Co-funded by the European Union

BIRGIT – training on Building InfoRmation models integrated with Geographical InformaTion

With the support of the Erasmus+ Program of the European Union Strategic Partnerships N° 2021-1-SE01-KA220-VET-000028000

Sensor Data Standards

Lecture Notes

Author(s)/Organisation(s):

Anders Östman (Novogit AB)

License

Version

Version 2.0

Date: 2025-04-29

Learning outcomes

At the end of this lecture, the learner is expected to be able to

- Explain core concepts, such as sensors, sensor networks.
- List different types of sensors and their usages.
- Describe the basic principles of accessing sensor networks.

With the support of the Erasmus+ Program of the European Union Strategic Partnerships N° 2021-1-SE01-KA220-VET-000028000

Expected competences when entering the lecture

- No specific pre-requisites required.

Summary

The aim of this lecture is to define the concepts of sensors and sensor network and to give some examples of their usages. The remaining focus of this lecture is then to describe how to access the outputs of the sensors and sensor networks, using OGC standards like Sensor Observation Services and OGC sensor API.

Expected Workload

16 slides with course learning content, 2 classroom hours, 0.2 ECTS (ECVET)

Revision History:

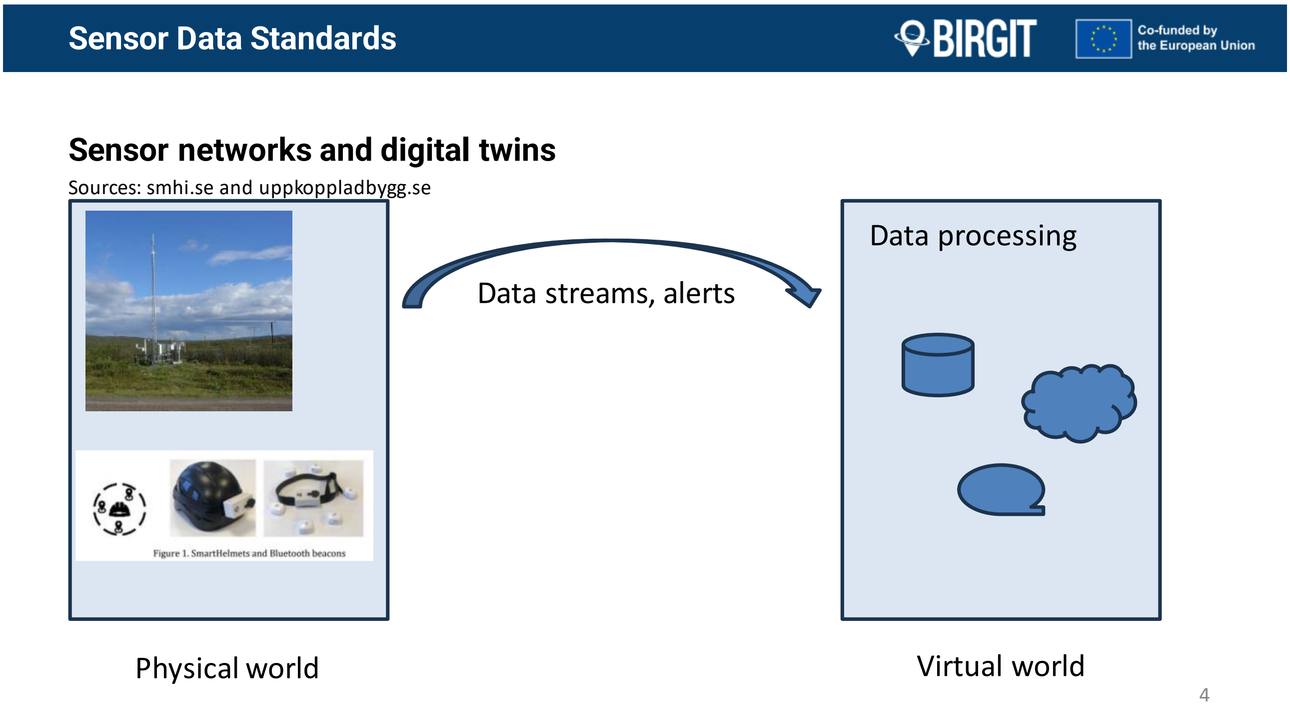
Revision	Date	Author(s)	Status	Description
0.1	2023-10-05	A. Östman	Final Draft	Lecture ready for review
0.2	2024-02-13	A. Östman	Final Draft	Update based on reviewers comments
1.0	2024-03-20	A Östman	Final	Update based on review and comments at sprint meeting
1.1.	2025-01-10	A Östman	Final	License change and ZigBee standard added
2.0	2025-04-29	A Östman	Final	Updated EU logo and disclaimer. Edited by T. Näslund

