

SUPSI

Managing and Sharing Sensor Time-Series Data with istSOS

Massimiliano Cannata, Milan Antonovic



Content

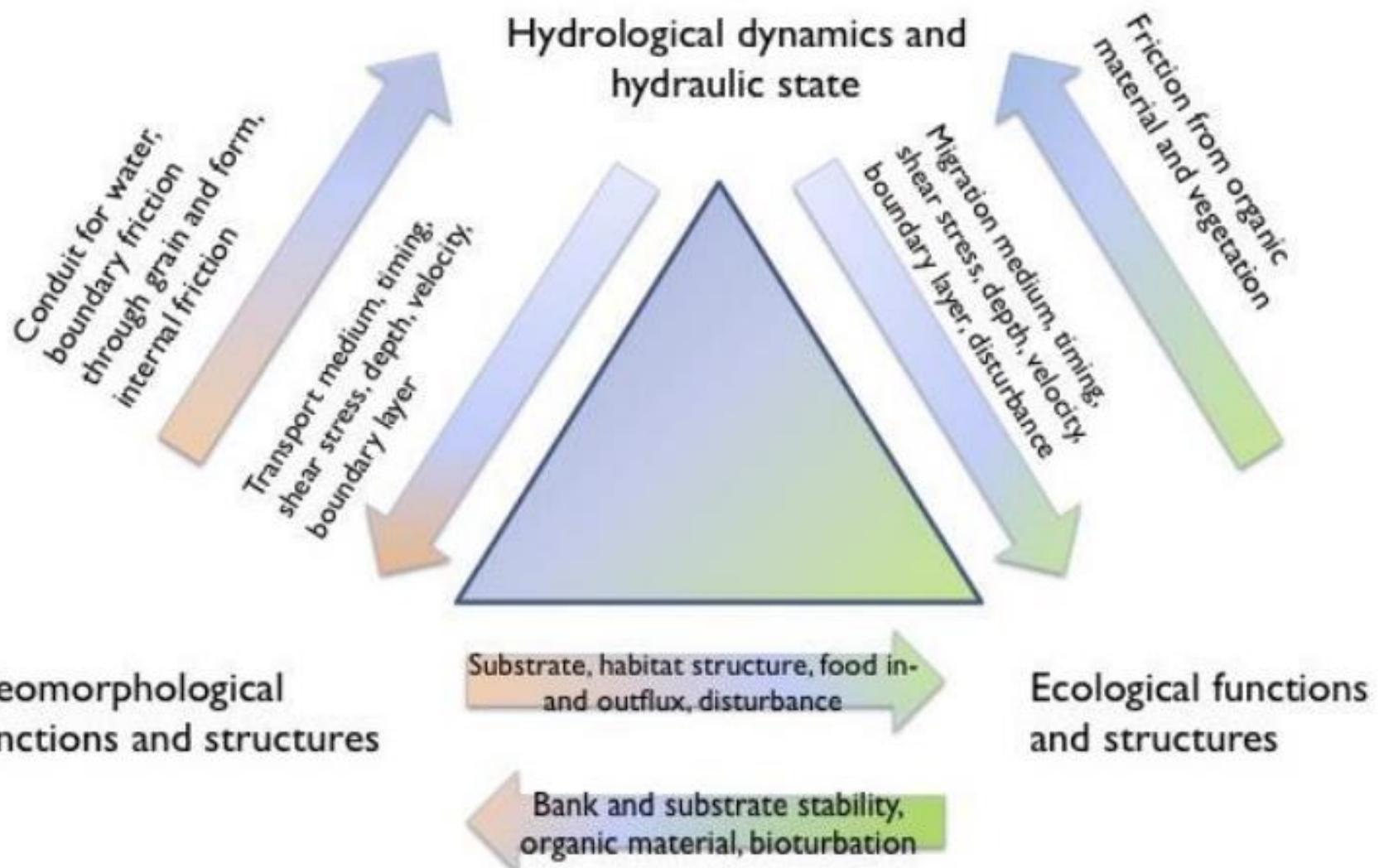
- Why do we need to monitor?
- The Locarno example
- HydroMet data monitoring system of Canton Ticino
- The application for the civil protection to support decision making
- istSOS: a SOS compliant software with special features

Why do we need to monitor?

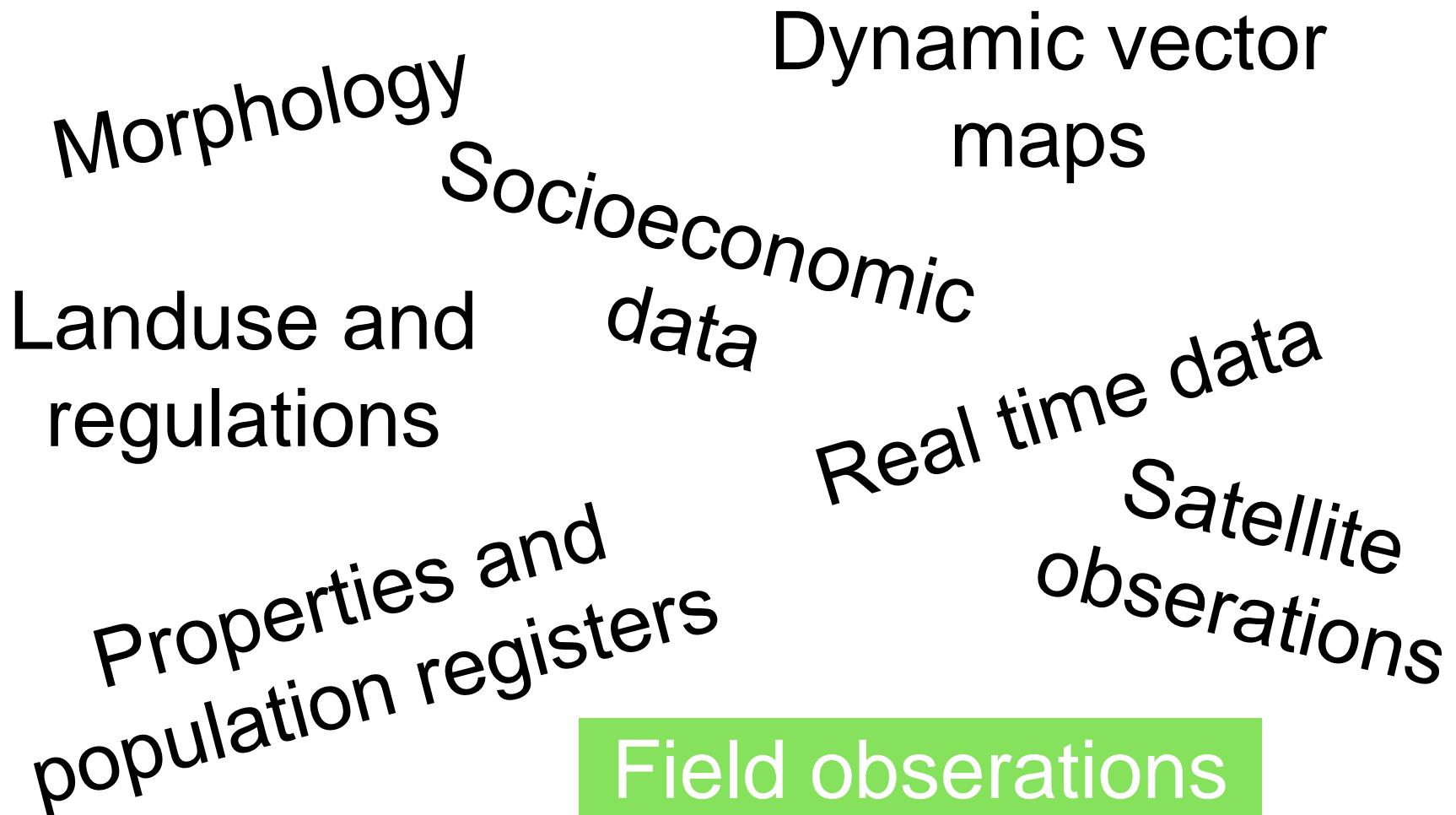
Solving societal challenges

Strengthening data production and the use of better data in policymaking and monitoring are becoming increasingly recognized as fundamental means for development

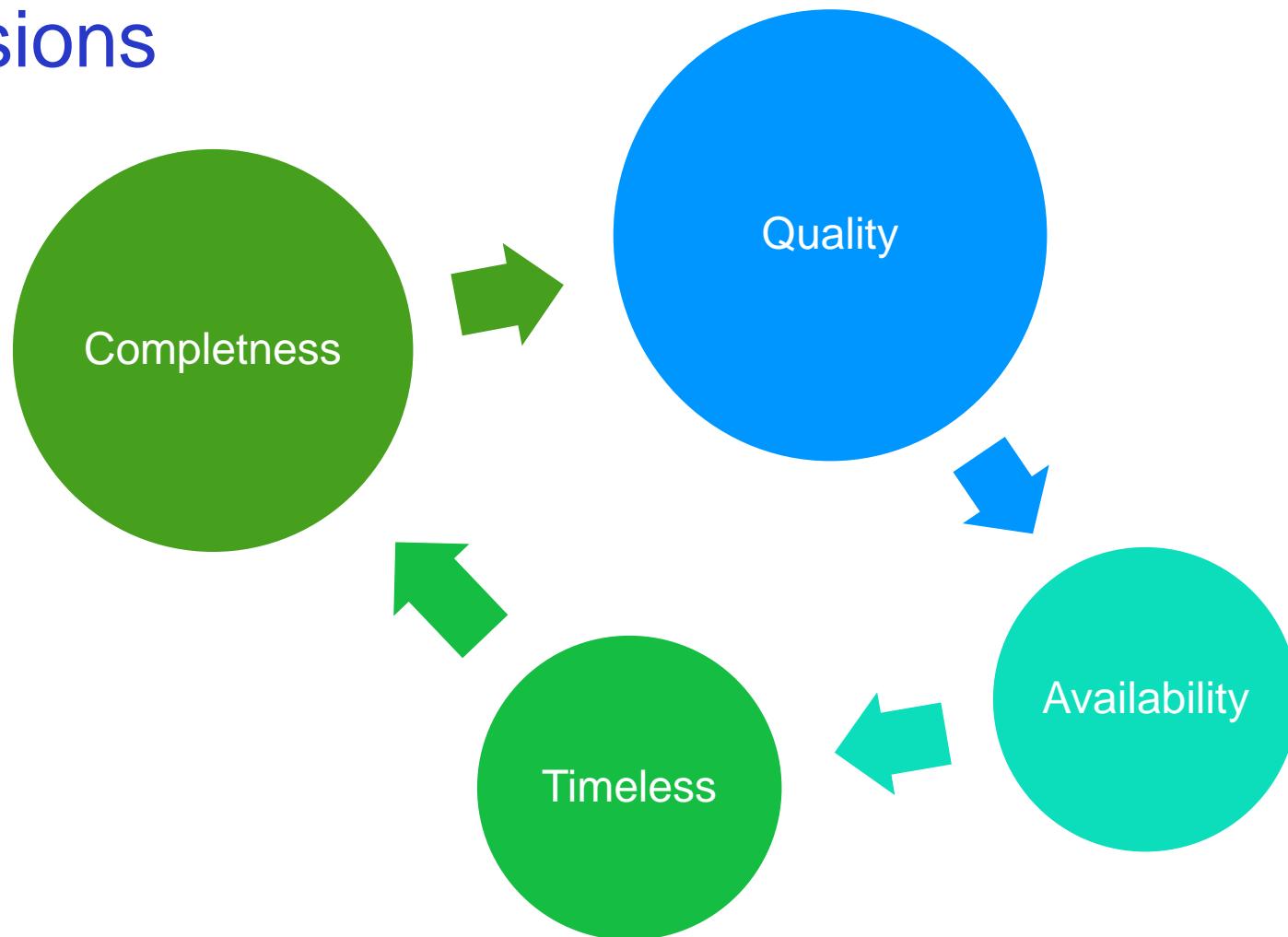
Water monitoring to understand the system



Understanding by (spatially) integrating multiple information source



Understanding data is a key aspects for wise decisions



Understand the situation and timely react

The Locarno example



Lake Verbano shores are exposed
to flooding events that cause
extensive damages

Locarno: almost flat area highly urbanized



Intense rainfall events compared to maximum discharge ($2,000 \text{ m}^3/\text{s}$)

