, and S. H. L. Liang, "**Testing of Sensor Observation Services: A Performance Evaluation**", *The First ACM SIGSPATIAL International Workshop on Sensor Web Enablement 2012 (SWE2012)*, Redondo Beach, California, ACM Digital Library, 2012.

#sensor and response time

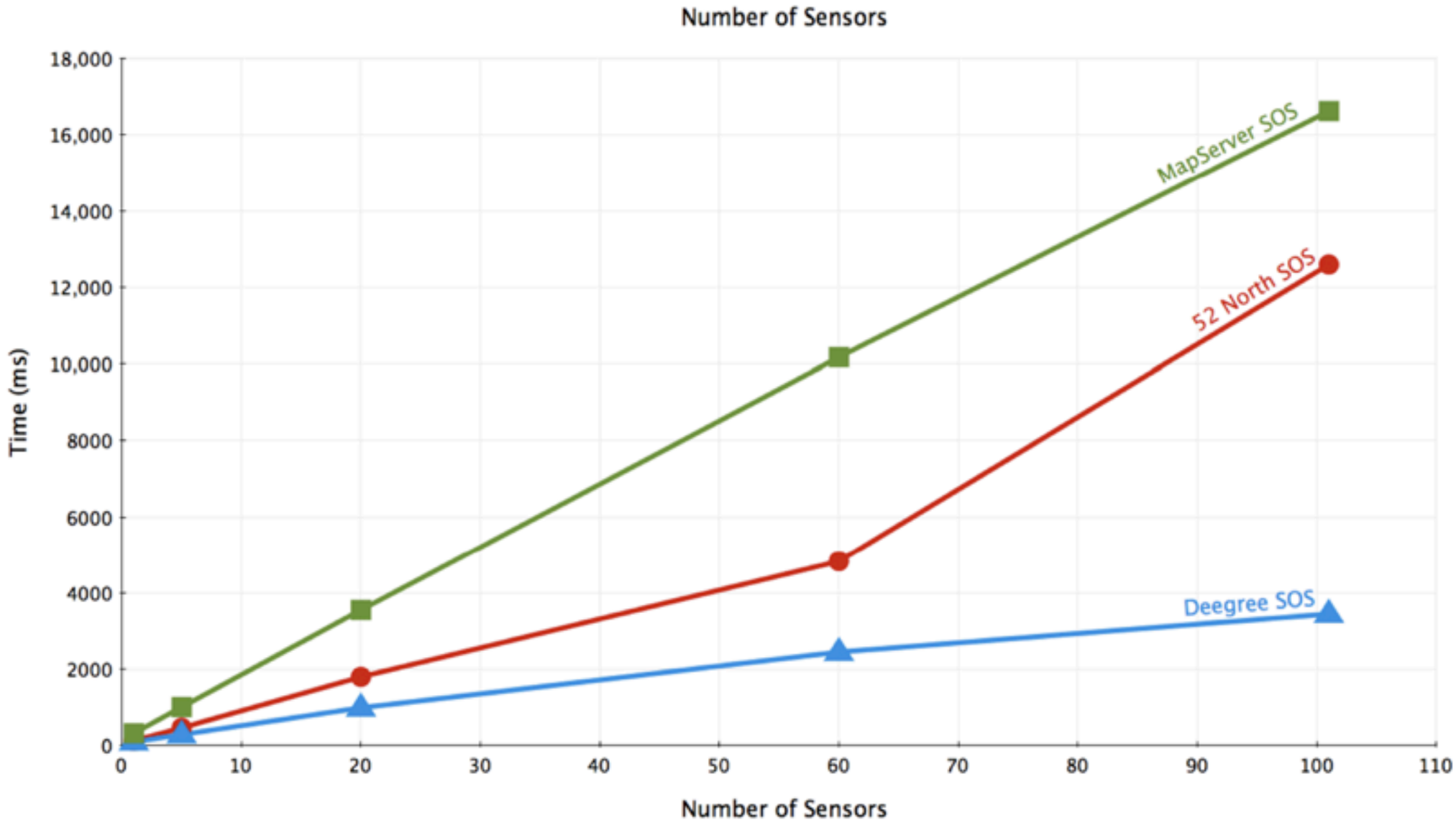


Figure 3. Response time (bottom) and size (top) for GetObservation requests based on number of sensors

Poorazizi, M. E., A. Hunter, and S. H. L. Liang, "**Testing of Sensor Observation Services: A Performance Evaluation**", *The First ACM SIGSPATIAL International Workshop on Sensor Web Enablement 2012 (SWE2012)*, Redondo Beach, California, ACM Digital Library, 2012.

Bbox size and response time

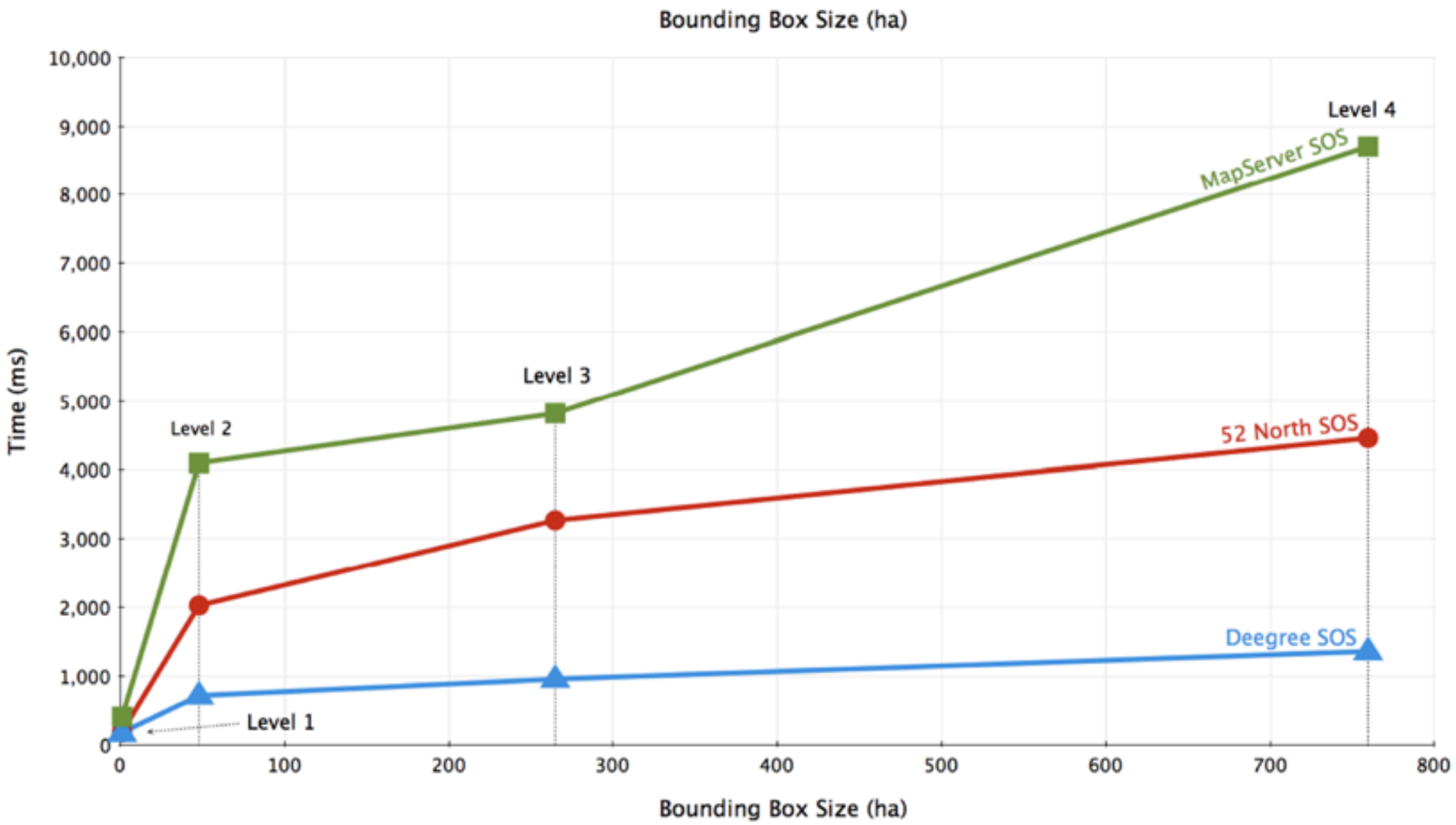


Figure 4. Response time (bottom) and size (top) for GetObservation requests based on bounding box levels

Poorazizi, M. E., A. Hunter, and S. H. L. Liang, "**Testing of Sensor Observation Services: A Performance Evaluation**", *The First ACM SIGSPATIAL International Workshop on Sensor Web Enablement 2012 (SWE2012)*, Redondo Beach, California, ACM Digital Library, 2012.

Temporal interval and response time

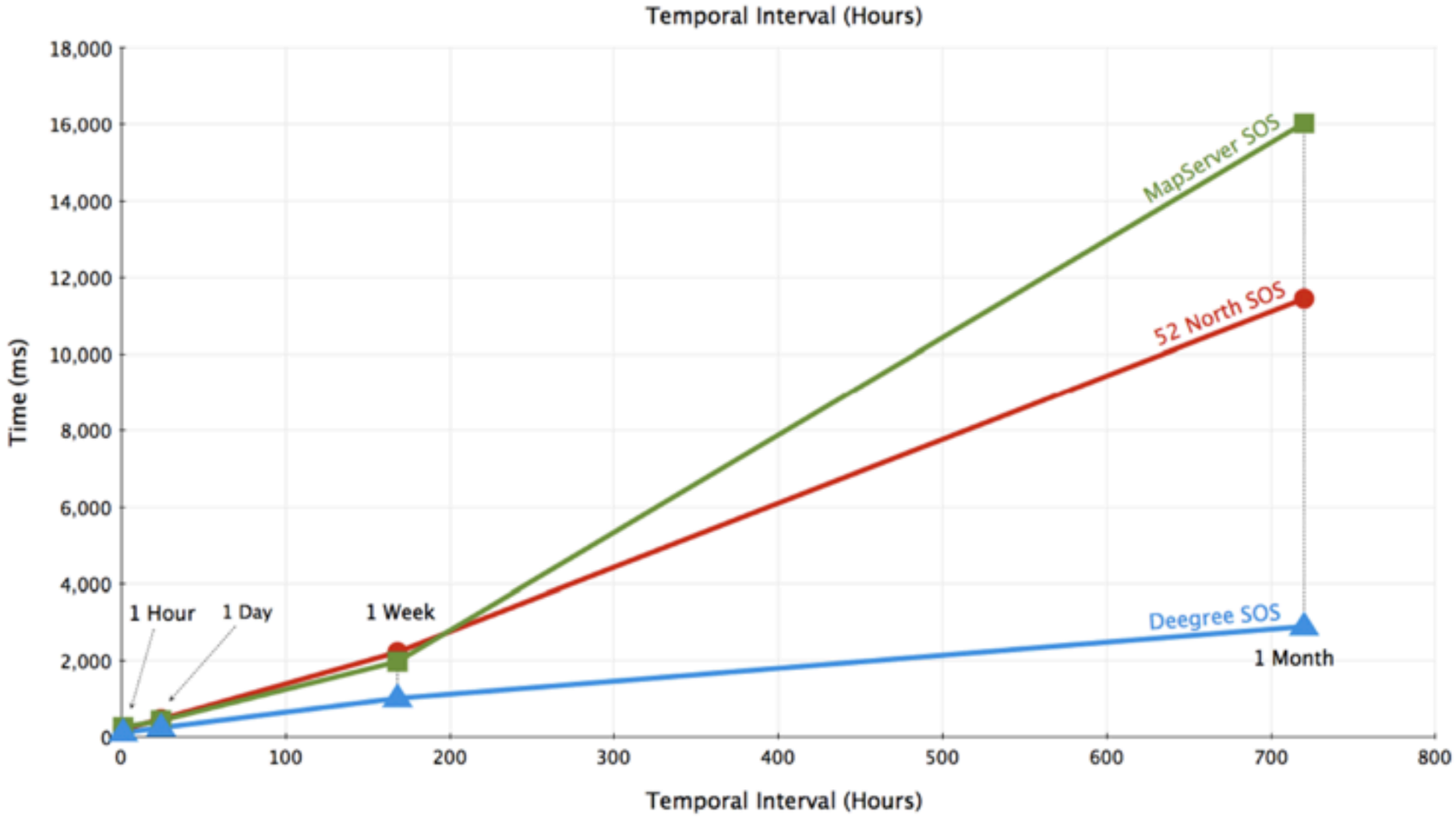
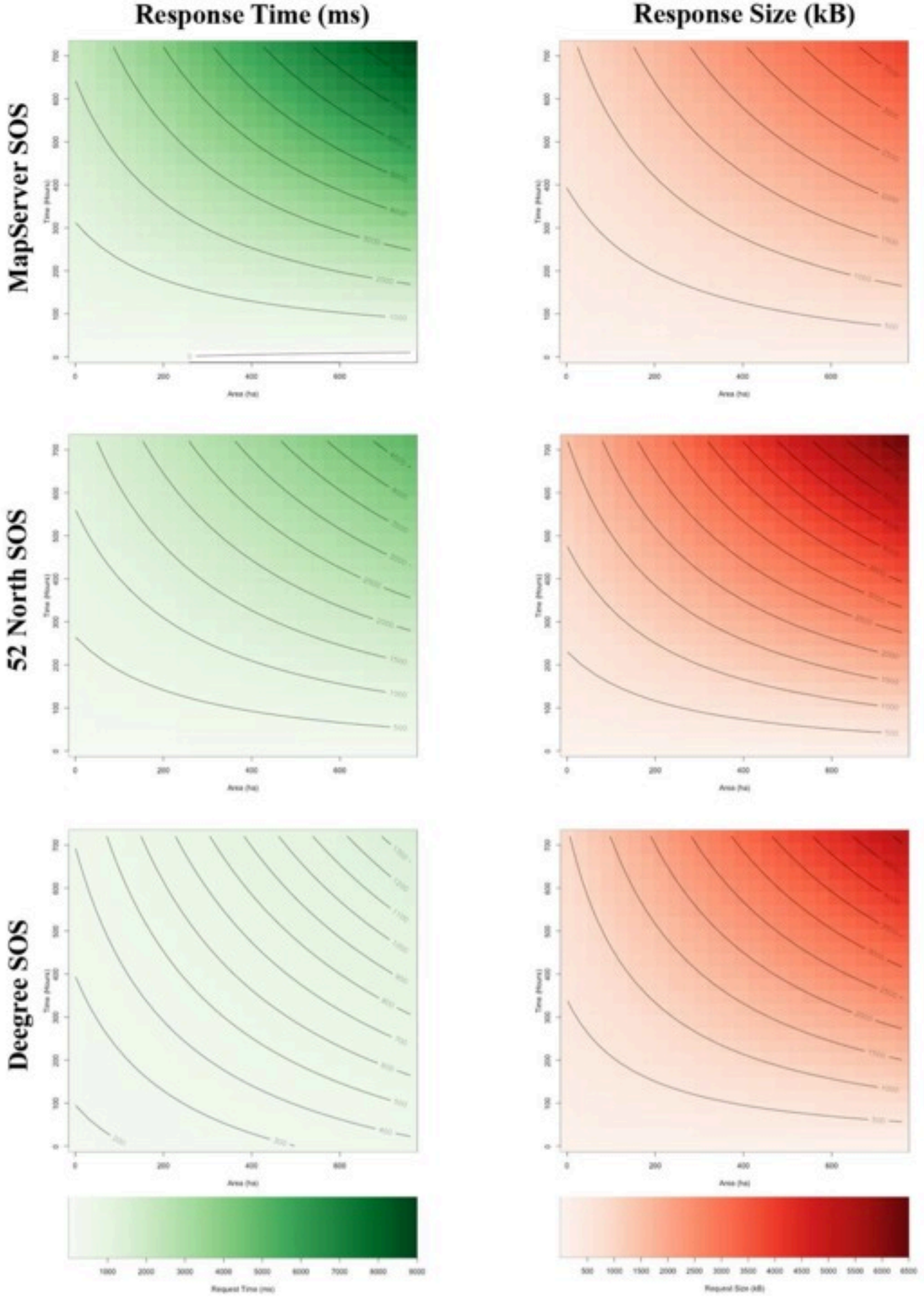


Figure 5. Response time (bottom) and size (top) for GetObservation request based on different time scales