Co-funded by
the European Union

BIRGIT

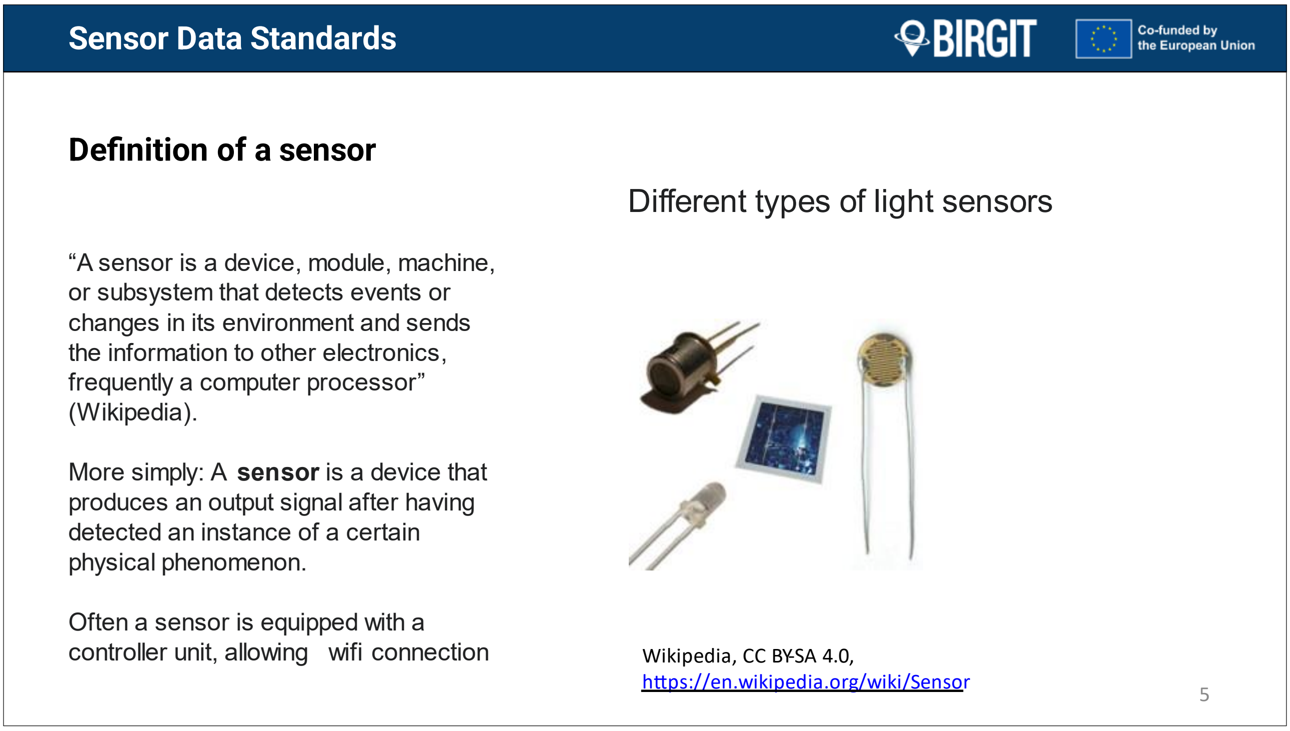
Contents

Sensor networks and digital twins



Sensors and sensor networks are one type of data sources for a digital urban twin. In this lecture we will focus on how sensors and sensor networks are designed and how one can access the data streams and alerts coming from the sensor network. In the next lecture and assignment, we will deal more with the usages of data from sensor networks.

Definition of a sensor



A sensor is a device that measures something. They are usually connected to a computer processor and can communicate with other devices through wifi, GSM or cable connections.

Earlier, analogue sensors were used, which required manual reading of meters. One example are the gauge meters, which measures the water level by a floating device. To connect such analogue sensors to a wireless sensor network, A/D (analogue to digital) conversion is required.

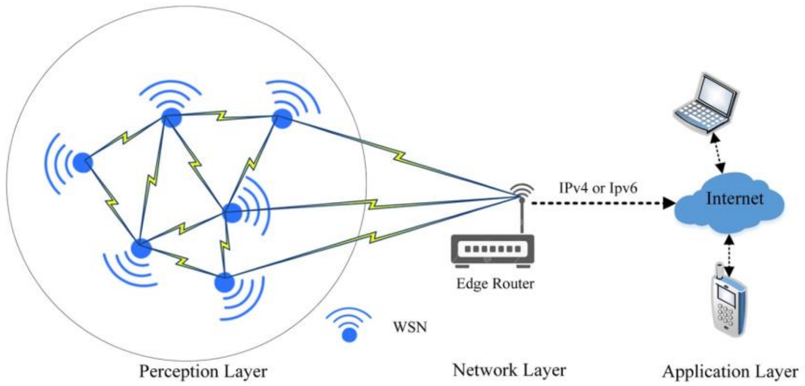
Digital sensors are now common in home-electronic kits and toys and can be connected to small processors, for instance in the Raspberry Pi-family, using the GPIO interface. These small processors often support Wi-Fi connections and can be programmed in Python.

Wireless sensor networks

Sensor Data Standards

BIRGIT Co-funded by the European Union

Wireless sensor networks



Source: Ahmad R, Wazirali R, Abu-Ain T. Machine Learning for Wireless Sensor Networks Security: An Overview of Challenges and Issues. *Sensors*. 2022; 22(13):4730. https://doi.org/10.3390/s22134730 6

A sensor is usually not operating alone but is a part of a wider system, a sensor network. The sensors may be of the same kind or be of different type. Each sensor has a controller unit, which controls the operations of this specific sensor. The controller unit then communicates with a central access point (edge router in the figure), which controls the operations of the entire sensor network.

Accessing the outputs of the sensors is facilitated through the central access point using specific predefined interfaces.

This lecture has a focus on the sensor data standards that are used within the GIS domain, facilitating access to the output of single sensors in a sensor network. These standards specify the form of the HTTP GET requests for accessing one or several sensors and how to interpret the results. In the BIM domain, the IFC standard do not focus on the access to sensors in such a detail as the OGC standards. However, it provides a framework for integrating sensors in BIM models.

Smart helmets

Sensor Data Standards

BIRGIT Co-funded by the European Union

Smart Helmets

Figure 1. SmartHelmets and Bluetooth beacons

- Temperature, accelerometers and gyroscope sensors
- Bluetooth transmission to beacons (sensor controller)
- Beacons connected to the access point of the sensor network by ordinary wifi connection

Source: Rudberg M, Sezer A.A. SmartHelmets and BuildingCloud technologies. https://www.uppkoppladbygg.se/media/amwgv2ch/ub_-white-paper-ncc-scharc-smart-helmets.pdf

7

Smart helmets were briefly introduced in the introductory lecture of digital urban twins. Each helmet is here equipped with three different sensors, namely a temperature sensor, an accelerometer, and a gyroscope. Output signals from these sensors are then transmitted to a beacon, using Bluetooth connection. This beacon acts as a sensor controller and communicates with the central access point of the network.