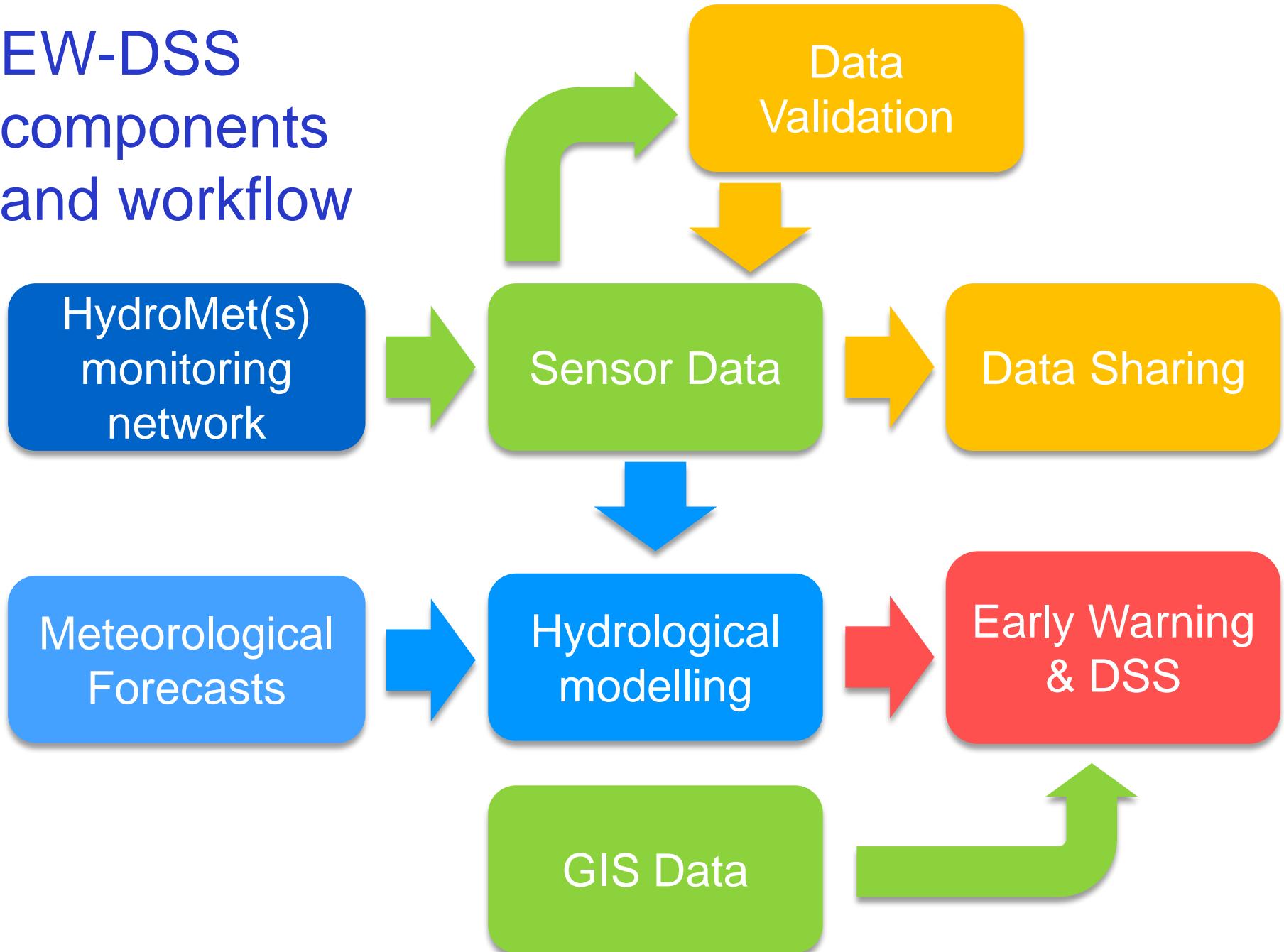
Large basins ($6'386 \text{ km}^2$) compared to limited lake area (213 km^2)

How to manage the risk

Building an Early Warning and
Decision Support System



EW-DSS components and workflow





HydroMet data monitoring system of Canton Ticino

Open Source Technologies

Open standard to
foster interoperability



Free and Open Source Software to
guarantee durability and flexibility



Past experiences
proprietary solutions showed
limitations that killed us 😞

Management of the HydroMet monitoring network of Canton Ticino, southern Switzerland

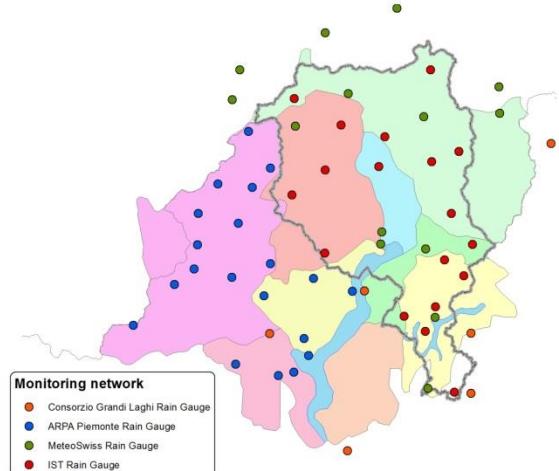


Real time data collection

In 2010, switch from analogue modem (collection every 4 hours) to GPRS sensor data transmission (real time)



Needs for smarter system for collecting and dispatch measurements



Looking for a solution...

Simple
Standard
Open
(possibly Python... shhh!)

The Sensor Observation Service standard

From <http://www.opengeospatial.org/standards/sos>

“The Sensor Observation Service standard is applicable to use cases in which sensor data needs to be managed in an interoperable way. This standard defines a Web service interface which allows querying observations, sensor metadata, as well as representations of observed features. Further, this standard defines means to register new sensors and to remove existing ones. Also, it defines operations to insert new sensor observations. This standard defines this functionality in a binding independent way; two bindings are specified in this document: a KVP binding and a SOAP binding (in version 2.0).”



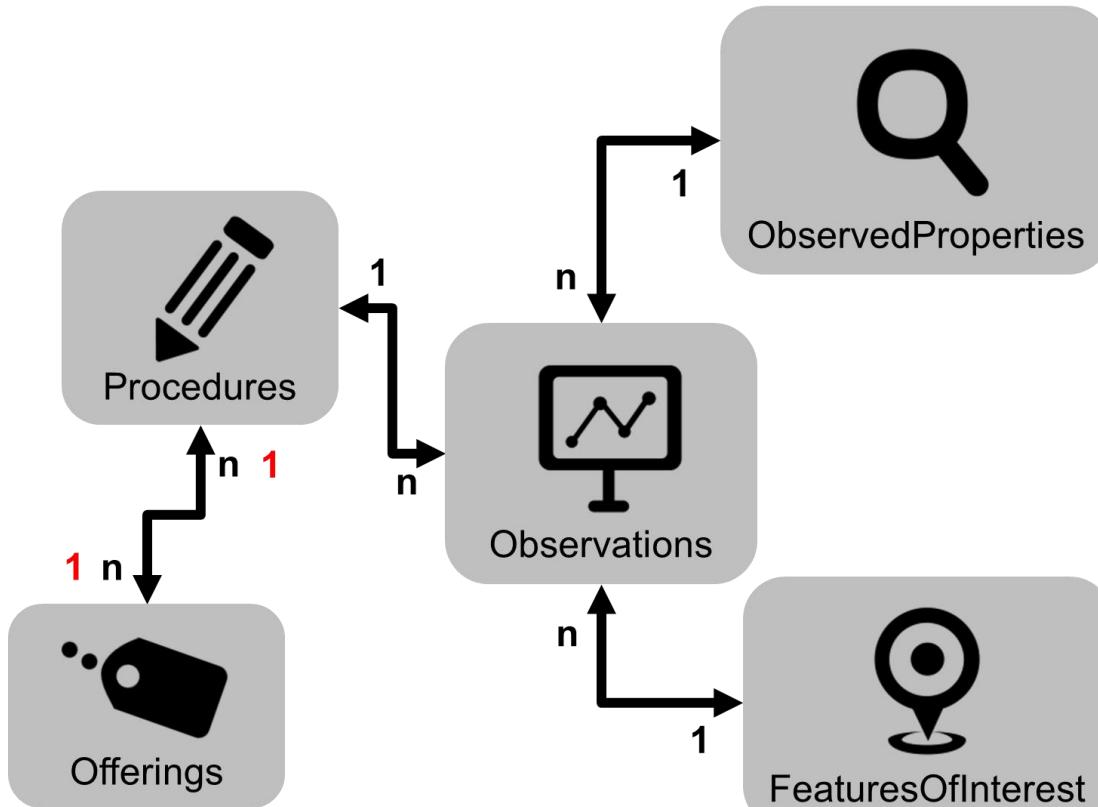
sos in the context of INSPIRE



Alexander Kotsev
www.jrc.ec.europa.eu

*Serving society
Stimulating innovation
Supporting legislation*

SOS (v1.0, v2.0)

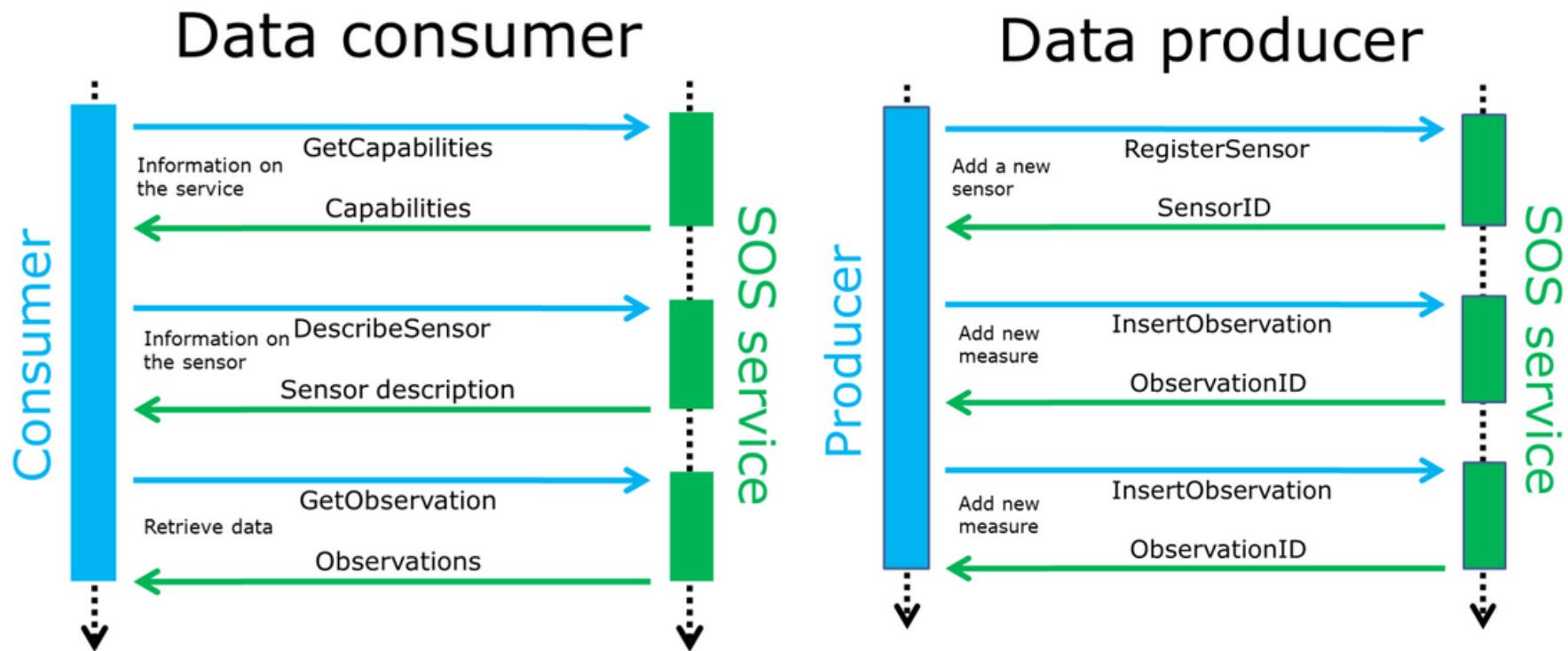


From system to users

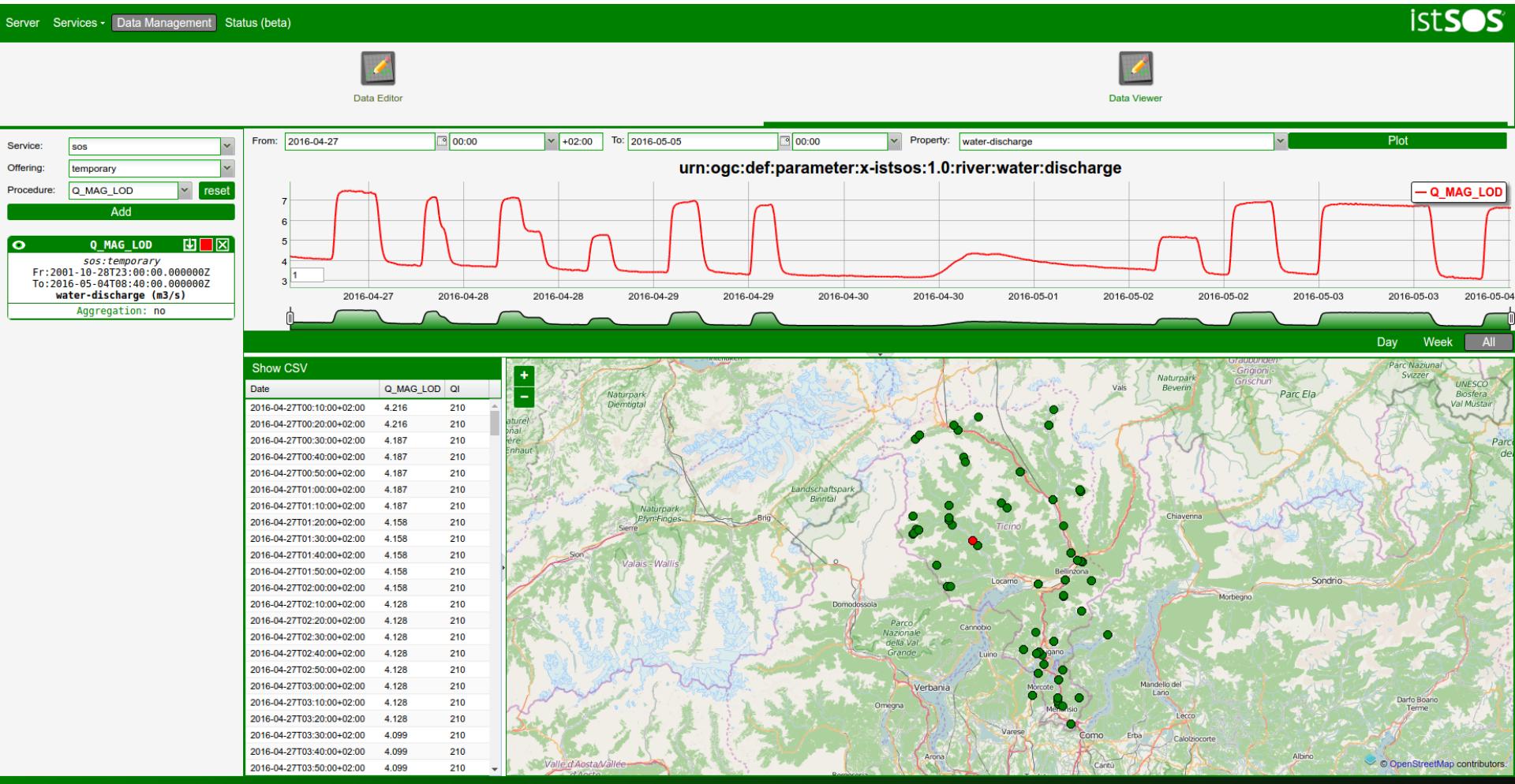
- GetCapabilities
- DescribeSensor
- GetObservation