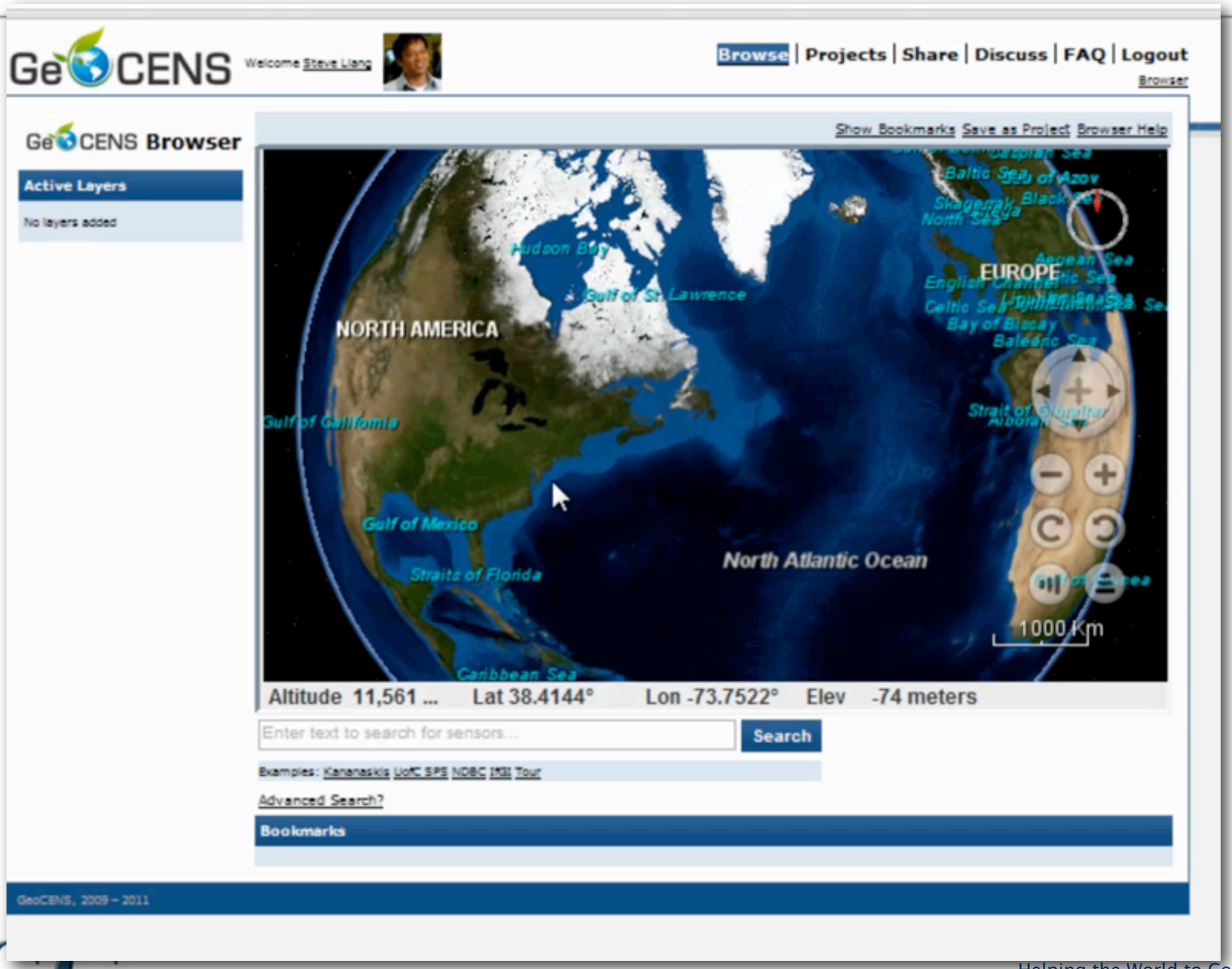
Poorazizi, M. E., A. Hunter, and S. H. L. Liang, "**Testing of Sensor Observation Services: A Performance Evaluation**", *The First ACM SIGSPATIAL International Workshop on Sensor Web Enablement 2012 (SWE2012)*, Redondo Beach, California, ACM Digital Library, 2012.



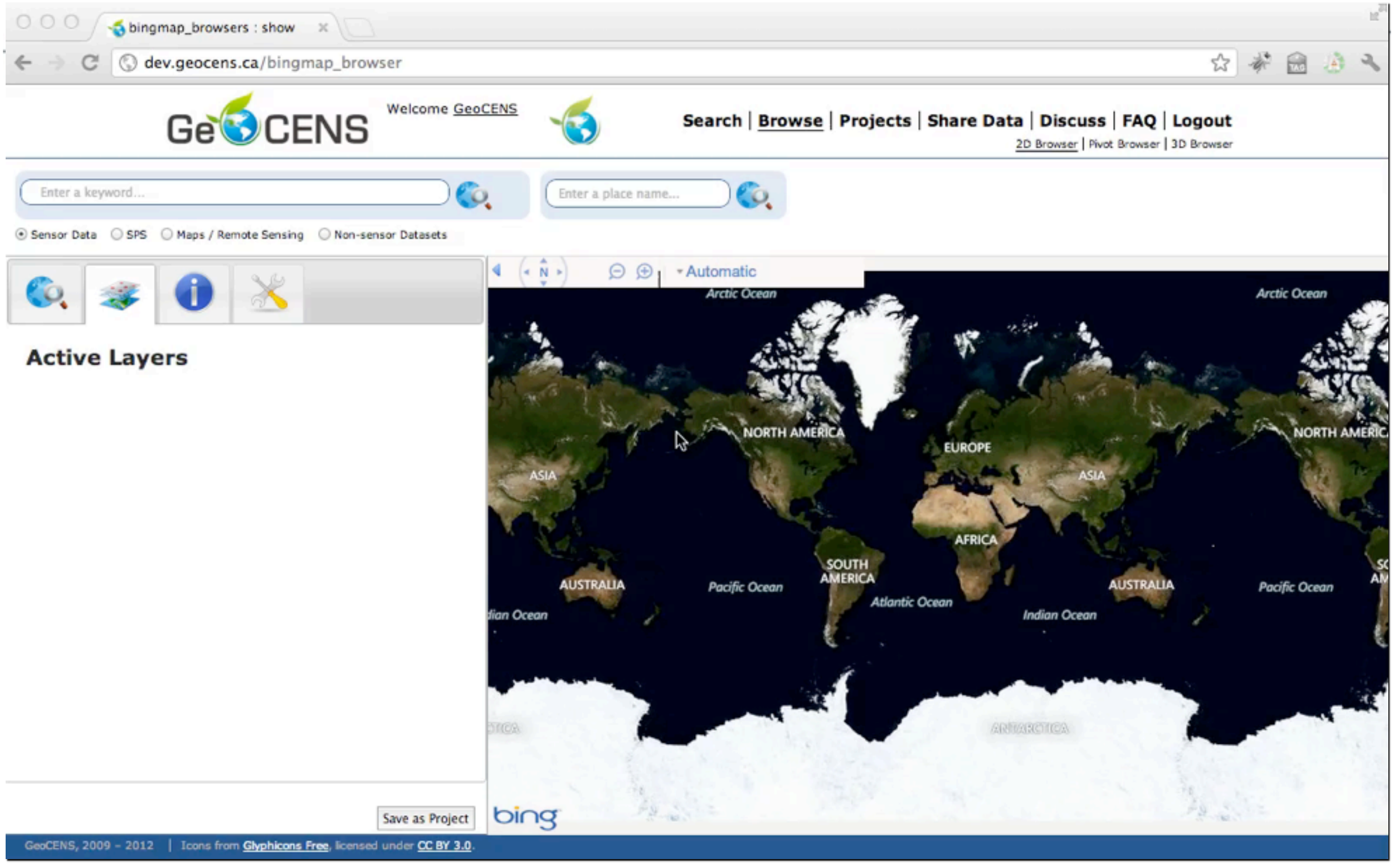
Poorazizi, M. E., A. Hunter, and S. H. L. Liang, "**Testing of Sensor Observation Services: A Performance Evaluation**", *The First ACM SIGSPATIAL International Workshop on Sensor Web Enablement 2012 (SWE2012)*, Redondo Beach, California, ACM Digital Library, 2012.

SWE Examples

GeoCENS 3D Sensor Web Browser



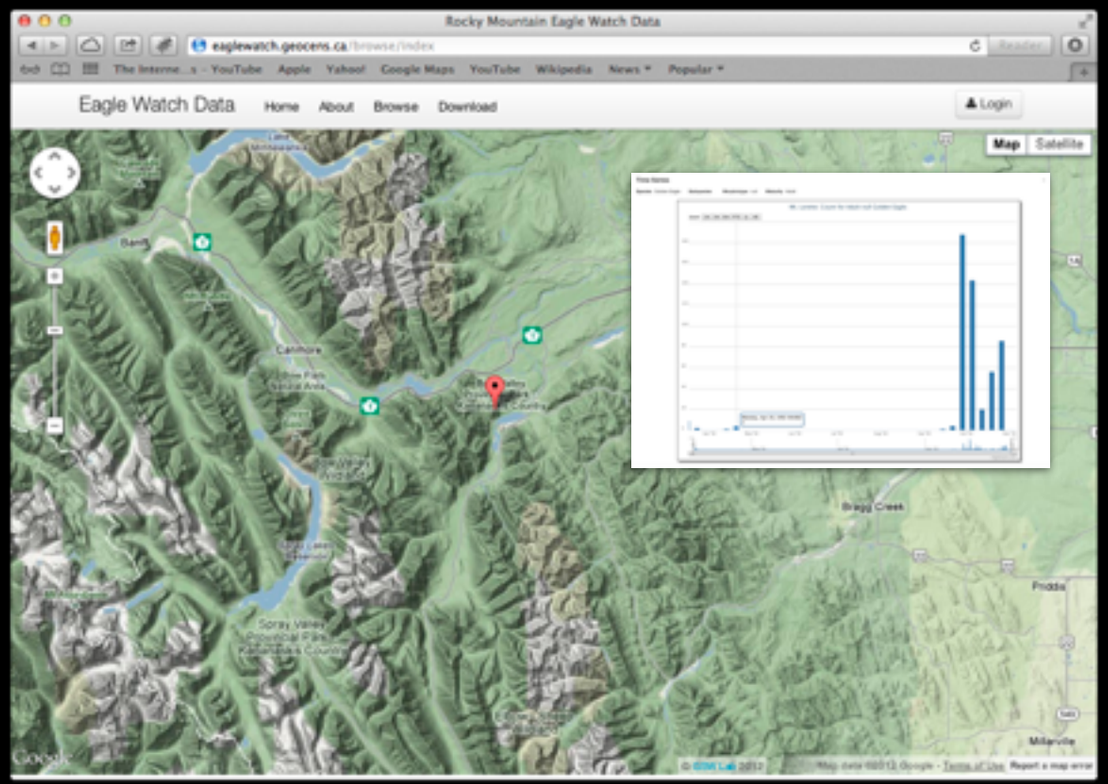
GeoCENS 2D Browser (http://dev.geocens.ca)



AAFC Real-time In-situ Soil-Moisture Monitoring for Agriculture

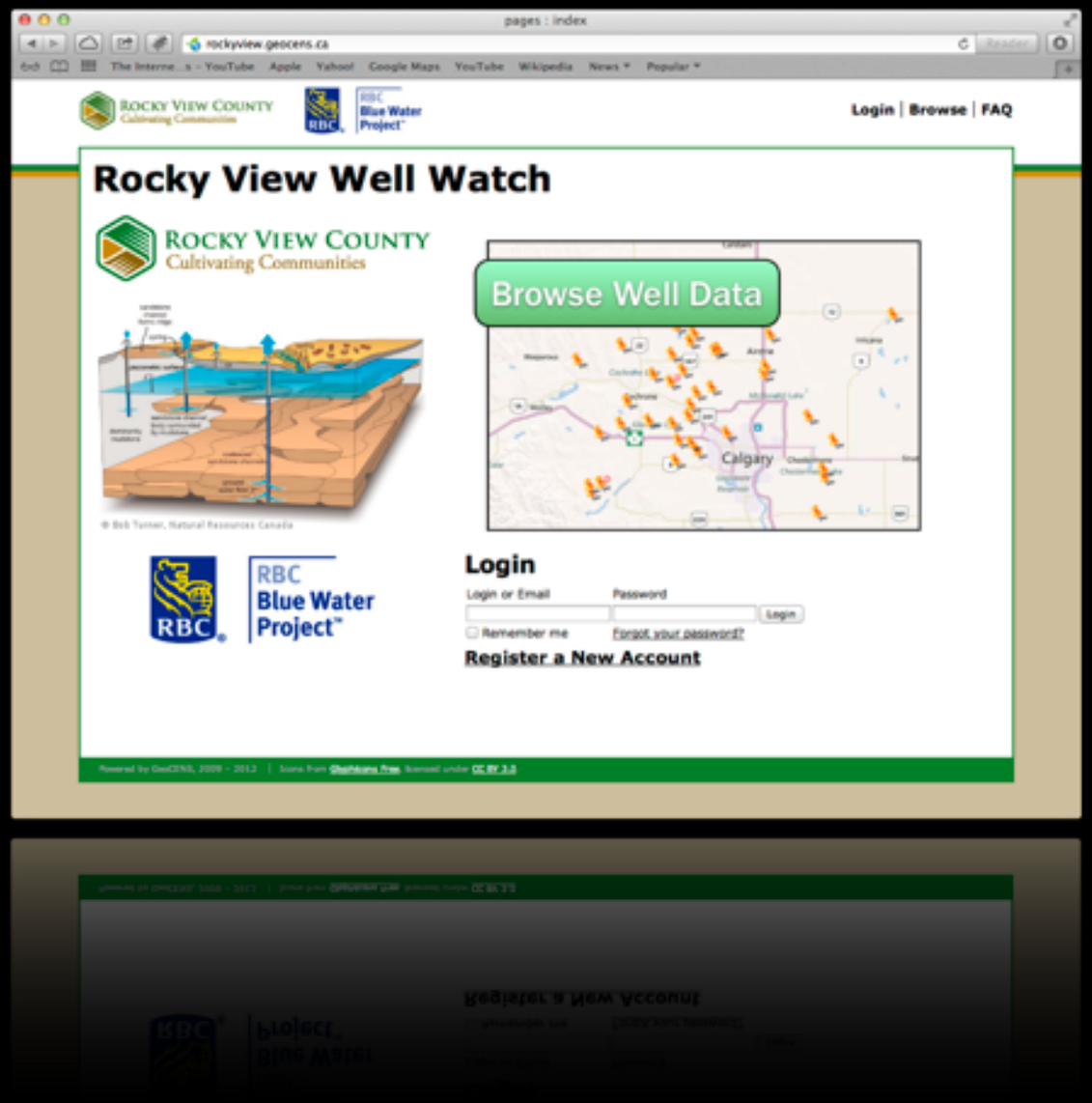
- Agriculture and Agri-Food Canada is deploying soil moisture monitoring networks across Canada
- empowered by GeoCENS
- more than 2,000 sensors online
- NASA JPL is connecting to RISMA for the SMAP satellite mission

Eagle Watch



- a citizen sensing project counting eagles for more than 20 years in the Rockies
- used to use paper and excel data sheets to record and share data
- the Eagle Watch portal ***significantly simplify*** the data entry, processing, and sharing work