This type of devices has many different potential usages. The use case referenced on the slide is using the helmets for accident detection on construction sites.

Air quality monitoring stations

Sensor Data Standards BIRGIT Co-funded by the European Union

Air Quality Monitoring Stations

- Only PM10 sensor in this specific case
- A/D converter and sensor controller in box
- Usually GSM connection to the access point of the sensor network

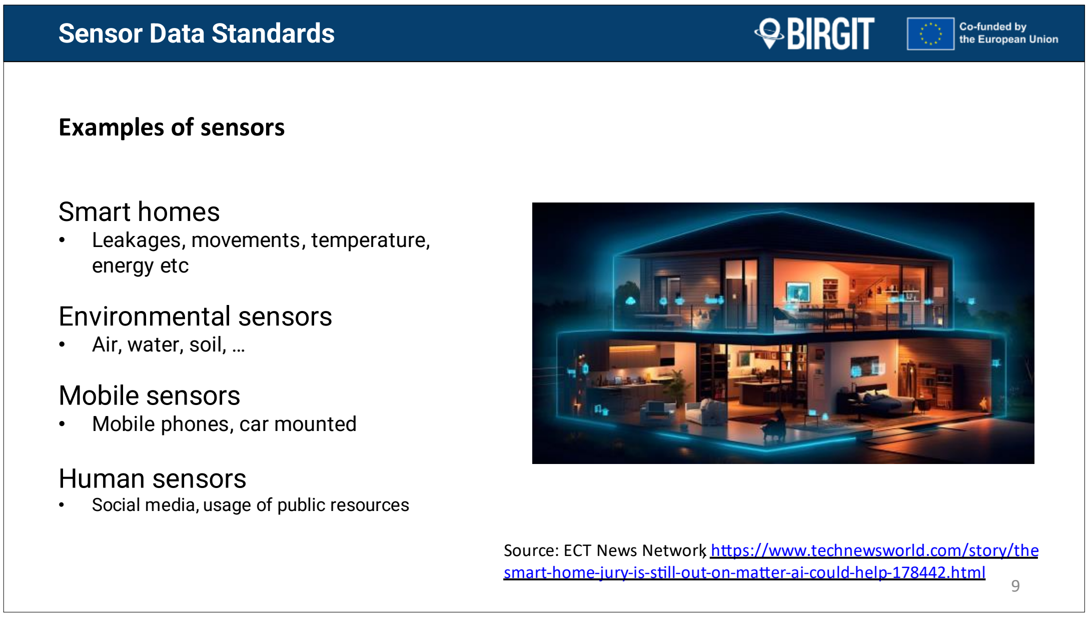
Image: Östra Sveriges Luftvårdsförbund.
https://oslvf.se/matningar_och_vaderstationer/

8

Another type of sensors are air quality sensors, which we will work further on in coming lectures. There is a global network of air quality monitoring stations, connected to a network. Each sensor is usually mounted in a box and placed at various locations by public authorities.

The specific sensor showed on this slide is in Södertälje, a city south of Stockholm, Sweden. The box has in this case only one type of sensor, namely to measure PM10 concentration (small particles with a diameter less than 10 micrometres). The box also has an AD-converter and equipment for transmitting data through the GSM network.

Examples of sensors



Here are some examples of the most common types of sensors. One purpose of this slide is to show the wide variety of sensors and sensor applications there are.

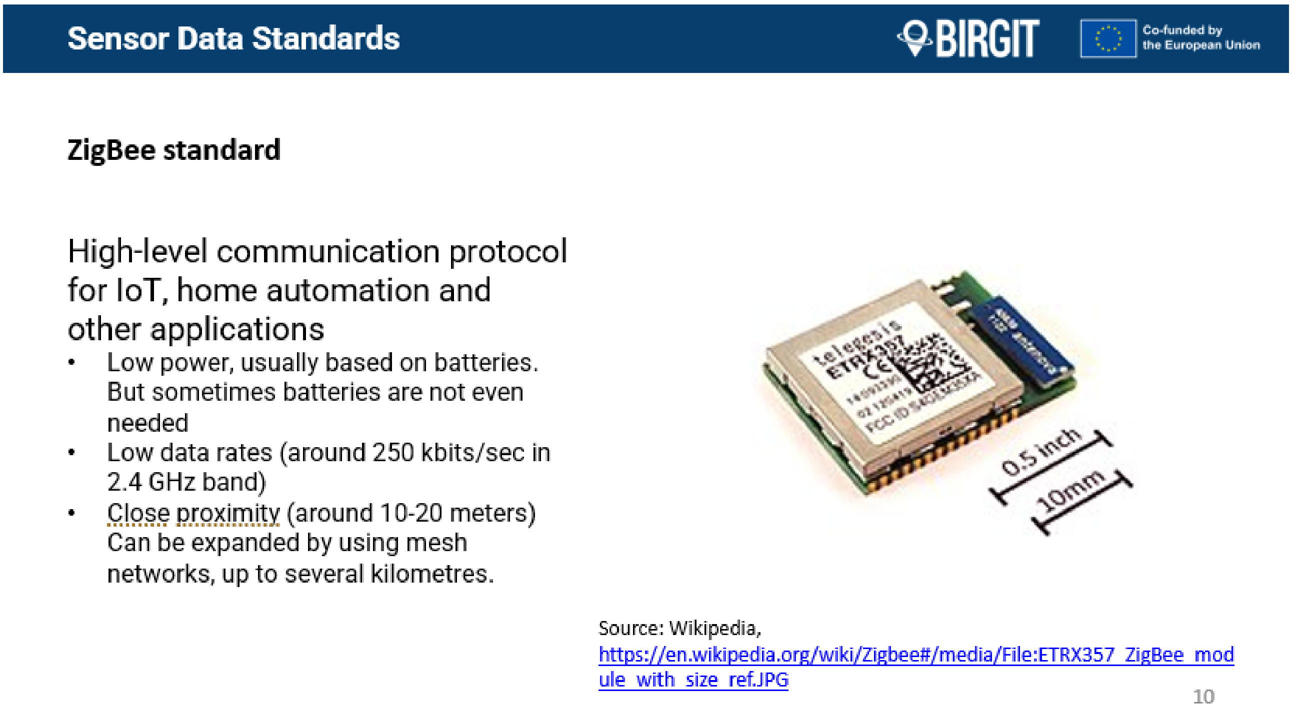
The smart homes sector is growing fast, and the sensors are supporting many different applications and needs, for instance detection of leakages in the water pipes, detecting movements indoors as well as outdoors, controlling room temperatures and energy consumption etc. In many cases, the sensors are connected to a central hub (access point) using a WLAN or cables.