- From sensors to system
- RegisterSensor
 - InsertObservation

SOS User types



istSOS service for Canton Ticino HydroMet



151 registered sensors

15 observed properties

(air-temperature, air-rainfall, water-height, water-height, air-humidity, water-discharge, water-height, air-pressure, air-radiation, water-conductivity, water-temperature, water-temperature, battery-tension, water-tension, air-relative_humidity)

40 years of data (1976-2016)

88.5 Mio registered observations

32 GB of database

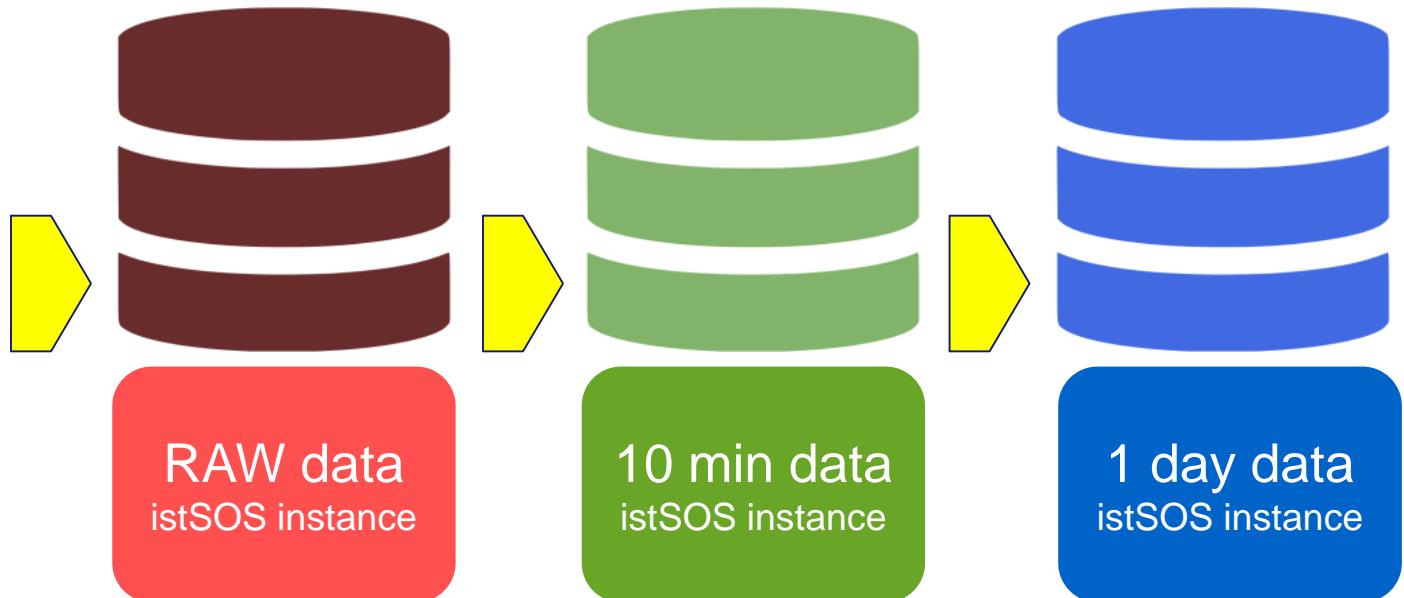
8 GB of data served in April

(4.5 GB with SOS requests & 3.21 GB with istSOS RESTful API)

5.7 Mio served requests in April

1 Internal server error response (500) in April

istSOS service for Canton Ticino HydroMet



OBS: 38 Mio
STAT: 151
SIZE: 19 GB
SOS: 1.4 GB
WA: 2.45 GB
Req: 2.4 Mio
Years: 26

OBS: 49,6 Mio
STAT: 151
SIZE: 13 GB
SOS: 3.1 GB
WA: 0.76 GB
Req: 3.3 Mio
Years: 26

OBS: 0.9 Mio
STAT: 151
SIZE: 184MB
SOS: 10 MB
WA: 3 MB
Req: 190
Years: 40

The application for
the civil protection to support
decision making

EWS components with open standard and open software

Hydro-meteo monitoring network
Python scripts
ASCII to SOS

Meteorological Forecasts
PostGIS & WPS

GIS Data
Geoserver WMS / WFS

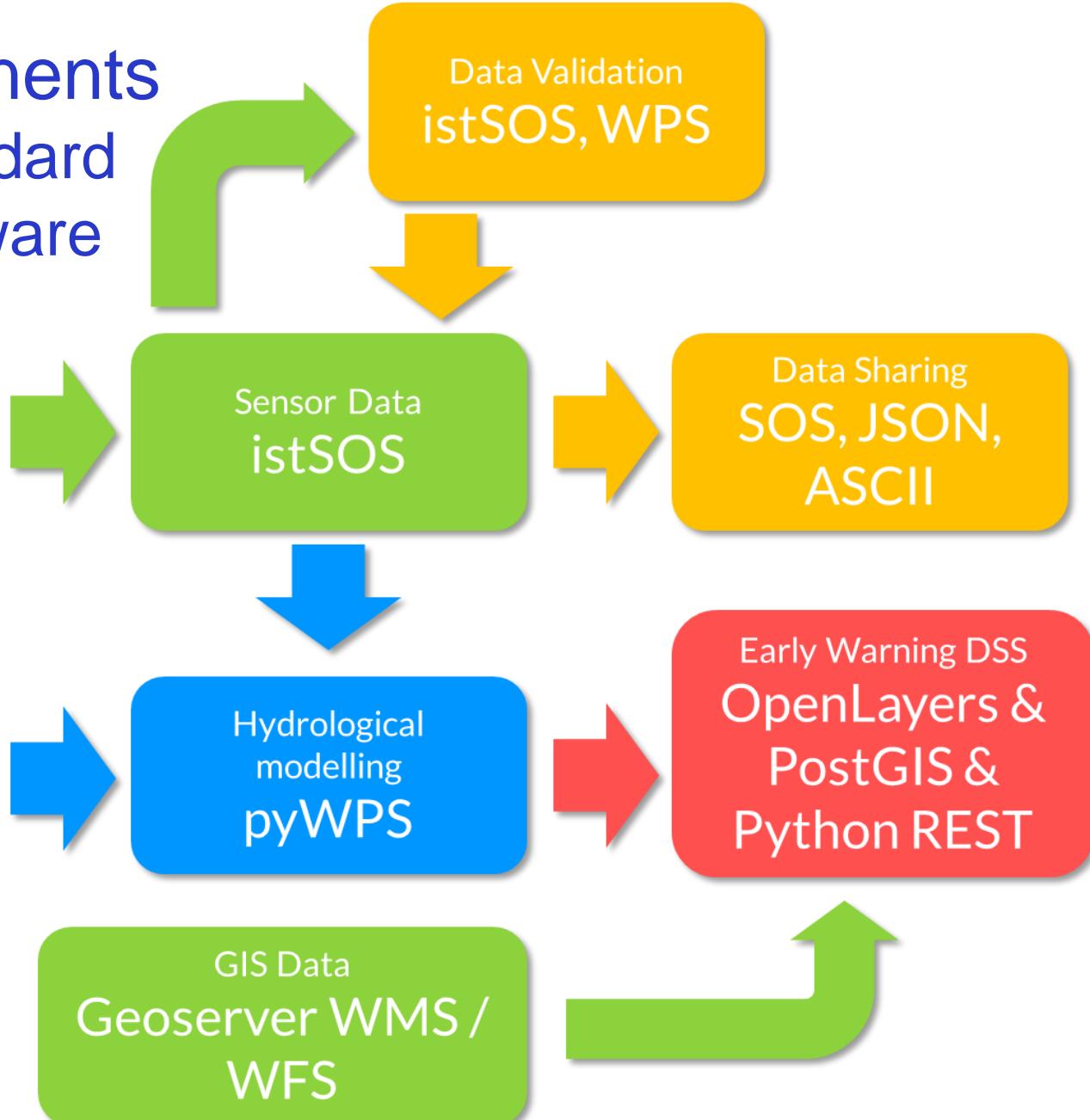
Data Validation
istSOS, WPS

Sensor Data
istSOS

Data Sharing
SOS, JSON, ASCII

Hydrological modelling
pyWPS

Early Warning DSS
OpenLayers &
PostGIS &
Python REST

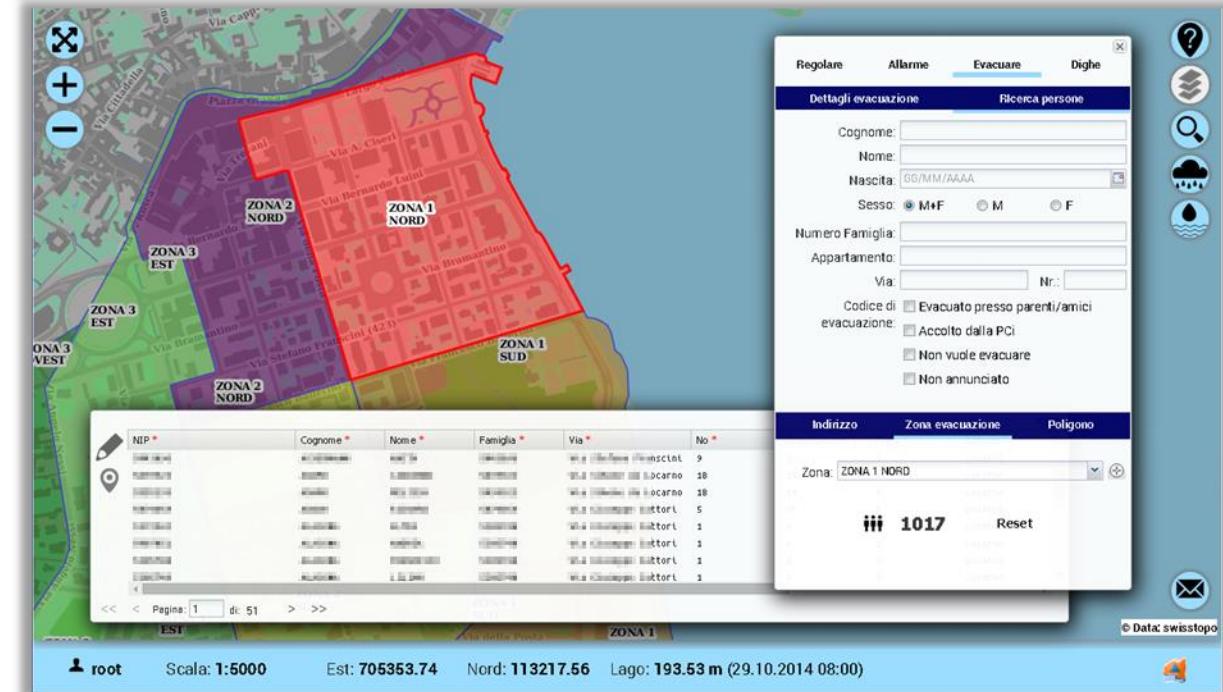
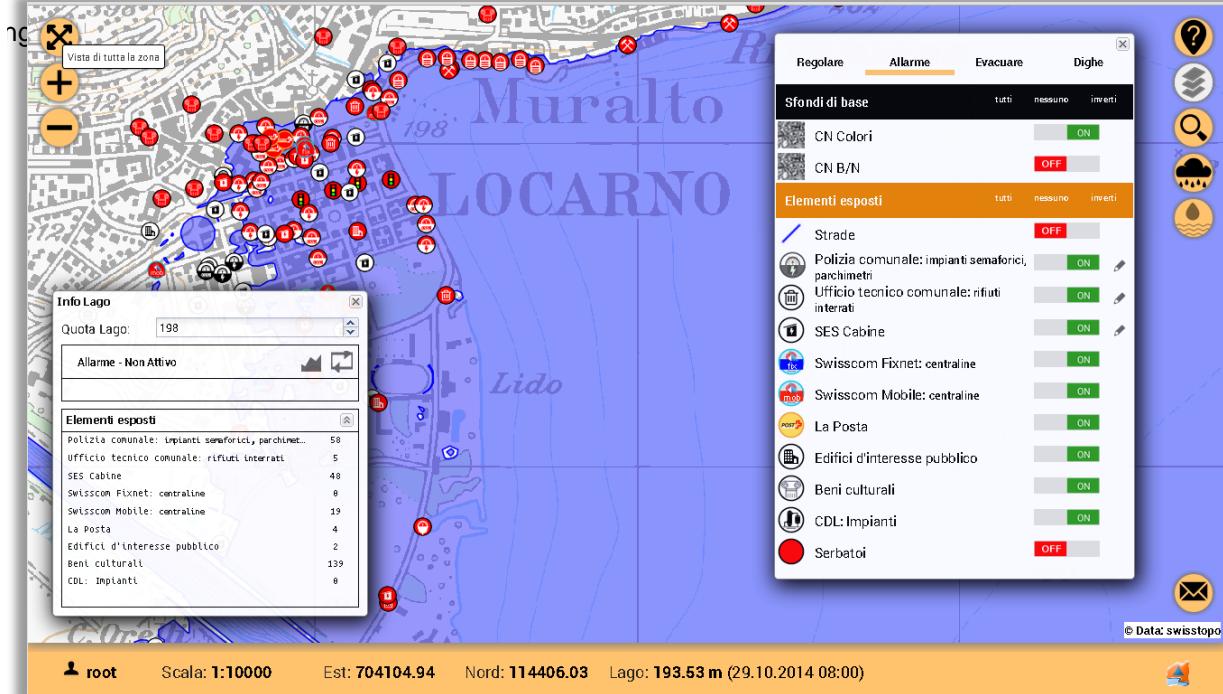


Mode «Regular»: access to data and funtions for resources management (search for parcel, address or person, query and eventually edit elements details, etc.)

