

GRASS and GFOSS USERS

The XII italian meeting

TRENTO, February 9-11, 2011

SOS: Sensor Observation Service

Massimiliano Cannata, Milan Antonovic

Scuola universitaria professionale
della Svizzera italiana

SUPSI

Istituto Scienze della Terra



Sensor Observation Service

contenuti

- Introduzione allo standard SOS
- Descrizione del software istSOS
- Installazione di istSOS
- Creazione di un servizio
- Comunicazione col servizio
- Osservazioni sullo standard
- Discussione finale

Introduzione allo standard SOS

Scuola universitaria professionale
della Svizzera italiana

SUPSI

Istituto Scienze della Terra

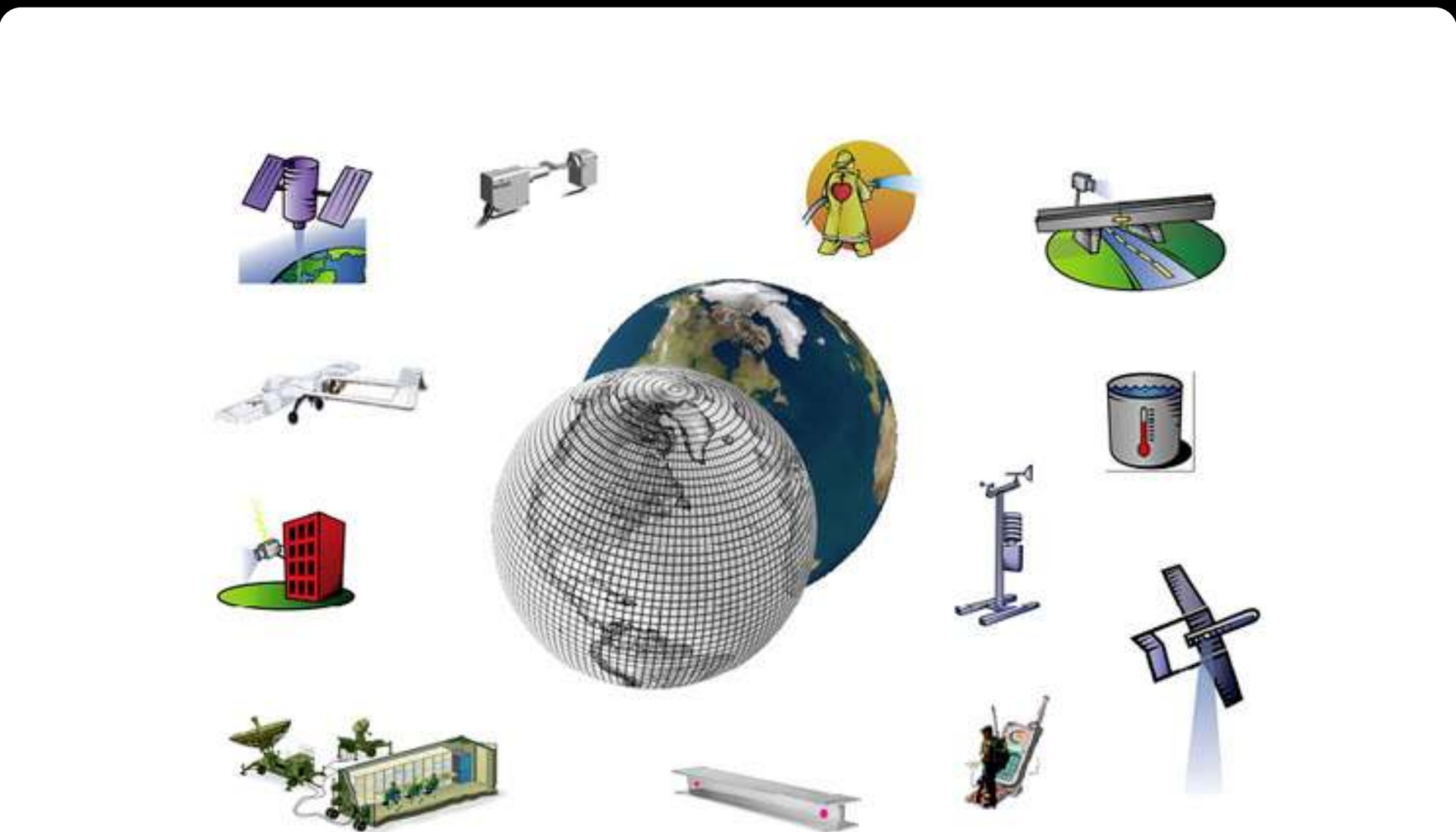


Sensor Observation Service

Sensori

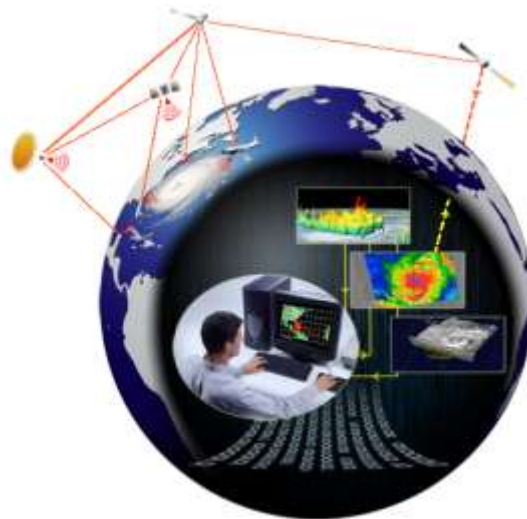
- I sensori sono ovunque:
 - case, supermercati, strade, fiumi, oceani, atmosfera, spazio, nostre tasche, etc...
- I sensori misurano diversi fenomeni:
 - temperatura dell'aria, pressione nelle tubazioni, qualità dell'acqua, stress di un edificio, radiazione solare, superficie fogliare, spostamento del terreno, etc...
- I sensori vengono usati per diversi scopi:
 - sorvegliare il territorio, monitorare un'infrastruttura, predire un pericolo, controllare un processo, etc.

...una pelle digitale per la terra



Il Sensor Web

L'idea del Sensor Web nasce come concetto diversi anni fa (Kevin Delin, NASA, 1997) ma solo oggi, seppur ancora lontani dalla sua realizzazione, si trova una prima implementazione di tale concetto!



Non siamo ancora pronti...

A fronte di tanti dati misurati in ogni istante e in ogni luogo del nostro pianeta ad oggi non siamo ancora in grado di sfruttare appieno questo enorme potenziale di informazioni.



reti di sensori
tra loro disconnesse



Mancanza di un
motore di ricerca



Mancanza di un
formato condiviso

Sensor Web: cos'è!

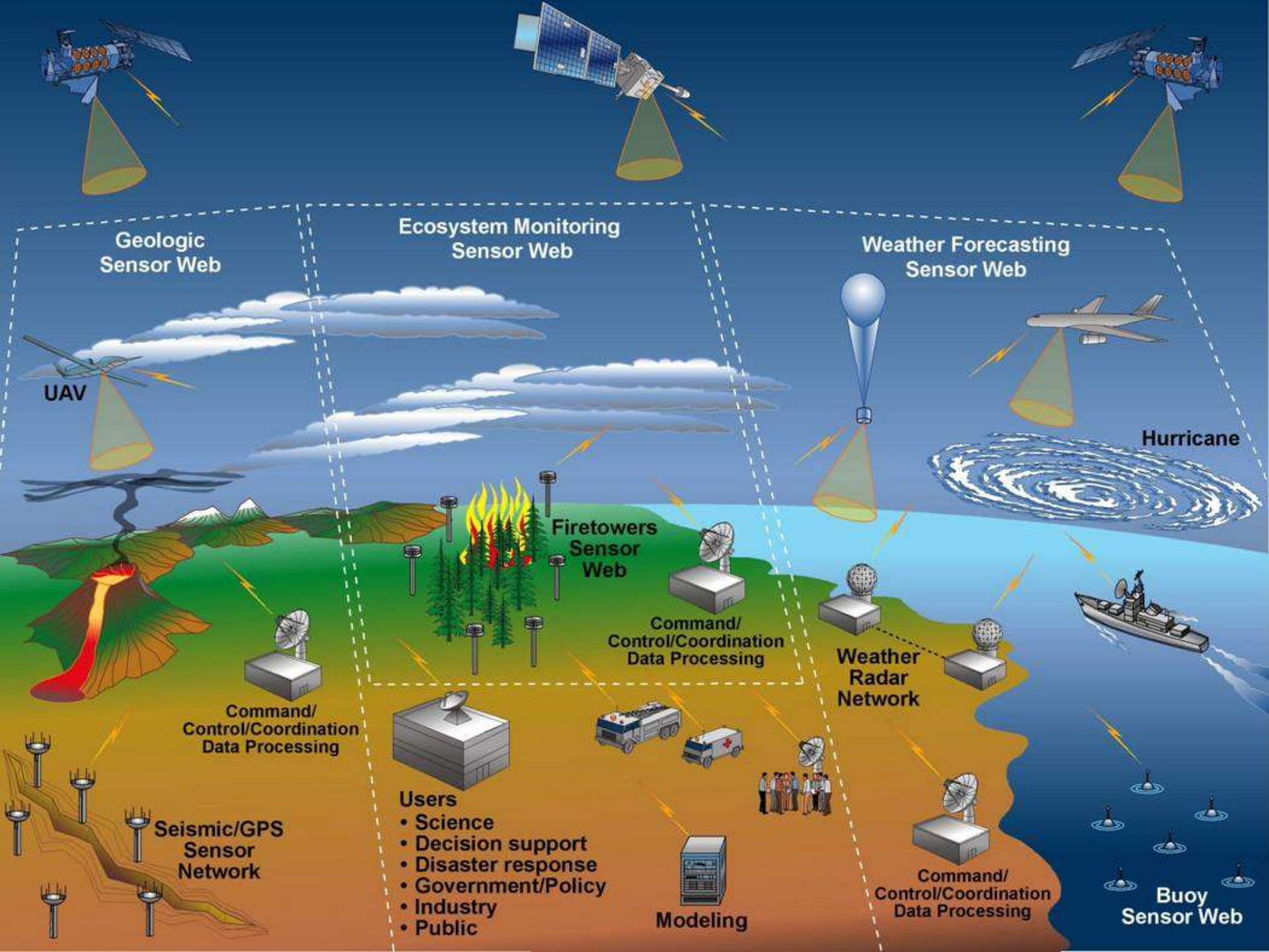
1. I sensori e le reti di sensori devono essere interconnesse ed accessibili tramite il Internet
2. Le informazioni sui sensori e le osservazioni devono essere accessibili attraverso servizi Web
3. I sensori devono essere descritti da un linguaggio standard e non proprietario tale da essere "autoesplicativo" sia per l'uomo che per le macchine

Sensor Web: cos'è!

4. I sensori ed i dati devono essere "ricercabili" tramite Internet in relazione a dati passati, presenti o futuri
5. Sensori, simulazioni e processi devono poter essere configurati ed attivati automaticamente tramite servizi Web
6. I software devono essere in grado di elaborare e localizzare le informazioni di nuovi sensori senza averne una conoscenza a priori

Sensor Web: cos'è!

7. Nuove informazioni devono poter essere generate dinamicamente integrando dati osservati
8. Allarmi devono poter essere generati sulla base di informazioni derivate dai sensori e disseminati per poter attivare azioni di intervento, anche automatiche
9. I sensori devono essere intelligenti, agendo come elementi che fanno parte di un ambiente che conoscono (reti intelligenti di sensori)



SWE: Sensor Web Enablement

- È un gruppo di lavoro dell'OGC (Open Geospatial Consortium) finalizzato allo sviluppo di:
 - Una tecnologia che abiliti la realizzazione del Sensor Web
 - Una suite di standard aperti e condivisi che definiscano la sintassi ed i servizi necessari per la realizzazione del Sensor Web
 - Un'architettura orientata ai servizi (SOA, Service Oriented Architecture) in grado di integrarsi con i principali approcci dell'*Information Technology*

SWE: Sensor Web Enablement

FORNITORI

Reti eterogenee di sensori

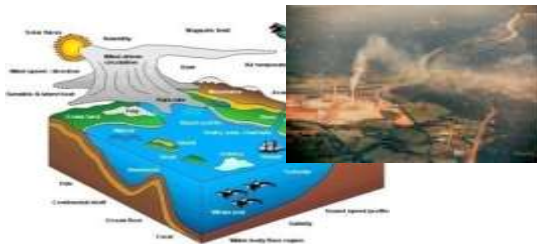


- sparsi
- diversi

- mobili/fissi
- estensibile

SPECIALISTI

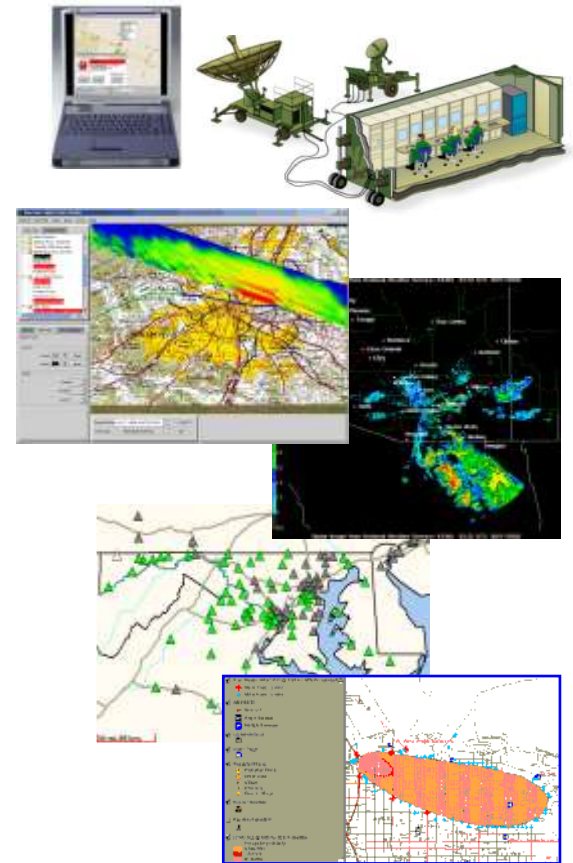
Modelli e simulazioni



- a cascata
- scala nazionale, regionale, urbana
- adattabili
- integrazione di dati

UTENTI

Supporto alle decisioni



- neutrale
- ampio

- flessibile
- adattivo

SWE

- ricerca
- accesso
- processamento
- notifica di allerte

Servizi Web e linguaggi basati su standard aperti (OGC, ISO, OASIS, IEEE)

SWE: i suoi *standards*

- **SWE Common**: modello di dati e codifiche per i sensori.
- **SensorML** (Sensor model Language): modello di dati e codifiche per i processi e le sue componenti.
- **O&M** (Observation and Measurements): modello di dati e codifiche per le osservazioni e le misure.
- **SOS** (Sensor Observation Service): modello ed interfaccia per la distribuzione di informazioni ed osservazioni.