The environmental sensors have been in use for several years. They were earlier based on manual readings, but now automatic readings are dominating. Many environmental legislations within the EU require some kind of monitoring of the environmental conditions, for instance the air quality monitoring program which we will discuss more in coming lectures.

A modern mobile phone is equipped with several different types of sensors, where the camera and the GPS may be the most used sensors. But there are also other sensors, like the accelerometer and a thermometer. A car is also equipped with sensors and these sensors also falls into the category of mobile sensors.

A fourth group of sensors often neglected are we humans, who observe or initiate events and conditions as we move around. As an example, local outbreaks of flu may be detected by an increasing usage of certain key words on social media (fever, flu, sick, ill, …). Another example is the usage of public services, for instance public transport, by the usage of ticket machines and access cards.

ZigBee standard



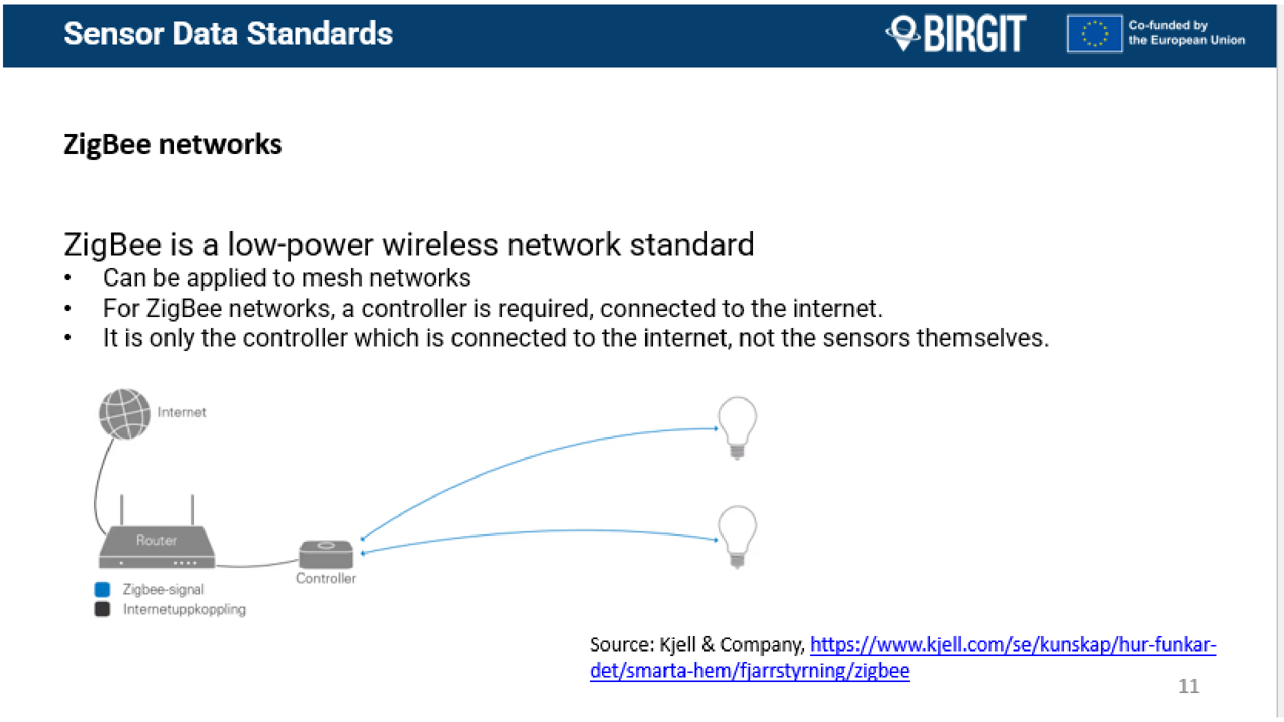
The first ZigBee standard was ratified in 2004. The most recent standard is ZigBee 3.0, which was released in 2017. The ZigBee standard address several new applications, for instance Internet of Things (IoT), smart homes as several other professional applications. Many sensor consumer products are based on ZigBee.

ZigBee networks are characterized by using low-power sensors and other units. This means that they operate on low power voltages. The image shows a unit where a ZigBee unit is mounted on top of a GPIO card. This unit may be driven by 3-4 1.5V batteries.

ZigBee units often communicate on the 2.4 GHz band. This limits the data transmission rate to around 250 kbit / sec. This limitation may cause restrictions on the topology of the sensor networks.

