



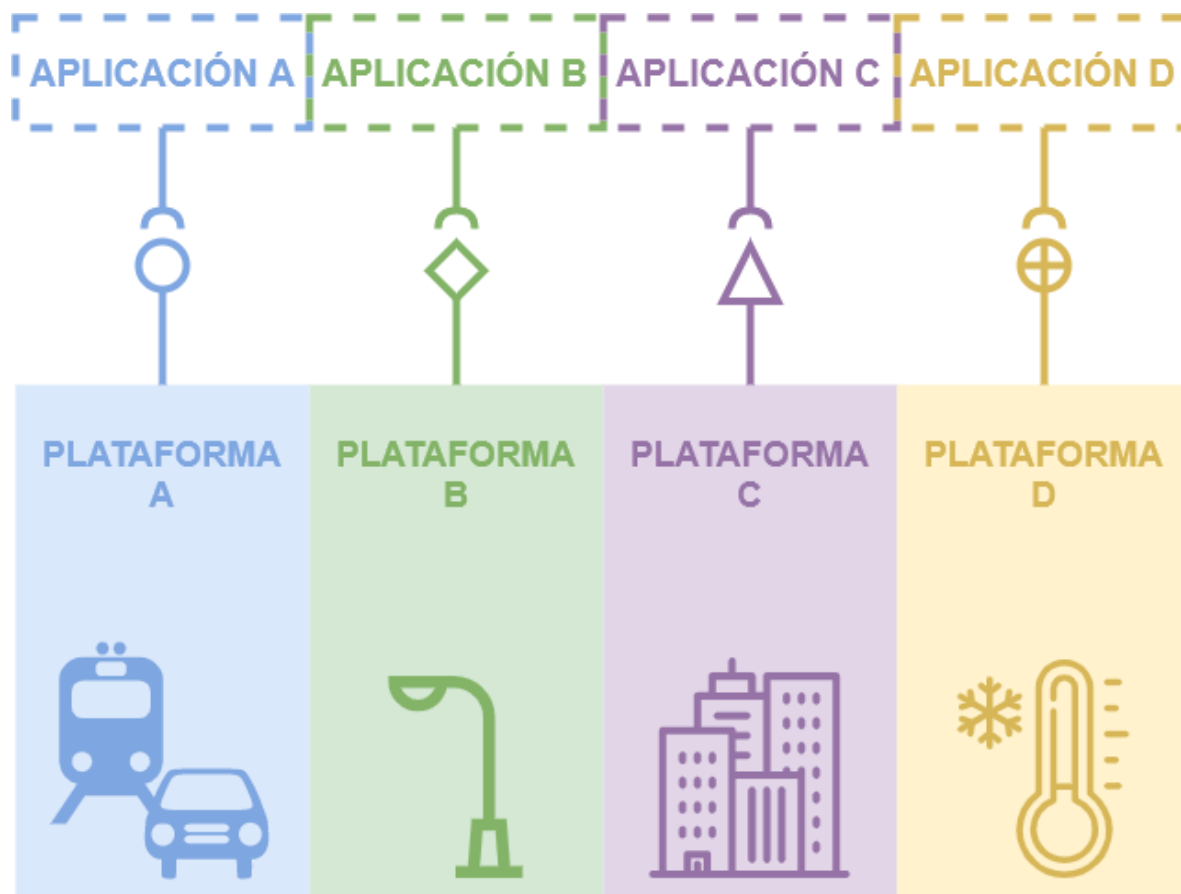
INTEROPERABILIDAD en *IoT* mediante ESTÁNDARES OGC “X FORO ILAF”