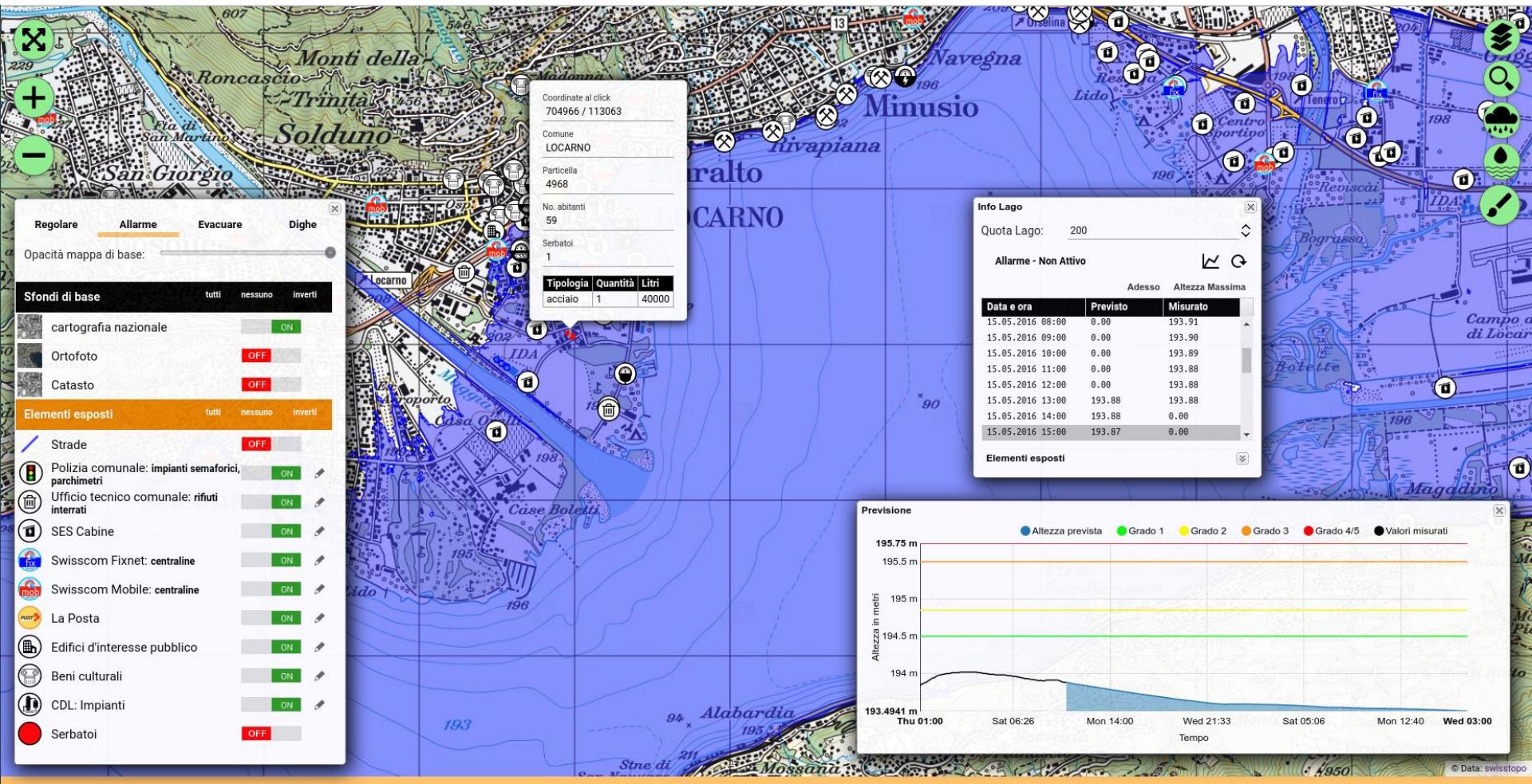
Mode «Alarm»: access to forecasted lake levels and exposed elements data (detect exposed elements with a given lake level, see plots of forecasted lake level, access instructions to secure exposed elements, etc.)

Mode «Evacuation»: access to details of population to manage evacuation (create evacuation event, search and locate person, count people in an area, assign people an evacaution status, etc.)

Mode «Dams»: access to dam-break hazard zones and wave arrival times



Access protected for data sensibility



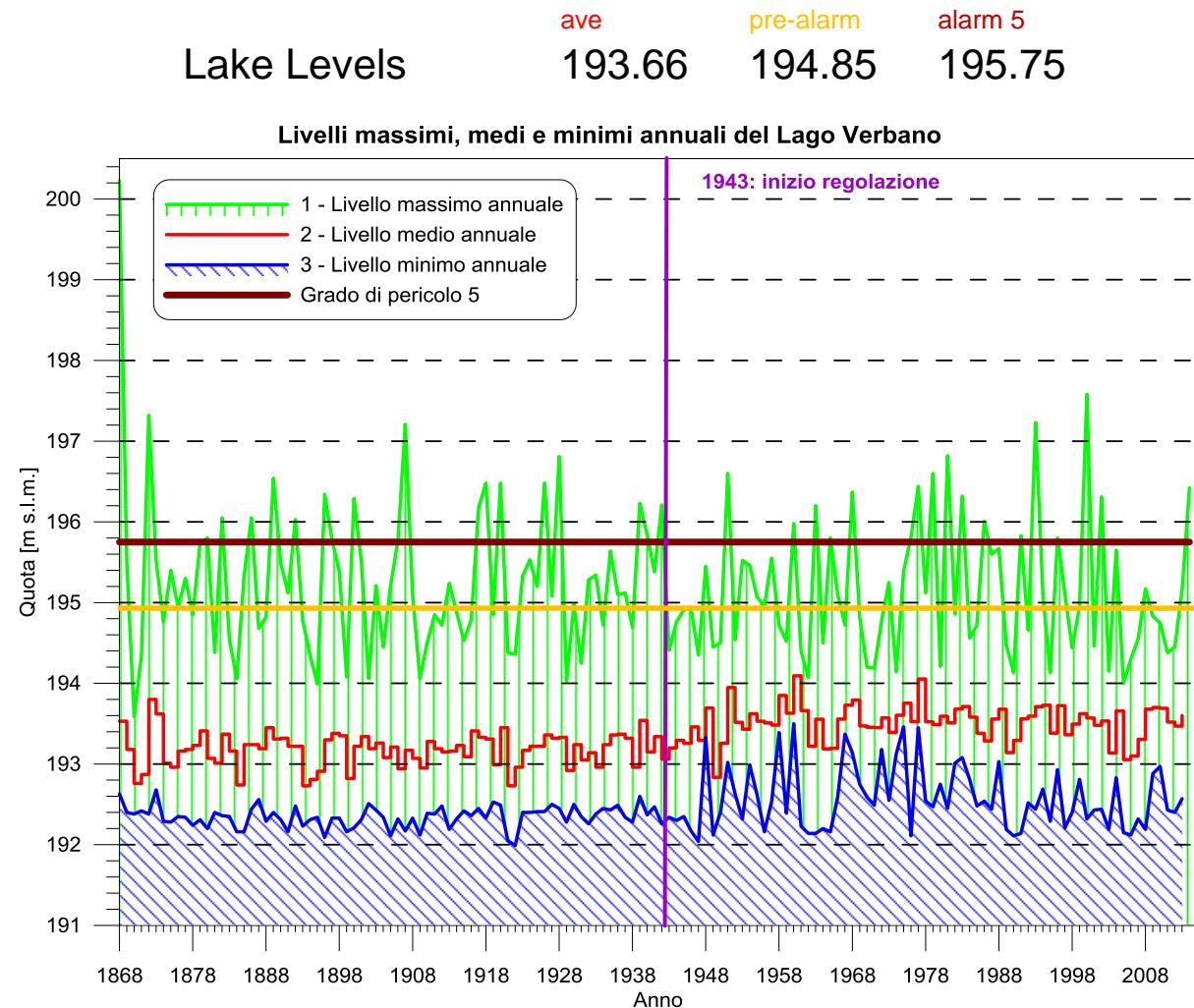
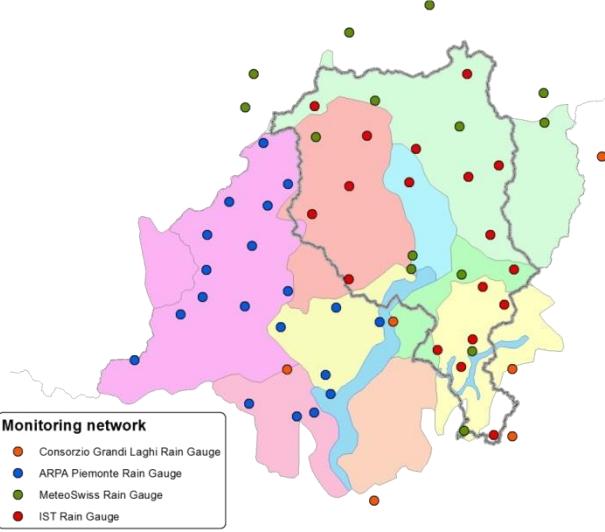
Lake Verbano events

Basin = 6,386 km²

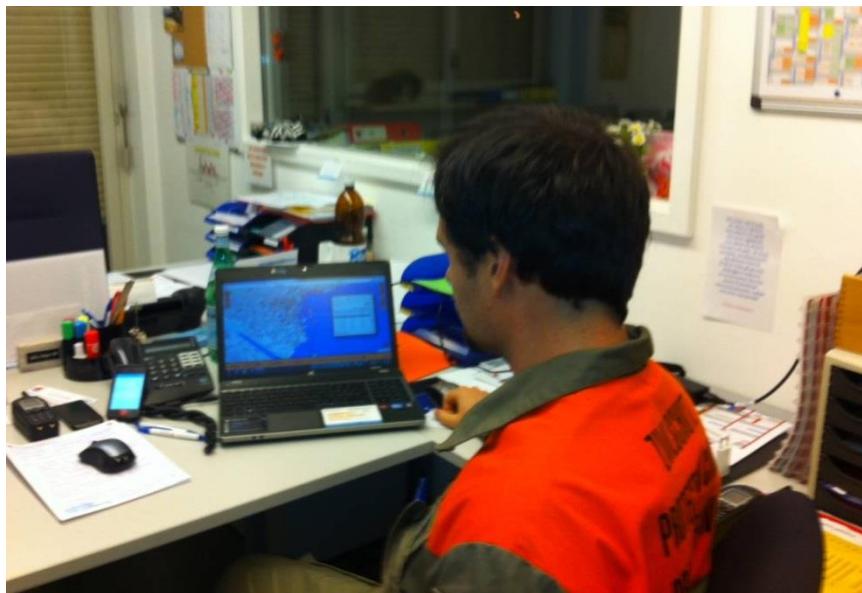
Lake = 213 km²

Discharge

max = ca. 2,000 m³/s



Flood of the 12-21.11.2014 max lake level 196.42 (alarm level 5)



81% 07:08

Dal Quotidiano:

10.11.2014: TORNA IL MALTEMPO, SI ATTENDE L'ESONDAZIONE DEI LAGHI

A screenshot of a mobile phone displaying a news article from 'Dal Quotidiano'. The article is dated 10.11.2014 and discusses the return of bad weather and the expected flooding of lakes. Below the text is a small video frame showing a reporter with a microphone interviewing a man in a green and red outdoor vest. The background of the video frame shows a map and some office equipment.

SITGAP2: Average Daily Statistics during the last flood

Unique Visitors: **50**
 Requests / Second: **2.5 - 30**

REQUESTS	Requests	Bandwidth
SITGAP Web Services	25'372	0.83 GB
WMS	190'112	2.57 GB
Total	215'484	3.4 GB

REQUESTS	%	ERROR	CODE
205132	95.20	OK - The request sent by the client was successful	200
9568	4.44	Not Modified - Resource has not been modified	304
401	0.19	Unauthorized - Request needs user authentication	401
365	0.17	Bad Request - The syntax of the request is invalid	400
5	0.002	Document Not Found - Requested resource could not be found	404
10	0.004	Internal Server Error	500

istSOS: a SOS compliant software with special features

The software

istSOS 2.0 documentation »

next | modules | index

istSOS[®]

Download istSOS v2.0

Welcome to istSOS-project

Free and Open Source Sensor Observation Service Data Management System



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

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

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

Web sections:

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

Enter search terms or a module, class or function name.