A third limitation is that the distance for communication is limited, to around 10 – 20 meters, but sometimes longer. But as we can see in the next slide, this distance may be extended by simple means.

ZigBee networks



ZigBee devices can also be configured into networks. This means that there is a controller, which handles the communication with the internet as well as with the ZigBee devices.

But we have a proximity limitation of around 20 – 100. To increase this distance, we can apply the mesh topology. This means that we can have intermediate nodes which receives data and transmit data to other nodes. In this way, mesh networks can be quite large, although the distance between each node may be just 20 meters.

But still, external access is made through the controller. The communication within the ZigBee network may then be encrypted and not accessed by external parties.

Accessing sensor networks

Sensor Data Standards BIRGIT Co-funded by the European Union

Accessing Sensor networks

- HTTP (Hypertext transfer protocol is the foundation of data communication for the World Wide Web
- HTTP GET is the most used HTTP requests. It is used for retrieving data. The request may be initiated by a mouse click on a hyperlink.
- There are also other HTTP requests, for instance POST, PUT, DELETE etc.
- A HTTP GET request can easily be issued using Python

```
import request

# The API endpoint
url = "https://mySensorNetowork.com/sensors"

# A GET request to the API
respone = requests.get(url)
```

10

The **Hypertext Transfer Protocol** (**HTTP**) is an application layer protocol in the Internet protocol suite model for distributed, collaborative, hypermedia information systems (Fielding, Nottingham, Reschke, 2022. HTTP Semantics. https://datatracker.ietf.org/doc/html/rfc9110). It is the foundation of data communication for the World Wide Web, where hypertext documents include hyperlinks to other resources that the user can easily access, for example by a mouse click or by tapping the screen in a web browser.

HTTP GET is the most used HTTP request. It simply sends such a request to the URL specified in the call, and the responding server returns the content of that URL. This request is for instance issued when you click on a hyperlink on a web page.

HTTP GET requests are often used also when accessing sensor networks. One problem is often what to do with the response you get. In many cases, some kind of programming is required.

The Python code given on this slide shows how to access a sensor network. The first statement says simply that you need to import a standard library facilitating the usage of HTTP requests. The next two statements specify the URL to be accessed and then finally a request is issued and the response is stored in the variable named “response”. The upcoming questions is then how the URL and the response should be structured. To simplify our work, we need standards.

Web API's

Sensor Data Standards BIRGIT Co-funded by the European Union

Web API's

- Remote Procedure Calls (RPC)
  - OGC Sensor Observation Standards
  - url=http://myAccessPoint?service=SOS&version=2.0.0& …
- Representational State Transfer (REST)
  - OGC SensorThings Standards
  - url=http://myAccessPoint/sensor
- Simple Object Access Protocol (SOAP)
  - Submitting complex messages using XML

11

The term "API" stands for Application Programming Interface and the intention is to provide uniform ways to access web resources from programming environments like Python and C#. The web API's are adding an addition layer on top of the HTTP standard.

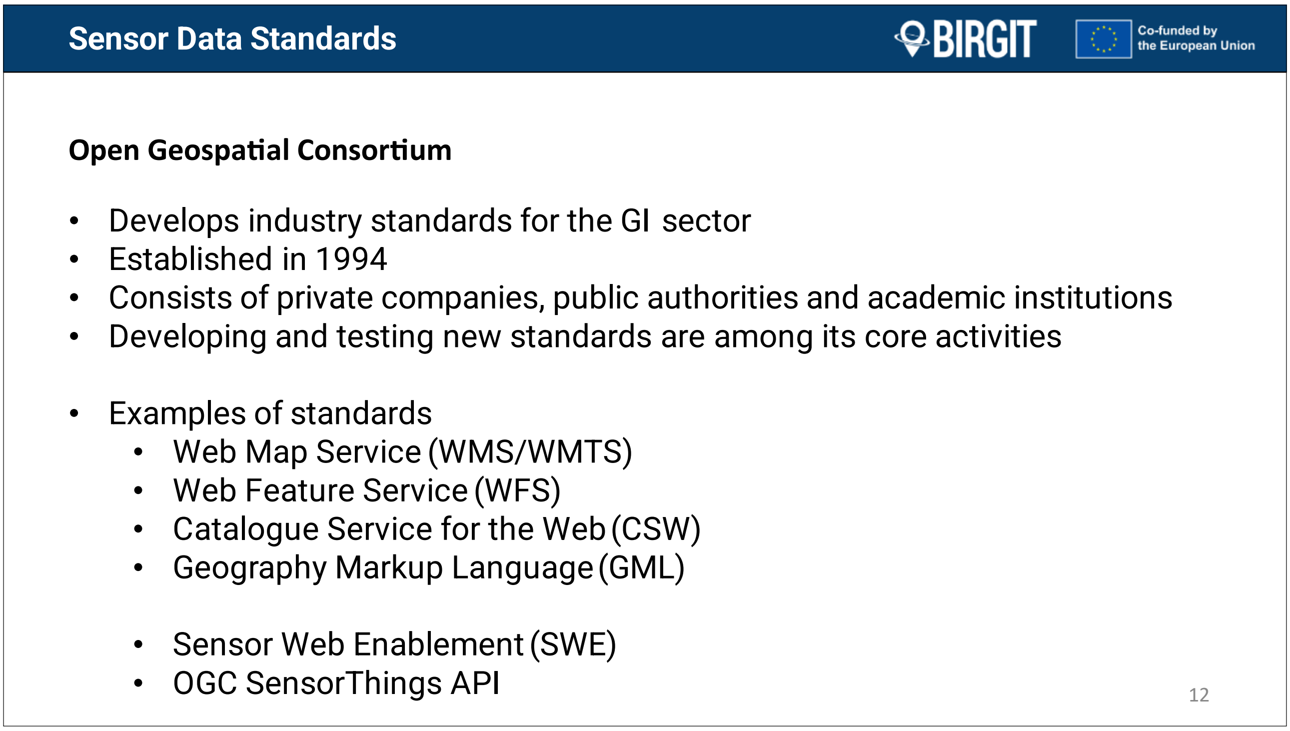
The OGC SWE (Sensor Web Enablement) standard suite is mainly based on RPC and SOAP (optional). The RPC calls have a question mark at the end of the URL of the remoted procedure. Then additional parameters may be given by key value pairs (parameter name and parameter value). This allows for quite complicated queries to the remote procedure. Many of the early OGC standards like WMS and WFS are based on RPC's.