Rockyview Well Watch

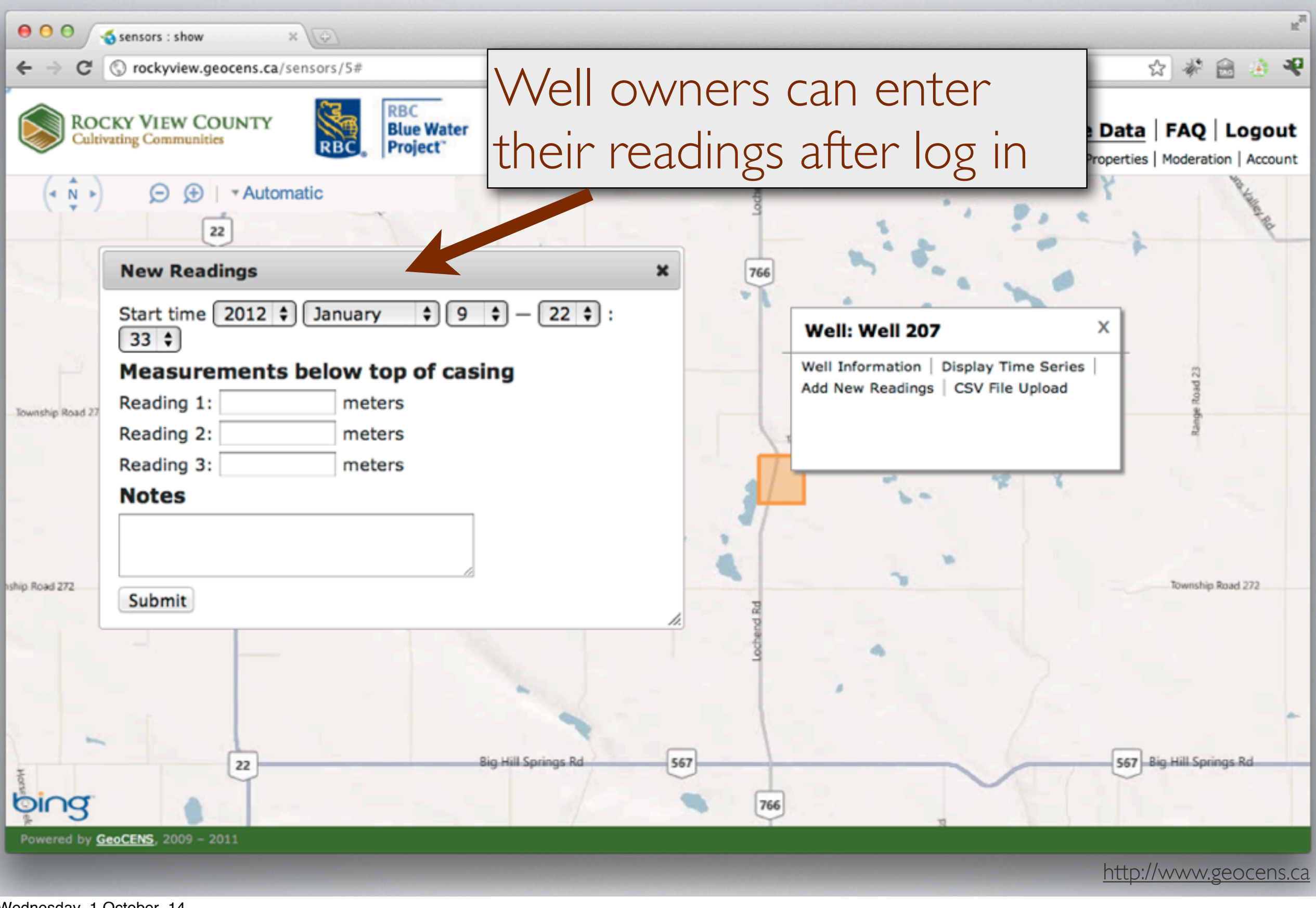


- A RBC Blue Water Project
- ***affordable*** and ***scalable*** long term groundwater monitoring
- more than 40 well owners
- data since 2008
- QA/QC by UofC Hydrologists

Project coordinator oversees all groundwater wells (QA/QC). She is the only person has the permission to edit and put notes for all wells.

http://www.geocens.ca

Well owners can enter their readings after log in



http://www.geocens.ca

sensors : show

rockyview.geocens.ca/sensors/4#

Rocky View County
Cultivating Communities

RBC Blue Water Project™

Welcome steve.liang

Browse | **Share Data** | **FAQ** | **Logout**

Sensor Data | Observed Properties | Moderation | Account

Automatic

Time Series Graph

Well 206 – Rocky View Groundwater

missing data

Well: Well 206

Well Information | Display Time Series

Add New Readings | CSV File Upload

Township Road 272

Big Hill Springs Rd

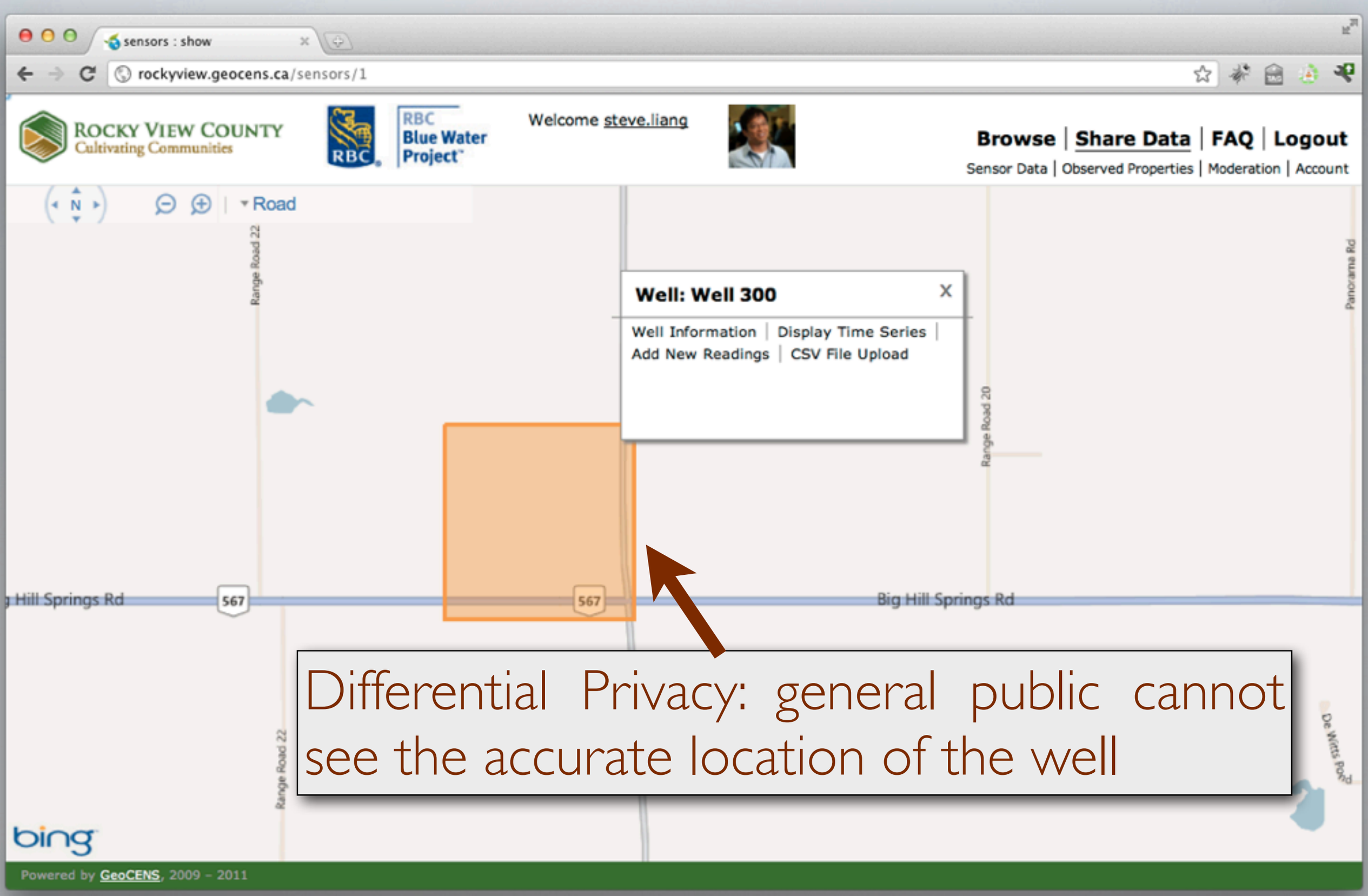
Symons Valley Rd

Horse Creek Rd

bing

Powered by GeoCENS, 2009 - 2011

http://www.geocens.ca



Differential Privacy: general public cannot see the accurate location of the well

http://www.geocens.ca

sensors : show

rockyview.geocens.ca/sensors/5#

Rocky View County
Cultivating Communities

RBC Blue Water Project

Welcome steve.liang

Browse | Share Data | FAQ | Logout

Sensor Data | Observed Properties | Moderation | Account

Automatic

Well Information

Name
Well 207

Description
Monitored by the University of Calgary using an automatic data logger.

Latitude *
51.32

Longitude *
-114.32

Elevation
1272.02 m

Casing Height
0.4 m

Approved for public viewing?
Yes

Available for public access?
Yes

* Latitude and longitude have been truncated for privacy reasons

Well: Well 207

Well Information | Display Time Series | Add New Readings | CSV File Upload

Well information (metadata)

Township Road 274

Range Road 42

Range Road 23

22

766

567

Big Hill Springs Rd

bing

Powered by GeoCENS, 2009 – 2011

http://www.geocens.ca

wellobsvs : index

Data management interface for individual readings

Browse | Share Data | FAQ | Logout

Listing of Well Readings

Well Name	Date and Time	Reading 1	Reading 2	Reading 3	Notes	SOS Upload	Approved	Actions
Well 21	2007-08-28 12:00	59.35	59.37	59.37		Completed	true	View Edit Delete
Well 21	2007-09-16 12:30	59.3	59.295	59.29	Pump shut off 6 hours before	Completed	true	View Edit Delete
Well 21	2007-10-01 13:05	59.33	59.325	59.235	Pump and valve shut off for 3 hours	Completed	true	View Edit Delete
Well 21	2007-10-19 14:10	59.285	59.285	59.29		Completed	true	View Edit Delete
Well 21	2007-11-01 15:15	59.32	59.32	59.32	some issues with meter.	Completed	true	View Edit Delete
Well 21	2007-12-14 14:00	59.34	59.34	59.34		Completed	true	View Edit Delete
Well 21	2008-01-14 15:00	59.29	59.28	59.29		Completed	true	View Edit Delete
Well 21	2008-02-14 15:30	59.34	59.34	59.34	To the top of well casing	Completed	true	View Edit Delete
Well 21	2008-02-29 16:00	59.3	59.3	59.3		Completed	true	View Edit Delete
Well 21	2008-04-01 15:15	59.36	59.36	59.36		Completed	true	View Edit Delete
Well 21	2008-04-16 12:00	59.38	59.38	59.377	to top of well casing, pump off 6 hours	Completed	true	View Edit Delete
Well 21	2008-04-29 12:00	59.34	59.34	59.335	Pump off 2 hours	Completed	true	View Edit Delete
Well 21	2008-06-02 15:00	59.385	59.375	59.38	Pump off 2 hours	Completed	true	View Edit Delete
Well 21	2008-06-17 12:00	59.325				Completed	true	View Edit Delete
Well 21	2008-07-06 12:00	59.31				Completed	true	View Edit Delete

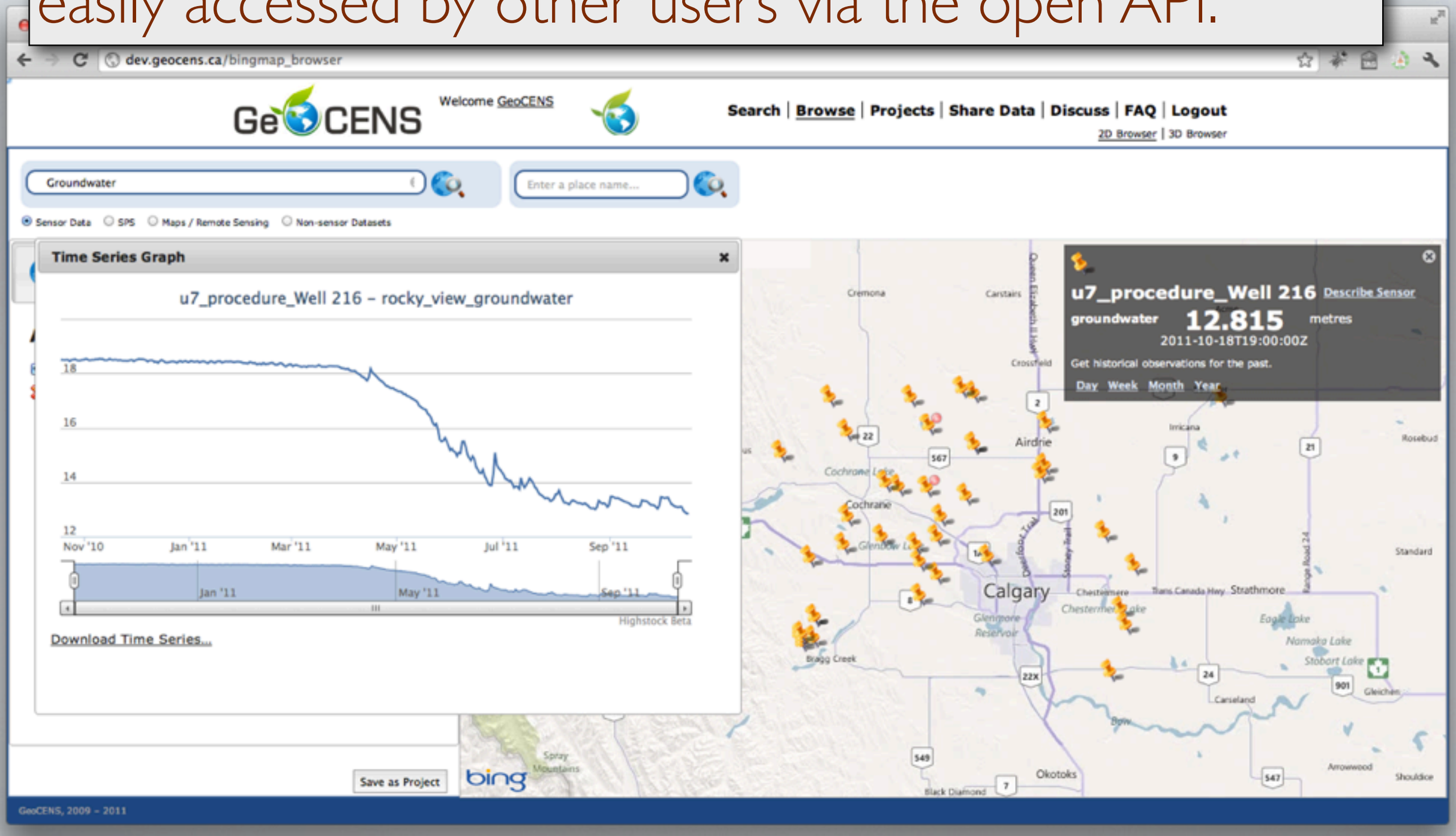
« Previous 1 2 3 4 5 6 Next »

New Well Reading

Events associated with the readings

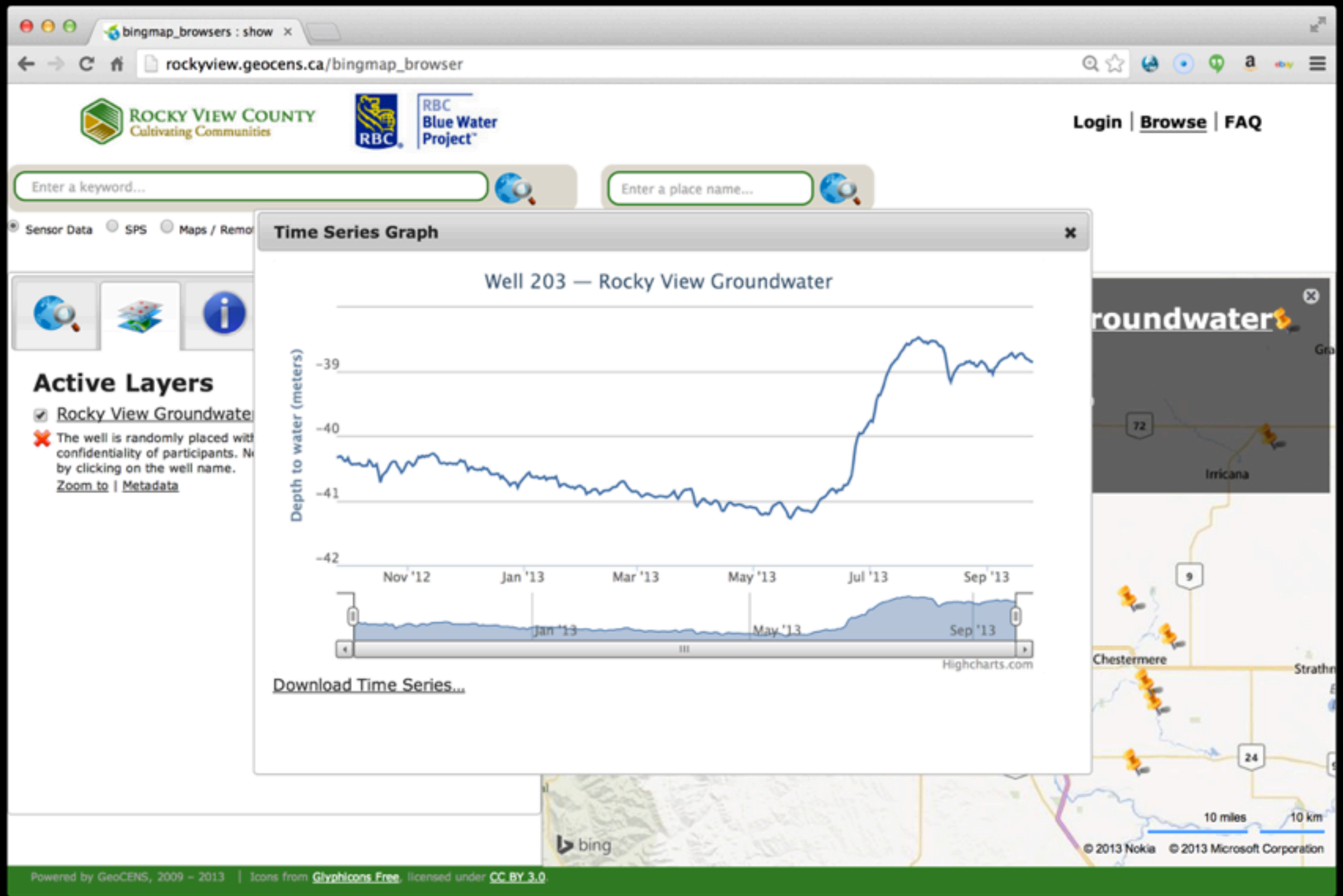
http://www.geocens.ca

GeoCENS is an open system. i.e., all readings are published in open and international standards. Can be easily accessed by other users via the open API.