Sergi Trilles

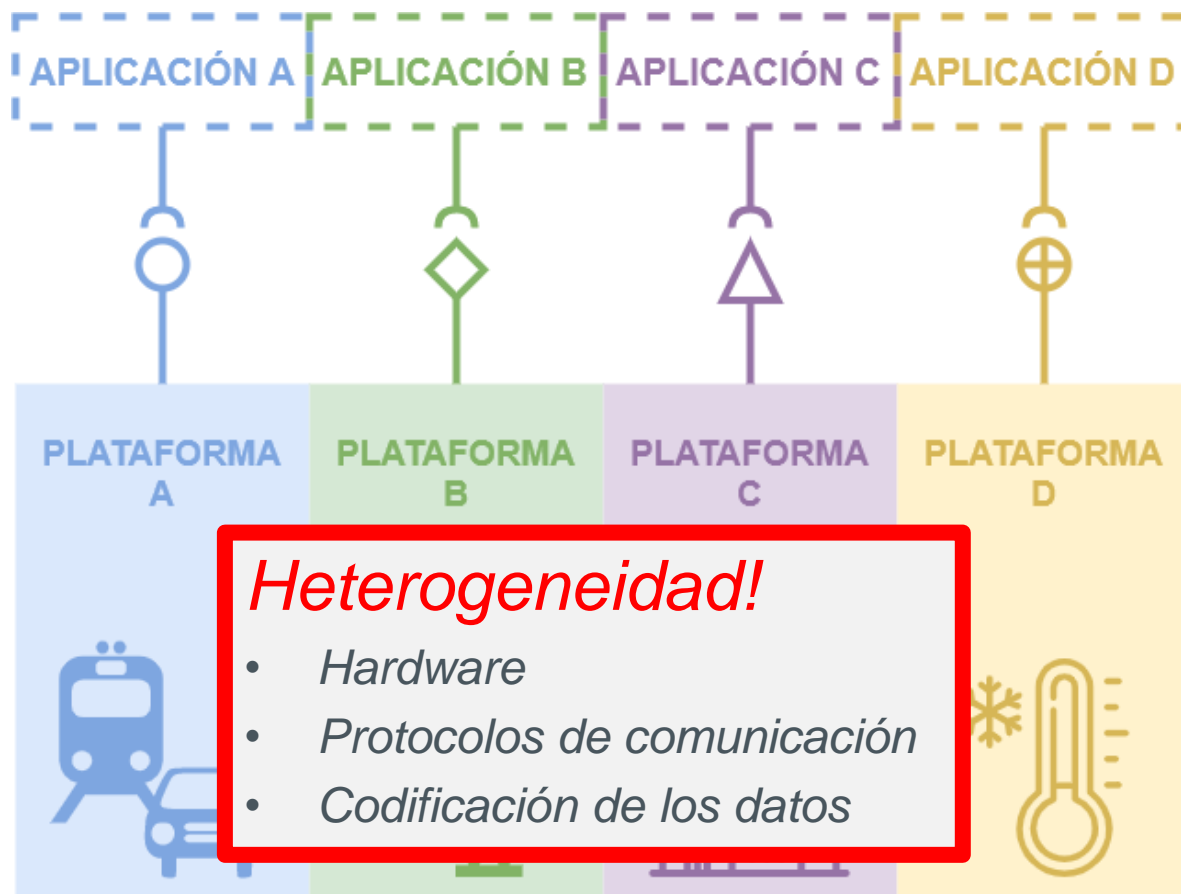
Universidad Jaime I

1. Introducción
2. *Propuestas OGC*
 - a) *OGC Sensor Observation Service*
 - b) *OGC SensorThings API*
3. *OGC SensorThings API vs SOS*
4. Conclusiones

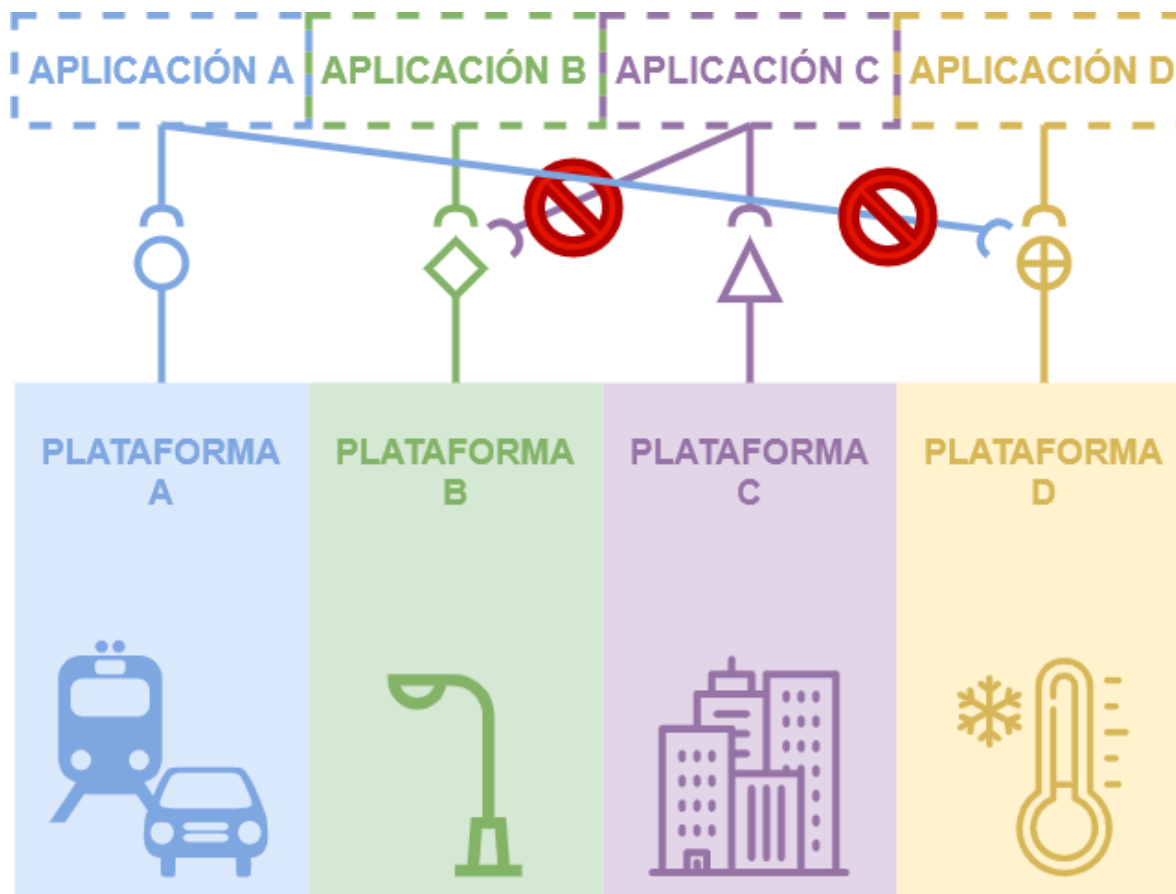
Introducción (I)