REST API's have received increased attention during the last years. It has some advantages, as compared to RPC and SOAP. One reason is that it simplifies the development of different mobile applications, at least to some extent.

The distinction between RPC's and REST calls are not that always clear. RPC's can comply with the REST paradigms and REST calls can access the resources through RPC's as well. So, the line between these two principles is really blurred.

Some sensor standards, such as the Sensor Web Enablement suite, also have a SOAP binding. Despite its name, it is not that simple as RPC's and REST calls. Instead, one have the possibility to include a lot of information in the call, by using XML documents, allowing for complex queries and other operations.

Open Geospatial Consortium (OGC)

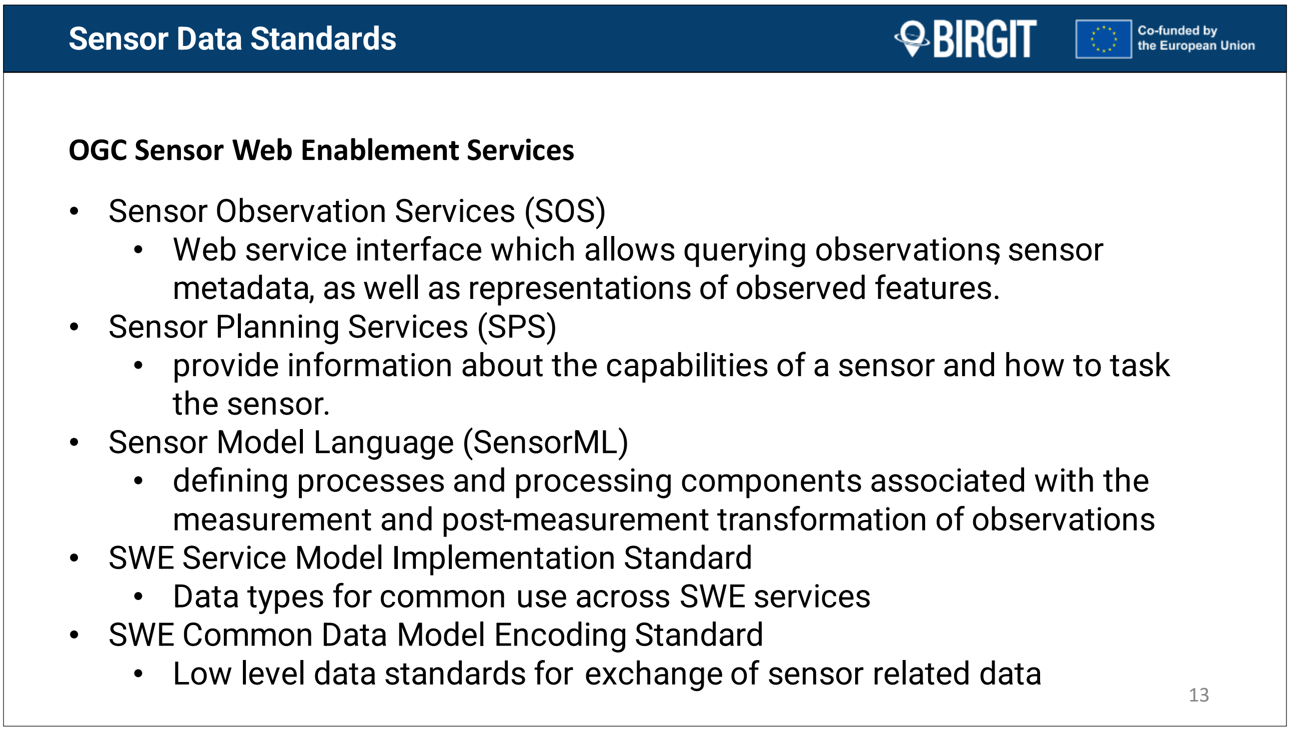


We will now look into the sensor standards often used in the GI industry. The Open Geospatial Consortium is the organisation specifying de-facto industry standards in the GI domain. It was established in 1994 and its members include private companies (software vendors, consultants etc), public authorities and academic institutions. Its aim is to specify technical standards that meet the needs of the GI industry.

There are also other standardisation bodies like ISO and CEN. They develop formal standards, which can be referenced by legislations. ISO acts on the international level while CEN acts at the European level. OGC has a close collaboration with these formal standardisation bodies, as well as with other developers of de-facto standards, such as W3C (World Wide Web Consortium). One large difference between OGC and ISO/CEN is that OGC also test the standards, aiming to ensure that the standards work in practice. The ISO/CEN processes are in principle limited to a paper-review process. Consequently, the overall procedure is then that OGC develop and test industry standards, which after a while become adopted by ISO and CEN. It should also be mentioned that the OGC standards are freely accessible, while the ISO and CEN standards are documents to be paid for. It should also be mentioned that there are some ISO standards that are not covered by OGC, for instance standards on metadata profiles and data quality.

In this lecture, we will discuss two different OGC sensor standards, namely the Sensor Web Enablement suite and the OGC SensorThing API.