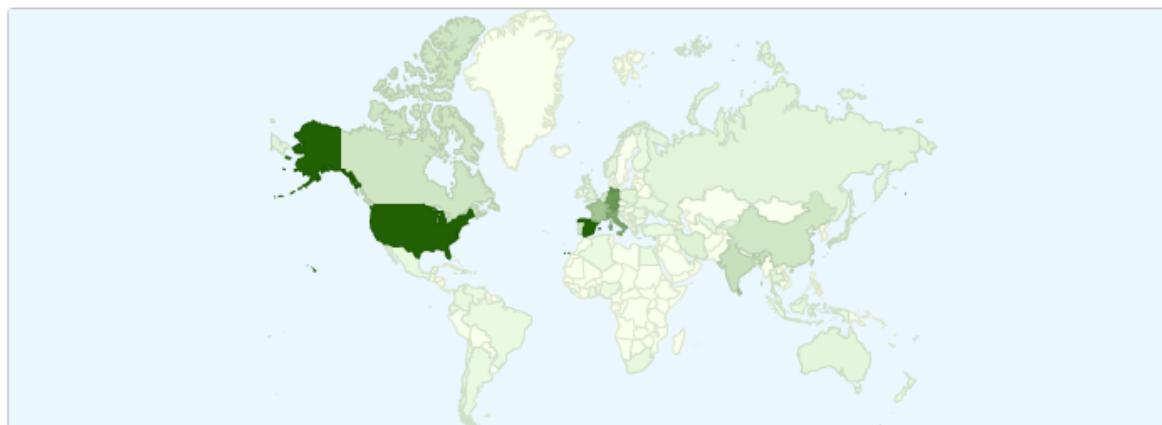
 Go

Open Source Software (GPL v2)

Brought to you by: mantonovic, maxicannata

[Summary](#)[Files](#)[Reviews](#)[Support](#)[Code](#)[Wiki](#)[Tickets](#)[Blog](#)[Admin](#)[Add New...](#)[Home \(Change File\)](#)

Date Range: 2016-01-01 to 2017-01-01

**DOWNLOADS****1,255**

In the selected date range

TOP COUNTRY**Spain**

18% of downloaders

TOP OS**Other**

41% of downloaders

OS downloads as: [Percent ▾](#)

Country ▾	Android ▾	Linux ▾	Macintosh ▾	Unknown ▾	Windows ▾	Total ▾
1. Spain	0%	13%	4%	78%	5%	231
2. United States	0%	3%	3%	89%	5%	230
3. Germany	0%	19%	1%	62%	18%	146
4. Italy	1%	42%	3%	8%	46%	118
5. France	0%	44%	29%	6%	21%	82
6. Sri Lanka	0%	84%	0%	0%	16%	68
7. Switzerland	0%	57%	16%	13%	14%	63
8. India	0%	38%	2%	0%	60%	40
9. Canada	0%	13%	19%	0%	68%	31
10. China	0%	0%	14%	3%	83%	29

Supported system types (up-to-now)

in-situ - fixed - point

in-situ - mobile - points

Supported versions

SOS v1.0.0
(core & transactional)

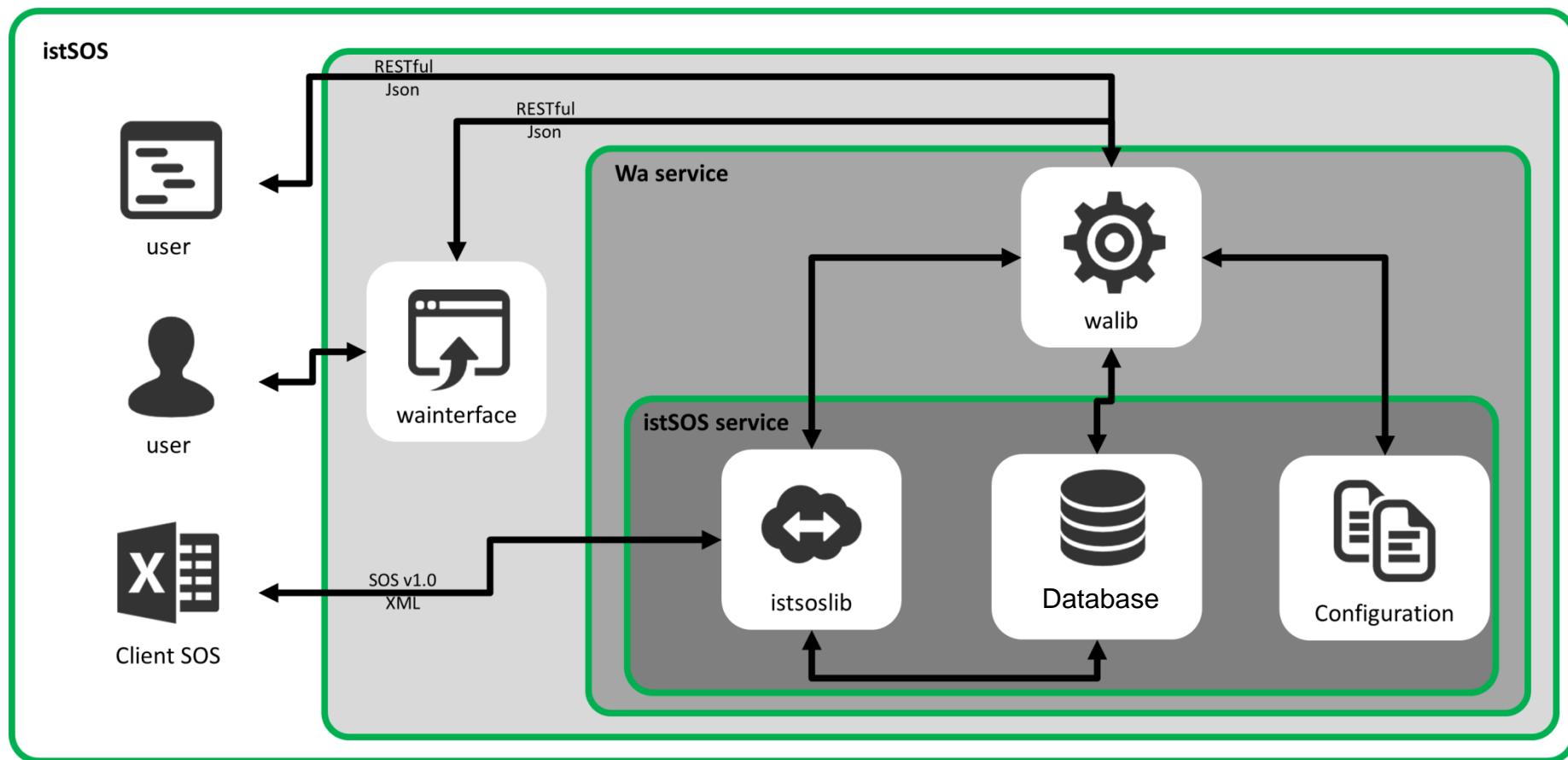
&

SOS v2.0
(core)

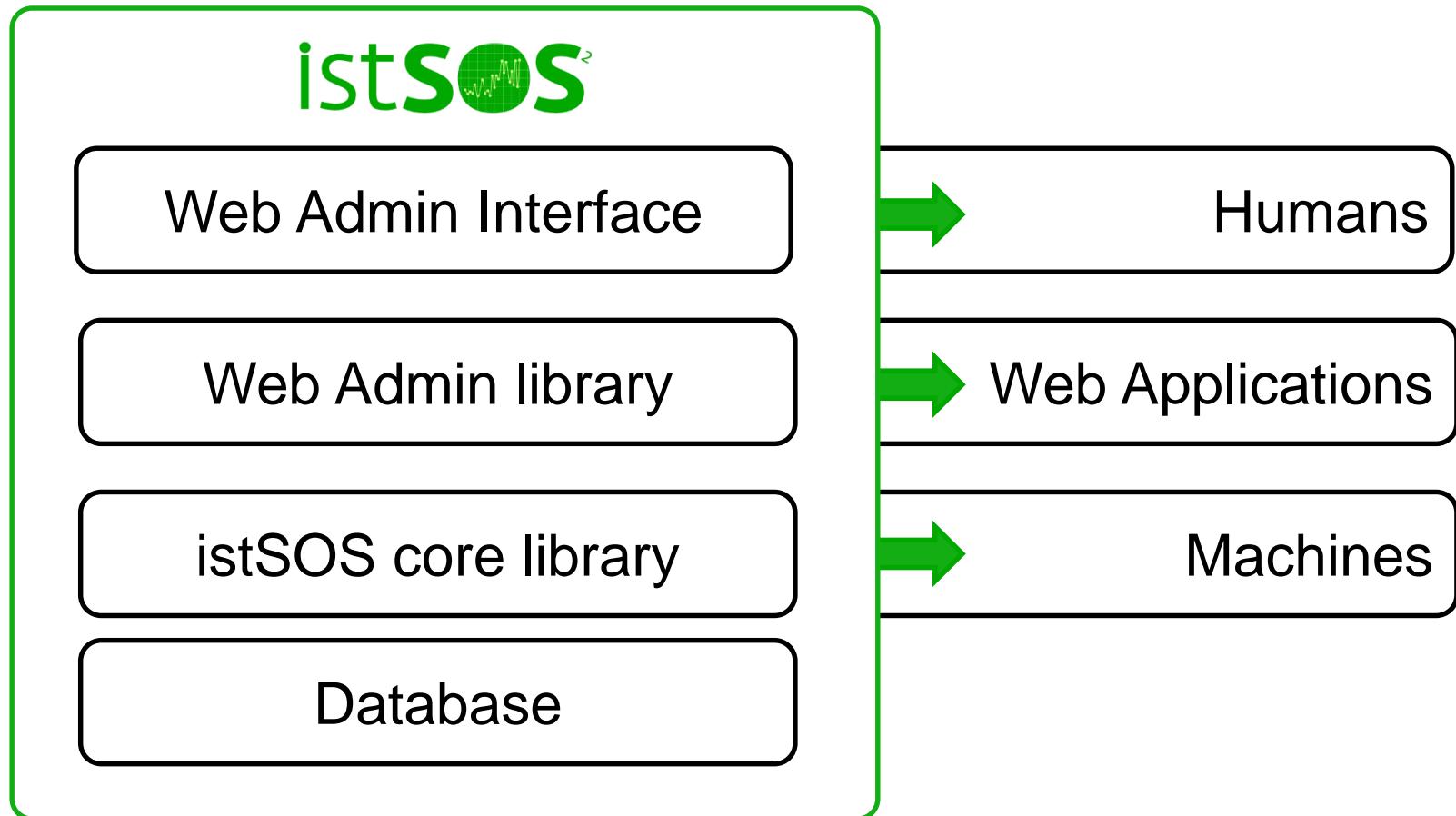
The istSOS software



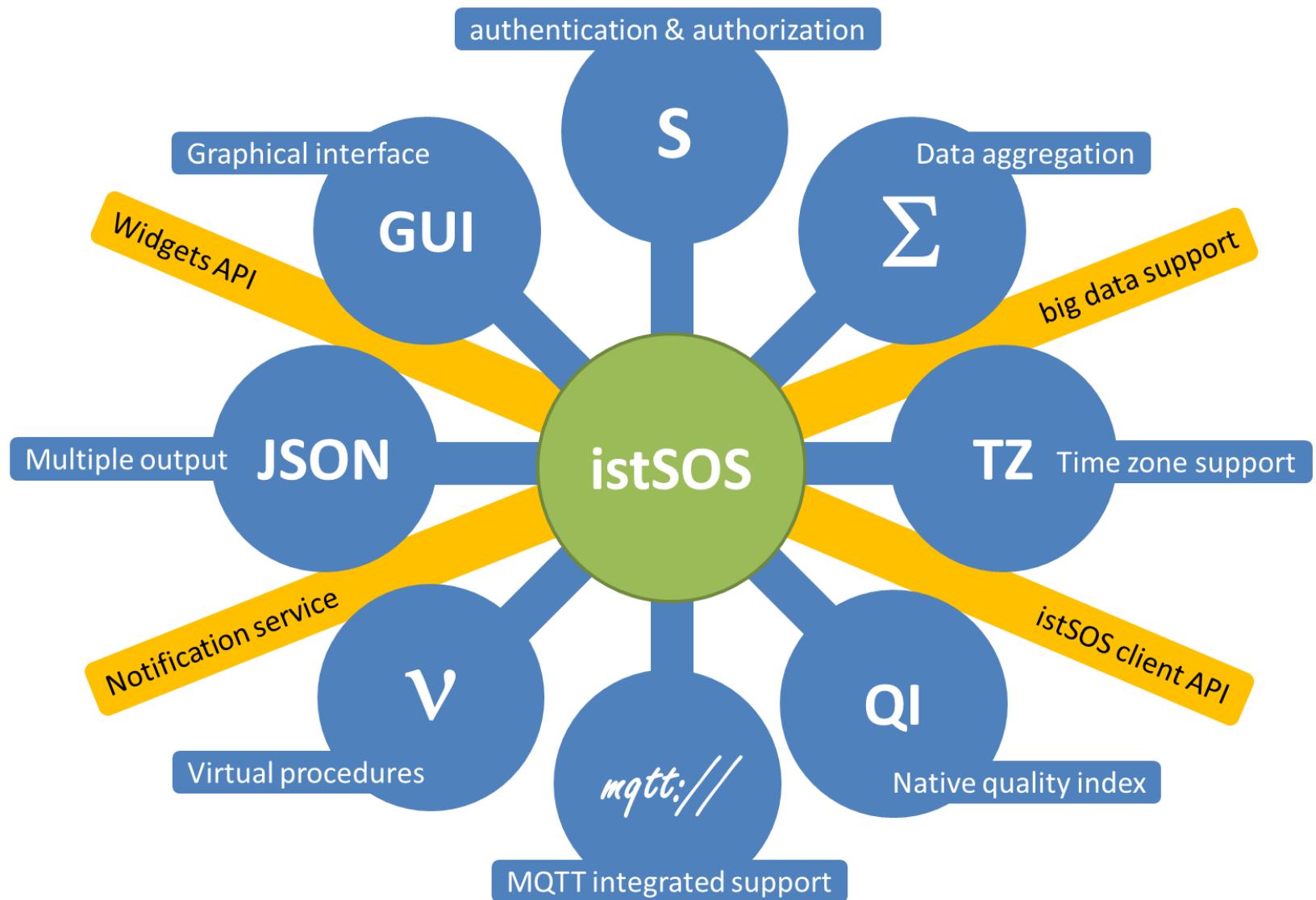
isodate
psycopg2
pytz



The istSOS software architecture



Some of the special features...