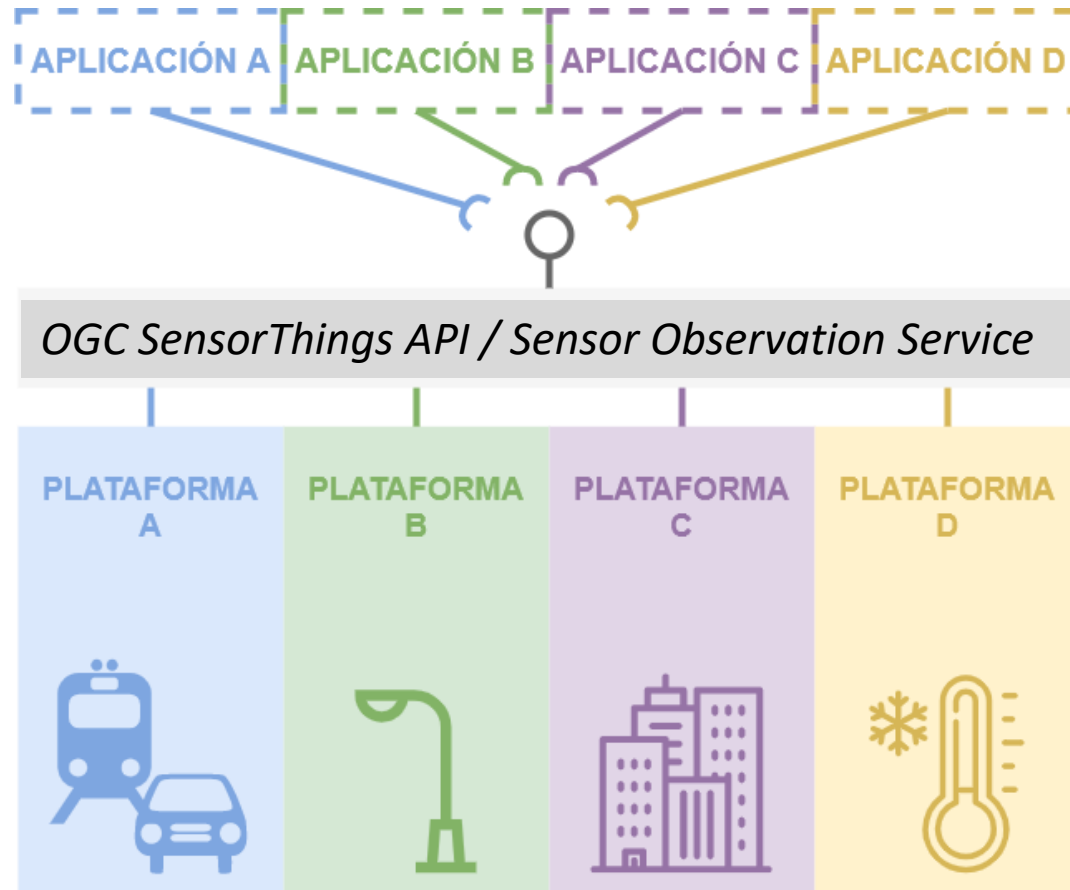
Introducción (II)



Introducción (III)



Introducción (IV)



OGC Sensor Observation Service (I)

- Estándar OGC desde 2007
- Pensado para dar soporte a *sensores*
- Forma parte del *OGC Sensor Web Enablement (SWE)*
- Tres modelos:
 - *O&M para definir observaciones*
 - *SensorML para definir el sensor*
 - *TransducerML para definir el actuador*
- Interfaz SOAP
- Formato de codificación *XML*
- Soporte para sensores *In-situ* o *Remote sensing*

- Conceptos:
 - *Offering*: los datos se agrupan en diferentes *offerings*, por tipo y sensor o grupo de ellos
 - *Procedure*: describe (metadatos) un sensor, una colección de sensores o un proceso que genera algunas observaciones
 - *Feature of Interest*: cada observación se vincula a una característica de interés (*FoI*), que generalmente describe dónde ocurrió el fenómeno observado
 - *ObservedProperty*: el fenómeno a medir, como: temperatura, dirección del viento; puede ser numérico, lógico, categórico o descriptivo
 - *Observation*: es el valor de una propiedad observada en un momento y lugar particular (*FoI*)

- Operaciones:
 - *GetCapabilities*: proporciona acceso a metadatos y detalles sobre las capacidades del servicio
 - *DescribeSensor*: solicita los metadatos del sensor en formato *SensorML*
 - *GetObservation*: devuelve las observaciones en formato *O&M*
 - *RegisterSensor*: permite a los usuarios registrar sensores
 - *InsertObservation*: inserta observaciones

- Implementaciones:
 - **istSOS**: codificada en **Python**, desarrollada en el Instituto de Ciencias de la Tierra en SUPSI (University of Applied Sciences and Arts of Southern Switzerland). Tiene una interfaz fácil de usar para la configuración del servicio, una API REST y exploración de datos. La última versión de istSOS2 (istSOS-2.3.1) se lanzó en febrero de 2016. Actualmente, istSOS3 está en desarrollo
 - **52North-SOS**: está implementada en **Java** y desarrollada por 52North GMBH, con sede en Münster, Alemania. Incluye una variedad de características extendidas, que incluyen soporte para el servicio de descarga INSPIRE y codificaciones XML especializadas, una API REST y una interfaz de cliente extensa para configuración de servicios y exploración de datos. Última versión SOS 4.4.7 (hace 13 días)

- Estándar OGC desde 2016
- Pensado para dar soporte a *IoT*
- Inspirado en *OGC Sensor Web Enablement (SWE)*
- Dos perfiles
 - *Sensing* aprobado en 2016
 - *Tasking* aprobado en 2019
- Interfaz RESTful (*HTTP*) *CREATE, READ, UPDATE y DELETE*
- Soporte *MQTT* y *CoAP*
- Formato de codificación *JSON*