http://www.geocens.ca

Calgary Flood 2013



OGC SensorThings API

SensorThings API Scope

- Based on the comprehensive OGC SWE framework
- Develop an efficient API that is more suitable for the resource-constrained IoT devices

SensorThings API FAQ

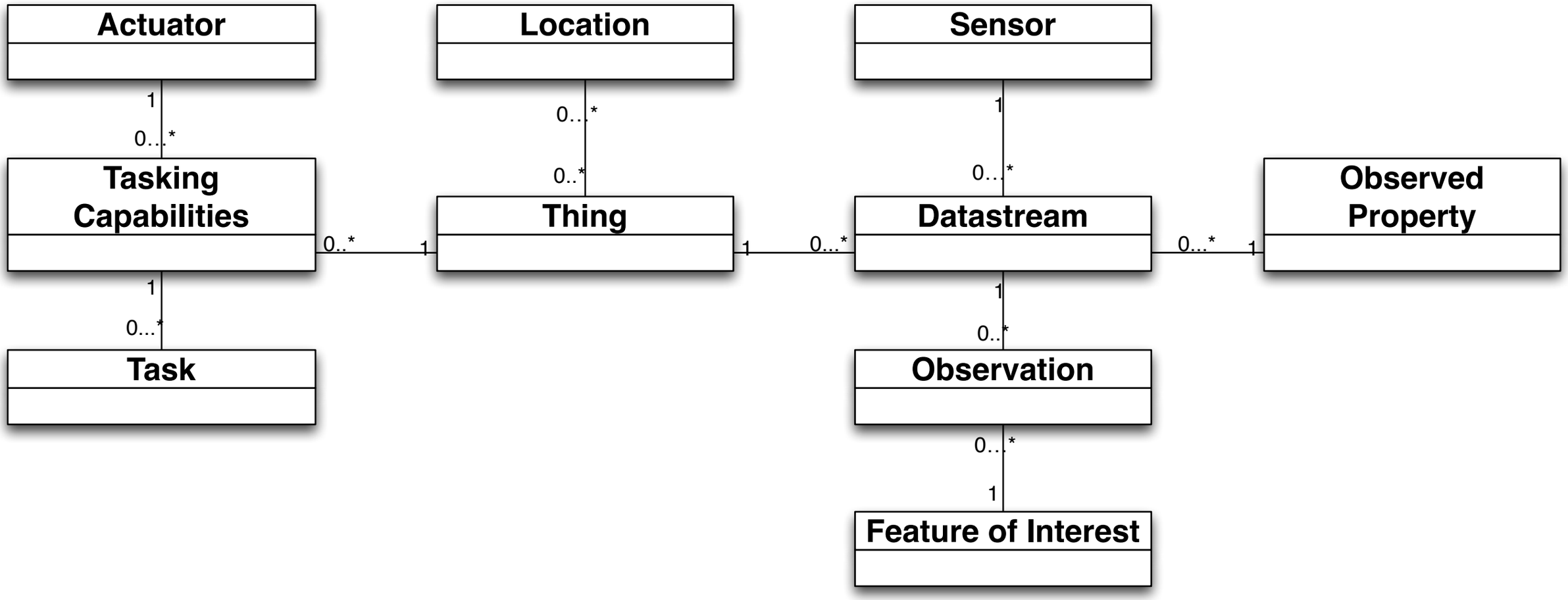
- Open standard (currently an OGC standard candidate)
- based on OGC/ISO O&M, SOS and SPS standards
- REST-like interfaces (inspired by OASIS OData)
- Easy-to-use and efficient JSON-encodings
- Comprehensive support for location information
- Support publish and subscribe (based on MQTT)
- Support Linked Data (based on JSON-LD)

SensorThings API Data Model

Similar Platforms

- Xively
- SensorCloud
- DeviceCloud
- ThingSpeak
- Onion IoT
- ThingWorx
- Axeda Machine Cloud
- ARM Sensinode
- Evrythng
- Ayla Networks
- TempoDB

SensorThings Data Model (Entities)



What is a Thing?

- A thing is an object of the physical world (physical things) or the information world (virtual things) that is capable of being identified and integrated into communication networks. (ITU-T Y.2060)

What is a Thing?

- Depending on your view point
- Take wearable devices as an example
  - from a health care professional's perspective
  - from a wearable bracelet maker's perspective

- Heart rate
- Respiration
- Activity

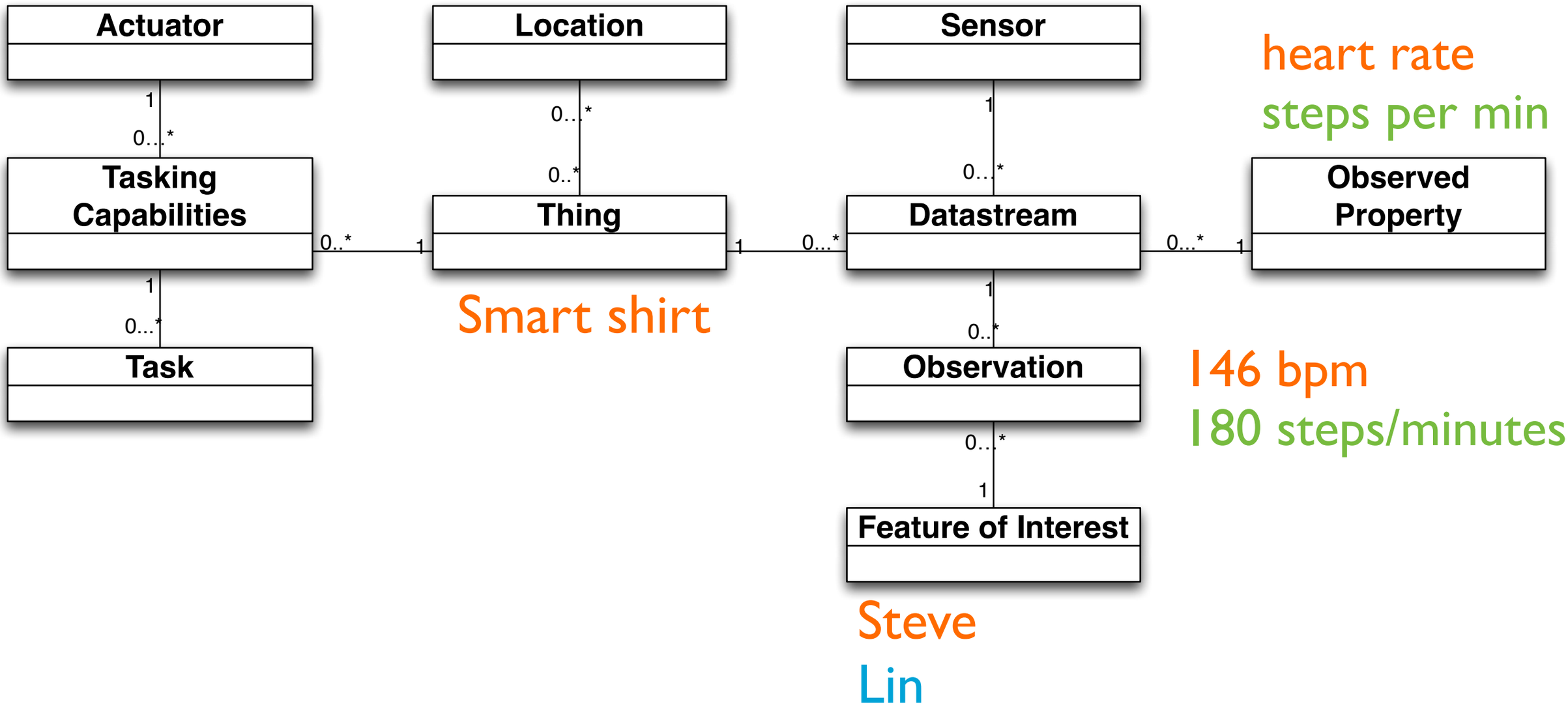
in gym

heart rate sensor

activity sensor

heart rate

steps per min



Smart shirt

146 bpm

180 steps/minutes

Steve

Lin

- CO2 Concentration
- Room Temperature
- Weight

Sensor focus view

master bathroom

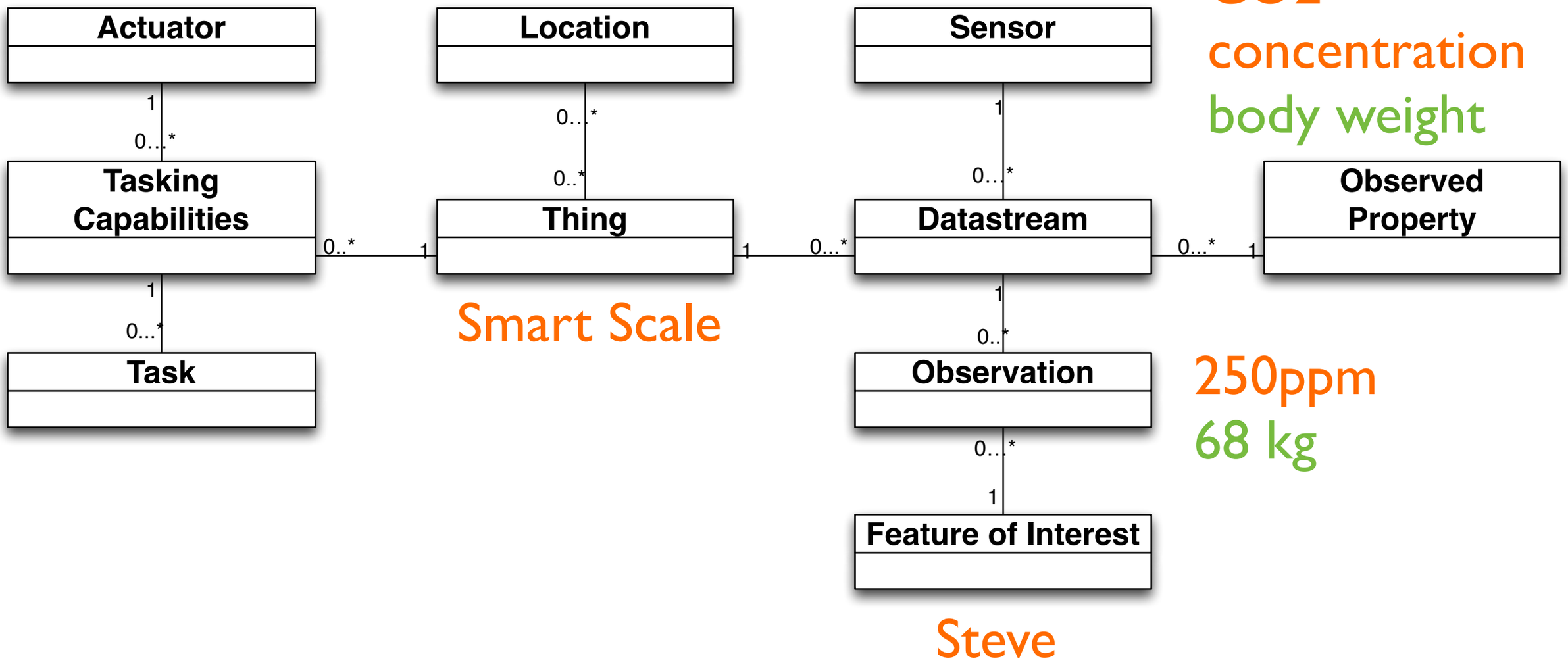
CO2 sensor

scale

CO2

concentration

body weight



Let's say you want to build a one-stop-shop for personal health data

- Weight

- Heart rate
- Respiration
- Activity

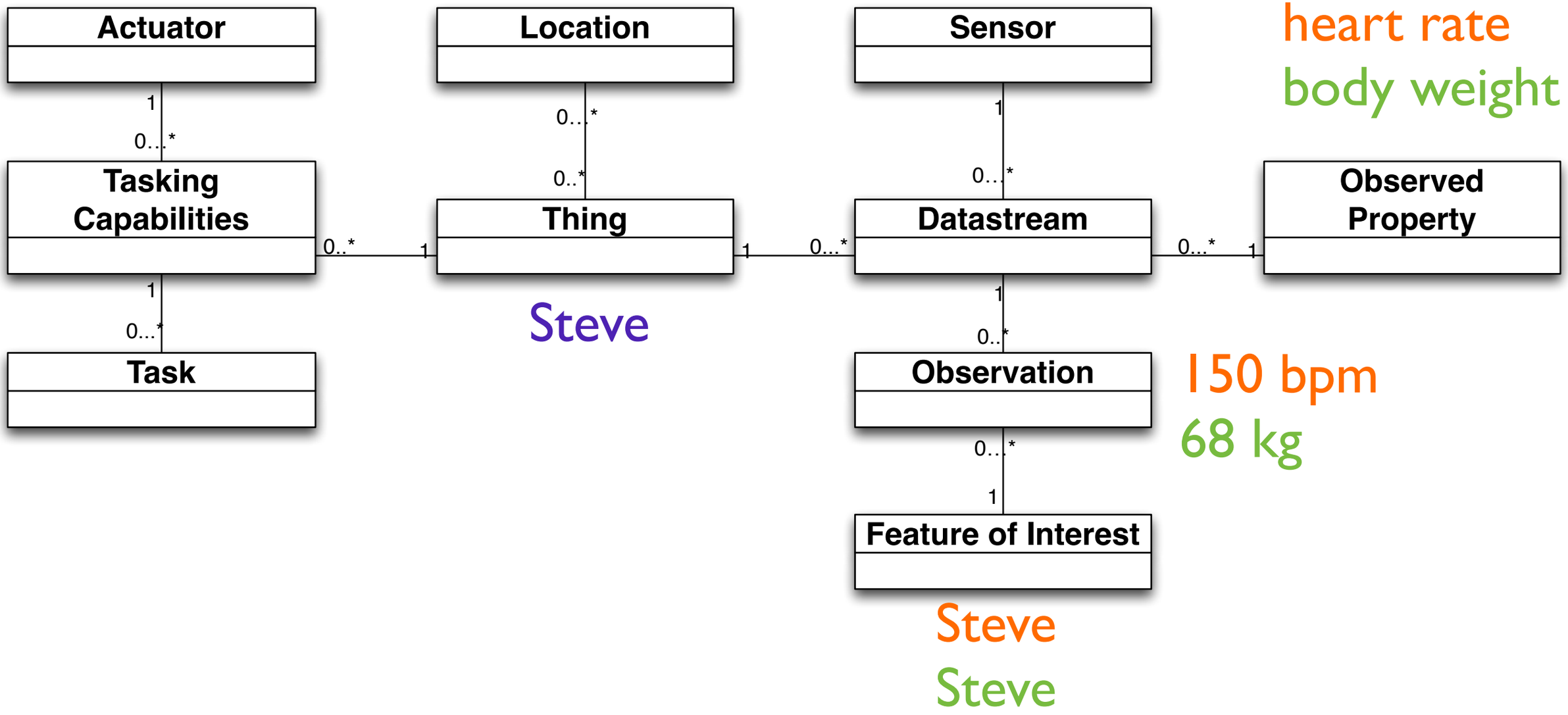
heart rate sensor

scale

gym

heart rate

body weight



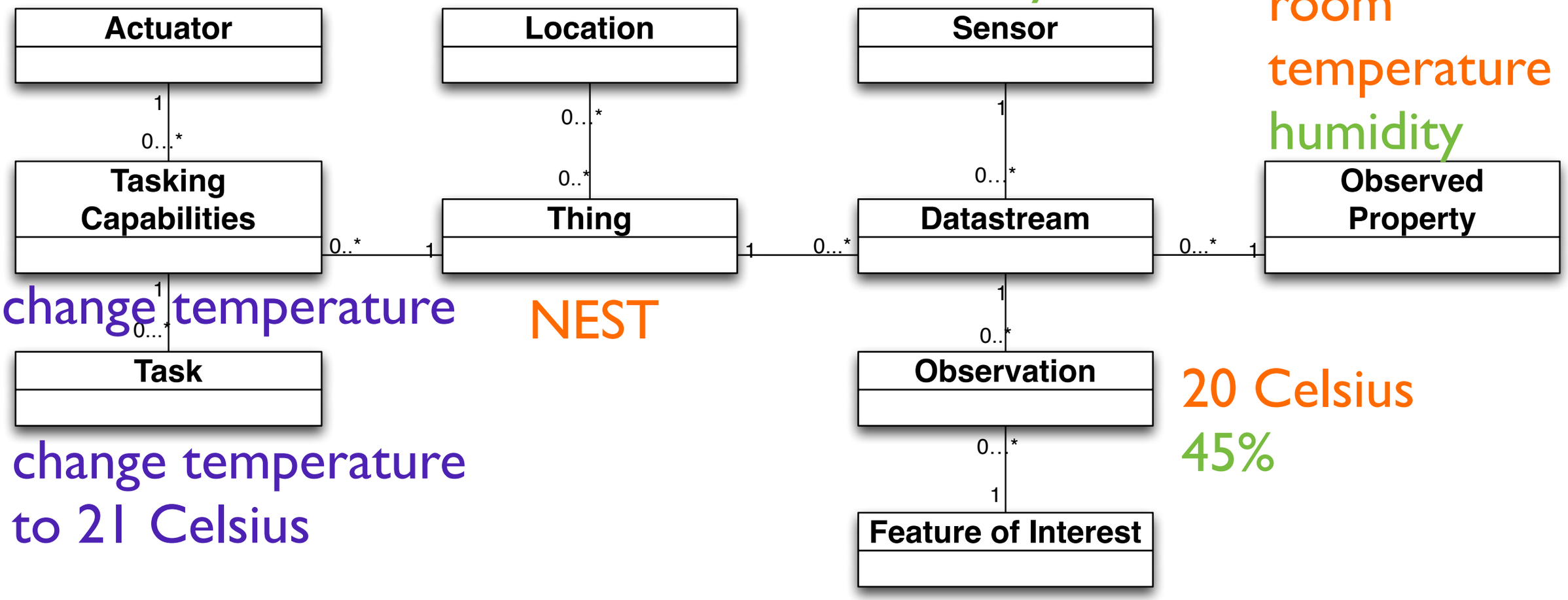
- Sensing:
  - Room Temperature
  - Humidity
- Tasking:
  - Control Temperature