OGC Sensor Web Enablement Services

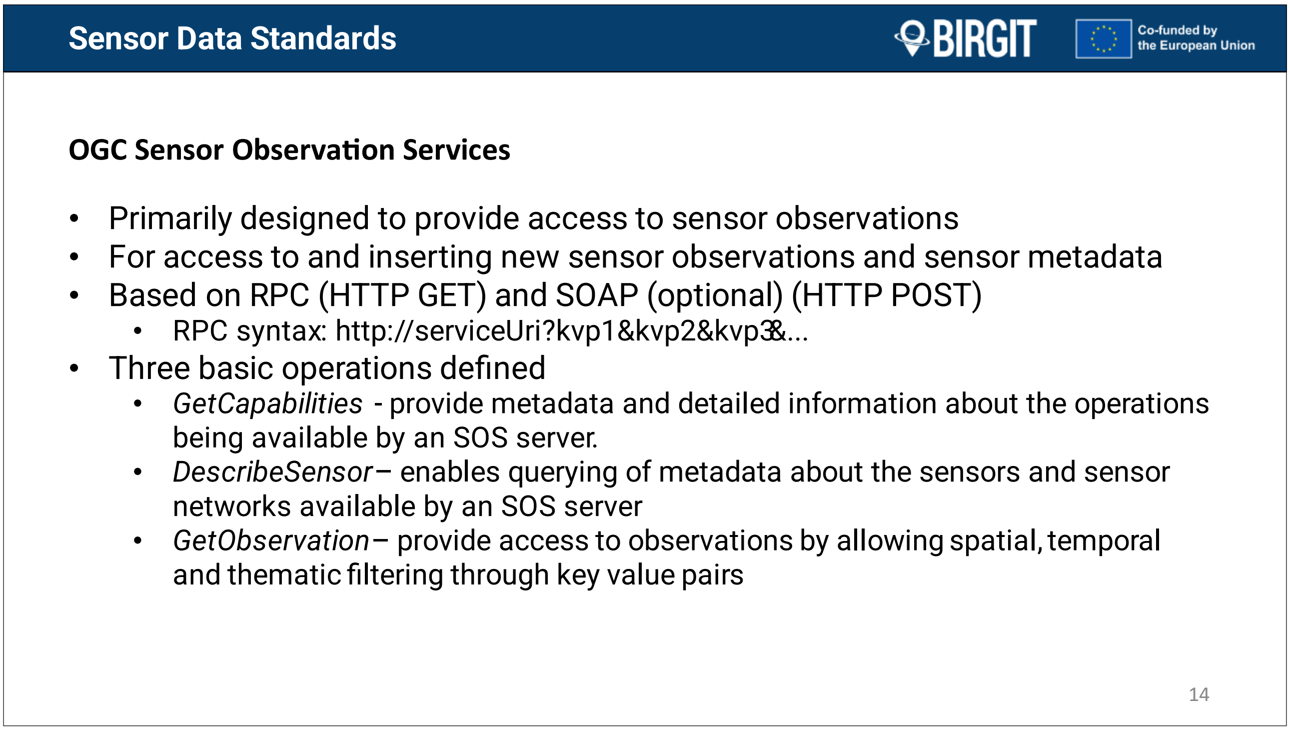


The OGC Sensor Web Enablement suite of standards consists of 5 different standards. The most used standard is the Sensor Observation Service standard (SOS), which query the sensors of the sensor network for most recent or historical data.

The Sensor Planning Service (SPS) is more aimed at the management of sensor networks, for instance for optimization purposes, while Sensor Model Language (SensorML) is a XML-based profile specifying the semantics of sensor characteristics and observations. In addition to that, we also have some low-level standards, specifying common data types and data models.

In this lecture, we will mainly discuss the SOS standard.

OGC Sensor Observation Services (SOS)



The SOS standard is primarily designed to provide access to sensor observations. It is based on the usage of HTTP GET and as optional also HTTP POST, through the SOAP binding.

The simplest way of accessing sensor observations is by using Remote Procedure Calls (RPC's) which are based on HTTP GET. A RPC consists of the URI to the service followed by a question mark. After the question mark, several KVP's (key value pairs) may be specified, giving additional information to the remote procedure. A KVP is a combination of a key and a value, for instance REQUEST=GetObservation.

The SOS standard may also be used for inserting new observations, new sensors and new sensor metadata. Since sensor metadata may be quite voluminous, the SOAP binding may be easier to use for this type of operations.

A SOS service shall provide at least three basic operations, namely GetCapabilities (an operation needed for all OGC RPC-type of standards, like WMS, WFS and so on),0 DescribeSensor which queries for sensor metadata and GetObservation, which query for observation data. The GetObservation operations also allows for filtering, where spatial filters (bounding box), temporal filters (before, between, after) as well as other filters may be used.

HTTP GET Request Example

Sensor Data Standards BIRGIT Co-funded by the European Union

HTTP GET request example

```
#          procedureID = sensorID

sosEndPoint = ”https://OurApiEndpoint?”
swesCommon = "service=SOS&version=2.0.0&"
getObservationString = "REQUEST=GetObservation&procedure="
+ procedureID + "&responseFormat=application/json"

sosUrl = sosEndpoint + swesCommon + getObservationString
response = requests.get(sosUrl)
```

15

In Python, an RPC request can be built up as follows.

The sosEndPoint variable is here set to the URL of the service.