OGC SensorThings API (II)

- Sensing Core* (basado en O&M) (I)

Thing: un objeto del mundo físico o virtual

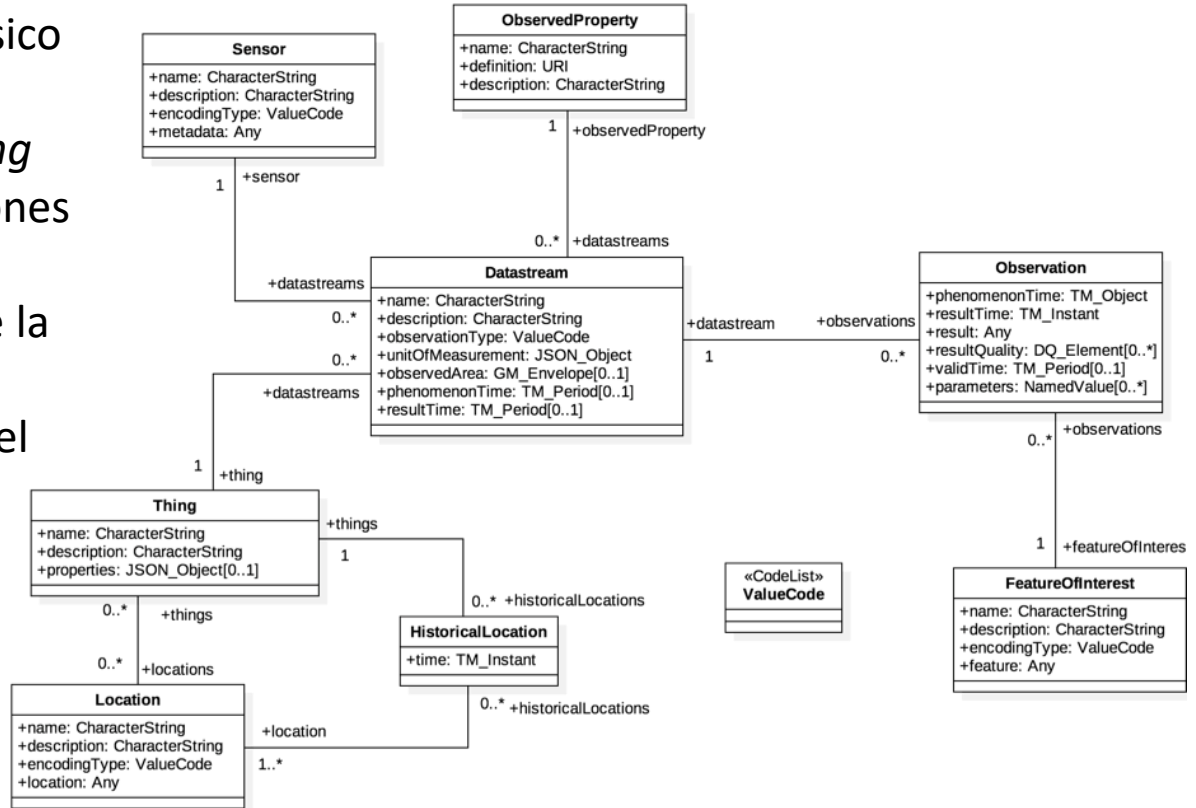
Locations: localización del *Thing*

HistoricalLocations: localizaciones históricas

Datastream: observaciones de la misma propiedad y sensor

ObservedProperty: especifica el fenómeno de una observación

Sensor: un instrumento que observa una propiedad o fenómeno

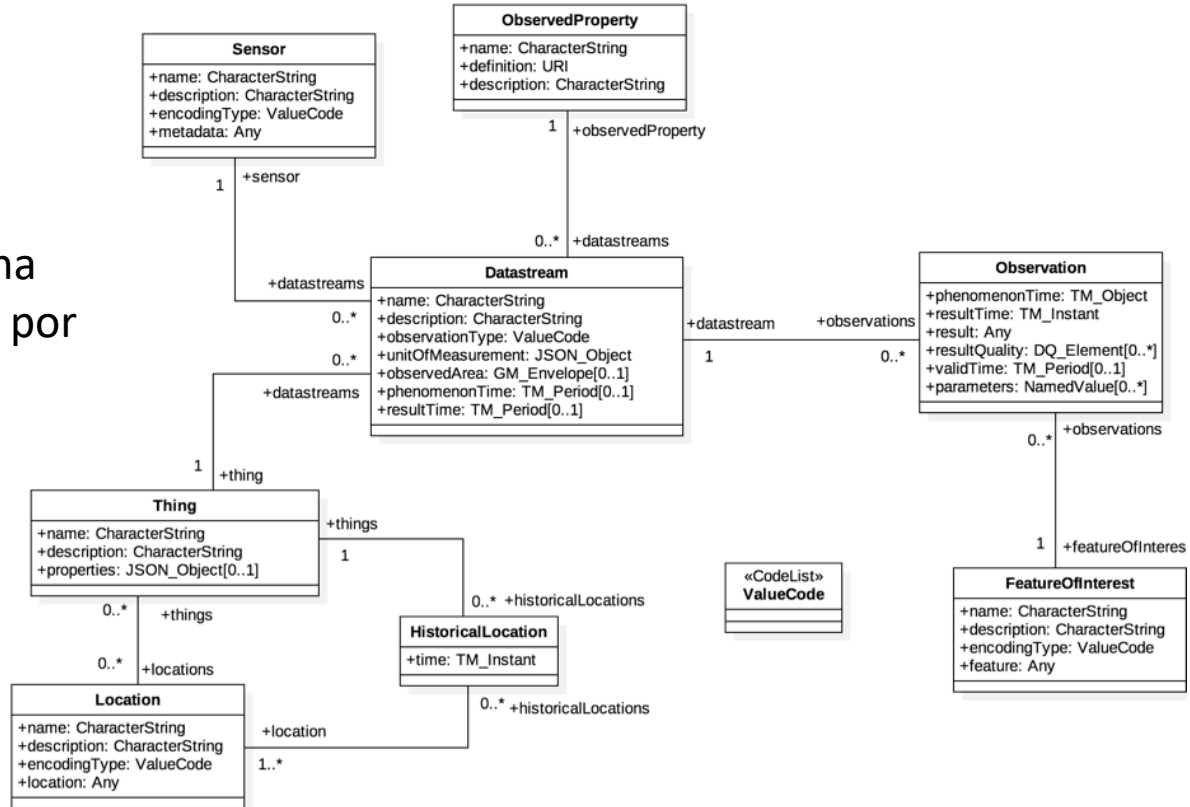


OGC SensorThings API (II)

- *Sensing Core* (basado en O&M) (II)

Observation: acto de medir y determinar el valor de una propiedad

FeatureOfInterest: una observación se describe con una característica de interés, como por ejemplo el lugar



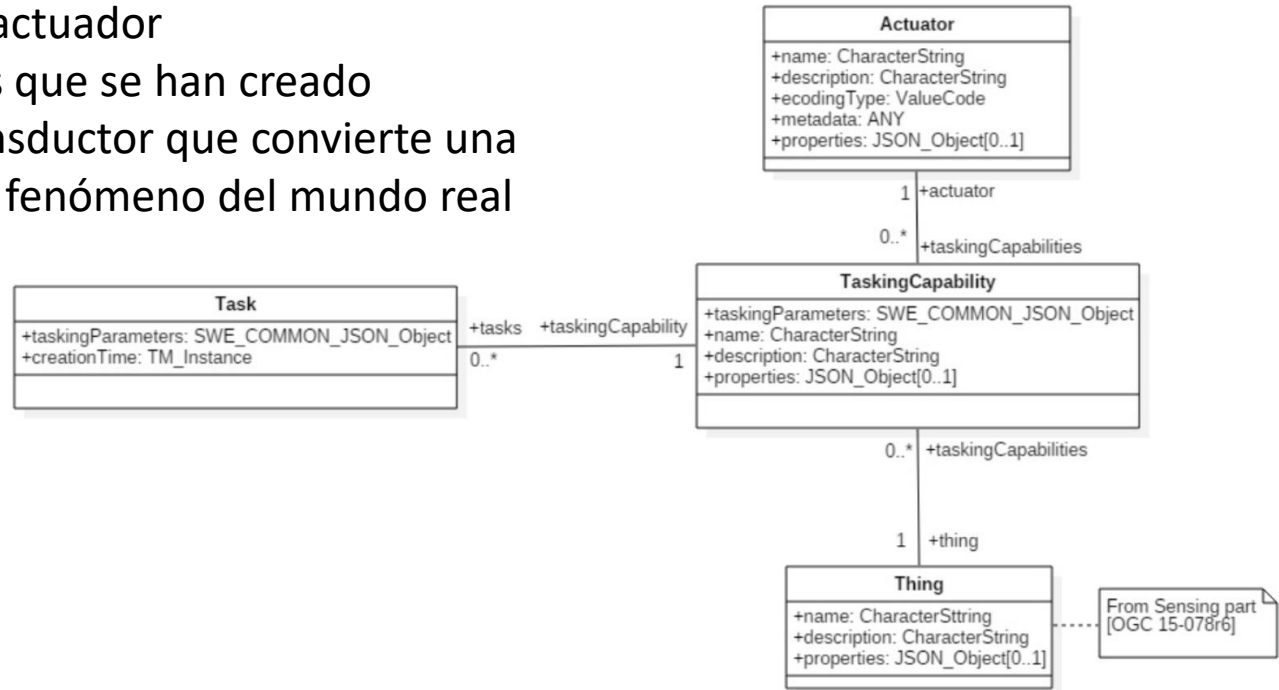
OGC SensorThings API (III)

- Tasking Core

TaskingCapabilities: especifica los parámetros aptos para tareas de un actuador

Task: colección de tareas que se han creado

Actuator: un tipo de transductor que convierte una señal en alguna acción o fenómeno del mundo real



OGC SensorThings API (IV)

`http://serviceURI/Things(id)/Datastreams?$stop=2&orderBy=ID`

The diagram illustrates the structure of the URL `http://serviceURI/Things(id)/Datastreams?$stop=2&orderBy=ID`. A horizontal line with three vertical tick marks below it separates the URL into three segments. The first segment, `http://serviceURI`, is labeled "service URI". The second segment, `/Things(id)/Datastreams`, is labeled "resource path". The third segment, `?$stop=2&orderBy=ID`, is labeled "query options".