SWE: i suoi *standards*

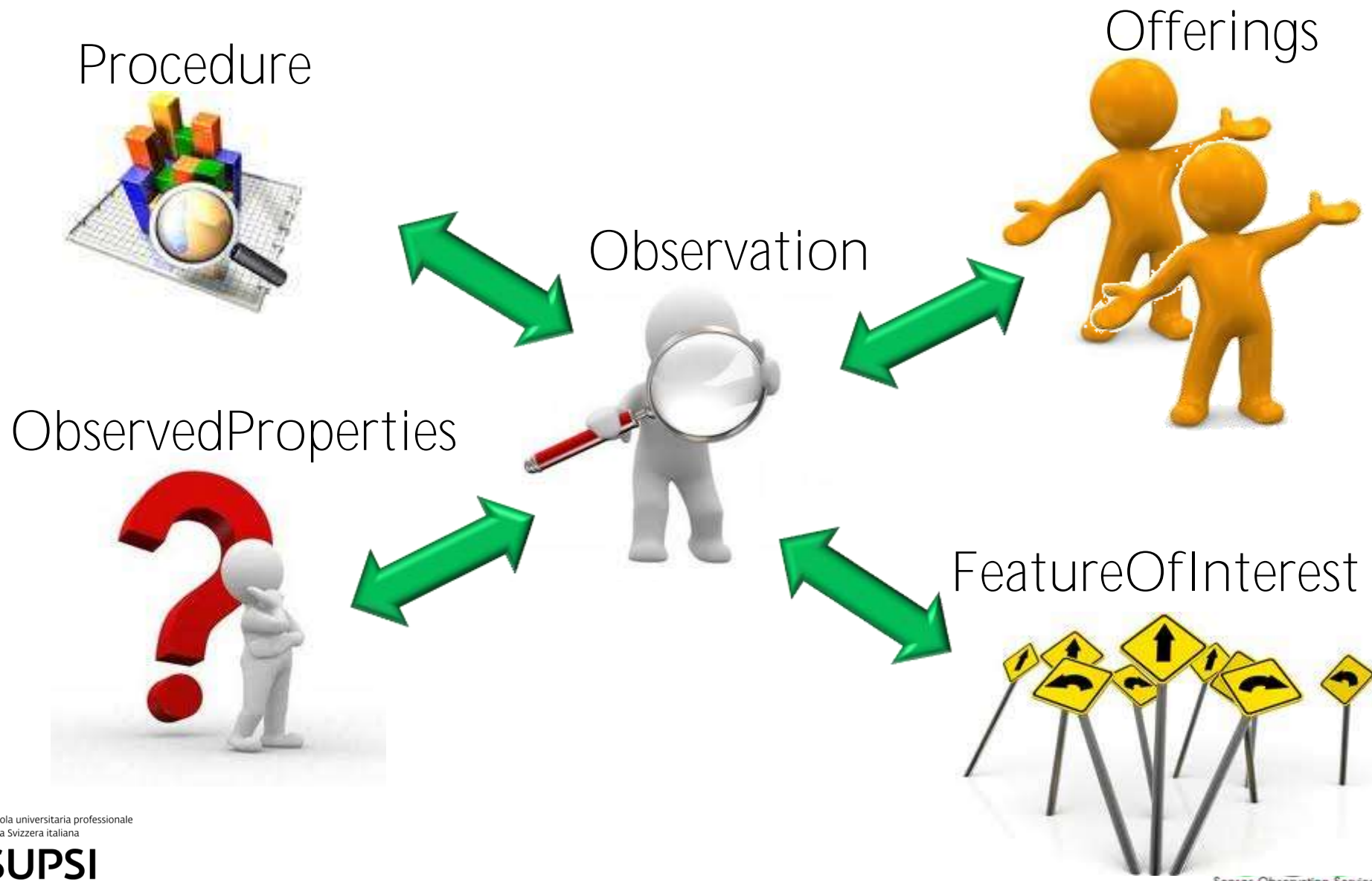
- **SPS** (Sensor Planning Service): modello ed interfaccia per l'esecuzione di operazioni da parte dei sensori.
- **SAS** (Sensor Alert Service): modello ed interfaccia per la notifica ad utenti registrati di informazioni legate ad un evento d'interesse basata sulla valutazione continua di osservazioni di sensori.
- **WNS** (Web Notification Service): modello ed interfaccia per la notifica ad utenti registrati di informazioni tramite diversi protocolli così come scelto dall'utente.

Sensor Observation Service

- L'SOS è un'interfaccia per accedere a dati rilevati da sensori. Questa prevede un servizio Web capace di rispondere a richieste di dati e informazioni di sensori secondo le specifiche riportate nel documento "OGC 06-009r6" (SOS versione 1.0)



Elementi chiave dell'SOS



Sensor Observation Service

Le possibili richieste supportate da un SOS sono:

Obbligatorie

(*core profile*)

1. GetCapabilities
2. DescribeSensor
3. GetObservation

Opzionali

(*transactional profile*)

4. RegisterSensor
5. InsertObservation

Opzionali

(*enhanced profile*)

6. GetFeatureOfInterest
7. GetResult
8. GetObservationByID
9. GetFeatureOfInterestTime
10. DescribeFeatureType
11. DescribeObservationType
12. DescribeResultModel

SOS: *core profile*

- **GetCapabilities:**
 - Serve per richiedere una descrizione del servizio.
 - Fornisce informazioni su chi mantiene il servizio, sulle operazioni supportate, sui fenomeni e gli elementi osservati, ad altre informazioni secondarie.
- **DescribeSensor:**
 - Serve per richiedere informazioni di un determinato sensore.
 - Fornisce informazioni su un sensore o gruppo di sensori utilizzati in una procedura secondo il formato SensorML.
- **GetObservation:**
 - Serve per richiedere osservazioni eventualmente specificando dei filtri.
 - Fornisce i dati rilevati dai sensori secondo il formato O&M.

SOS: *transactional profile*

- RegisterSensor :
 - Serve per richiedere la registrazione di un nuovo sensore.
- InsertObservation :
 - Serve per richiedere l'inserimento di nuove osservazioni relative ad un sensore.

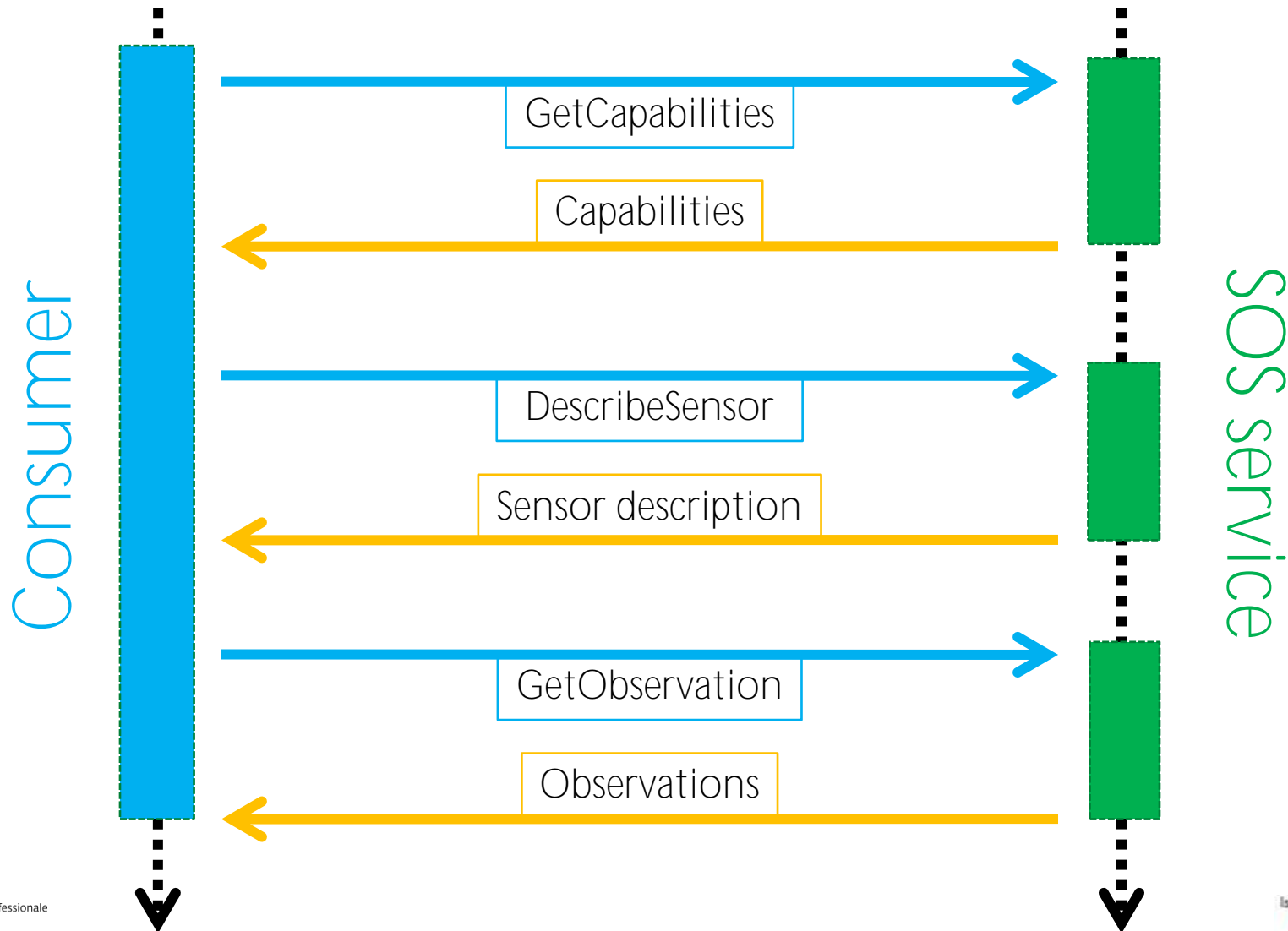
SOS: *enhanced profile*

- **GetFeatureOfInterest:**
 - Serve per richiedere un elemento osservato.
 - Fornisce l'elemento geometrico secondo il formato GML.
- **GetResult:**
 - Serve per richiedere ripetutamente osservazioni senza dover sottomettere una piena richiesta GetObservation.
 - Fornisce i dati rilevati dai sensori secondo un formato "ridotto" dell'O&M
- **GetObservationByID:**
 - Serve per richiedere osservazioni secondo un determinato ID.
 - Fornisce i dati rilevati dai sensori secondo il formato O&M.