furnace

Steve's house

thermometer
humidity sensor

room
temperature
humidity



change temperature

NEST

20 Celsius
45%

change temperature
to 21 Celsius

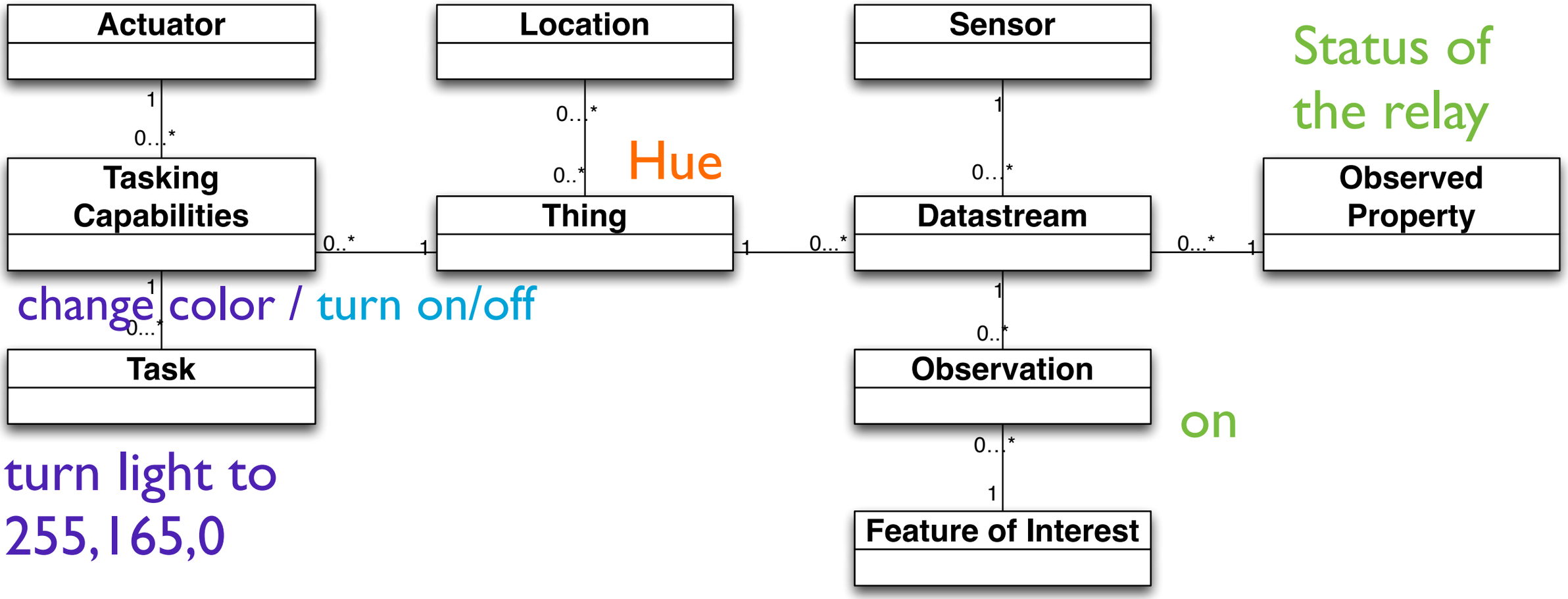
Steve's house

- Sensing:
  - Status (on/off)
- Tasking:
  - Turn on/off
  - Change color (R,G,B)

Hue light bulb

Steve's living room

Relay



Hue

Status of the relay

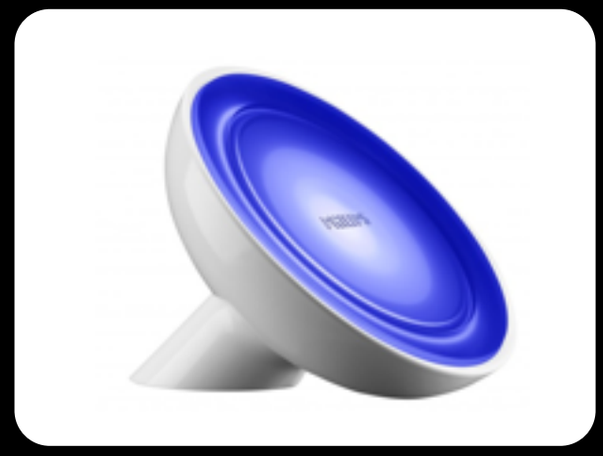
change color / turn on/off

turn light to 255,165,0

on

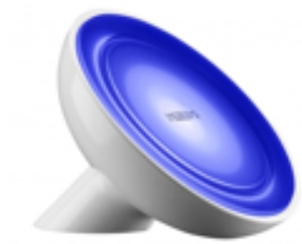
Steve's living room

Let's say you want to build a smart building management system



- Sensing:
  - Room Temperature
  - Humidity
- Tasking:
  - Control Temperature

- Sensing:
  - Status (on/off)
- Tasking:
  - Turn on/off
  - Change color (R,G,B)

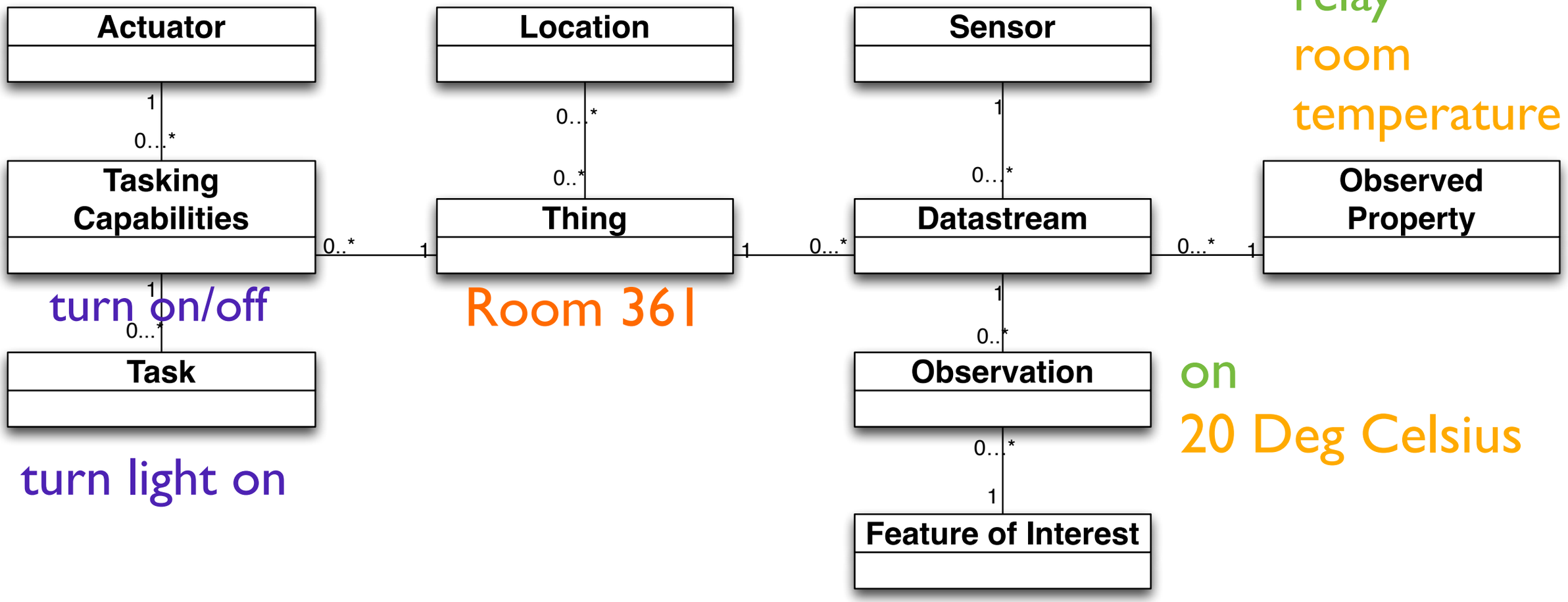


Hue light bulb

location of Room 361

Relay
Thermometer

Status of the
relay
room
temperature



turn on/off

Room 361

on
20 Deg Celsius

turn light on

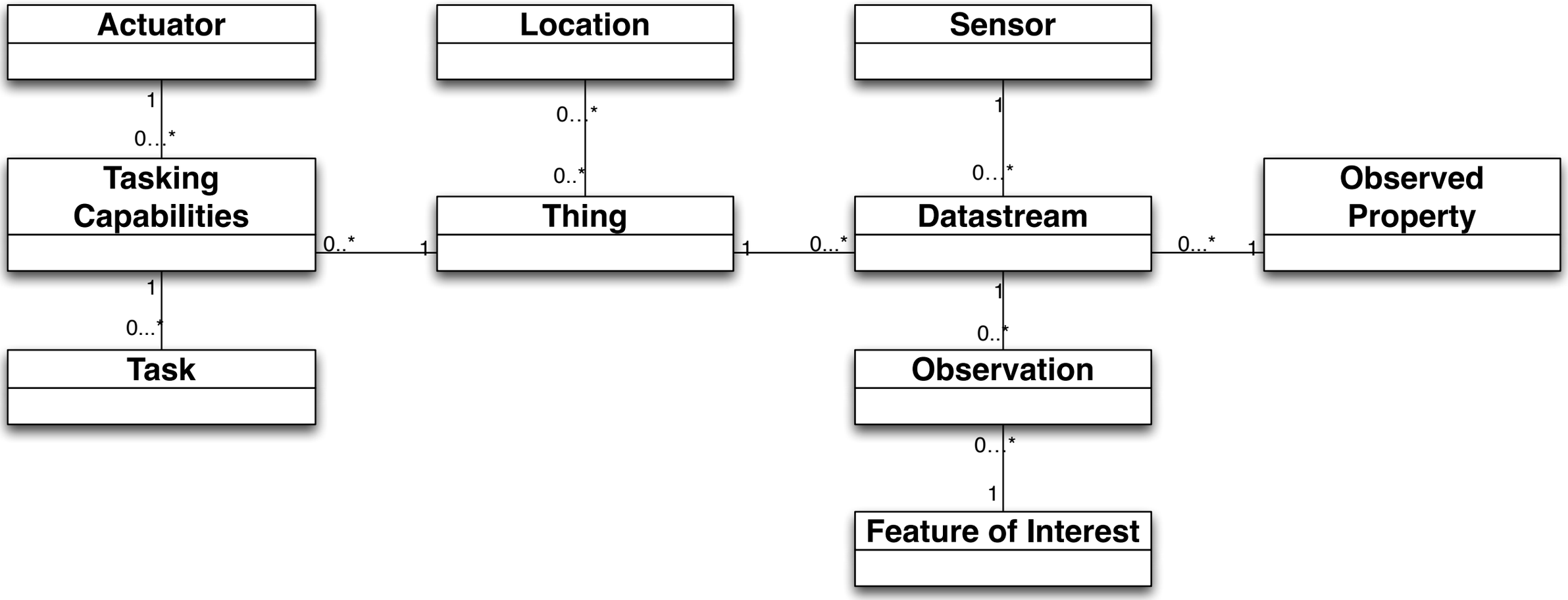
Room 361

- What is your THING depending on your use case.
- SensorThings API supports multiple views.

SensorThings is REST-like

- Each resource is represented by an unique URI
- CRUD each resource with HTTP verbs
  - CREATE - HTTP POST
  - READ - HTTP GET
  - UPDATE - HTTP PUT
  - DELETE - HTTP DELETE

SensorThings Data Model (Entities)



Entity Common Properties

Name	Definition	Data type	Multiplicity and use
ID	ID is the system-generated identifier of an entity. ID is unique among the entities of the same entity type in a SensorThings service.	Any	One (mandatory)
Self-Link	Self-Link is the absolute URL of an entity that is unique among all other entities.	URL	One (mandatory)
Association-Link	Association-Link is the relative URL showing the related entities in other entity types. Only the Self-Links of related entities are returned when resolving Association-Links.	Relative URL	One-to-many
Navigation-Link	Navigation-Link is the relative URL that retrieves content of related entities.	Relative URL	One-to-many

Thing

Table 6-2. Properties of a Thing entity

Name	Definition	Data type	Multiplicity and use
description	This is a short description of the corresponding Thing entity.	Character String	One (mandatory)
properties	A JSON Object containing user-annotated properties as key-value pairs.	JSON Object	Zero-to-one
trajectory	The trajectory of the Thing. This property contains the previous and current locations of the Thing with their time.	JSON Array	Zero-to-one

Tasking Profile

Sensing Profile

Location

id: Any
encodingType: ValueCode
location:{ }

Sensor

id: Any
encodingyype: ValueCode

metadata: { }

Actuator

id: Any
metadata: { }

Datastream

id:Any
description: text
observationType: ValueCode
resultType: ValueCode (or responseFormat?)
unitOfMeasurement: JSON (include description and URI definition)

observedArea:GM_Envelope[0…1]
phenomenonTime: TMPeriod[0…1]
resultTime: TM_Period[0…1]

(featureOfInterestType: ValueCode[]. We decide not to have it, because FoI can be different types in SensorThings)

Observed Property

id:Any
name: text
definition: iri
description: text

Tasking Capabilities

id: Any
acceptableParameters:
acceptableParameter[*]
deviceProtocol: HTTP or CoAP Protocol

Thing

id: Any
description: text
properties:{ }

trajectory: { }

Task

id: Any
input: Input[*]
time: TM_Object
status: text

Observation

id:Any
phenomenonTime: TM_Object
result: Any

resultQuality*: DQ_Element[0…*]
resultTime*: TM_Instant
validTime*: TM_Period[0…1]
parameters: NamedValue[0…*]

Feature of Interest

id:Any
description: text
encodingType: ValueCode
feature: { }

*Note: In the case of in-situ sensing, FoI and Location is the same.