- *Implementaciones:*

- **GOST:** GO Sensor Things (*GOST*) fue desarrollado por Geodan utilizando el lenguaje de programación Golang (Go). El desarrollo de la implementación aún está en curso, pero presenta la funcionalidad completa *OGC SensorThings API Sensing* e incluye un *dashboard* para la visualización de datos.
- **Mozilla-SensorThings:** Mozilla Corporation comenzó a desarrollar una implementación de *OGC SensorThings API*, pero el desarrollo se ha estancado desde febrero de 2017.
- **FROST:** Fraunhofer Open-Source Sensor Things (*FROST*) fue desarrollado por Fraunhofer IOSB en Java. Los desarrolladores de *FROST* continúan trabajando en el proyecto para ampliar sus características y completar el estándar *OGC SensorThings API*.

OGC SensorThings API vs SOS (I)

	OGC SensorThings API	OGC Sensor Observation Service
Codificación	JSON	XML
Estilo de arquitectura	Resource Oriented Architecture	Service Oriented Architecture
Binding	REST	SOAP
Insertar sensores o observaciones	HTTP (CRUD)	Utiliza operaciones específicas, p. ej: <i>RegisterSensor()</i> , <i>InsertObservation()</i>
Eliminar sensores	HTTP DELETE	Utiliza operaciones específicas, p. ej: <i>DeleteSensor()</i>
Paginación	\$stop, \$skip, \$nextLink	No soportado
Soporte paradigma Pub/Sub	MQTT	No soportado
Actualización de los sensores u observaciones	HTTP PATCH y JSON PATCH	No soportado
Eliminar observaciones	HTTP DELETE	No soportado
Soporte Linked-data	JSON-LD	No soportado
Múltiples observaciones	\$select	No soportado

OGC SensorThings API vs SOS (II)

Insertar una observación en SOS

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <env:Envelope
3   xmlns:env="http://www.w3.org/2003/05/soap-envelope"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.w3.org/2003/05/soap-envelope
5   http://www.w3.org/2003/05/soap-envelope/soap-envelope.xsd">
6   <env:Body>
7     <sos:InsertObservation
8       xmlns:sos="http://www.opengis.net/sos/2.0"
9       xmlns:swe="http://www.opengis.net/swe/2.0"
10      xmlns:swe="http://www.opengis.net/swe/2.0"
11      xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
12      xmlns:gml="http://www.opengis.net/gml/3.2"
13      xmlns:xlink="http://www.w3.org/1999/xlink"
14      xmlns:om="http://www.opengis.net/om/2.0"
15      xmlns:sams="http://www.opengis.net/samplingSpatial/2.0"
16      http://www.opengis.net/samplingSpatial/2.0" service="SOS" version="2.0.0"
17      xsi:schemaLocation="http://www.opengis.net/sos/2.0 http://schemas.opengis.net/sos/2.0/sos.xsd
18      http://www.opengis.net/samplingSpatial/2.0 http://schemas.opengis.net/samplingSpatial/2.0/spatialSamplingFeature.xsd">
19     <!-- multiple offerings are possible -->
20     <sos:offering>http://www.52north.org/test/offering/9</sos:offering>
21     <sos:observation>
22       <om:Observation gml:id="o1">
23         <om:type xlink:href="http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement"/>
24         <om:phenomenonTime>
25           <gml:TimeInstant gml:id="phenomenonTime">
26             <gml:timePosition>2012-11-19T17:46:15.000+00:00</gml:timePosition>
27           </gml:TimeInstant>
28         </om:phenomenonTime>
29         <om:resultTime xlink:href="#phenomenonTime"/>
30         <om:procedure xlink:href="http://www.52north.org/test/procedure/9"/>
31         <om:parameter>
32           <om:NamedValue>
33             <om:name xlink:href="http://www.opengis.net/def/param-name/OGC-OM/2.0/samplingGeometry"/>
34             <om:value xsi:type="gml:GeometryPropertyType">
35               <gml:Point gml:id="SamplingPoint1">
36                 <gml:description>description</gml:description>
37                 <gml:identifier codeSpace="">hereIdentifier</gml:identifier>
38                 <gml:name>hereIam</gml:name>
39                 <gml:pos srsName="http://www.opengis.net/def/crs/EPSG/0/4326">52.9 7.52</gml:pos>
40               </gml:Point>
41             </om:value>
42           </om:NamedValue>
43         </om:parameter>
44         <om:observedProperty xlink:href="http://www.52north.org/test/observableProperty/9_3"/>
45         <om:featureOfInterest>
46           <sams:SpatialSamplingFeature gml:id="ssf_test_feature_9">
47             <gml:identifier codeSpace="">http://www.52north.org/test/featureOfInterest/9</gml:identifier>
48             <gml:name>S2North</gml:name>
49             <sf:type xlink:href="http://www.opengis.net/def/samplingFeatureType/OGC-OM/2.0/SF_SamplingPoint"/>
50             <sf:sampledFeature xlink:href="http://www.52north.org/test/featureOfInterest/1"/>
51             <sams:shape>
52               <gml:Point gml:id="test_feature_9">
53                 <gml:pos srsName="http://www.opengis.net/def/crs/EPSG/0/4326">51.935101100104916
54                 </gml:pos>
55               </gml:Point>
56             </sams:shape>
57             </sams:SpatialSamplingFeature>
58           </om:featureOfInterest>
59           <om:result xsi:type="gml:MeasureType" uom="test_unit_9_3">0.29</om:result>
60         </om:Observation>
61       </sos:observation>
62     </sos:InsertObservation>
63   </env:Body>
64 </env:Envelope>
```

Insertar una observación en SensorThings API