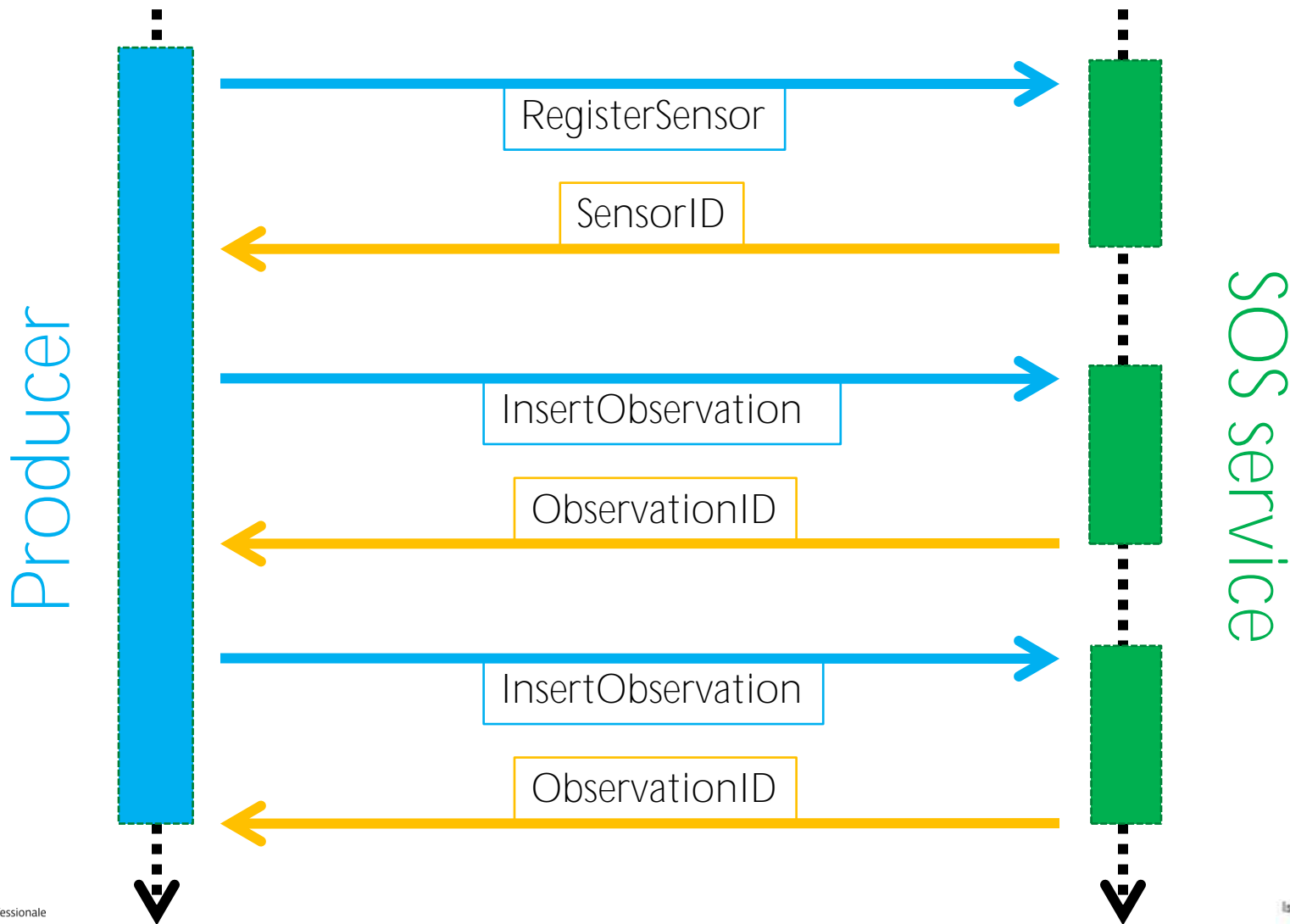
SOS: *enhanced profile*

- GetFeatureOfInterestTime
 - Serve per richiedere in quale periodo il servizio ha osservazioni legate ad un certo elemento d'interesse.
- DescribeFeatureType
 - Serve per descrivere lo schema XML con il quale un elemento d'interesse è rappresentato
- DescribeObservationType
 - Serve per recuperare lo schema XML con il quale quella osservazione è rappresentata (può essere uno schema particolare legato ad un campo applicativo, es. chimica, meteo, etc..)
- DescribeResultModel
 - Serve per recuperare lo schema XML utilizzato per rappresentare l'elemento result rappresentato nell'observation (GetObservation)

Data consumer



Data producer



il linguaggio

- Il linguaggio utilizzato per comunicare con il servizio SOS è l'**XML** (eXtensible Markup Language). A dispetto del nome non si tratta propriamente di un linguaggio, ma di un *meta linguaggio*, cioè un linguaggio per costruire altri linguaggi.
- L'XML consente di separare il contenuto dalla forma in modo che sia: facilmente comprensibile, indipendente dal tipo di software, ed estremamente versatile.

istSOS (descrizione del software)

Scuola universitaria professionale
della Svizzera italiana

SUPSI

Istituto Scienze della Terra



Sensor Observation Service

istSOS

is the SOS implementation by the
Istituto scienze della Terra
(Institute of Earth sciences)



<http://sites.google.com/site/istsosproject>

Licenza GPL v2

- This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Tecnologia

istSOS è interamente sviluppato in
Python e si basa su
Apache/Mod_WSGI,
PostgreSQL/PostGIS e GDAL/OGR



isodate

APACHE
HTTP SERVER



mod_wsgi



psycopg2



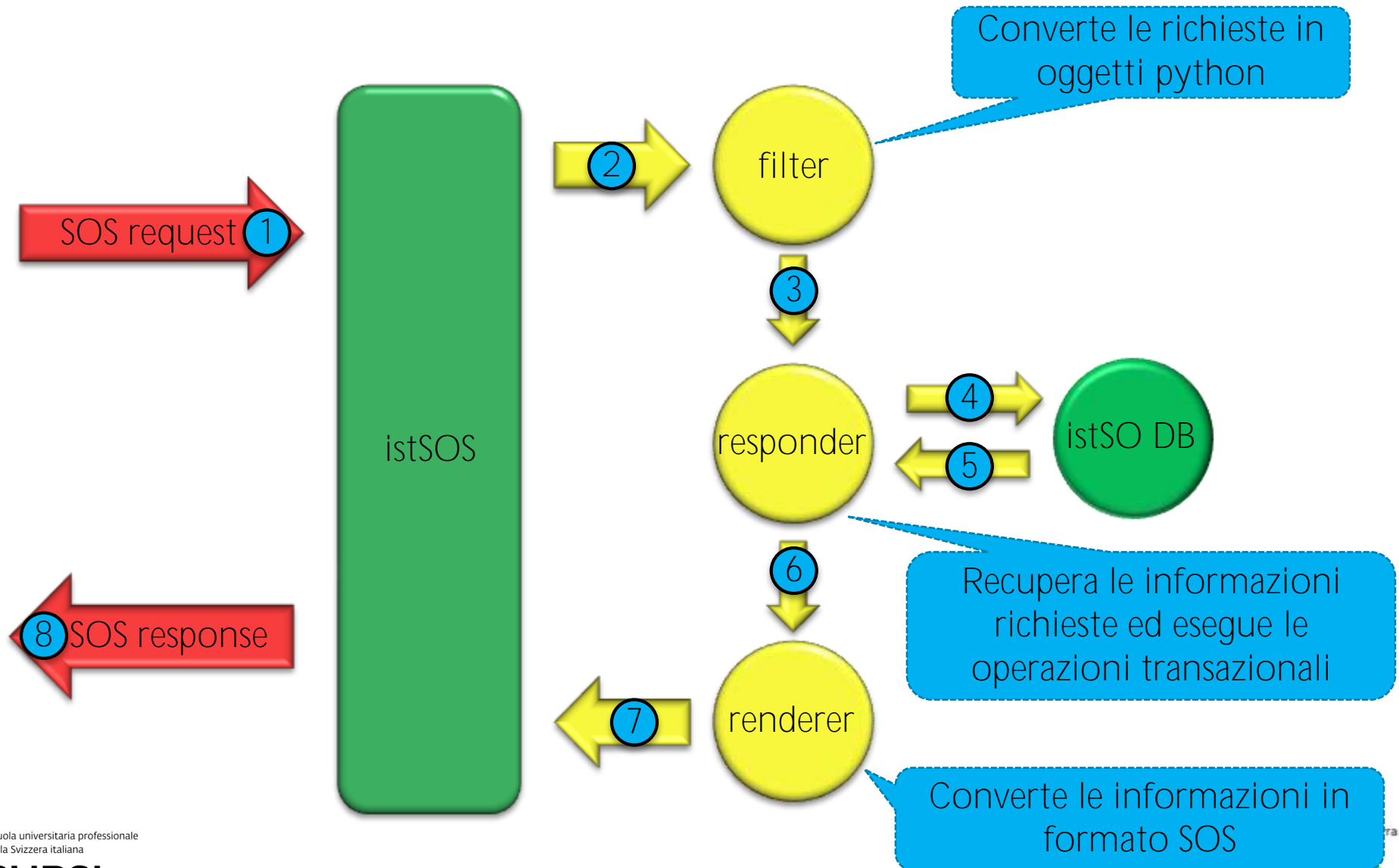
Design pattern

istSOS è stato implementato con un design pattern (schema di progettazione) di tipo “factory”

Questo consente di
istanziare
automaticamente oggetti
diversi in funzione del
tipo di richiesta.



istSOS: struttura



Richieste supportate

- CoreProfile:
 - GetCapabilities
 - DescribeSensor
 - GetObservation
- Transactional Profile:
 - RegisterSensor
 - InsertObservation
- Enhanced profile:
 - GetFeatureOfInterest

Non ancora sviluppate

- Enhanced profile:
 - GetResult
 - GetObservationByID
 - GetFeatureOfInterestTime
 - DescribeFeatureType
 - DescribeObservationType
 - DescribeResultModel

Sensori

Per istSOS vale

sensore == serie temporale

quindi:

“un insieme di valori fisso per ciascun istante”

T,P Sensor



R Sensor



XYZ Sensor

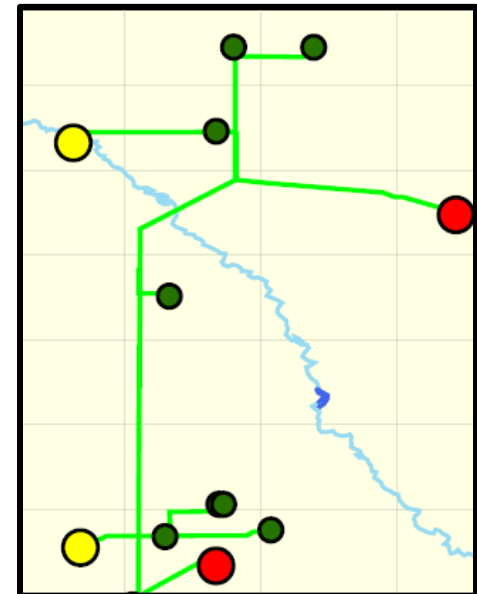


Istituto Scienze della Terra
IST SOS
Sensor Observation Service

Istituto Scienze della Terra
IST SOS
Sensor Observation Service

FOI: featureOfInterest

- Secondo le specifiche può essere qualsiasi cosa ma per istSOS è la geometria osservata e mai il mezzo (arie, acqua, etc..) (es.: un punto, una rete, una regione)
- Lo schema istSOS permette di associare una sola FOI per ciascuna Procedure



Tipi di sensori

- Il modello di dati è differente per ciascuna tipologia di sensore, così istSOS ha deciso di distinguere i sensori secondo i seguenti criteri :
 - Tipo di osservazione:
 - Discreta – distribuita
 - stazionarietà:
 - fissa – mobile
 - Modalità di campionamento:
 - in situ – remota

Tipi di sensori supportati

Fino ad oggi istSOS supporta due tipi di sensori

"fixpoint"

=

in situ – fisso – discreto puntuale

Posizione di campionamento data
dal rispettivo FOI

[2010-09-05T12:10+02:00,
127,0.44]



"mobilepoint"

=

in situ – mobile- discreto puntuale

Posizione di campionamento data da una
tripletta x,y,z

[2010-09-05T12:10+02:00,
697812,78562,873.23,12.7]



istSOS (installazione)

Scuola universitaria professionale
della Svizzera italiana

SUPSI

Istituto Scienze della Terra



Sensor Observation Service

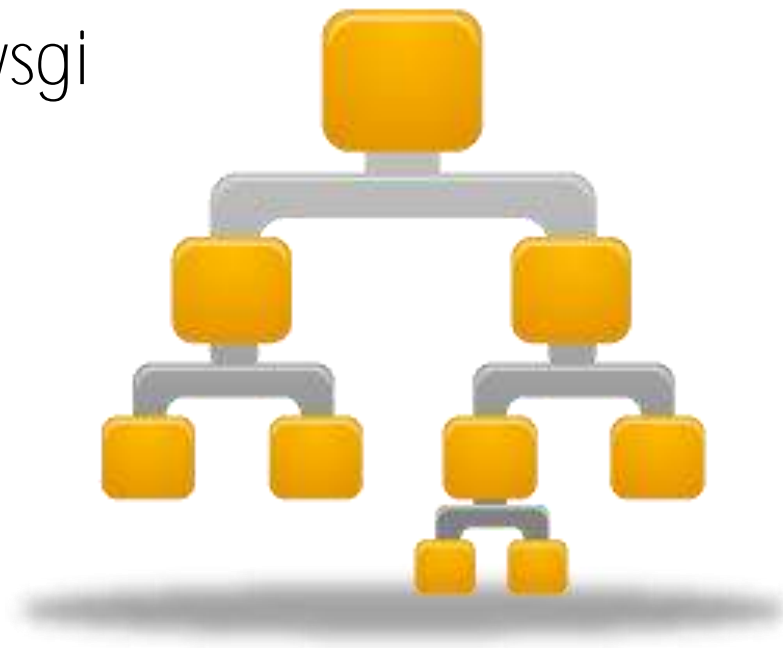
Installazione su Ubuntu

1. Installazione dipendenze
2. Installazione dello schema
3. Installazione delle librerie
4. Configurazione di apache
5. Configurazione di istSOS