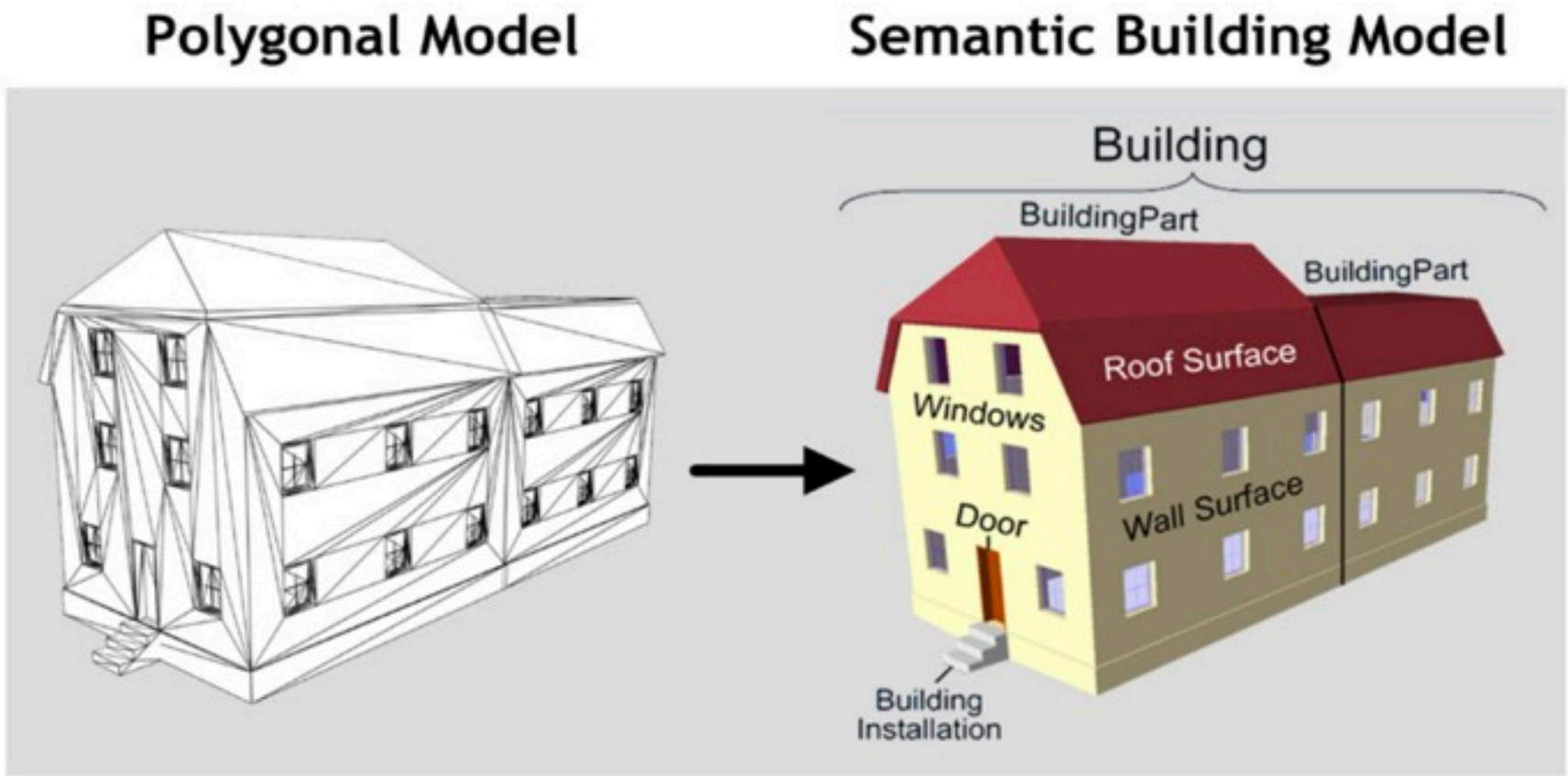
Location

Table 6-4. Properties of a Location entity

Name	Definition	Data type	Multiplicity and use
encodingType	The encoding type of the location property. Its value is one of the ValueCode enumeration (see Chapter x.x for the available ValueCode).	ValueCode	One (mandatory)
location	The geographical position of the Location. The location type is defined by EncodingType.	Any (depending on the value of the EncodingType)	One (mandatory)

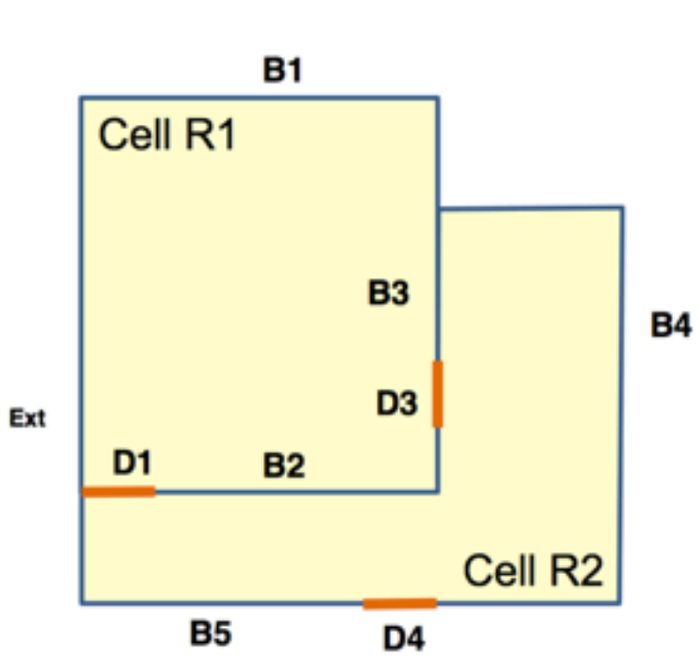
OGC CityGML

– Semantic 3D city and building models provide
  - the geometry and
  - a thematic differentiation of the indoor areas (at least separation in building parts, storeys and rooms)

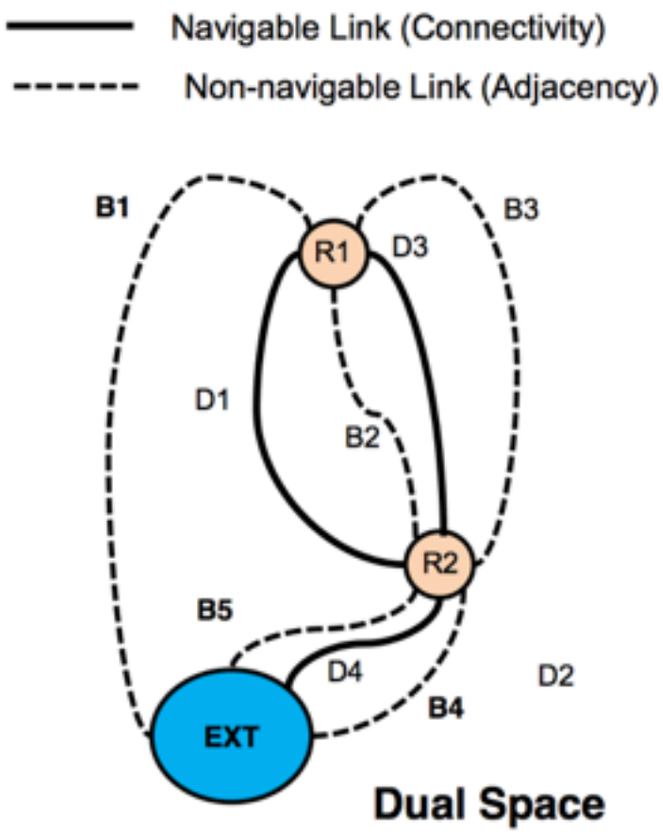


OGC Indoor GML

- Goal of IndoorGML
  - Common schema framework for Interoperability between indoor navigation applications
  - Enable many indoor applications, e.g., emergency control, visually handicapped guidance, indoor robots, etc..



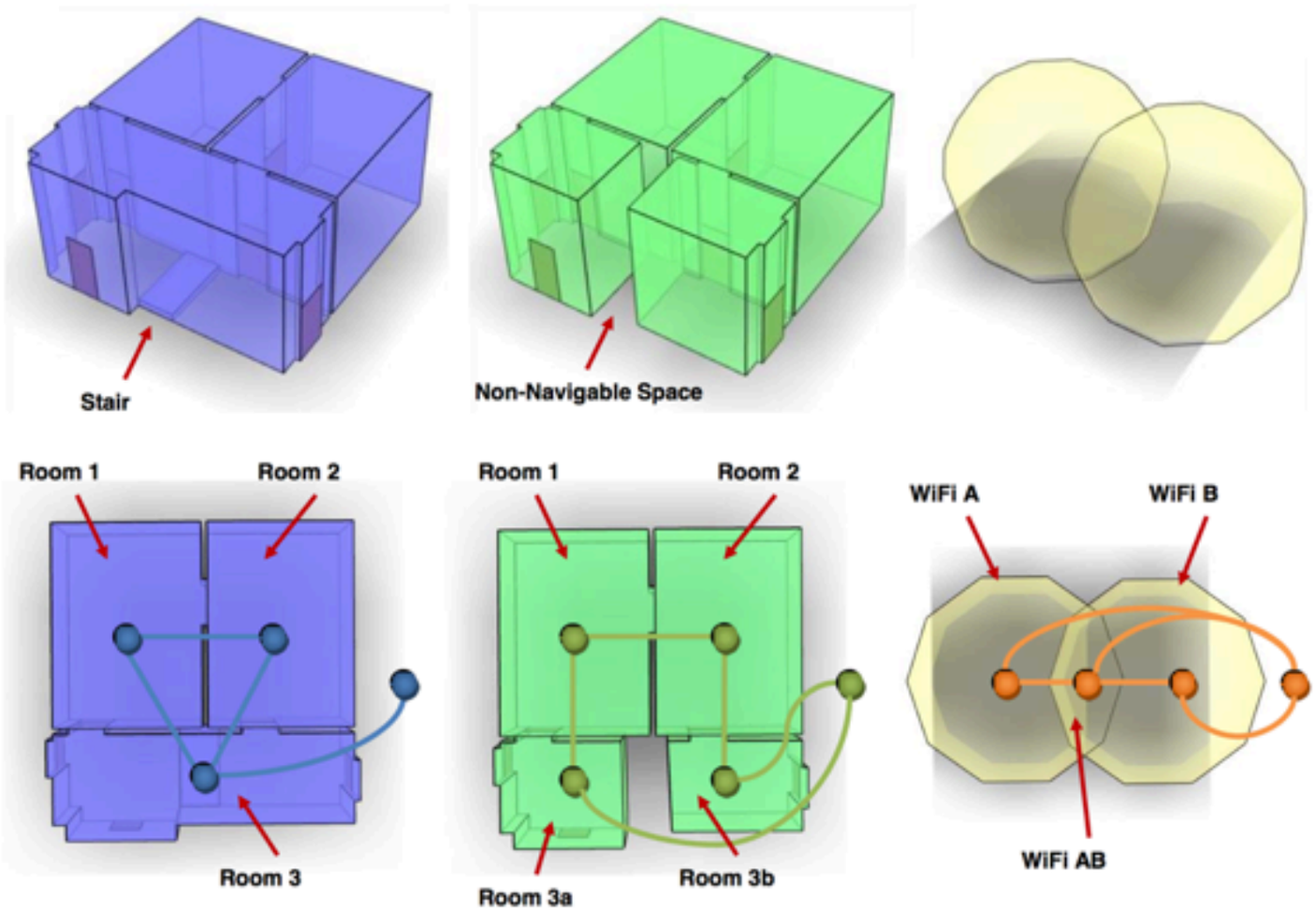
Topographic Space



Dual Space

Indoor GML: Topology and Multi-Layered Space Model

- Two components
  - Topology between symbolic space units
  - Multi-Layered Space Model



last known Location of the Thing


```
{
   "id":1,
   "Self-Link":"http://demoURL:8080/SensorThings_V1.0/Locations(1)",
   "Things":{
      "Association-Link":"Locations(1)/$links/Things",
      "Navigation-Link":"Locations(1)/Things"
   },
   "encodingType":"http://example.com/location_types#GeoJSON",
   "location":{
      "type":"Point",
      "coordinates":[
         100.0,
         50.0
      ]
   }
}
```

Multiple representation of the last known Location of a Thing

```
{
    "id":2,
    "Self-Link":"http://demoURL:8080/SensorThings_V1.0/Locations(2)",
    "Things":{
       "Association-Link":"Locations(1)/$links/Things",
       "Navigation-Link":"Locations(1)/Things"
    },
    "encodingType":"http://example.com/location_types#cityGML",
    "location":{
        some CityGML
    }
}
```

still under development

A **Datastream** groups a collection of Observations and the Observations in a Datastream must measure the same ObservedProperty (*i.e.*, one phenomenon) and are produced by the same Sensor.

Table 6-6. Properties of a Datastream entity

Name	Definition	Data type	Multiplicity and use
description	This is the description of the Datastream entity.	Character String	One (mandatory)
unitOfMeasurement	A JSON Object containing three key-value pairs. The "Name" property presents the full name of the UnitOfMeasurement; the "Symbol" property shows the textual form of the unit symbol; and the "Definition" contains the IRI defining the UnitOfMeasurement.	JSON Object	One (mandatory)
observationType	The type of observation (with unique result type), which is used by the service to encode observations.	ValueCode see OGC 10-025r1 Table 6	One (mandatory)
observedArea	The spatial bounding box of the spatial extent of all FeaturesOfInterest that belong to Observations associated with this Datastream.	GeoJSON Polygon (GM_Envelope see ISO-19107)	Zero-to-one
phenomenonTime	The temporal bounding box of the phenomenon times of all observations belonging to this Datastream.	TM_Period see ISO-19108	Zero-to-one
resultTime	The temporal bounding box of the result times of all observations belonging to this Datastream.	TM_Period see ISO-19108	Zero-to-one

observationType

- http://www.opengis.net/def/observationType/OGC- OM/2.0/OM_CategoryObservation
  - type='gml:ReferenceType'
- http://www.opengis.net/def/observationType/OGC- OM/2.0/OM_CountObservation
  - type='xs:integer'
- http://www.opengis.net/def/observationType/OGC- OM/2.0/OM_Measurement
  - type='gml:MeasureType'
- http://www.opengis.net/def/observationType/OGC- OM/2.0/OM_TruthObservation
  - type='xs:boolean

OGC 10-025

Table 5 — Map of UML classes in O&M v2.0 to XML elements in OMXML

O&M v2.0	OMXML	O&M v2.0	OMXML
OM_CategoryObservation	om:OM_Observation	ObservationContext	om:ObservationContext
OM_ComplexObservation	om:OM_Observation	PreparationStep	spec:PreparationStep
OM_CountObservation	om:OM_Observation	SamplingFeatureComplex	sam:SamplingFeatureComplex
OM_DiscreteCoverageObservation	om:OM_Observation[1]	SF_SamplingCurve	sams:SF_SpatialSamplingFeature
OM_GeometryObservation	om:OM_Observation	SF_SamplingFeature	sam:SF_SamplingFeature
OM_Measurement	om:OM_Observation	SF_SamplingFeatureCollection	sam:SF_SamplingFeatureCollection
OM_Observation	om:OM_Observation	SF_SamplingPoint	sams:SF_SpatialSamplingFeature
OM_PointCoverageObservation	om:OM_Observation[1]	SF_SamplingSolid	sams:SF_SpatialSamplingFeature
OM_TemporalObservation	om:OM_Observation	SF_SamplingSurface	sams:SF_SpatialSamplingFeature
OM_TimeSeriesObservation	om:OM_Observation[1]	SF_SpatialSamplingFeature	sams:SF_SpatialSamplingFeature
OM_TruthObservation	om:OM_Observation	SF_Specimen	spec:SF_Specimen

4.13 [1] Coverage observations are implemented by the generic om:OM_Observation but there are no specific conformance tests.

OGC 10-025

Table 6 — Map of UML classes in O&M v2.0 to OGC observation-type names and observation result-types

O&M v2.0	OGC Name	Content of result in OMXML[1]
OM_CategoryObservation	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_CategoryObservation	type='gml:ReferenceType'
OM_ComplexObservation	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_ComplexObservation	swe:DataRecord
OM_CountObservation	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_CountObservation	type='xs:integer'
OM_DiscreteCoverageObservation	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_DiscreteCoverageObservation	-[2] or a reference using xlink attributes
OM_GeometryObservation	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_GeometryObservation	gml:AbstractGeometry[3]
OM_Measurement	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement	type='gml:MeasureType'
OM_Observation	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Observation	type='xs:anyType'[4]
OM_PointCoverageObservation	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_PointCoverageObservation	-[2] or a reference using xlink attributes
OM_TemporalObservation	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_TemporalObservation	gml:AbstractTimeObject[3]
OM_TimeSeriesObservation	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_TimeSeriesObservation	-[2] or a reference using xlink attributes
OM_TruthObservation	http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_TruthObservation	type='xs:boolean

[1] An XML element name indicates that the result in this case has element content; type=' ' indicates that the result has 'simple' (non-element) content.

[2] gml:AbstractDiscreteCoverage, cvi:CV_AbstractDiscreteCoverage, gml:MultiPointCoverage, cvi:CV_DiscretePointCoverage are available for encoding coverages in XML, but the preferred OGC encoding for coverages is under revision.

[3] A concrete member of the substitution group shall substitute for an abstract element.

[4] Any well-formed XML can substitute for xs:anyType.