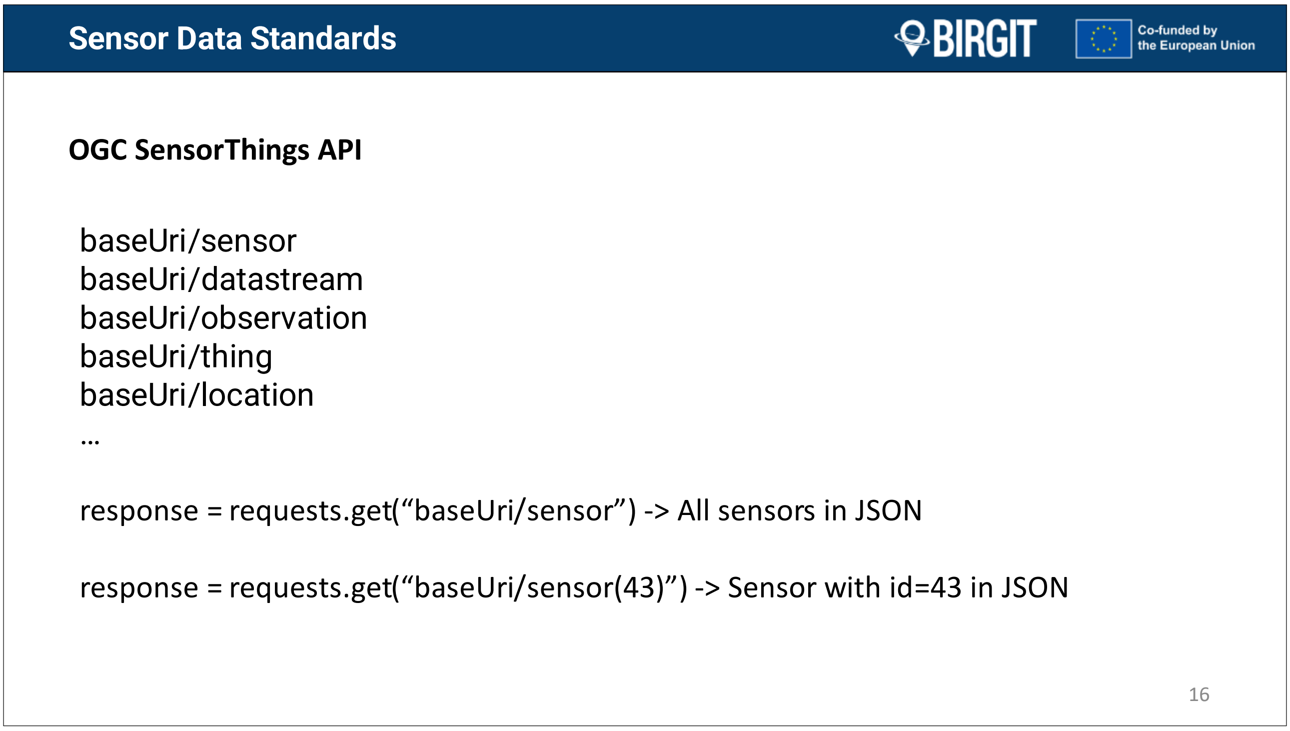
The names and meaning of the key value pairs are defined in the SOS standard. The swesCommon variable is here set to the KVP’s that are common for all SWE operations.

The getObservationString is then set to the additional KVP’s to be added to our RPC. Of special interest may be the responseFormat KVP, where we state that we wish to receive the data in json format. This requires of course that the service can respond in json format, something which is mentioned in the service metadata received by the GetCapabilities operation.

The term “procedure” is here used to specify the ID-value of each sensor.

Then finally, the entire URL is created by adding all these three strings together and then sent to the SOS server by the requests.get method.

OGC SensorThings API



The OGC SensorThings standard is a newer standard and base on the modern usage of REST API's. We will here look at these REST API's from an application programming perspective, that is someone aiming to use a specific web resource in a Python program.

One way of looking at the REST API's is to view them as web pages, structured in a hierarchical manner. In case you are looking for data in json format, you may consider each resource as a web page consisting of one json file.

In the example on this slide, we assume we have a database with information about sensors in a sensor network. The characteristic of a certain sensor is described by the subfolder "/sensor". Other subfolders are then used for accessing data streams and single observations.

But this is only the way we look at the data from the usage perspective. Often, all information is stored in one database, but the architecture of our system is decoupled from the access to the resources. This means a possibility to change the system architecture without having to change all applications accessing the resources.

Co-funded by the European Union

BIRGIT

Handling sensor networks in QGIS

Sensor Data Standards

BIRGIT

Co-funded by the European Union

Handling sensor networks in QGIS

- There is currently no support for SWE services in QGIS
  - Python scripting required
  - Pre-analysis of service metadata required
- Plug-in for SensorThings API available in QGIS
  - Very few such services exists

17

A QGIS plug-in for SWE services was earlier available, but since the latest upgrades of the software, such plug-ins are not available any longer. This means that to be able to access SWE services, some programming in Python is required (or another suitable language). This also means that an analysis of the service metadata is required before the Python scripts can be developed.

Recent developments include however a plugin for the SensorThings API. Unfortunately, since the standard is so new, very few services exist today (October 2023) which use the standard.

Co-funded by the European Union

BIRGIT

References

Fielding, Nottingham, Reschke, 2022. HTTP Semantics. https://datatracker.ietf.org/doc/html/rfc9110

OASIS, 2020. OData version 4.01. Part 2: URL Conventions. https://docs.oasis-open.org/odata/odata/v4.01/odata-v4.01-part2-url-conventions.pdf

OGC, 2012. OGC Sensor Observation Service Interface Standard, OGC 12-006. https://www.ogc.org/standard/sos/

OGC, 2021. OGC SensorThings API Part 1. Sensing Version 1.1. https://docs.ogc.org/is/18-088/18-088.html

Stowe M, 2015. Undisturbed REST. https://www.mulesoft.com/lp/ebook/api/restbook.

Ullo S.L, Sinha G.R, 2020. Advances in Smart Environment Monitoring Systems Using IoT and Sensors. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7309034/