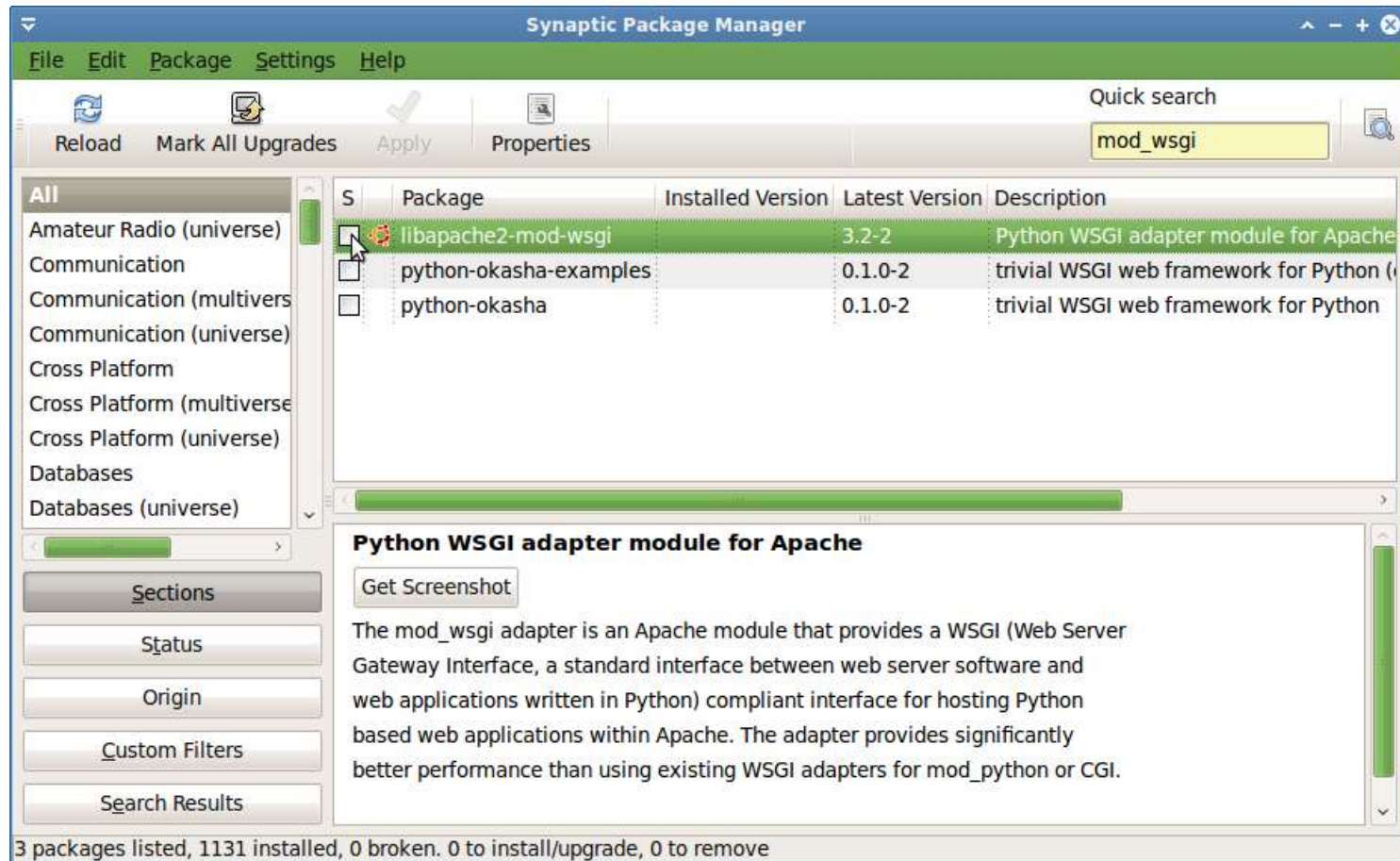
dipendenze

- Requisiti di base:
 - Python (2.6.x)
 - PostgreSQL/PostGIS
 - Apache (2.x >) con mod_wsgi
- Pacchetti Python:
 - psycopg2
 - isodate
 - GDAL