```
{
  "phenomenonTime": "2017-02-07T18:02:00.000Z",
  "resultTime" : "2017-02-07T18:02:05.000Z",
  "result" : 21.6,
  "Datastream":{"@iot.id":8}
}
```

OGC SensorThings API vs SOS (III)

Insertar un sensor en SOS

```
<?xml version="1.0" encoding="UTF-8"?>
<root xmlns:sos="http://www.opengis.net/sos" xmlns:ogc="http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sos http://www.opengis.net/sos/sos.xsd http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement.xsd">
  <name>TemperatureMonitoringSystem</name>
  <description>Sensor system monitoring area temperature</description>
  <properties>
    <DeploymentCondition>
      <name>DeploymentCondition</name>
      <value>Deployed in a third floor balcony</value>
    </DeploymentCondition>
    <CaseUsed>
      <name>CaseUsed</name>
      <value>Radiation shield</value>
    </CaseUsed>
  </properties>
  <locations>
    <location>
      <name>UofC CCIT</name>
      <description>University of Calgary, CCIT building</description>
      <encodingType>application/vnd.geo+json</encodingType>
      <location>
        <type>Point</type>
        <coordinates>[-114.133, 51.08]</coordinates>
      </location>
    </location>
  </locations>
  <datastreams>
    <datastream>
      <name>AirTemperatureDS</name>
      <description>Datastream for recording temperature</description>
      <observationType>http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement</observationType>
      <unitOfMeasurement>
        <name>DegreeCelsius</name>
        <symbol>degC</symbol>
        <definition>http://www.qudt.org/qudt/owl/1.0.0/unit/Instances.html#DegreeCelsius</definition>
      </unitOfMeasurement>
      <observedProperty>
        <name>AreaTemperature</name>
        <description>The degree or intensity of heat present in the area</description>
        <definition>http://www.qudt.org/qudt/owl/1.0.0/quantity/Instances.html#AreaTemperature</definition>
      </observedProperty>
      <sensor>
        <name>DHT22</name>
        <description>DHT22 temperature sensor</description>
        <encodingType>application/pdf</encodingType>
        <metadata>https://cdn-shop.adafruit.com/datasheets/DHT22.pdf</metadata>
      </sensor>
    </datastream>
  </datastreams>
</root>
```

Insertar Thing en SensorThings API