Sensor

Table 6-8. Properties of a Sensor entity

Name	Definition	Data type	Multiplicity and use
encodingType	The encoding type of the location property. Its value is one of the ValueCode enumeration (see Chapter x.x for the available ValueCode).	ValueCode	One (mandatory)
metadata	The detailed description of the Sensor or system. The metadata type is defined by EncodingType.	Any (depending on the value of the encodingType)	One (mandatory)

ObservedProperty

Table 6-10. Properties of an ObservedProperty entity

Name	Definition	Data type	Multiplicity and use
name	The name of the ObservedProperty.	Character String	One (mandatory)
definition	The IRI of the ObservedProperty. Dereferencing this IRI should result in a representation of the definition of the ObservedProperty.	IRI	One (mandatory)
description	A description about the ObservedProperty.	Character String	Zero-to-one

```
{
   "ID":1,
   "Self-Link":"http://.../SensorThings_V1.0/ObservedProperties(1)",
   "Datastreams":{
      "Association-Link":"ObservedProperties(1)/$links/Datastreams",
      "Navigation-Link":"ObservedProperties(1)/Datastreams"
   },
   “Name”:”DewPoint Temperature”,
   "Definition":”http://examples.com#DewPointTemperature”   ,
   “Description”:”The dewpoint temperature is the temperature to which the air must be cooled, at constant pressure, for dew to form. As the grass and other objects near the ground cool to the dewpoint, some of the water vapor in the atmosphere condenses into liquid water on the objects.”
}
```

Observation

Table 6-12. Properties of an Observation entity

Name	Definition	Data type	Multiplicity and use
phenomenonTime	The time point/period of when the Observation happens.	TM_Object	One (mandatory)
result	The estimated value of an ObservedProperty from the Observation.	Any (depends on the ObservationType)	One (mandatory)
resultTime	The time of the observation's result was generated.	TM_Instant	Zero-to-one
resultQuality	Describes the quality of the result.	DQ_Element	Zero-to-many
validTime	The time period during which the result may be used.	TM_Period	Zero-to-one
parameters	Key-value pairs showing the environmental conditions during measurement.	NamedValues in a JSON Array	Zero-to-one

```
{
   "id":1,
   "Self-Link":"http://.../SensorThings_V1.0/Observations(1)",
   "featureOfInterest":{
      "Association-Link":"Observations(1)/$links/FeatureOfInterest",
      "Navigation-Link":"Observations(1)/FeatureOfInterest"
   },
   "datastream":{
      "Association-Link":"Observations(1)/$links/Datastream",
      "Navigation-Link":"Observations(1)/Datastream"
   },
   "phenomenonTime":"2012-06-26T03:42:02-0600",
   "result": "70.4"
}
```

Feature of Interest

An observation results in a value being assigned to a phenomenon. The phenomenon is a property of a feature, the latter being the feature-of-interest of the observation. In the case of an in-situ sensor or Observations being attributes of the Thing, the FeatureOfInterest may be the Thing itself. For remote sensors, the FeatureOfInterest may be the geographical area or volume that is being sensed

Feature of Interest

Table 6-14. Properties of a FeatureOfInterest entity

Name	Definition	Data type	Multiplicity and use
description	A description about the FeatureOfInterest.	Character String	One (mandatory)
encodingType	The encoding type of the content property. It can be GeoJSON or other data type.	ValueCode	One (mandatory)
feature	The geographical position of the FeatureOfInterest. The content is open to accommodate GeoJSON or other data types.	JSON Object	One (mandatory)

Tasking Capabilities

Table 6-16. Properties of a TaskingCapability entity

Name	Definition	Data type	Multiplicity and use
description	This is a short description of the TaskingCapability.	Character String	One (mandatory)
acceptableParameters	A TaskingCapability may have zero-to-many AcceptableParameters. Users can understand the acceptable input of the TaskingCapability.	AcceptableParameters in a JSON Array see Table 6-17	Zero-to-one
deviceProtocol	The DeviceProtocol is not visible to users but needs to be registered into the SensorThings service when creating the TaskingCapability. The DeviceProtocol describes the communication protocol of sending tasks to the TaskingCapability.	HTTPProtocol see Table 6-18	One (mandatory)

SensorThings API URL Pattern

Usage 1: No resource path

- If a user request does not include any resource path, the SensorThings service returns a list of collection names as the service document.

URI Pattern: SERVICE_ROOT_URI

Response: A list of collection names.

Example: http://example.com/SensorThings_V1.0/

Usage 2: Address to an entity collection

- To address to an entity collection, users can simply put the collection name after the service root URI. The service returns a list of all the entities in the specified collection.

URI Pattern: SERVICE_ROOT_URI/COLLECTION_NAME

Response: A list of all entities (with all the properties) in the specified collection (if no service-driven pagination imposed).

Example: http://examples.com/SensorThings_V1.0/**Things**

Usage 3: Address to an entity in a collection

Users can address to a specific entity in a collection by place the unique identifier of the entity between brace symbol "()" and put after the collection name. The service then returns the entity with all its properties.

URI Pattern: SERVICE_ROOT_URI/COLLECTION_NAME(KEY)

Response: An entity (with all its properties) that holds the specified key in the collection.

Example: http://examples.com/SensorThings_V1.0/**Things(1)**

Usage 4: Address to a property of an entity

- Users can address to a property of an entity by specifying the property name after the URI addressing to the entity. The service then returns the value of the specified property.

URI Pattern: SERVICE_ROOT_URI/COLLECTION_NAME(KEY)/PROPERTY_NAME

Response: The specified property of an entity that holds the key in the collection.

Example: http://examples.com/SensorThings_V1.0/Observations(1)/Time

Usage 5: Address to navigation property (link)

- As the entities in different collections may hold some relationships, users can request the linked entities by addressing to a navigation property of an entity. The service then returns one or many entities that hold a certain relationship with the specified entity.

URI Pattern: SERVICE_ROOT_URI/COLLECTION_NAME(KEY_OF_ENTITY)/LINK_NAME

Response: One or many entities that holds a certain relationship with the specified entity.

Example: http://demo.student.geocens.ca:8080/SensorThings_V1.0/Datastreams(1)/Observations returns all the observations in the datastream.

Usage 6: Address to an association link

- As the entities in different collections may hold some relationships, users can request the linked entities' Self-Links by addressing to an association link of an entity. Association-Link is the relative URL showing the related entities in other entity types. Only the Self-Links of related entities are returned when resolving Association links.

URI Pattern: SERVICE_ROOT_URI/COLLECTION_NAME(KEY_OF_ENTITY)/$links/LINK_NAME

Response: One or many relative URLs of the entities that holds a certain relationship with the specified entity.

Example: http://examples.com/SensorThings_V1.0/Datastreams(1)/$links/Observations returns all the Self-Links of the observations in the datastream.

Usage 7: Nested resource path

- As users can use navigation properties to link from one collection to another, users can further extend the resource path with unique identifiers, properties, or links (i.e., Usage 3 to 5).

Example 1: http://examples.com/SensorThings_V1.0/Datastreams(1)/Observations(1) returns a specific observation in the datastream.

Example 2: http://examples.com/SensorThings_V1.0/Datastreams(1)/Observations(1)/Time returns the "Time" property of the specified observation in the datastream.

Example 3: http://examples.com/SensorThings_V1.0/Datastreams(1)/Observations(1)/FeatureOfInterest returns the feature of interest entity of the specified observation in the datastream.

$orderby

- **Example 1:** http://examples.com/SensorThings_V1.0/Observations?$orderby=resultValue returns all Observations ordered by the ResultValue property in ascending order.

- **Example 2:** http://examples.com/SensorThings_V1.0/Observations?$orderby=Datastreams/id desc,time returns all Observations ordered by the ID property of the linked Datastream entry in descending order, then by the Time property of Observations in ascending order.

$top

Example 1: http://examples.com/SensorThings_V1.0/Things?$top=5 returns only the first five entities in the Things collection.

Example 2: http://examples.com/SensorThings_V1.0/Observations?$top=5&$orderby=time desc returns the first five Observation entries after sorted by the Time property in descending order.

Note: if the $top value exceeds the service-driven pagination limitation (*i.e.*, the largest number of entities the service can return in a single response), the $top query option should be discarded and the pagination limitation should be imposed.

$skip

The $skip query option specifies that the result must not include the first n entities, where n is a non-negative integer value specified by the $skip query option.

Example 1: http://examples.com/SensorThings_V1.0/Things?$skip=5 returns Thing entities starting with the sixth Thing entity in the Things collection.

Example 2: http://examples.com/SensorThings_V1.0/Observations?$skip=2&$top=2&$orderby=time returns the third and fourth Observation entities from the collection of all Observation entities when the collection is sorted by the Time property in ascending order.

$filter

A URI with a $filter query option identifies a subset of the entries from the collection of entries identified by the resource path of the URI. The subset is determined by selecting only the entries that satisfy the predicate expression specified by the $filter. The value of the $filter option is a Boolean expression.

The expression language that is used in $filter operators supports references to properties and literals. The literal values can be strings enclosed in single quotes, numbers and boolean values (true or false) or datetime values represented as datetime'ISO 8601 time string'.

- Example 1: http://examples.com/SensorThings_V1.0/Observations?$filter=resultValue lt 10.00 returns all Observations whose ResultValue is less than 10.00.

  In addition, users may use the properties of linked entities in the $filter predicate. The following are examples of the possible uses of the $filter in the data model of the SensorThings service.

- Example 2: http://examples.com/SensorThings_V1.0/Observations?$filter=Datastream/ObservedProperty/unitOfMeasurement eq 'Celsius' returns all Observations that UnitOfMeasurement (a property of ObservedProperty) is 'Celsius'.

- Example 3: http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things?$filter=Locations/Geometry st_within(POLYGON ((30 10, 10 20, 20 40, 40 40, 30 10))) returns Things whose current or past locations were in the polygon defined in the Well-Known Text (WKT).

- Example 4: http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things?$filter=Datastreams/Observations/FeatureOfInterest/id eq 'FOI_1' and Datastreams/Observations/time ge datetime'2010-06-01T00:00:00Z' and Datastreams/Observations/time le datetime'2010-07-01T00:00:00Z' returns Things that have any observations on a feature of interest with a unique identifier equals to 'FOI_1' in June 2010.

Table 8-1. Built-in Filter Operators

Operator	Description	Example
Logical Operators		
eq	Equal	/ObservedProperties?$filter=UnitOfMeasurement eq 'Celsius'
ne	Not equal	/FeaturesOfInterest?$filter=Geometry/type ne 'Polygon'
gt	Greater than	/Observations?$filter=ResultValue gt 20.0
ge	Greater than or equal	/Observations?$filter=ResultValue ge 20.0
lt	Less than	/Observations?$filter=ResultValue lt 100
le	Less than or equal	/Observations?$filter=ResultValue le 100
and	Logical and	/Observations?$filter=ResultValue le 3.5 and FeatureOfInterest/ID eq 'FOI_1'
or	Logical or	/Observations?$filter=ResultValue gt 20 or ResultValue le 3.5
not	Logical negation	/Things?$filter=not startswith(Description,'test')
Arithmetic Operators		
add	Addition	/Observations?$filter=ResultValue add 5 gt 10
sub	Subtraction	/Observations?$filter=ResultValue sub 5 gt 10
mul	Multiplication	/Observations?$filter=ResultValue mul 2 gt 2000
div	Division	/Observations?$filter=ResultValue div 2 gt 4
mod	Modulo	/Observations?$filter=ResultValue mod 2 eq 0
Grouping Operators		
( )	Precedence grouping	/Observations?$filter=(ResultValue sub 5) gt 10

$expand

The $expand system query option indicates the related entities that must be represented inline. The value of the $expand query option must be a comma separated list of navigation property names. Additionally each navigation property can be followed by a forward slash and another navigation property to enable identifying a multi-level relationship.

Example 1: http://examples.com/SensorThings_V1.0/Things?$expand=Locations returns the collection of Things as well as each of the Datastreams associated with each Thing entity.

- **Example 2:** http://examples.com/SensorThings_V1.0/Things?$expand=Datastreams/ObservedProperty returns the collection of Things, the Datastreams associated with each Thing, and the ObservedProperty associated with each Datastream.

- **Example 3:** http://demo.student.geocens.ca:8080/SensorThings_V1.0/Datastreams(1)?$expand=Observations,ObservedProperty returns the Datastream whose ID is 1 as well as the Observations and ObservedProperty associated with this Datastream entity.

$select

A data service URI with a $select query option identifies the same set of entries as a URI without a $select query option; however, the value of $select specifies that a response from an SensorThings service should return a subset of the properties. The value of a $select query option is a comma-separated list of selection clauses. Each selection clause may be a property name (including navigation property names).

- **Example 1:** http://examples.com/SensorThings_V1.0/Observations?$select=resultValue,time returns only the resultValue and time properties for each Observation entity.
- **Example 2:** http://examples.com/SensorThings_V1.0/Datastreams(1)?$select=id,Observations&$expand=Observations/FeatureOfInterest returns the id properties of the Datastream entity, and all the properties of the entity identified by the Observations and FeatureOfInterest navigation properties.

Data Array (not stable)

- In SensorThings services, users are able to request for multiple Observation entities and format the entities in the **DataArray** format. When a SensorThings service returns a DataArray response, the service groups Observation entities by Datastream, which means the Observation entities that link to the same Datastream are aggregated in one DataArray.
- In order to request for DataArray, users must include the query option "$resultModel=Observation" when requesting observation entities. For example, http://example.com/Observations?**$resultModel=Observation**

```
GET /SensorThings_V1.0/Observations HTTP/1.1 200 OK
Host: example.com:8080
Content-Type: application/json

{
  "Observations":[
    {
      "Datastream":{
        "Navigation-Link":"Datastreams(1)"
      },
      "count":3,
      "Components":["id","phenomenonTime","result"],
      "DataArray":[
        [1,"2005-08-05T12:21:13Z",20.0],
        [2,"2005-08-05T12:22:08Z",30.0],
        [3,"2005-08-05T12:22:54Z",0.0]
      ]
    }
  ]
```

CreateObservations (not stable)

- Users can invoke the CreateObservations action by sending a HTTP POST request to the SERVICE_ROOT_URL/CreateObservations. For example, http://sensorthings.example.com/CreateObservations.
- The message body aggregates observations by Datastreams, which means all the Observations linked to one Datastream are aggregated in one JSON object. The parameters of each JSON object are shown in the following table.
- As an Observation links to one FeatureOfInterest, to establish the link between an Observation and a FeatureOfInterest, users should include the FeatureOfInterest IDs in the DataArray. If no FeatureOfInterest ID presented, the FeatureOfInterest will be created based on the Location entities of the linked Thing entity by default.

```
POST /SensorThings_V1.0/CreateObservations HTTP/1.1
Host: demo.student.geocens.ca:8080
Content-Type: application/json

[
  {
    "Datastream":{"id":1},
    "count":3,
    "Components":["phenomenonTime","result","FeatureOfInterest/id"],
    "DataArray":[
      ["2005-08-05T12:21:13Z",20.0,1],
      ["2005-08-05T12:22:08Z",30.0,1],
      ["2005-08-05T12:22:54Z",0.0,2]
    ]
  }
]
```

POST /SensorThings_V1.0/CreateObservations HTTP/1.1 201 Created
Host: example.com:8080
Content-Type: application/json

```
[
  [
   "http://example.com/SensorThings_V1.0/Observations(1)",
   "error",
   "http://example.com/SensorThings_V1.0/Observations(2)"
  ]
]
```

SensorThings Resources

- GitHub
- Open Source libraries
- Videos
  - https://www.youtube.com/watch?v=AdFKRF3V8SI

SensorThings and MQTT

- Transport Protocol
- Topic
  - relative path
  - Things(1)/Datastreams(1)/Observations