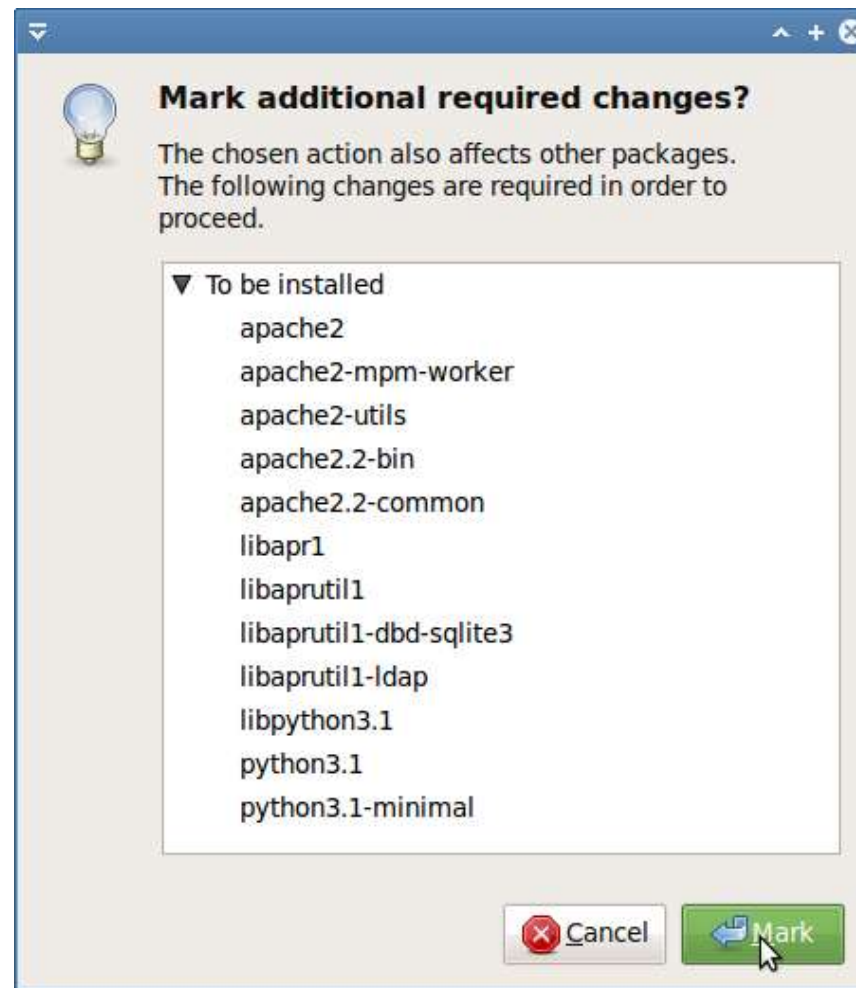
Apache e mod_wsgi

- Synaptic: libapache2-mod-wsgi



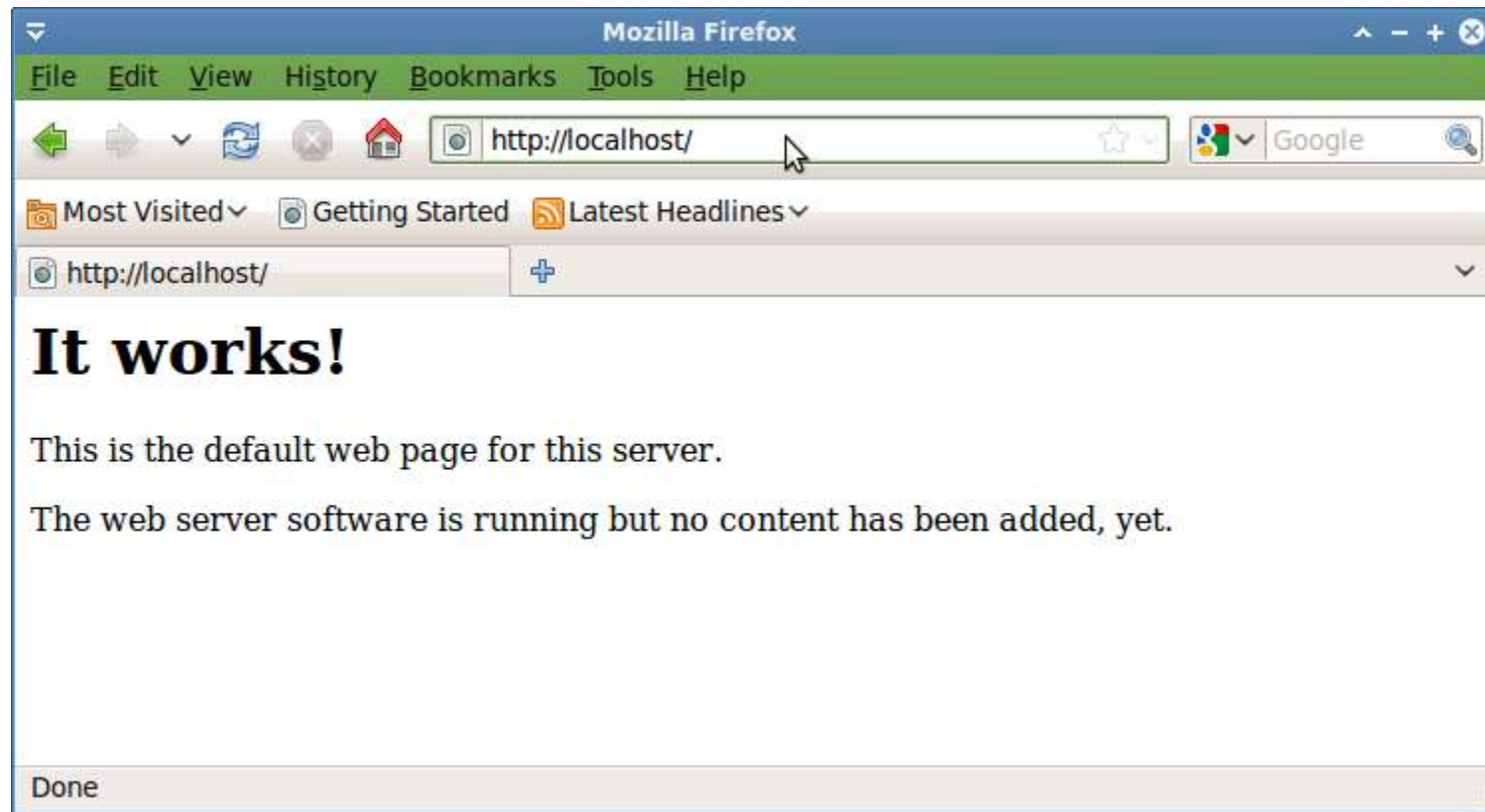
Apache e mod_wsgi

- Dipendenze installate



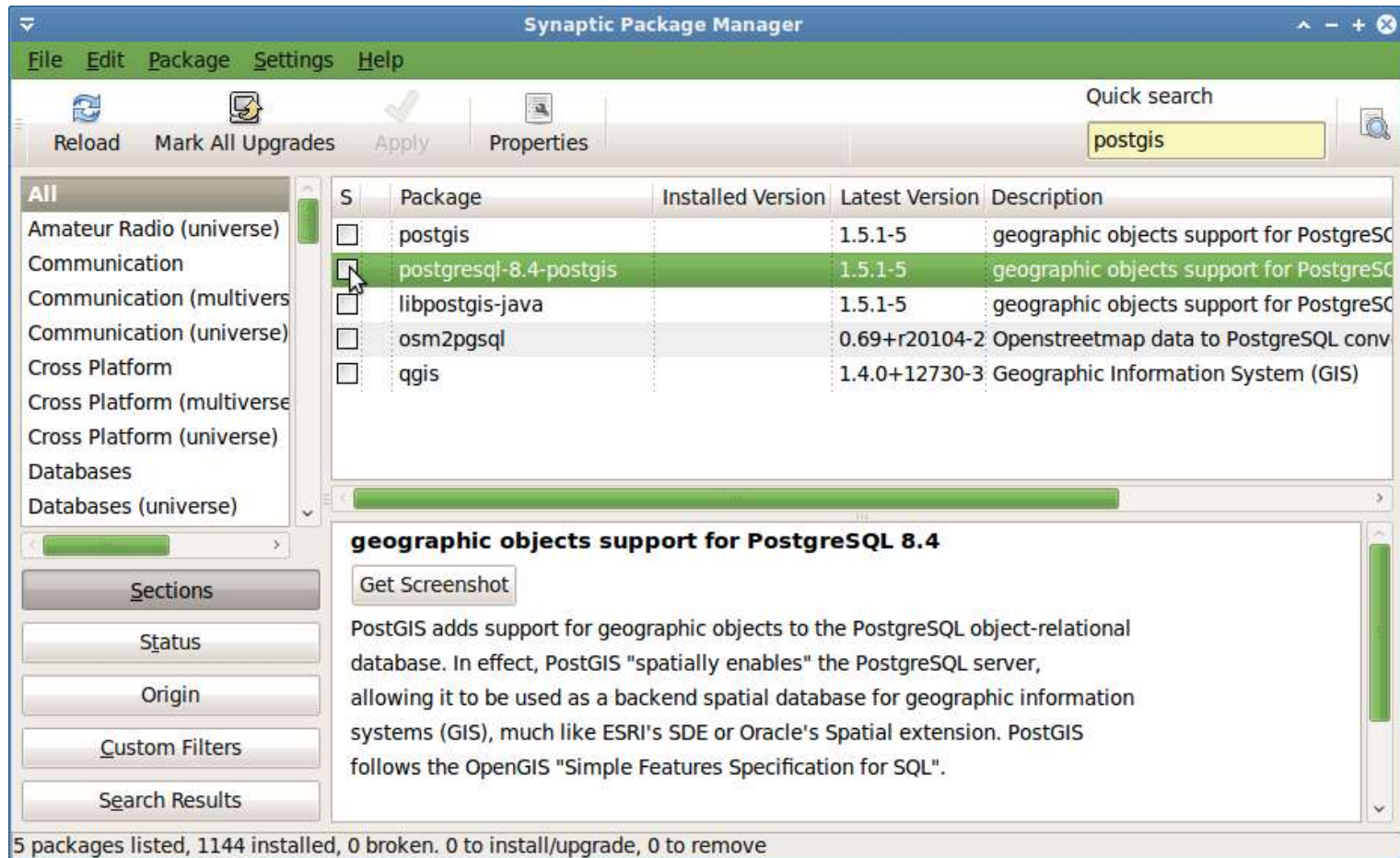
Test di Apache

- Verifichiamo funzionamento di Apache



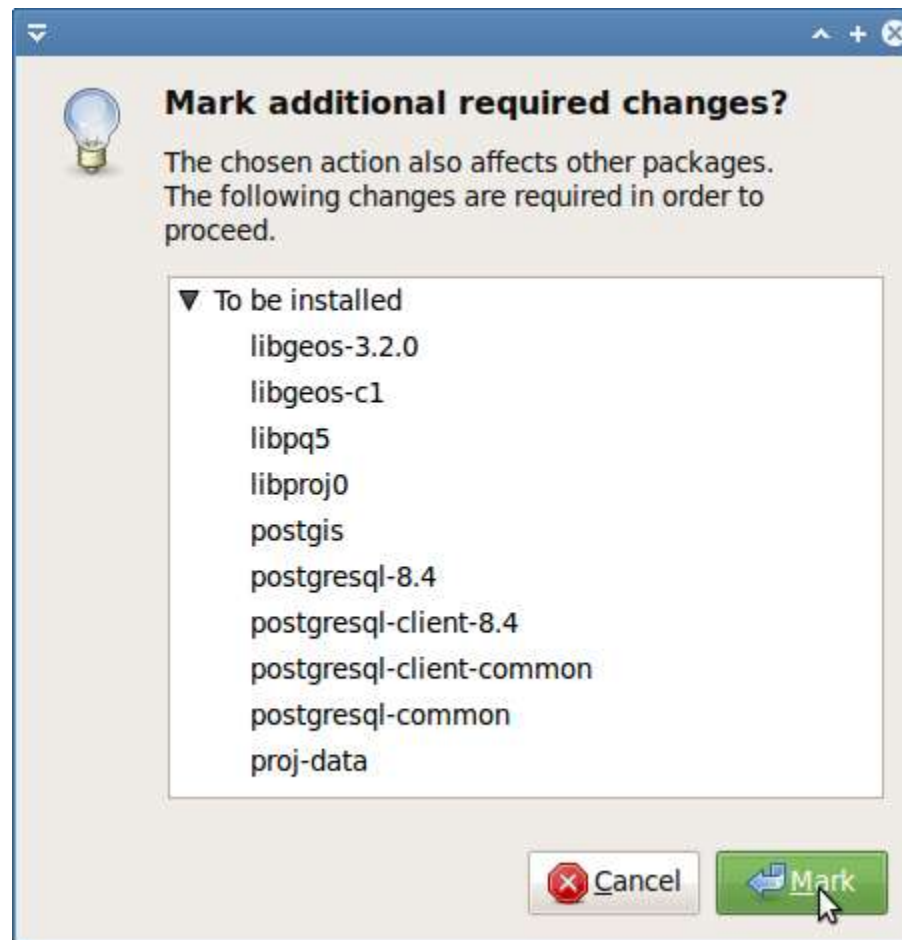
PostgreSQL / PostGIS

- Synaptic: postgresql-8.4-postgis



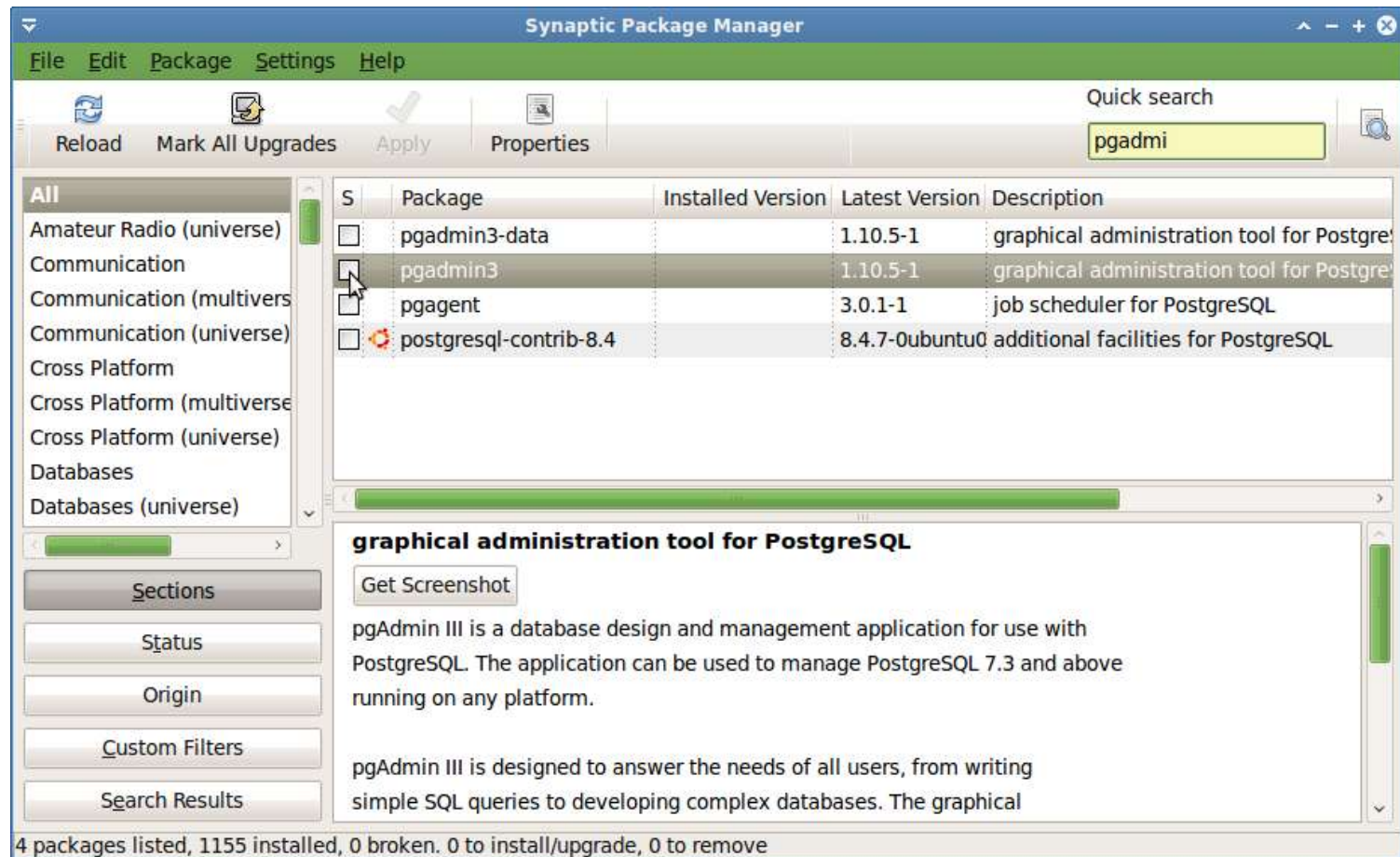
PostgreSQL / PostGIS

- Dipendenze installate



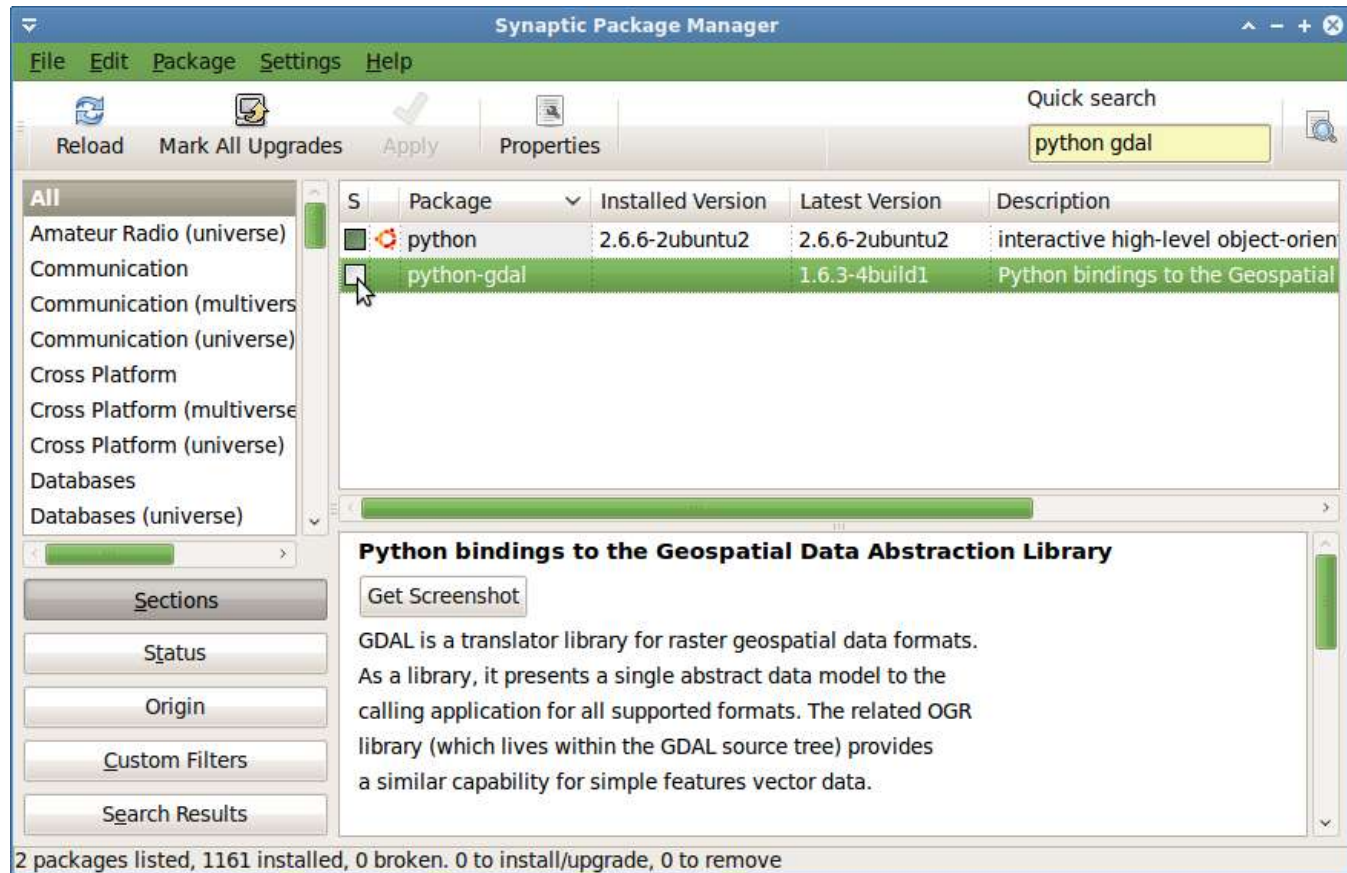
PgAdmin3 (utilities)