```
{
  "name": "Temperature Monitoring System",
  "description": "Sensor system monitoring area temperature",
  "properties": {
    "Deployment Condition": "Deployed in a third floor balcony",
    "Case Used": "Radiation shield"
  },
  "Locations": [
    {
      "name": "UofC CCIT",
      "description": "University of Calgary, CCIT building",
      "encodingType": "application/vnd.geo+json",
      "location": {
        "type": "Point",
        "coordinates": [-114.133, 51.08]
      }
    }
  ],
  "Datastreams": [
    {
      "name": "Air Temperature DS",
      "description": "Datastream for recording temperature",
      "observationType": "http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement",
      "unitOfMeasurement": {
        "name": "Degree Celsius",
        "symbol": "degC",
        "definition": "http://www.qudt.org/qudt/owl/1.0.0/unit/Instances.html#DegreeCelsius"
      },
      "ObservedProperty": {
        "name": "Area Temperature",
        "description": "The degree or intensity of heat present in the area",
        "definition": "http://www.qudt.org/qudt/owl/1.0.0/quantity/Instances.html#AreaTemperature"
      },
      "Sensor": {
        "name": "DHT22",
        "description": "DHT22 temperature sensor",
        "encodingType": "application/pdf",
        "metadata": "https://cdn-shop.adafruit.com/datasheets/DHT22.pdf"
      }
    }
  ]
}
```



OGC SensorThings API vs SOS (IV)

Devolver una observación en SOS

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <env:Envelope
3   xmlns:env="http://www.w3.org/2003/05/soap-envelope"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.w3.org/2003/05/soap-envelope
5     http://www.w3.org/2003/05/soap-envelope/soap-envelope.xsd">
6   <env:Body>
7     <cos:GetObservation
8       xmlns:sos="http://www.opengis.net/sos/2.0"
9       xmlns:fes="http://www.opengis.net/fes/2.0"
10      xmlns:gml="http://www.opengis.net/gml/3.2"
11      xmlns:swe="http://www.opengis.net/swe/2.0"
12      xmlns:xlink="http://www.w3.org/1999/xlink"
13      xmlns:sms="http://www.opengis.net/sms/2.0" service="SOS" version="2.0.0"
14      xsi:schemaLocation="http://www.opengis.net/sos/2.0 http://schemas.opengis.net/sos/2.0/sos.xsd">
15     <!-- optional -->
16     <swes:extension xsi:type="swes:BooleanDefinition" definition="MergeObservationsIntoDataArray" csw:defaultValue="true" />
17     </swes:extension>
18     <!-- optional -->
19     <swes:extension xsi:type="swes:CountPropertyType" csw:count="4326" />
20     </swes:extension>
21     <!-- optional -->
22     <swes:extension xsi:type="swes:TextPropertyType" csw:language="per" />
23     </swes:extension>
24     <!-- optional -->
25     <swes:extension xsi:type="swes:BooleanPropertyType" csw:identifier="returnHumanReadableIdentifier" />
26     </swes:extension>
27     <!-- optional, multiple values possible -->
28     <cos:procedure>http://www.52north.org/test/procedure/1/</cos:procedure>
29     <cos:procedure>http://www.52north.org/test/procedure/2/</cos:procedure>
30     <cos:procedure>http://www.52north.org/test/procedure/6/</cos:procedure>
31     <!-- optional, multiple values possible -->
32     <cos:offering>http://www.52north.org/test/offering/1/</cos:offering>
33     <cos:offering>http://www.52north.org/test/offering/2/</cos:offering>
34     <cos:offering>http://www.52north.org/test/offering/6/</cos:offering>
35     <!-- optional, multiple values possible -->
36     <cos:observableProperty>http://www.52north.org/test/observableProperty/1/</cos:observableProperty>
37     <cos:observableProperty>http://www.52north.org/test/observableProperty/2/</cos:observableProperty>
38     <cos:observableProperty>http://www.52north.org/test/observableProperty/6/</cos:observableProperty>
39     <!-- optional -->
40     <cos:temporalFilter>
41       <fes:During>
42         <fes:ValueReference>phenomenonTime</fes:ValueReference>
43         <gml:TimePeriod gml:id="tp_1">
44           <gml:beginPosition>2012-11-19T14:00:00.000+01:00</gml:beginPosition>
45           <gml:endPosition>2012-11-19T15:00:00.000+01:00</gml:endPosition>
46         </gml:TimePeriod>
47       </fes:During>
48     </cos:temporalFilter>
49     <!-- optional -->
50     <cos:temporalFilter>
51       <fes:Equals>
52         <fes:ValueReference>phenomenonTime</fes:ValueReference>
53         <gml:TimeInstant gml:id="ti_1">
54           <gml:timePosition>2012-11-19T14:00:00.000+01:00</gml:timePosition>
55         </gml:TimeInstant>
56       </fes:Equals>
57     </cos:temporalFilter>
58     <!-- optional -->
59     <cos:temporalFilter>
60       <fes:Equals>
61         <fes:ValueReference>phenomenonTime</fes:ValueReference>
62         <gml:TimeInstant gml:id="ti_1">
63           <gml:timePosition>2012-11-19T14:00:00.000+01:00</gml:timePosition>
64         </gml:TimeInstant>
65       </fes:Equals>
66     </cos:temporalFilter>
67     <!-- optional -->
68     <cos:temporalFilter>
69       <fes:Equals>
70         <fes:ValueReference>resultTime</fes:ValueReference>
71         <gml:TimeInstant gml:id="ti_1">
72           <gml:timePosition>2012-11-19T14:00:00.000+01:00</gml:timePosition>
73         </gml:TimeInstant>
74       </fes:Equals>
75     </cos:temporalFilter>
76     <!-- optional -->
77     <cos:featureOfInterest>http://www.52north.org/test/featureOfInterest/1/</cos:featureOfInterest>
78     <cos:featureOfInterest>http://www.52north.org/test/featureOfInterest/2/</cos:featureOfInterest>
79     <cos:featureOfInterest>http://www.52north.org/test/featureOfInterest/6/</cos:featureOfInterest>
80     <!-- optional -->
81     <cos:spatialFilter>
82       <fes:BBOX>
83         <fes:ValueReference>om:featureOfInterest/sms:SF_SpatialSamplingFeature/sms:shape
84         </fes:ValueReference>
85         <gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
86           <gml:lowerCorner>0 0</gml:lowerCorner>
87           <gml:upperCorner>60 60</gml:upperCorner>
88         </gml:Envelope>
89       </fes:BBOX>
90     </cos:spatialFilter>
91     <!-- Spatial filter for Spatial Filtering Profile -->
92     <!-- optional -->
93     <cos:spatialFilter>
94       <fes:BBOX>
95         <fes:ValueReference>http://www.opengis.net/req/omml/2.0/data/samplingGeometry
96         </fes:ValueReference>
97         <gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
98           <gml:lowerCorner>0 0</gml:lowerCorner>
99           <gml:upperCorner>60 60</gml:upperCorner>
100        </gml:Envelope>
101      </fes:BBOX>
102    </cos:spatialFilter>
103    <!-- optional -->
104    <cos:responseFormat>http://www.opengis.net/om/2.0/cos:responseFormat
105  </cos:responseFormat>
106 </env:Body>
107 </env:Envelope>
```

Devolver una observación en SensorThings API

```
{
  "value": [
    {
      "@iot.id": 1595550,
      "@iot.selfLink": "https://server/Observations(1595550)",
      "phenomenonTime": "2017-02-16T21:55:12.841Z",
      "result": "7",
      "resultTime": null,
      "Datastream@iot.navigationLink": "https://server/Observations(1595550/v1.0/Observations(1595550)/Datastream",
      "FeatureOfInterest@iot.navigationLink": "https://server/Observations(1595550)/FeatureOfInterest"
    }
  ]
}
```

MQTT (SensorThing API) vs HTTP (SOS)

		3G		Wifi	
		HTTPS	MQTT	HTTPS	MQTT
receive	messages / hour	1,708	160,278	3,628	263,314
	% battery / msg	0.01709	0.00010	0.00095	0.00002
	msgs (note losses)	240 / 1024	1024 / 1024	524 / 1024	1024 / 1024
send	msg / hour	1,926	21,685	5,229	23,184
	% battery / msg	0.00975	0.00082	0.00104	0.00016

Fuente: <https://www.slideshare.net/Eurotechchannel/eth-mqtt-brokeriotreadydevnation2015>

OGC SensorThings API ofrece más funciones, mejor experiencia al desarrollador, mejor eficiencia y escalabilidad sobre los dispositivos *IoT* actuales

Gracias por su atención

correo: stilles@uji.es
web: <http://www3.uji.es/~stilles>