SensorThings API and JSON-LD

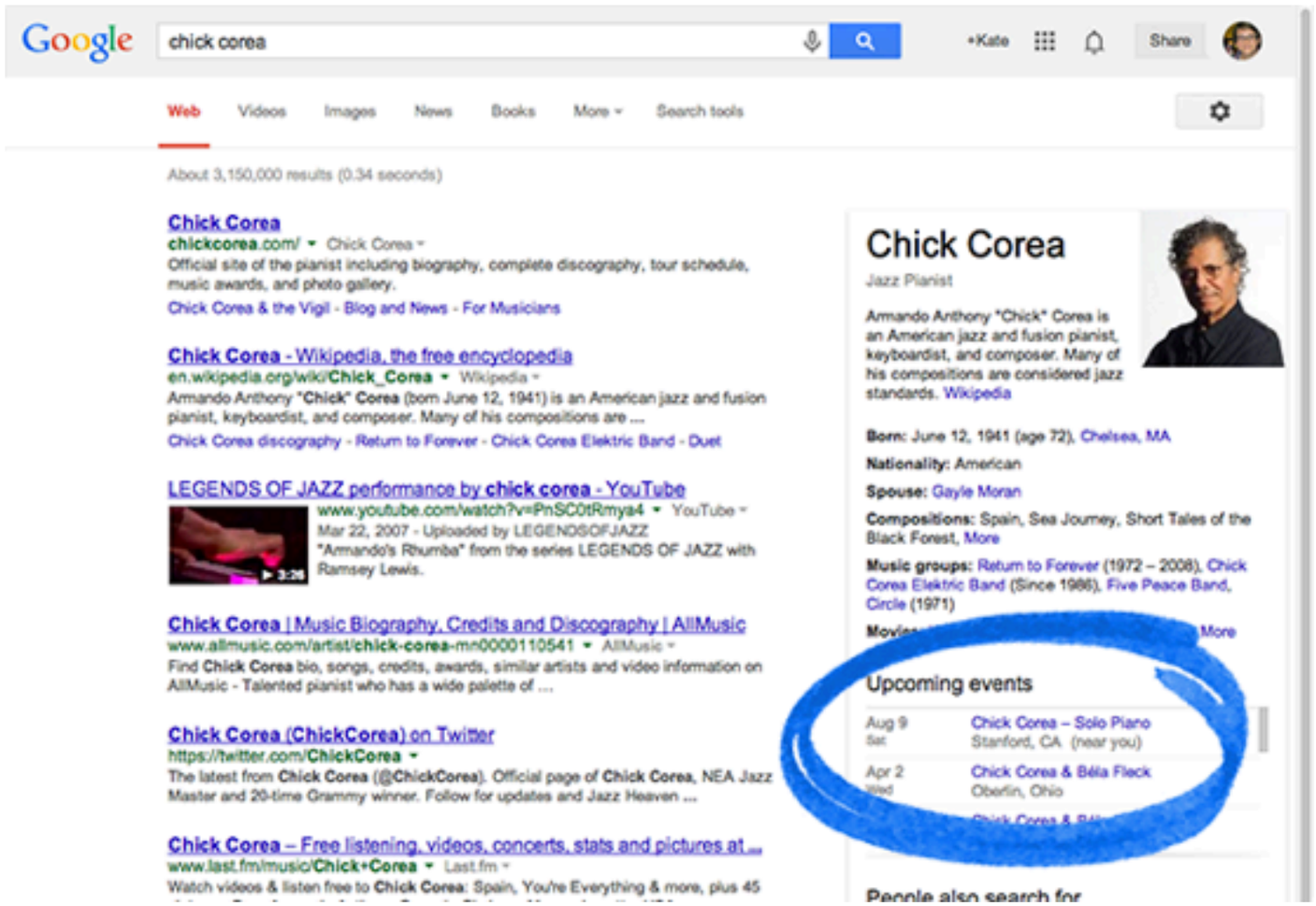
Objectives

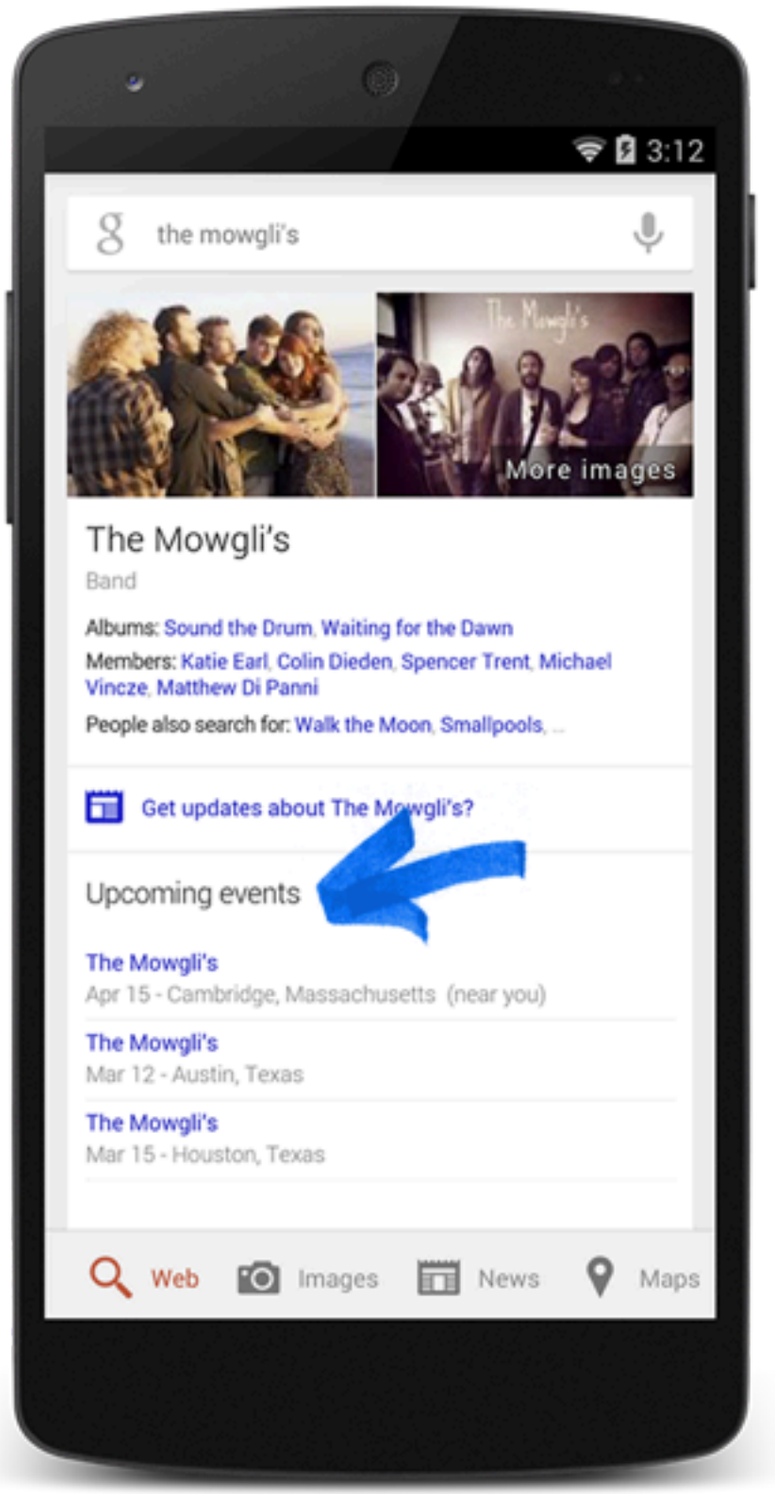
- Supporting JSON-LD makes SensorThings API Linked Data.
  - Linked Data is machines and human readable
  - Self-describing

- JSON-LD is an optional for SensorThings API

JSON-LD Background

- JSON-LD is a lightweight Linked Data format.
- JSON-LD is suitable for the web programming environments.
- The keys (terms) in the JSON-LD can be parse as URIs without ambiguity.
- The different terms can be mapped by their URIs.
- JSON-LD is widely used by major Web companies
  - –Google, Groupon, etc.


Google
chick corea
+Kate
Share
Web
Videos
Images
News
Books
More
Search tools
About 3,150,000 results (0.34 seconds)
Chick Corea
chickcorea.com/ - Chick Corea -
Official site of the pianist including biography, complete discography, tour schedule, music awards, and photo gallery.
Chick Corea & the Vigil - Blog and News - For Musicians
Chick Corea - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Chick_Corea - Wikipedia -
Armando Anthony "Chick" Corea (born June 12, 1941) is an American jazz and fusion pianist, keyboardist, and composer. Many of his compositions are ....
Chick Corea discography - Return to Forever - Chick Corea Elektric Band - Duet
LEGENDS OF JAZZ performance by chick corea - YouTube
www.youtube.com/watch?v=PnSC0tRmya4 - YouTube -
Mar 22, 2007 - Uploaded by LEGENDSOFJAZZ
"Armando's Rhumba" from the series LEGENDS OF JAZZ with Ramsey Lewis.
Chick Corea | Music Biography, Credits and Discography | AllMusic
www.allmusic.com/artist/chick-corea-mn0000110541 - AllMusic -
Find Chick Corea bio, songs, credits, awards, similar artists and video information on AllMusic - Talented pianist who has a wide palette of ....
Chick Corea (ChickCorea) on Twitter
https://twitter.com/ChickCorea -
The latest from Chick Corea (@ChickCorea). Official page of Chick Corea, NEA Jazz Master and 20-time Grammy winner. Follow for updates and Jazz Heaven ...
Chick Corea – Free listening, videos, concerts, stats and pictures at ...
www.last.fm/music/Chick+Corea - Last.fm -
Watch videos & listen free to Chick Corea: Spain, You're Everything & more, plus 45
Chick Corea
Jazz Pianist
Armando Anthony "Chick" Corea is an American jazz and fusion pianist, keyboardist, and composer. Many of his compositions are considered jazz standards. Wikipedia
Born: June 12, 1941 (age 72), Chelsea, MA
Nationality: American
Spouse: Gayle Moran
Compositions: Spain, Sea Journey, Short Tales of the Black Forest, More
Music groups: Return to Forever (1972 – 2008), Chick Corea Elektric Band (Since 1986), Five Peace Band, Circle (1971)
More
Upcoming events
Aug 9 Sat
Chick Corea – Solo Piano
Stanford, CA (near you)
Apr 2
Chick Corea & Béla Fleck
Oberlin, Ohio
People also search for


3:12
the mowgli's
More images
The Mowgli's
Band
Albums: Sound the Drum, Waiting for the Dawn
Members: Katie Earl, Colin Dieden, Spencer Trent, Michael Vincze, Matthew Di Panni
People also search for: Walk the Moon, Smallpools, ...
Get updates about The Mowgli's?
Upcoming events
The Mowgli's
Apr 15 - Cambridge, Massachusetts (near you)
The Mowgli's
Mar 12 - Austin, Texas
The Mowgli's
Mar 15 - Houston, Texas
Web
Images
News
Maps

- <script type="application/ld+json">

```
[{
   "@context" : "http://schema.org",
   "@type" : "MusicEvent",
   "name" : "B.B. King",
   "startDate" : "2014-04-12T19:30",
   "location" : {
      "@type" : "Place",
      "name" : "Lupo's Heartbreak Hotel",
      "address" : "79 Washington St., Providence, RI"
   },
   "offers" : {
      "@type" : "Offer",
      "url" : "https://www.etix.com/ticket/1771656"
   }
},...
```

How to integrate JSON-LD into SensorThings

1. Define “@context” to SensorThings API

	Ways	Reasons
Offline	Referencing an JSON-LD context from a JSON document via an HTTP Link Header (Fig 1)	1) The original data won’t be changed.
	Referencing the remote context with parsing the "@context" in the retrieved JSON-LD document (Fig 2)	1) Change a little bit in the original data. 2) JSON-LD format is intuitive.
Inline	Put the whole context in the data {“@context”: {…….}, original data } (Fig 3)	1) Change the original data. 2) JSON-LD format is intuitive. 3) The data size is big.

Fig 1.

```
GET /ordinary-json-document.json HTTP/1.1
Host: example.com
Accept: application/ld+json,application/json,*/*;q=0.1

====================================

HTTP/1.1 200 OK
...
Content-Type: application/json
Link: <http://demo.student.geocens.ca/contexts/Thing.jsonld>; rel="http://
www.w3.org/ns/json-ld#context"; type="application/ld+json"
{
 "Description": "This is an Arduino",
 "Self-Link": "http://demo.student.geocens.ca:8080/SensorThings_V1.0/
Things(1)",
 ...
}
```

Fig 2

```
{"@context":" http://demo.student.geocens.ca/contexts/Thing.jsonld",
 "Description": " This an Arduino",
 "Self-Link":     "http://demo.student.geocens.ca:8080/SensorThings_V1.0/
Things(1)",
 ...
}
```

Fig 3


```
{
 "@context":{
  "Self-Link":"@id",
  "Description":"http://schema.org/description",
  "Locations":"http://ogc-iot.github.io/ogc-iot-api/datamodel#Locations",
  "@base":"http://demo.student.geocens.ca:8080/SensorThings_V1.0/",
  "Association-Link":{"@id":"http://ogc-iot.github.io/ogc-iot-api/datamodel#Association-Link","@type":"@id"},
  "Navigation-Link":{"@id":"http://ogc-iot.github.io/ogc-iot-api/datamodel#Navigation-Link","@type":"@id"},
  "ID":"http://ogc-iot.github.io/ogc-iot-api/datamodel#ID",
  "Datastreams":{"@id":"http://ogc-iot.github.io/ogc-iot-api/datamodel#Datastreams","@type":"@id"},
  "TaskingCapabilities":{"@id":"http://ogc-iot.github.io/ogc-iot-api/datamodel#TaskingCapabilities","@type":"@id"},
  "Metadata":{"@id ":" http://ogc-iot.github.io/ogc-iot-api/datamodel#Metadata","@type":"@id"},
  "Properties":{"@id ":" http://ogc-iot.github.io/ogc-iot-api/datamodel#Properties","@type":"@id"},
  "SensorML":"http://example.com/vacab#:SensorML",
  "type":"@type"
 },
 "Description":"thing_1",
 "Self-Link":"http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things(1)",
  . . .
}
```

SensorThings API
JSON-LD example

Thing Context


```
{
 "@context":{
  "Self-Link":"@id",
  "Description":"http://schema.org/description",
  "Locations":"http://ogc-iot.github.io/ogc-iot-api/datamodel#Locations",
  "@base":"http://demo.student.geocens.ca:8080/SensorThings_V1.0/",
  "Association-Link":{"@id":"http://ogc-iot.github.io/ogc-iot-api/datamodel#Association-Link","@type":"@id"},
  "Navigation-Link":{"@id":"http://ogc-iot.github.io/ogc-iot-api/datamodel#Navigation-Link","@type":"@id"},
  "ID":"http://ogc-iot.github.io/ogc-iot-api/datamodel#ID",
  "Datastreams":{"@id":"http://ogc-iot.github.io/ogc-iot-api/datamodel#Datastreams","@type":"@id"},
  "TaskingCapabilities":{"@id":"http://ogc-iot.github.io/ogc-iot-api/datamodel#TaskingCapabilities","@type":"@id"},
  "Metadata":{"@id ":" http://ogc-iot.github.io/ogc-iot-api/datamodel#Metadata","@type":"@id"},
  "Properties":{"@id ":" http://ogc-iot.github.io/ogc-iot-api/datamodel#Properties","@type":"@id"},
  "SensorML":"http://example.com/vacab#:SensorML",
  "type":"@type"
 }
}
```

Thing (referencing a remote context by putting @context at the top)


```
{
"@context":"http:// demo.student.geocens.ca/contexts/Thing.jsonld ",
"Description":"thing_1",
"Self-Link":"http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things(1)",
"ID":1,
"Metadata":{
                "type":"SensorML"
},
"Locations":{
                "Association-Link":"Things(1)/$links/Locations",
                "Navigation-Link":"Things(1)/Locations"
},
"Datastreams":{
                "Association-Link":"Things(1)/$links/Datastreams",
                "Navigation-Link":"Things(1)/Datastreams"
},
"TaskingCapabilities":{
                "Association-Link":"Things(1)/$links/TaskingCapabilities",
                "Navigation-Link":"Things(1)/TaskingCapabilities"
},
"Properties":{
    "Owner":"John Doe",
    "Color":"White"
  }
}
```

RDF Triples of the Thing example

<http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things(1)> < http://ogc-iot.github.io/ogc-iot-api/datamodel#Metadata> _:c14n0 .
<http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things(1)> < http://ogc-iot.github.io/ogc-iot-api/datamodel#Properties> _:c14n3 .
<http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things(1)> <http://ogc-iot.github.io/ogc-iot-api/datamodel#Datastreams> _:c14n4 .
<http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things(1)> <http://ogc-iot.github.io/ogc-iot-api/datamodel#ID> "1"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things(1)> <http://ogc-iot.github.io/ogc-iot-api/datamodel#Locations> _:c14n2 .
<http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things(1)> <http://ogc-iot.github.io/ogc-iot-api/datamodel#TaskingCapabilities> _:c14n1 .
<http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things(1)> < http://ogc-iot.github.io/ogc-iot-api/datamodel#Description > "thing_1" .
_:c14n0 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://example.com/vacab#:SensorML> .
_:c14n1 <http://ogc-iot.github.io/ogc-iot-api/datamodel#Association-Link> <http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things(1)/$links/TaskingCapabilities> .
_:c14n1 <http://ogc-iot.github.io/ogc-iot-api/datamodel#Navigation-Link> <http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things(1)/TaskingCapabilities> .
_:c14n2 <http://ogc-iot.github.io/ogc-iot-api/datamodel#Association-Link> <http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things(1)/$links/Locations> .
_:c14n2 <http://ogc-iot.github.io/ogc-iot-api/datamodel#Navigation-Link> <http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things(1)/Locations> .
_:c14n4 <http://ogc-iot.github.io/ogc-iot-api/datamodel#Association-Link> <http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things(1)/$links/Datastreams> .
_:c14n4 <http://ogc-iot.github.io/ogc-iot-api/datamodel#Navigation-Link> <http://demo.student.geocens.ca:8080/SensorThings_V1.0/Things(1)/Datastreams> .

Color Meaning : **Subject** **Prediction** **Object** **Value of Object**