- Synaptic: pgadmin3



Python-gdal

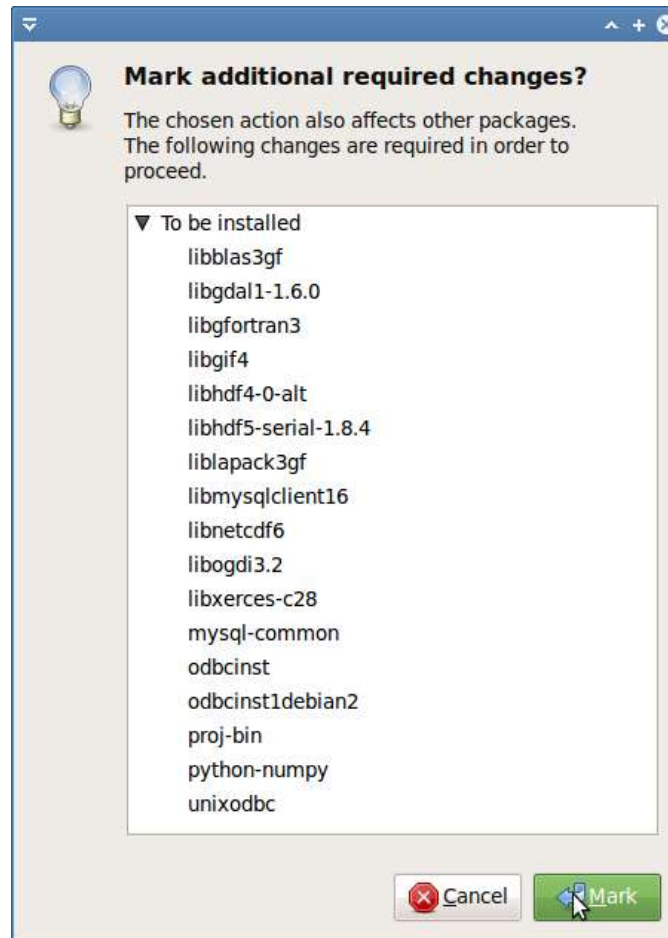
- Synaptic: python-gdal



Con easy_install gdal è corrotto :- (

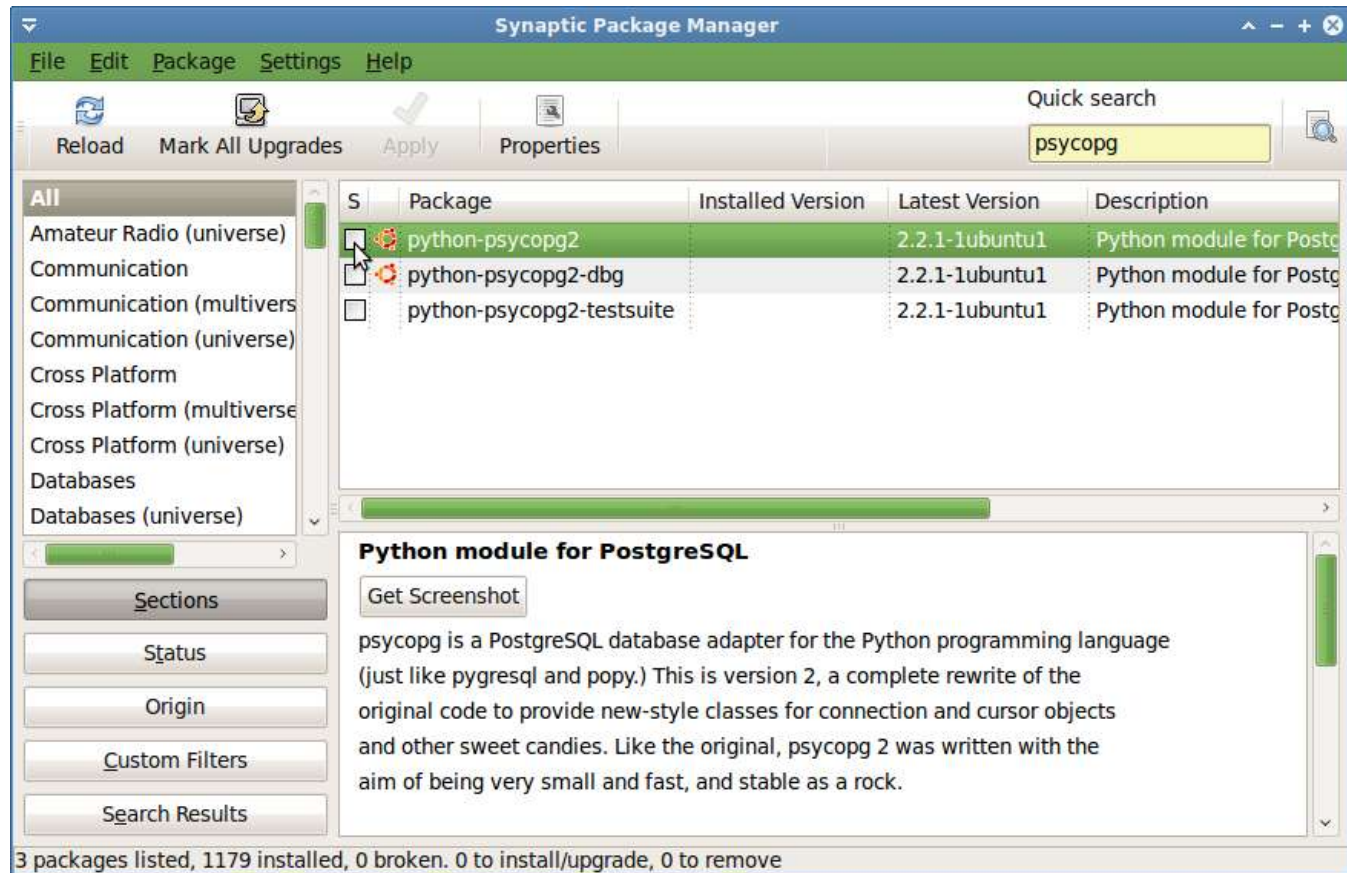
Python-gdal

- Dipendenze installate



psycpg2

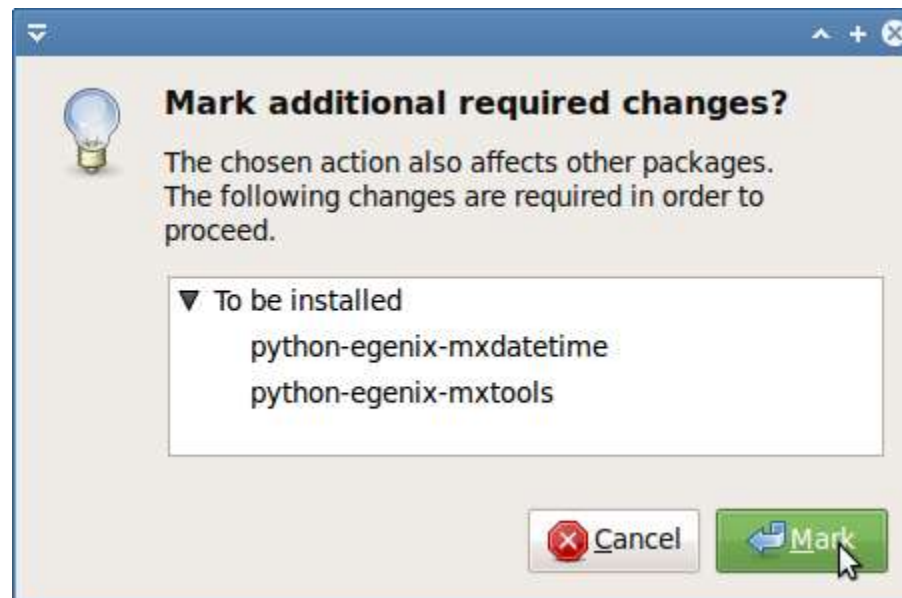
- Synaptic: python-psycpg2



Con easy_install è corrotto :- (

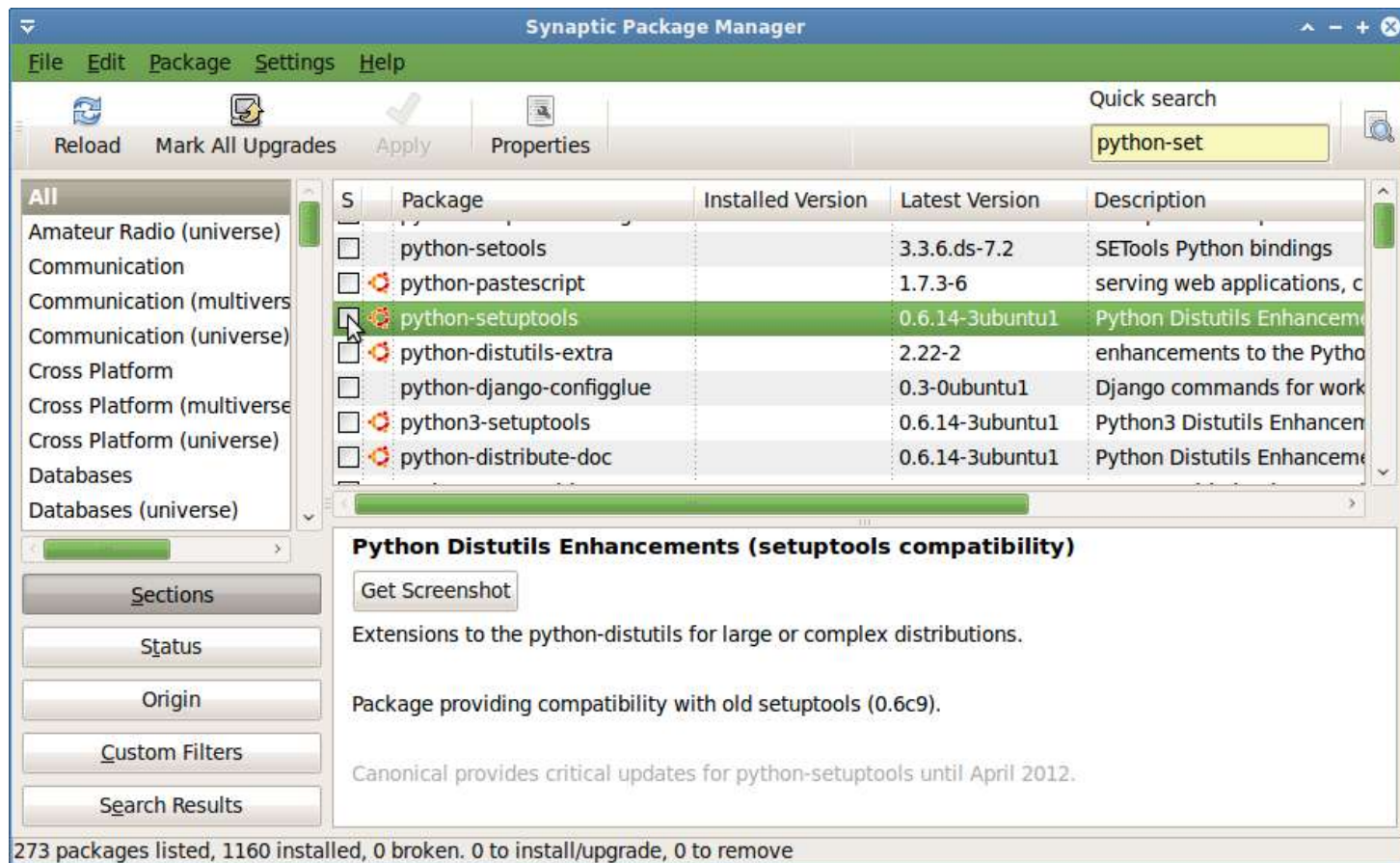
psycopg2

- Dipendenze installate



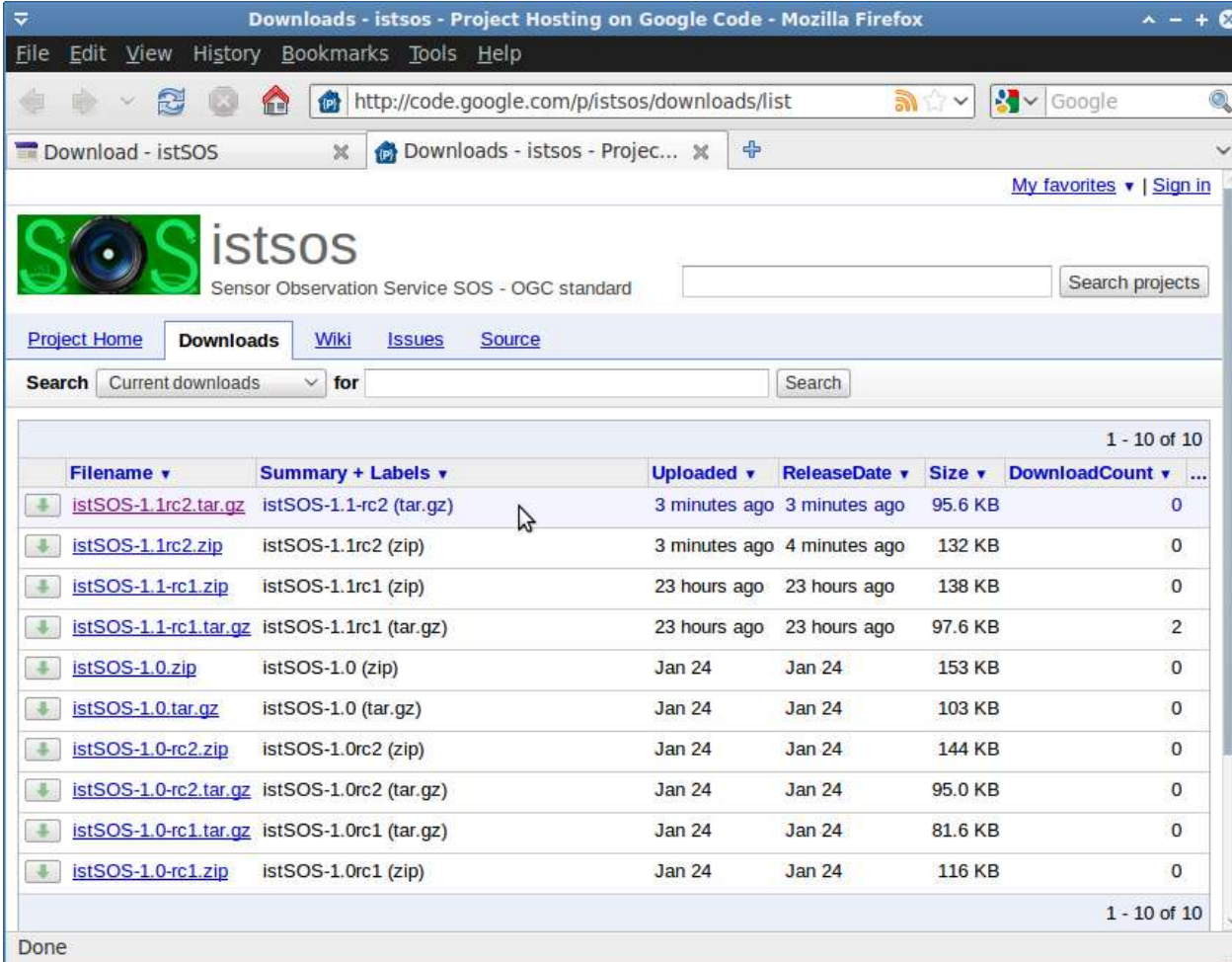
setuptools

- Synaptic: python-setuptools



Download istSOS

<http://code.google.com/p/istsos>



The screenshot shows the Google Code project page for istSOS. The browser window is titled "Downloads - istsos - Project Hosting on Google Code - Mozilla Firefox". The address bar shows the URL <http://code.google.com/p/istsos/downloads/list>. The page header includes the istSOS logo and the text "Sensor Observation Service SOS - OGC standard". Below the header, there are tabs for "Project Home", "Downloads", "Wiki", "Issues", and "Source". The "Downloads" tab is selected, and a search bar is visible. The main content area displays a table of downloads, showing 10 items. The table has columns for Filename, Summary + Labels, Uploaded, ReleaseDate, Size, and DownloadCount. The files listed are various versions of istSOS-1.1-rc2, istSOS-1.1-rc1, istSOS-1.0, and istSOS-1.0-rc2, available in both tar.gz and zip formats. The most recent files are istSOS-1.1-rc2.tar.gz and istSOS-1.1-rc2.zip, both uploaded 3 minutes ago.

Filename	Summary + Labels	Uploaded	ReleaseDate	Size	DownloadCount
istSOS-1.1-rc2.tar.gz	istSOS-1.1-rc2 (tar.gz)	3 minutes ago	3 minutes ago	95.6 KB	0
istSOS-1.1-rc2.zip	istSOS-1.1-rc2 (zip)	3 minutes ago	4 minutes ago	132 KB	0
istSOS-1.1-rc1.zip	istSOS-1.1-rc1 (zip)	23 hours ago	23 hours ago	138 KB	0
istSOS-1.1-rc1.tar.gz	istSOS-1.1-rc1 (tar.gz)	23 hours ago	23 hours ago	97.6 KB	2
istSOS-1.0.zip	istSOS-1.0 (zip)	Jan 24	Jan 24	153 KB	0
istSOS-1.0.tar.gz	istSOS-1.0 (tar.gz)	Jan 24	Jan 24	103 KB	0
istSOS-1.0-rc2.zip	istSOS-1.0-rc2 (zip)	Jan 24	Jan 24	144 KB	0
istSOS-1.0-rc2.tar.gz	istSOS-1.0-rc2 (tar.gz)	Jan 24	Jan 24	95.0 KB	0
istSOS-1.0-rc1.tar.gz	istSOS-1.0-rc1 (tar.gz)	Jan 24	Jan 24	81.6 KB	0
istSOS-1.0-rc1.zip	istSOS-1.0-rc1 (zip)	Jan 24	Jan 24	116 KB	0

istSOS package

sosConfig.py

file di configurazione

sos.py

interfaccia Web

istSOS

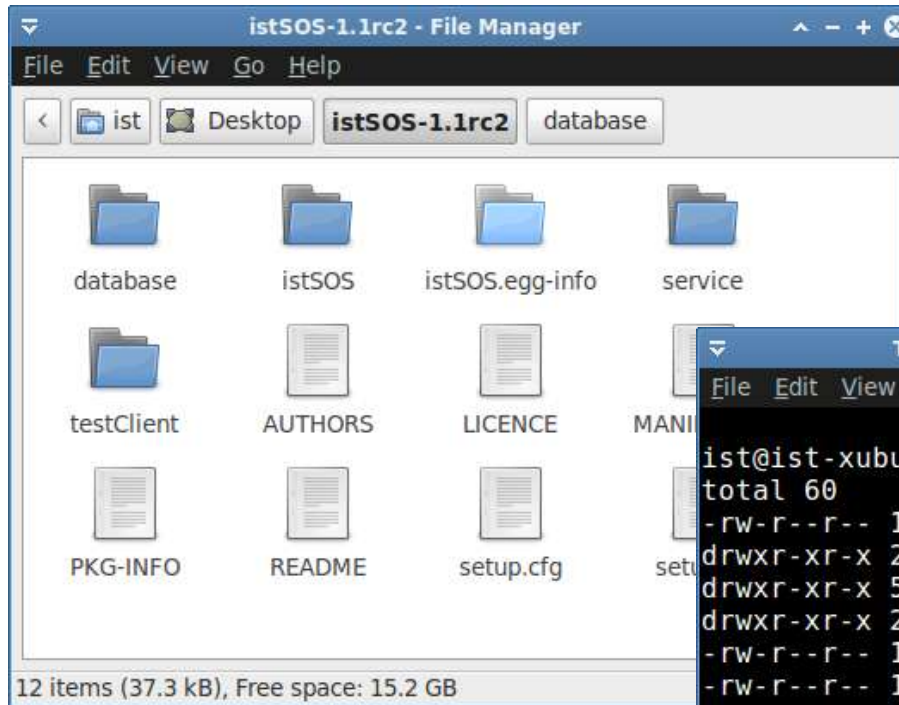
libreria istSOS

sos_schema.sql

schema PostGIS

istSOS

- Decomprimiamo e lanciamo un terminal...



The screenshot shows a terminal window titled "Terminal - ist@ist-xubuntu: ~/Desktop/istSOS-1.1rc2". The terminal has a menu bar with "File", "Edit", "View", "Terminal", "Go", and "Help". The command "ls -l" has been executed, and the output is displayed as follows:

```
ist@ist-xubuntu:~/Desktop/istSOS-1.1rc2$ ls -l
total 60
-rw-r--r-- 1 ist ist 189 2010-10-07 15:30 AUTHORS
drwxr-xr-x 2 ist ist 4096 2011-02-08 14:44 database