Easy installation and initialization and management using GUI (WebAdmin Interface)

The screenshot displays the istSOS WebAdmin Interface. At the top, a navigation bar includes icons for About istSOS, Status, Database, Service provider, Service identification (highlighted in green), Coordinates system, GetObservation Configuration, Proxy Configuration, New service, and Delete service. Below the bar, a breadcrumb trail shows 'default > Service identification' with a 'Submit' button.

The main area is titled 'Service Identification' and contains the following fields:

Title:	IST Sensor Observation Service
Abstract:	hydro-meteorological monitoring network
Keywords:	SOS,IST,SUPSI
Fees:	NONE
Access constraints:	NONE
URN authority:	x-istsos
URN version:	1.0

Below this, there are two panes. The left pane, titled 'Choose procedure', shows a table of data for 'T_LUGANO' with columns for Date, air-temperature, and qualityIndex. The right pane is a 'Plot' showing air-temperature time series for LOCARNO (red line) and T_LUGANO (green line) from January 28 to February 2, 2013. The plot includes a zoomed-in inset at the bottom.

Date	air-temperature	qualityIndex
2013-01-28T00:10:00.000000+0000	1.379	200
2013-01-28T00:20:00.000000+0000	1.284	200
2013-01-28T00:30:00.000000+0000	0.948	200
2013-01-28T00:40:00.000000+0000	1.249	200
2013-01-28T00:50:00.000000+0000	1.61	200
2013-01-28T01:00:00.000000+0000	2.051	200
2013-01-28T01:10:00.000000+0000	1.884	200
2013-01-28T01:20:00.000000+0000	1.476	200
2013-01-28T01:30:00.000000+0000	1.375	200
2013-01-28T01:40:00.000000+0000	1.278	200
2013-01-28T01:50:00.000000+0000	1.449	200
2013-01-28T02:00:00.000000+0000	1.394	200
2013-01-28T02:10:00.000000+0000	1.041	200
2013-01-28T02:20:00.000000+0000	1.334	200
2013-01-28T02:30:00.000000+0000	1.113	200

Data viewer

Server Services - Data Management Status (beta) **istSOS**

Data Editor Data Viewer

Service: **sos** From: 2016-05-06 00:00 +02:00 To: 2016-05-14 00:00 Property: water-discharge

Offering: temporary Procedure: **Q_CUC_POR** reset Plot **Q_CUC_POR**

urn:ogc:def:parameter:x-istsos:1.0:river:water:discharge

Q_CUC_POR **sos:temporary**
Fr:1999-12-31T23:00:00.00000Z
To:2016-05-13T16:30:00.00000Z
water-discharge (m3/s)
Aggregation: no

Plot showing water-discharge (m3/s) over time from May 6 to May 13, 2016. The plot shows a low baseline discharge of approximately 1 m3/s until May 11, followed by a sharp peak reaching up to 12 m3/s on May 11, before gradually decreasing.

Show CSV

Date	Q_CUC_POR	QI
2016-05-12T01:30:00+02:00	9.235	210
2016-05-12T01:40:00+02:00	9.969	210
2016-05-12T01:50:00+02:00	10.733	210
2016-05-12T02:00:00+02:00	10.82	210
2016-05-12T02:10:00+02:00	11.574	210
2016-05-12T02:20:00+02:00	11.619	210
2016-05-12T02:30:00+02:00	11.801	210
2016-05-12T02:40:00+02:00	11.394	210
2016-05-12T02:50:00+02:00	10.907	210
2016-05-12T03:00:00+02:00	10.951	210
2016-05-12T03:10:00+02:00	10.475	210
2016-05-12T03:20:00+02:00	10.518	210
2016-05-12T03:30:00+02:00	10.178	210
2016-05-12T03:40:00+02:00	9.762	210
2016-05-12T03:50:00+02:00	9.235	210
2016-05-12T04:00:00+02:00	9.315	210
2016-05-12T04:10:00+02:00	8.997	210

Map showing the location of the monitoring station (blue dot) near Porlezza, Italy. The map displays the town, surrounding roads (SS340, SP10, SP14), and natural features like the Lago di Piano and the Val d'Alpe. The station is located along the SS340 road between Molzano and Carlazzo.

Output formats from GetObservation request

XML – CSV - JSON

```
<com:ObservationCollection xmlns:sos="http://www.opengis.net/sos/1.0" xmlns:om="http://www.opengis.net/om/1.0"
  xmlns:swe="http://www.opengis.net/swe/1.0.1" xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/om/1.0 http://schemas.opengis.net/om/1.0.0/om.xsd">
  <gml:description>
    temporary offering to hold self-registered procedures/sensors waiting for service administration acceptance
  </gml:description>
  <gml:name>temporary</gml:name>
  <com:member>
    <com:Observation>
      <gml:name>LOCARNO</gml:name>
      <com:samplingTime>
        <gml:TimePeriod>
          <gml:beginPosition>2013-01-01T00:10:00.000000+0100</gml:beginPosition>
          <gml:endPosition>2013-02-05T00:20:00.000000+0100</gml:endPosition>
          <gml:duration>P34DT23H50M</gml:duration>
        </gml:TimePeriod>
      </com:samplingTime>
      <com:procedure xlink:href="urn:ogc:procedure:LOCARNO:00000000000000000000000000000000">
        <com:observedProperty>
          <swe:CompositePhenomenon gml:id="timeSeriesOfObservations">
            <gml:name>timeSeriesOfObservations</gml:name>
            <swe:component xlink:href="urn:ogc:procedure:LOCARNO:00000000000000000000000000000001">
              <swe:component xlink:href="urn:ogc:procedure:LOCARNO:00000000000000000000000000000002">
                <swe:component xlink:href="urn:ogc:procedure:LOCARNO:00000000000000000000000000000003">
                  <swe:CompositePhenomenon>
                    <gml:id>705120.300000000046566,1</gml:id>
                  <gml:coordinates>
                    705120.300000000046566,1
                  </gml:coordinates>
                </swe:component>
              </swe:component>
            </swe:component>
          </swe:CompositePhenomenon>
        </com:observedProperty>
      <com:featureOfInterest xlink:href="urn:ogc:featureOfInterest:LOCARNO:00000000000000000000000000000004">
        <gml:FeatureCollection>
          <gml:location>
            <gml:Point srsName="EPSG:2178">
              <gml:coordinates>
                705120.300000000046566,1
              </gml:coordinates>
            </gml:Point>
          </gml:location>
        </gml:FeatureCollection>
      </com:featureOfInterest>
    </com:Observation>
  </com:member>
</com:ObservationCollection>
```

- ObservationCollection: {
 - member: [
 - {
 name: "LOCARNO",
 - samplingTime: {
 duration: "P34DT23H50M",
 beginPosition: "2013-01-01T00:10:00.000000+0100",
 endPosition: "2013-02-05T00:20:00.000000+0100",
 },
 - result: {
 - dataArray: {
 elementCount: "3",
 - values: [
 [
 "2013-01-01T00:10:00.000000+0100",
 "0.000000",
 "4.000000"
],
 [
 "2013-01-01T00:20:00.000000+0100",
 "0.000000",
 "3.000000"
]
],
 - }
 },
 },
 },
],
}

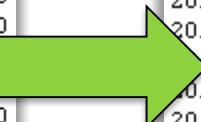
Aggregation on-the-fly with GetObservation request

AVG, SUM, MAX, MIN

Extra parameters: aggregateInterval = PT1H & aggregateFunction = AVG

```
2013-01-01T01:00:00.000000+0100,LOCARNO,3.000000
2013-01-01T01:10:00.000000+0100,LOCARNO,2.500000
2013-01-01T01:20:00.000000+0100,LOCARNO,3.400000
2013-01-01T01:30:00.000000+0100,LOCARNO,3.100000
2013-01-01T01:40:00.000000+0100,LOCARNO,3.100000
2013-01-01T01:50:00.000000+0100,LOCARNO,2.700000
2013-01-01T02:00:00.000000+0100,LOCARNO,2.300000
2013-01-01T02:10:00.000000+0100,LOCARNO,3.700000
2013-01-01T02:20:00.000000+0100,LOCARNO,2.800000
2013-01-01T02:30:00.000000+0100,LOCARNO,2.700000
2013-01-01T02:40:00.000000+0100,LOCARNO,2.200000
2013-01-01T02:50:00.000000+0100,LOCARNO,2.300000
2013-01-01T03:00:00.000000+0100,LOCARNO,2.200000
2013-01-01T03:10:00.000000+0100,LOCARNO,2.800000
2013-01-01T03:20:00.000000+0100,LOCARNO,2.400000
2013-01-01T03:30:00.000000+0100,LOCARNO,2.700000
2013-01-01T03:40:00.000000+0100,LOCARNO,2.300000
2013-01-01T03:50:00.000000+0100,LOCARNO,2.600000
2013-01-01T04:00:00.000000+0100,LOCARNO,2.300000
2013-01-01T04:10:00.000000+0100,LOCARNO,3.100000
2013-01-01T04:20:00.000000+0100,LOCARNO,2.700000
2013-01-01T04:30:00.000000+0100,LOCARNO,2.300000
2013-01-01T04:40:00.000000+0100,LOCARNO,2.500000
2013-01-01T04:50:00.000000+0100,LOCARNO,2.500000
2013-01-01T05:00:00.000000+0100,LOCARNO,2.300000
2013-01-01T05:10:00.000000+0100,LOCARNO,2.500000
2013-01-01T05:20:00.000000+0100,LOCARNO,1.400000
2013-01-01T05:30:00.000000+0100,LOCARNO,2.300000
2013-01-01T05:40:00.000000+0100,LOCARNO,1.500000
2013-01-01T05:50:00.000000+0100,LOCARNO,1.600000
```

```
2013-01-01T02:00:00.000000+0100,LOCARNO,2.9166666666666667
2013-01-01T03:00:00.000000+0100,LOCARNO,2.6500000000000000
2013-01-01T04:00:00.000000+0100,LOCARNO,2.5166666666666667
2013-01-01T05:00:00.000000+0100,LOCARNO,2.5666666666666667
2013-01-01T06:00:00.000000+0100,LOCARNO,1.9833333333333333
2013-01-01T07:00:00.000000+0100,LOCARNO,2.1166666666666667
2013-01-01T08:00:00.000000+0100,LOCARNO,2.0000000000000000
2013-01-01T09:00:00.000000+0100,LOCARNO,2.3166666666666667
2013-01-01T10:00:00.000000+0100,LOCARNO,2.3166666666666667
2013-01-01T11:00:00.000000+0100,LOCARNO,2.9166666666666667
2013-01-01T12:00:00.000000+0100,LOCARNO,4.0833333333333333
2013-01-01T13:00:00.000000+0100,LOCARNO,4.0000000000000000
2013-01-01T14:00:00.000000+0100,LOCARNO,4.5500000000000000
2013-01-01T15:00:00.000000+0100,LOCARNO,4.9333333333333333
2013-01-01T16:00:00.000000+0100,LOCARNO,4.5500000000000000
2013-01-01T17:00:00.000000+0100,LOCARNO,4.3000000000000000
2013-01-01T18:00:00.000000+0100,LOCARNO,4.4666666666666667
2013-01-01T19:00:00.000000+0100,LOCARNO,4.7000000000000000
2013-01-01T20:00:00.000000+0100,LOCARNO,4.3333333333333333
2013-01-01T21:00:00.000000+0100,LOCARNO,4.2000000000000000
2013-01-01T22:00:00.000000+0100,LOCARNO,4.1166666666666667
2013-01-01T23:00:00.000000+0100,LOCARNO,4.3333333333333333
2013-01-02T00:00:00.000000+0100,LOCARNO,4.0000000000000000
2013-01-02T01:00:00.000000+0100,LOCARNO,3.4000000000000000
2013-01-02T02:00:00.000000+0100,LOCARNO,2.8333333333333333
2013-01-02T03:00:00.000000+0100,LOCARNO,2.7666666666666667
2013-01-02T04:00:00.000000+0100,LOCARNO,2.6000000000000000
2013-01-02T05:00:00.000000+0100,LOCARNO,2.8333333333333333
2013-01-02T06:00:00.000000+0100,LOCARNO,2.7333333333333333
2013-01-02T07:00:00.000000+0100,LOCARNO,2.4666666666666667
```



Real Time Quality Check

Configuration of quality index on the observed property level or sensor specific.

The screenshot shows the configuration interface for an observed property named "air-temperature". The "Observed property" section includes fields for Name (air-temperature), Definition URN (urn:ogc:def:parameter:x-istsos:1.0:meteo:air:temperature), Description (air temperature at 2 meters above terrain), and Statistical QI (set to "Between" with values -50 and 50). Below this is an "Outputs" section listing the same property definition. A modal dialog titled "GetObservation configuration:" contains settings for Max request interval (hours) (0), Transactional request logger (checkbox checked), Default quality index (100 - raw), Correct quality index (110 - acceptable), and Stat. quality index (200 - reasonable).