drwxr-xr-x 5 ist ist 4096 2011-02-08 14:44 istSOS
drwxr-xr-x 2 ist ist 4096 2011-02-08 14:44 istSOS.egg-info
-rw-r--r-- 1 ist ist 15402 2010-10-07 15:30 LICENCE
-rw-r--r-- 1 ist ist 207 2011-02-08 14:43 MANIFEST.in
-rw-r--r-- 1 ist ist 733 2011-02-08 14:44 PKG-INFO
-rw-r--r-- 1 ist ist 261 2010-10-07 15:30 README
drwxr-xr-x 4 ist ist 4096 2011-02-08 14:44 service
-rw-r--r-- 1 ist ist 59 2011-02-08 14:44 setup.cfg
-rw-r--r-- 1 ist ist 894 2011-02-08 14:43 setup.py
drwxr-xr-x 3 ist ist 4096 2011-02-08 14:44 testClient
```


Librerie istSOS

Submodules

- [istSOS.filters](#)
 - [istSOS.filters.DS_filter](#)
 - [istSOS.filters.GC_filter](#)
 - [istSOS.filters.GF_filter](#)
 - [istSOS.filters.GO_filter](#)
 - [istSOS.filters.IO_filter](#)
 - [istSOS.filters.RS_filter](#)
 - [istSOS.filters.USD_filter](#)
 - [istSOS.filters.factory_filters](#)
 - [istSOS.filters.filter](#)
- [istSOS.renderers](#)
 - [istSOS.renderers.DSresponseRender](#)
 - [istSOS.renderers.GCresponseRender](#)
 - [istSOS.renderers.GFresponseRender](#)
 - [istSOS.renderers.GOresponseRender](#)
 - [istSOS.renderers.IOresponseRender](#)
 - [istSOS.renderers.RSresponseRender](#)
 - [istSOS.renderers.USDresponseRender](#)
 - [istSOS.renderers.factory_render](#)
- [istSOS.responders](#)
 - [istSOS.responders.DSresponse](#)
 - [istSOS.responders.GCresponse](#)
 - [istSOS.responders.GFresponse](#)
 - [istSOS.responders.GOresponse](#)
 - [istSOS.responders.IOresponse](#)
 - [istSOS.responders.RSresponse](#)
 - [istSOS.responders.USDresponse](#)
 - [istSOS.responders.factory_response](#)
- [istSOS.sosDatabase](#)
- [istSOS.sosException](#)



python setup.py install

Installiamo istSOS

```
Terminal - ist@ist-xubuntu: ~/Desktop/istSOS-1.1
File Edit View Terminal Go Help
ist@ist-xubuntu:~/Desktop/istSOS-1.1$ sudo python setup.py install
```

```
Terminal - ist@ist-xubuntu: ~/Desktop/istSOS-1.1
File Edit View Terminal Go Help
Using /usr/local/lib/python2.6/dist-packages/isodate-0.4.3-py2.6.egg
Searching for pycopg2==2.2.1
Best match: pycopg2 2.2.1
Adding pycopg2 2.2.1 to easy-install.pth file

Using /usr/lib/python2.6/dist-packages
Finished processing dependencies for istSOS==1.1
ist@ist-xubuntu:~/Desktop/istSOS-1.1$
```

istSOS (creazione del servizio)

Scuola universitaria professionale
della Svizzera italiana

SUPSI

Istituto Scienze della Terra



Sensor Observation Service

creazione del servizio

1. Creazione del DB
2. Configurazione istSOS
3. Configurazione di Apache
4. Test di funzionamento

Inizializzazione database

- > #impostiamo la password del DB
- > sudo su postgres
- > psql -c "ALTER USER postgres
PASSWORD '1234'"
- > # creiamo il DB
- > createdb sos
- > createlang plpgsql sos

Inizializzazione database

```
> # rendiamo il DB spaziale  
> cd /usr/share/postgresql/8.4/contrib/postgis-1.5/  
> psql -d sos -f postgis.sql  
> psql -d sos -f spatial_ref_sys.sql
```