Automatically data quality check assigning quality index on each observation based on specific constraints

Name	Definition	uom	From
air-temperature	urn:ogc:def:parameter:x-istsos:1.0:meteo:air:temperature	°C	-40

GetObservation configuration:	
Max request interval (hours):	0
Transactional request logger:	<input checked="" type="checkbox"/> Enabled if checked
Default quality index:	100 - raw (the format is correct)
Correct quality index:	110 - acceptable (the value is acceptable for the observed property)
Stat. quality index:	200 - reasonable (the value is in a resonable range for that observed property and sensor)

Quality index example

Raw data

```
urn:ogc:def:parameter:x-istsos:1.0:time:iso8601,urn:ogc:def:parameter:x-  
istsos:1.0:meteo:air:temperature  
2013-02-05T00:10:00.000000+0100,-45.000000  
2013-02-05T00:20:00.000000+0100,80.000000  
2013-02-05T00:30:00.000000+0100,1.56100
```

Assignment of quality indices during insertion:

```
2013-02-04T23:00:00.000000+0100,T_LUGANO,1.803000,200  
2013-02-04T23:10:00.000000+0100,T_LUGANO,2.211000,200  
2013-02-04T23:20:00.000000+0100,T_LUGANO,1.901000,200  
2013-02-04T23:30:00.000000+0100,T_LUGANO,1.910000,200  
2013-02-04T23:40:00.000000+0100,T_LUGANO,1.613000,200  
2013-02-04T23:50:00.000000+0100,T_LUGANO,1.560000,200  
2013-02-05T00:00:00.000000+0100,T_LUGANO,1.599000,200  
2013-02-05T00:10:00.000000+0100,T_LUGANO,-45.000000,110  
2013-02-05T00:20:00.000000+0100,T_LUGANO,80.000000,100  
2013-02-05T00:30:00.000000+0100,T_LUGANO,1.561000,200
```

Meaningful data (Level 1)

Raw data

Statistically sound (Level 1 + 2)

Time-Zone support

eventTime

2013-01-01T01:00:00+01 / 2013-01-07T01:00:00+01



2013-01-01T02:00:00+02 / 2013-01-07T02:00:00+02

eventTime

2013-01-01T02:10:00.000000+0200,LOCARNO,2.500000

2013-01-01T02:20:00.000000+0200,LOCARNO,3.400000

2013-01-01T02:30:00.000000+0200,LOCARNO,3.100000

2013-01-01T02:40:00.000000+0200,LOCARNO,3.100000

2013-01-01T02:50:00.000000+0200,LOCARNO,2.700000

2013-01-01T03:00:00.000000+0200,LOCARNO,3.300000

2013-01-01T03:10:00.000000+0200,LOCARNO,3.700000

2013-01-01T03:20:00.000000+0200,LOCARNO,2.800000

2013-01-01T03:30:00.000000+0200,LOCARNO,2.700000

2013-01-01T03:40:00.000000+0200,LOCARNO,2.200000

2013-01-01T03:50:00.000000+0200,LOCARNO,2.300000

2013-01-01T04:00:00.000000+0200,LOCARNO,2.200000

2013-01-01T04:10:00.000000+0200,LOCARNO,2.800000

2013-01-01T04:20:00.000000+0200,LOCARNO,2.400000

2013-01-01T04:30:00.000000+0200,LOCARNO,2.700000

2013-01-01T04:40:00.000000+0200,LOCARNO,2.300000

2013-01-01T04:50:00.000000+0200,LOCARNO,2.600000

2013-01-01T05:00:00.000000+0200,LOCARNO,2.300000

2013-01-01T05:10:00.000000+0200,LOCARNO,3.100000

2013-01-01T05:20:00.000000+0200,LOCARNO,2.700000

2013-01-01T05:30:00.000000+0200,LOCARNO,2.300000



2013-01-01T01:10:00.000000+0100,LOCARNO,2.500000

2013-01-01T01:20:00.000000+0100,LOCARNO,3.400000

2013-01-01T01:30:00.000000+0100,LOCARNO,3.100000

2013-01-01T01:40:00.000000+0100,LOCARNO,3.100000

2013-01-01T01:50:00.000000+0100,LOCARNO,2.700000

2013-01-01T02:00:00.000000+0100,LOCARNO,3.300000

2013-01-01T02:10:00.000000+0100,LOCARNO,3.700000

2013-01-01T02:20:00.000000+0100,LOCARNO,2.800000

2013-01-01T02:30:00.000000+0100,LOCARNO,2.700000

2013-01-01T02:40:00.000000+0100,LOCARNO,2.200000

2013-01-01T02:50:00.000000+0100,LOCARNO,2.300000

2013-01-01T03:00:00.000000+0100,LOCARNO,2.200000

2013-01-01T03:10:00.000000+0100,LOCARNO,2.800000

2013-01-01T03:20:00.000000+0100,LOCARNO,2.400000

2013-01-01T03:30:00.000000+0100,LOCARNO,2.700000

2013-01-01T03:40:00.000000+0100,LOCARNO,2.300000

2013-01-01T03:50:00.000000+0100,LOCARNO,2.600000

2013-01-01T04:00:00.000000+0100,LOCARNO,2.300000

2013-01-01T04:10:00.000000+0100,LOCARNO,3.100000

2013-01-01T04:20:00.000000+0100,LOCARNO,2.700000

2013-01-01T04:30:00.000000+0100,LOCARNO,2.300000

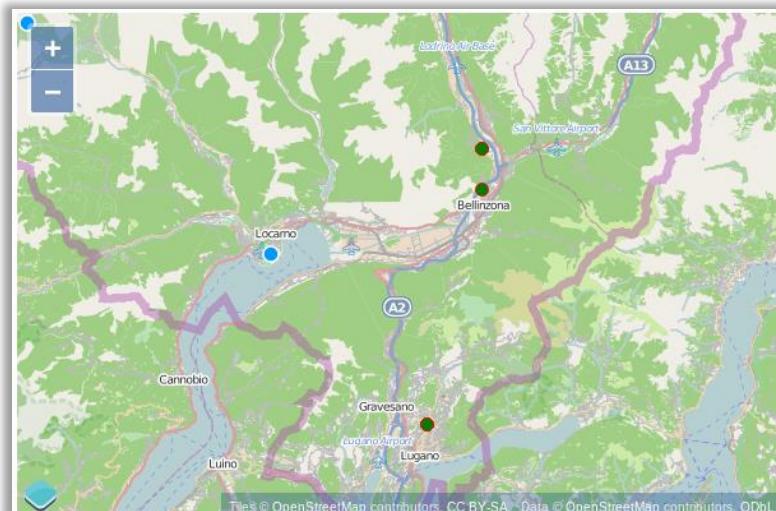
2013-01-01T04:40:00.000000+0100,LOCARNO,2.500000

JSON RESTful API

RESTful API for accessing and managing istSOS:

- Get service status, test database connection, initialize new services
- Get, insert, update and delete:
 - sensors,
 - offerings,
 - observed properties,
 - quality indexes,
 - unit of measures,
 - service metadata,
 - observations,
 - features of interest,
 - supported epsg,
 - etc...

WALib supporting Sensors Mapping and Charting Observations



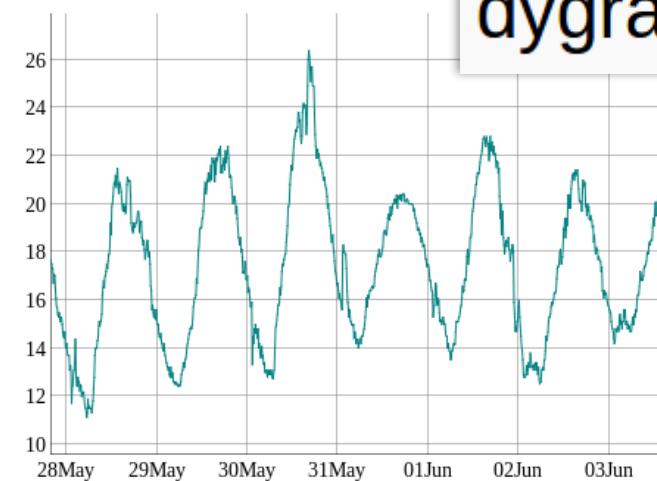
OpenLayers 3

LOCARNO

Begin: 2014-05-03T16:30:00+0200
End: 2014-06-03T19:50:00+0200

Observed properties:

1. air-temperature
2. air-rainfall



```
var vector = new ol.layer.Vector({
  source: new ol.source.Vector({
    parser: new ol.parser.GeoJSON(),
    url: '/istsos/wa/istsos/services/demo/procedures/operations/geojson?epsg=4326'
  })
});
```

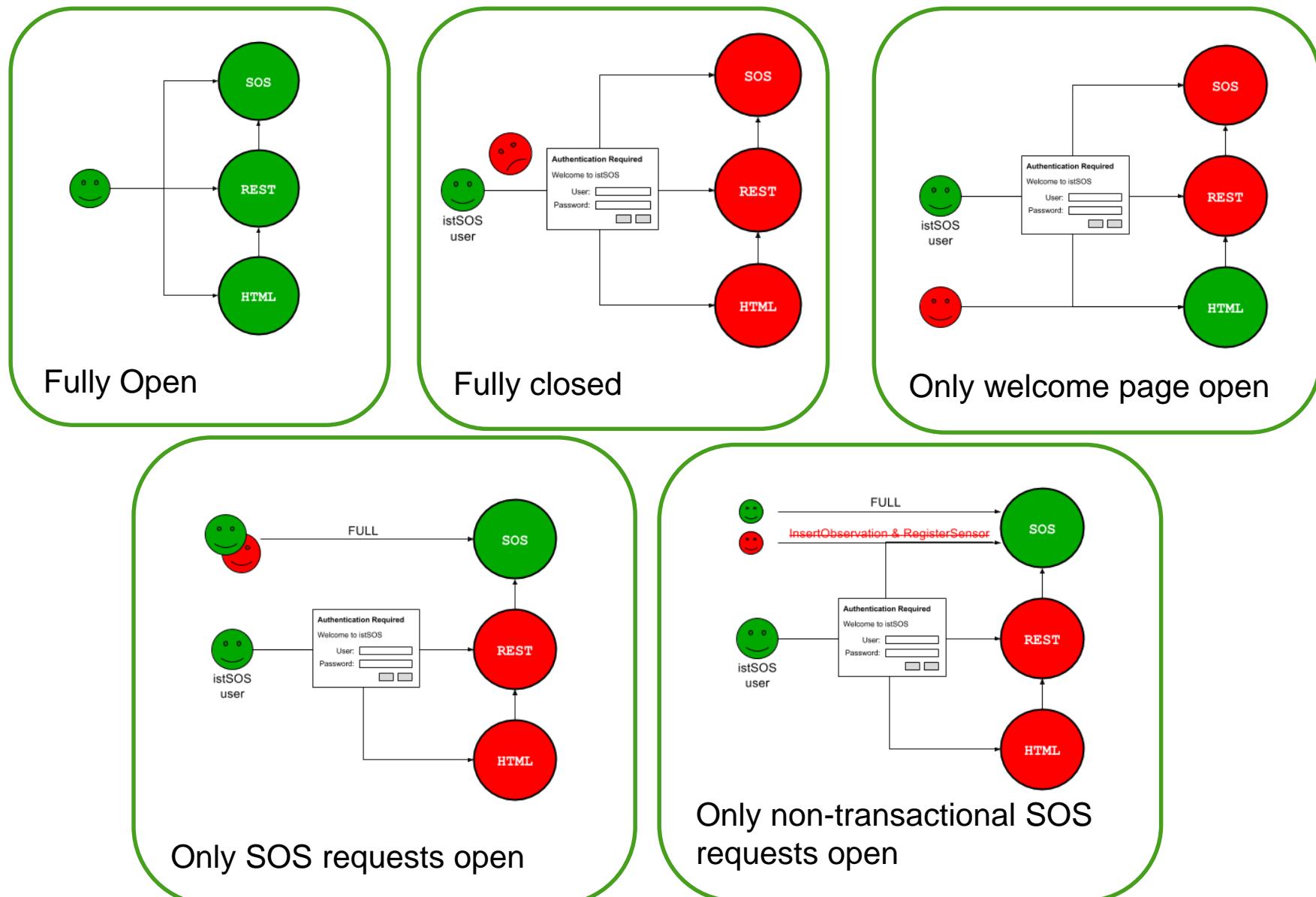
Virtual procedures (EVAPOTRANSPIRATION)

The diagram illustrates the process of creating a virtual procedure. On the left, a box labeled "GRABOW" contains "Real Sensor" data: Air Temperature, Relative humidity, Wind velocity, and Solar radiation. An arrow points to the right, labeled "FAO56 Calculation". On the right, a box labeled "V_GRABOW" contains "Virtual procedure" data: Reference evapotranspiration. Below the boxes is a screenshot of the istSOS interface. The top navigation bar includes Service identifier, Service identification, Coordinates system, GetObservation Configuration, Proxy Configuration, Offerings, Procedures, Virtual Procedures (which is selected), New procedure, Observed properties, Units of measures, and Data quality. A dropdown menu shows "Virtual procedure": "V_GRABOW". The main area shows Python code for the "execute" method of the "V_GRABOW" class:

```
1 from istboslib.responders.GOresponse import VirtualProcess
2 import FAO56
3
4 class istvp(VirtualProcess):
5     procedures = [
6         "GRABOW": [
7             "urn:ogc:def:parameter:x-istsos:1.0:meteo:air:temperature",
8             "urn:ogc:def:parameter:x-istsos:1.0:meteo:air:humidity:relative",
9             "urn:ogc:def:parameter:x-istsos:1.0:meteo:air:wind:velocity",
10            "urn:ogc:def:parameter:x-istsos:1.0:meteo:solar:radiation"
11        ]
12    }
13    def execute(self):
14        data = self.getData("GRABOW")
15        data_out = []
16        for rec in data:
17            if self.filter.qualityIndex == True:
18                # rec is a list:
19                # [0]=time, [1]=T, [2]=Tqi, [3]=RH, [4]=RHqi,
20                # [5]=u2, [6]=u2qi, [7]=Rs, [8]=Rsqi
21                etp = FAO56.ET0(isodate = str(rec[0]),
22                               T=float(rec[1]),
23                               RH=float(rec[3]),
24                               u2=float(rec[5]),
25                               Rs=float(rec[7])*0.0036, # W/m2 to MJ/(m2*h)
26                               lat=22.67,
27                               lon=51.25,
28                               z=177)
29                data_out.append([rec[0], etp, min([rec[2], rec[4], rec[6], rec[8]]))]
```

A Python logo is displayed next to the word "python". At the bottom are two green buttons: "Delete Code File" and "Save the Python Code".

(NEW) Security configurations



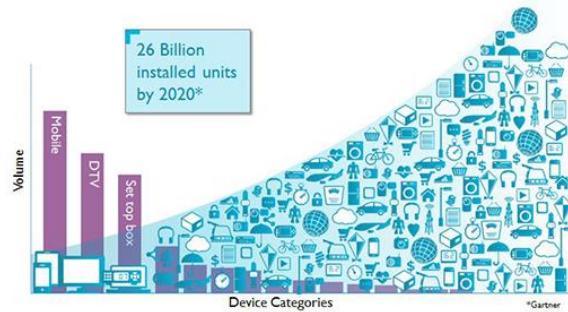
(NEW) User profiles

admin:
users with this
role have
access to all the
istSOS features

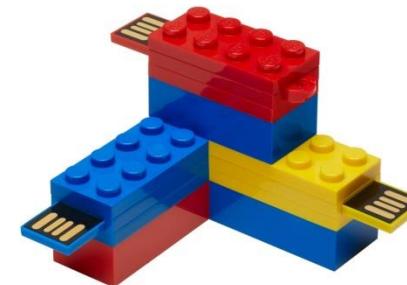
datamanager:
users with this
role can modify
only measures
and procedures
metadata

visitor:
users with this
role can only
view measures

istSOS one enabler of the IOT



The Internet of Things (IoT) is the collection of billions of end devices intelligently connected and interoperating with servers and services.



Combine “Things” like Lego blocks to create applications, services, values....



How to chose the blocks? what are they? how do they fits together? We need well known language: standards for machine – machine interaction!

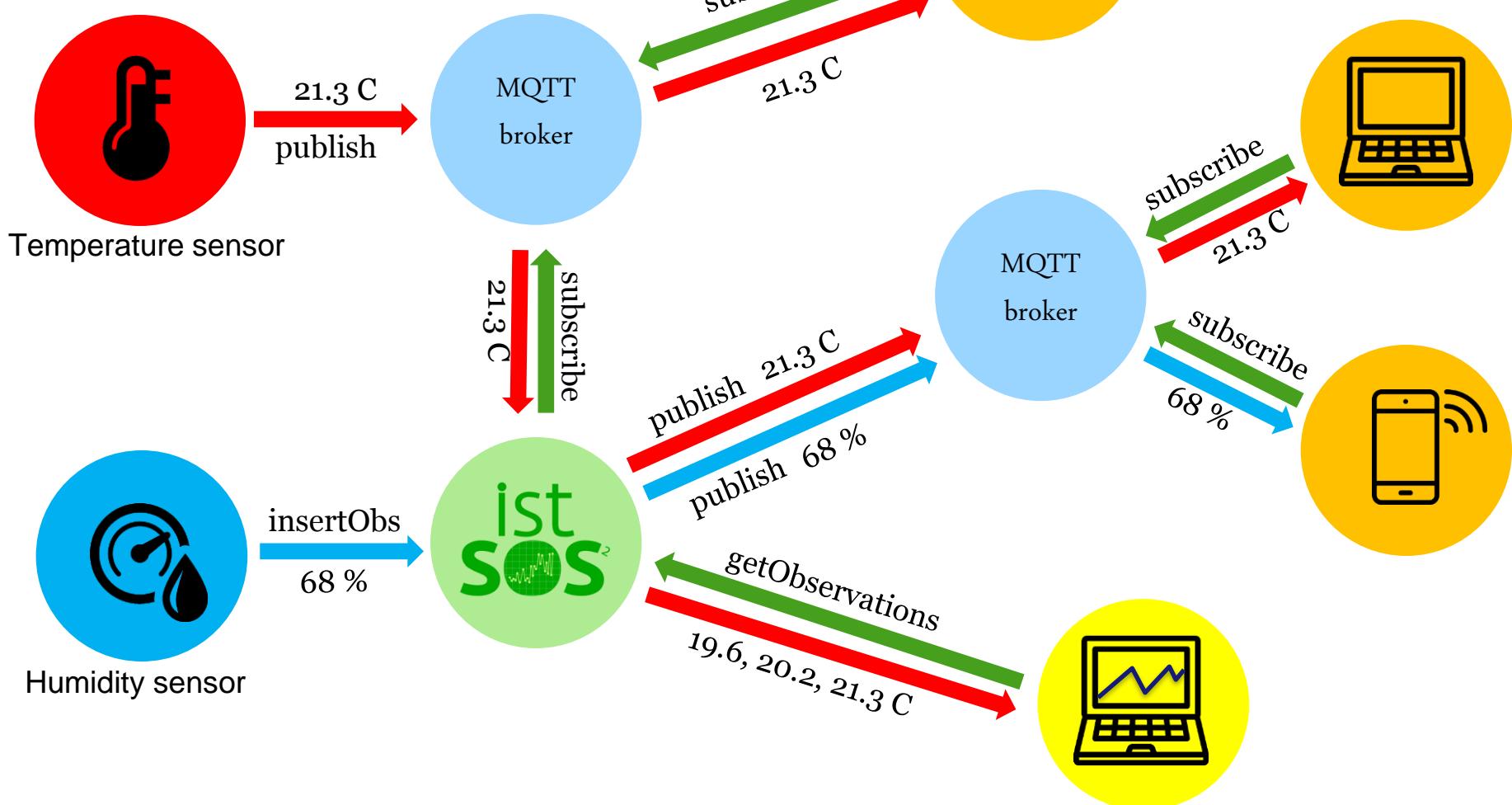
SOS is an OGC standard to:

- register new sensor (THING) to the Web,
- storing new observations
- retrieving desired data

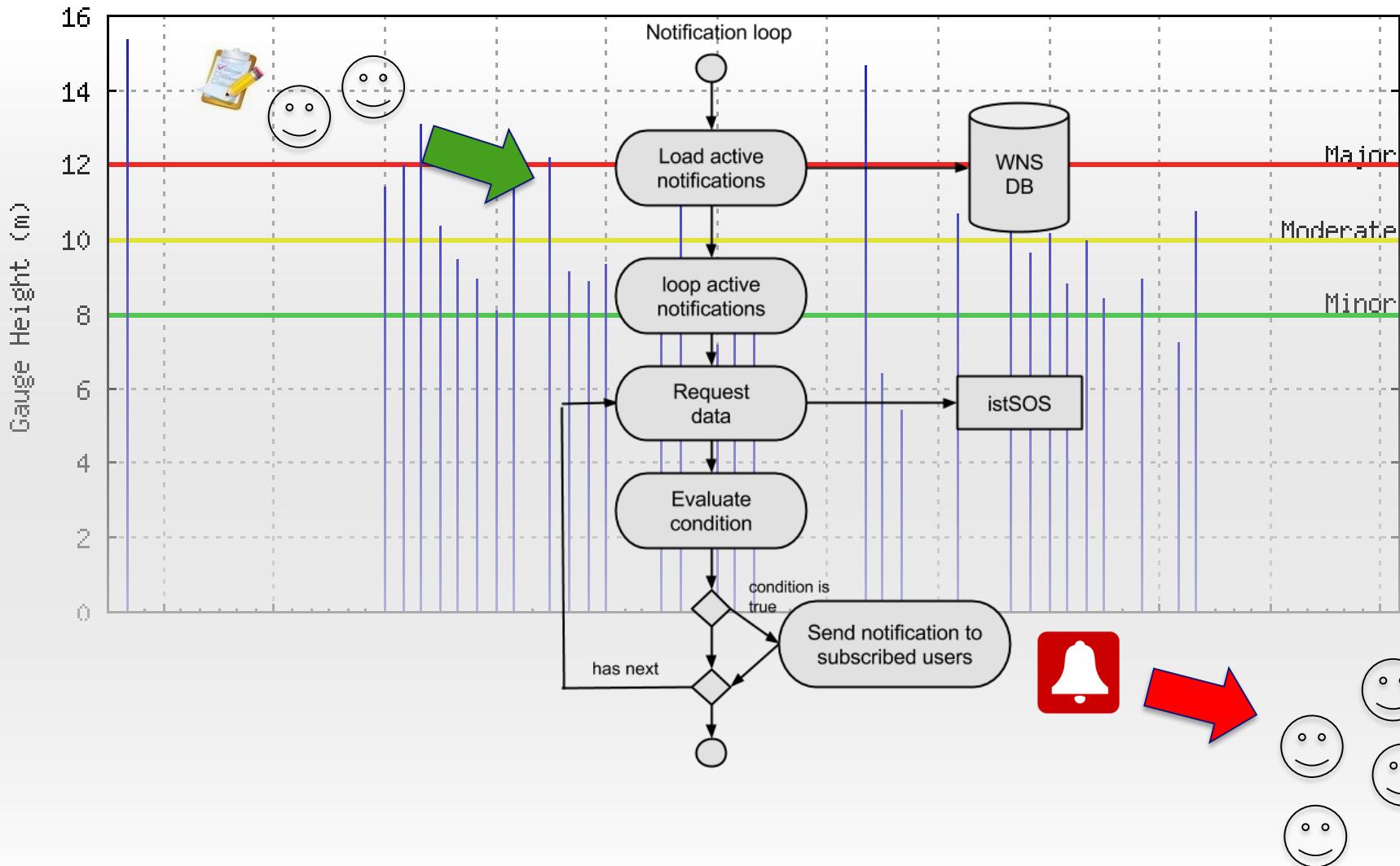
istSOS[®]

(NEW) MQTT support

- IoT - Event based behaviour
- Time series analysis



(2014) istSOS Alert & Notification Service



(2016) Google Summer of Code

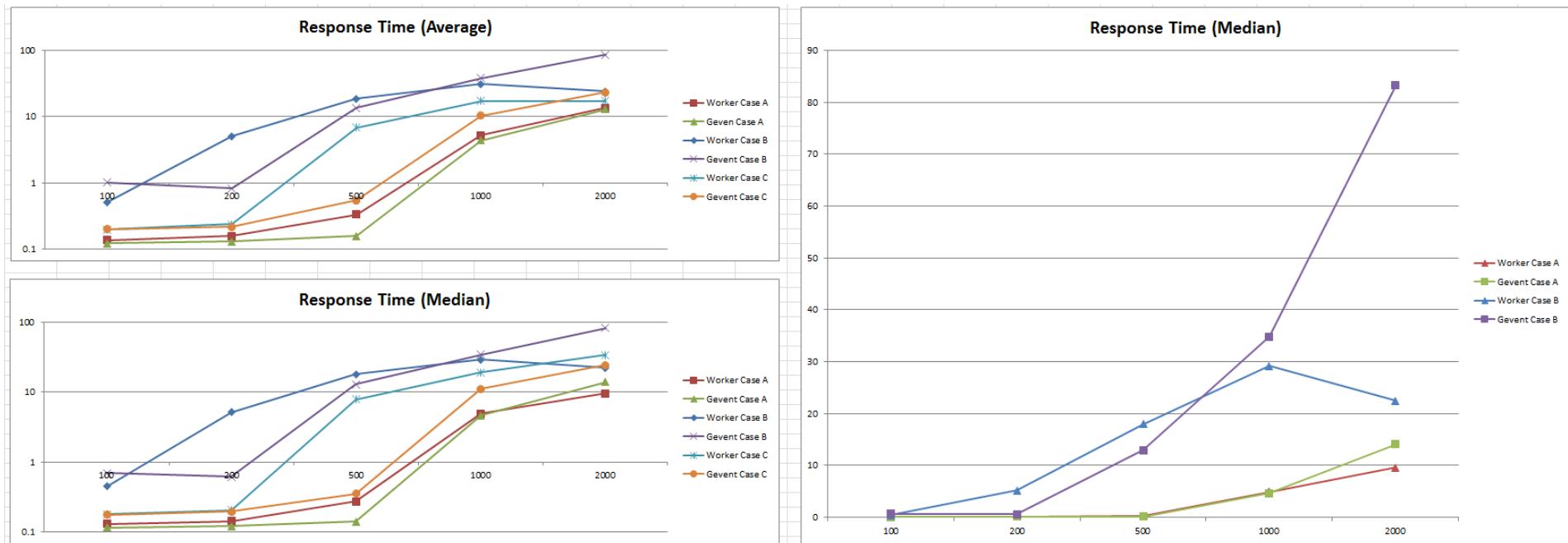
- **istSOS widget**
HTML embeddable elements
- **istSOS Android API**
library to build-up Android application with enabled istSOS features
- **VistSOS client API**
library to enable complex visualization of istSOS time-series data

Next research steps

istSOS loading test to identify bottlenecks

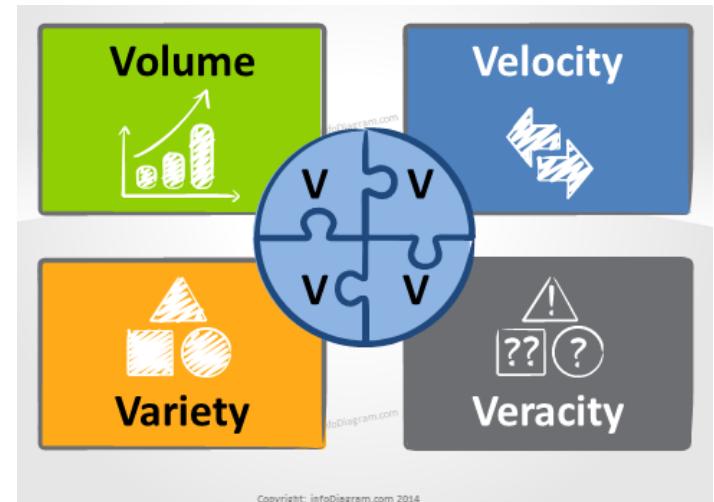
PAPER IN PROGRESS

- 3 different cases
- 2 different WGSI server solution
- 2 different user types
- 5 different concurrent users scenarios (100, 200, 500, 1'000, 2'000)



Big data issue

- Deploying the system worldwide we can imagine million of sensors with billion of centralized data to be served and ingested
- We need a scalable and adaptable and fast:
a solution for big data !



Thanks

istSOS

<http://istsos.org>