Inizializzazione database

creiamo lo schema

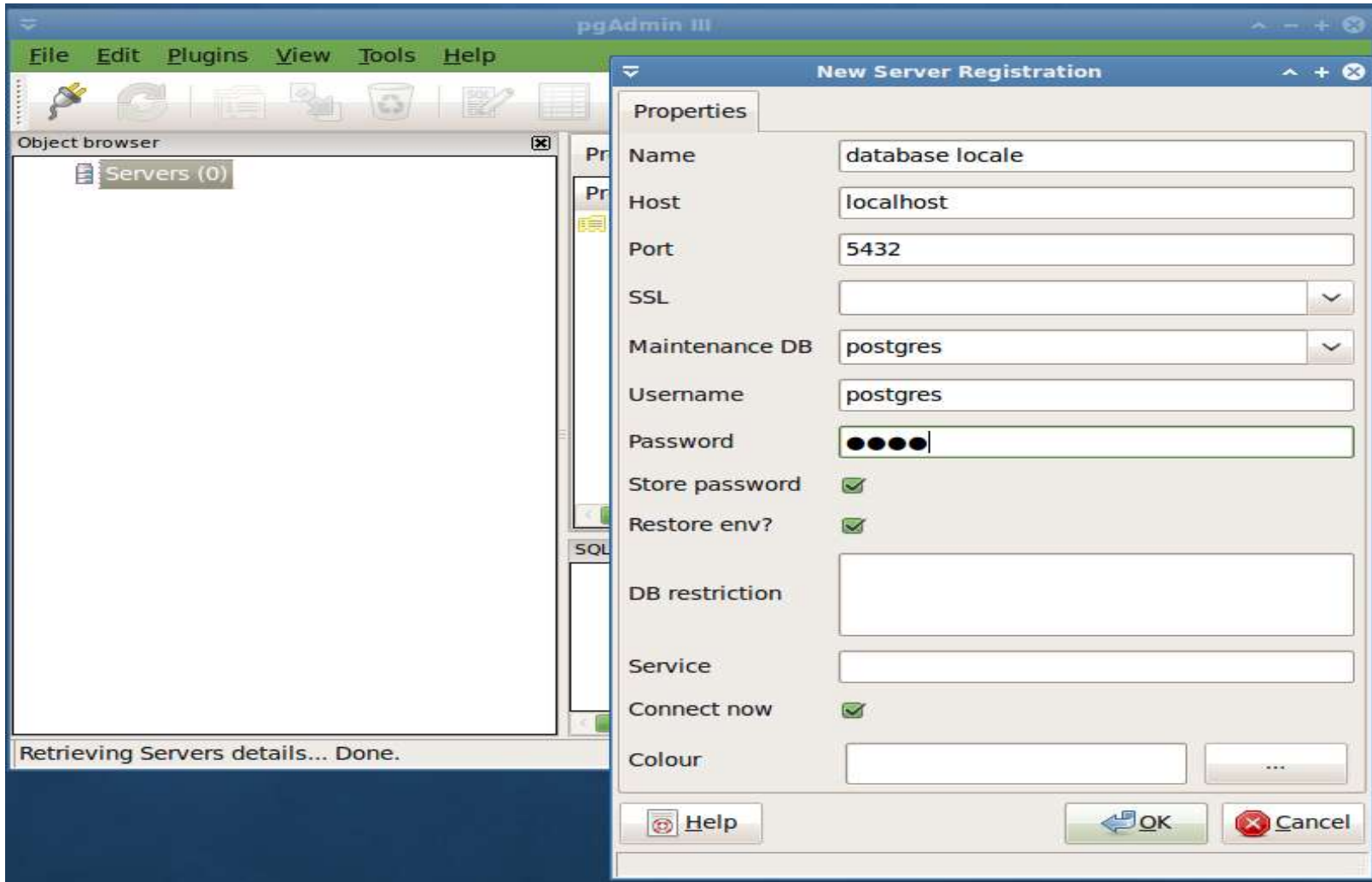
> cd .../istSOS-1.1rc2/database/

> psql -d sos -f sos_schema.sql

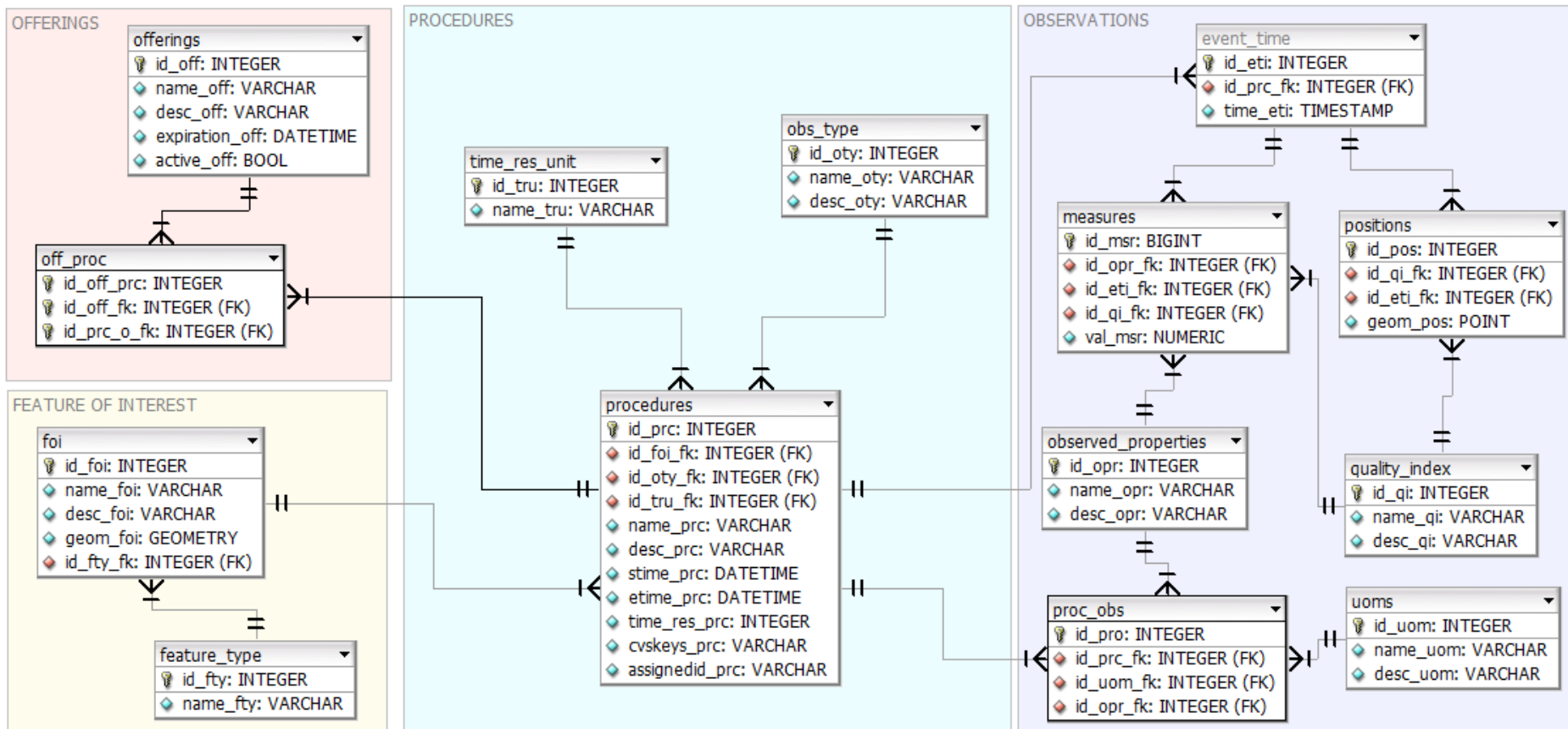
> #logout da postgres !!!

> exit

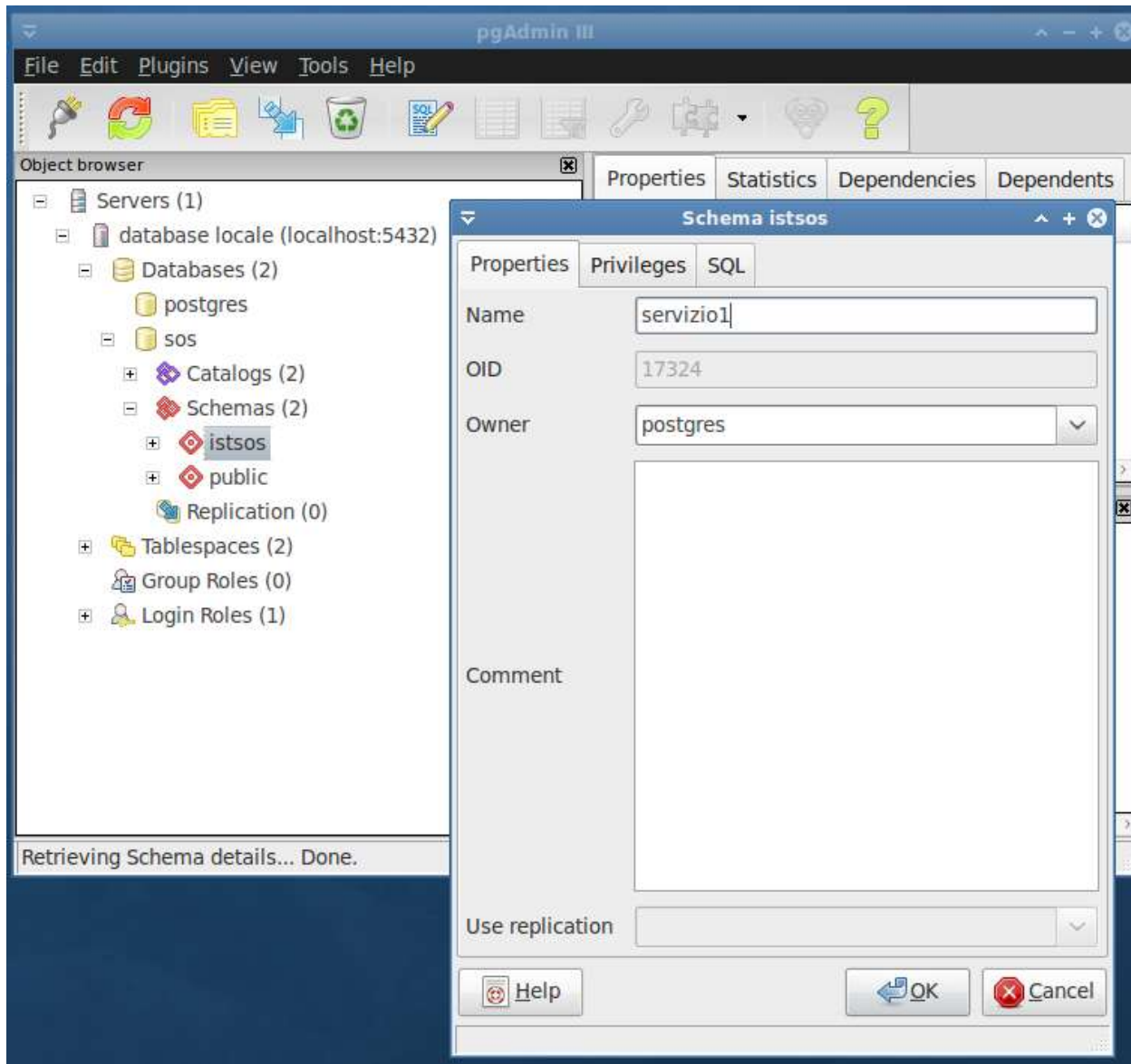
Configurazione PgAdmin



Vediamo il DB istSOS



Rinominiamo lo schema



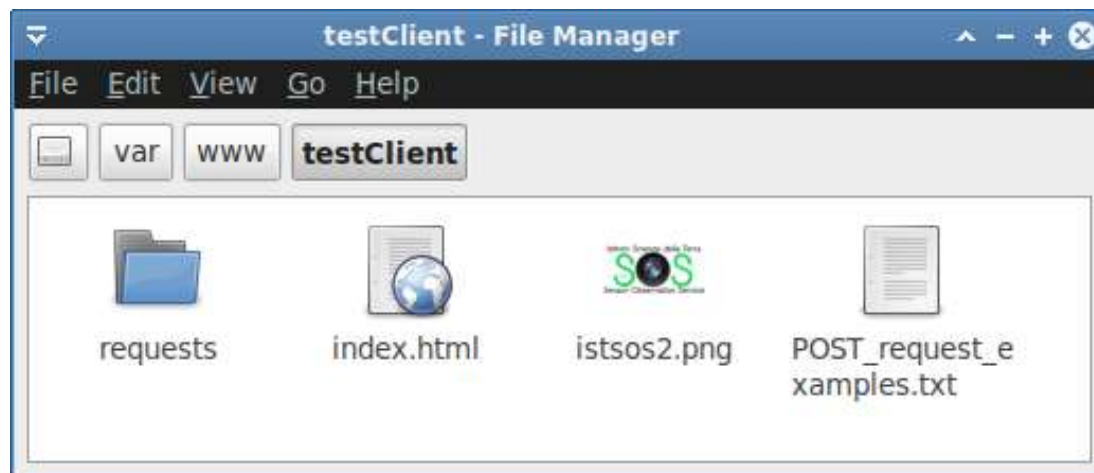
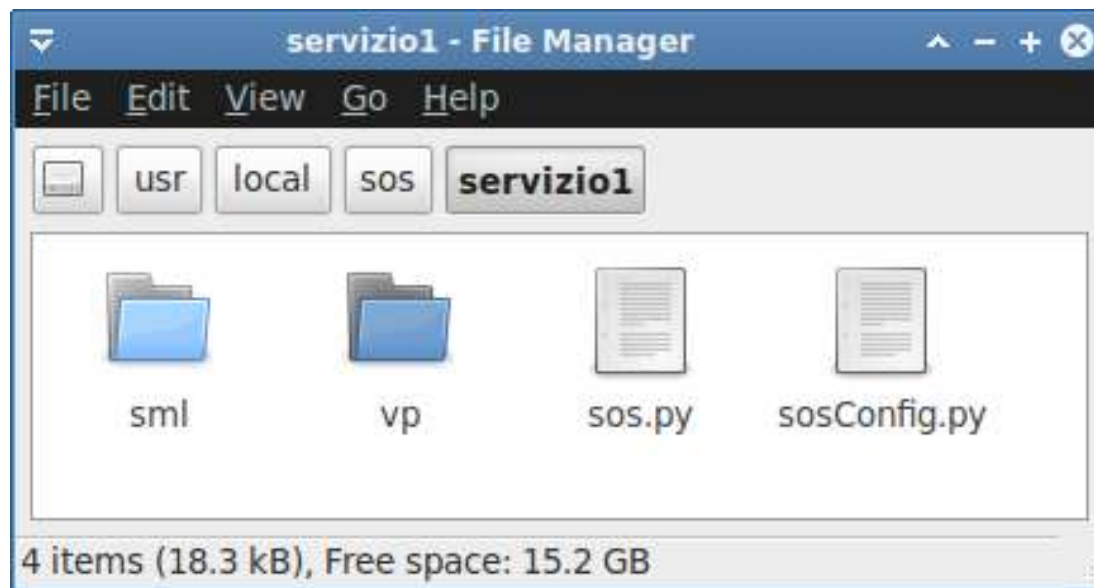
Creazione struttura servizio

- > # creiamo lo spazio per i servizi sos
- > sudo su
- > mkdir /usr/local/sos
- > mkdir /usr/local/sos/servizio1

Creazione struttura servizio

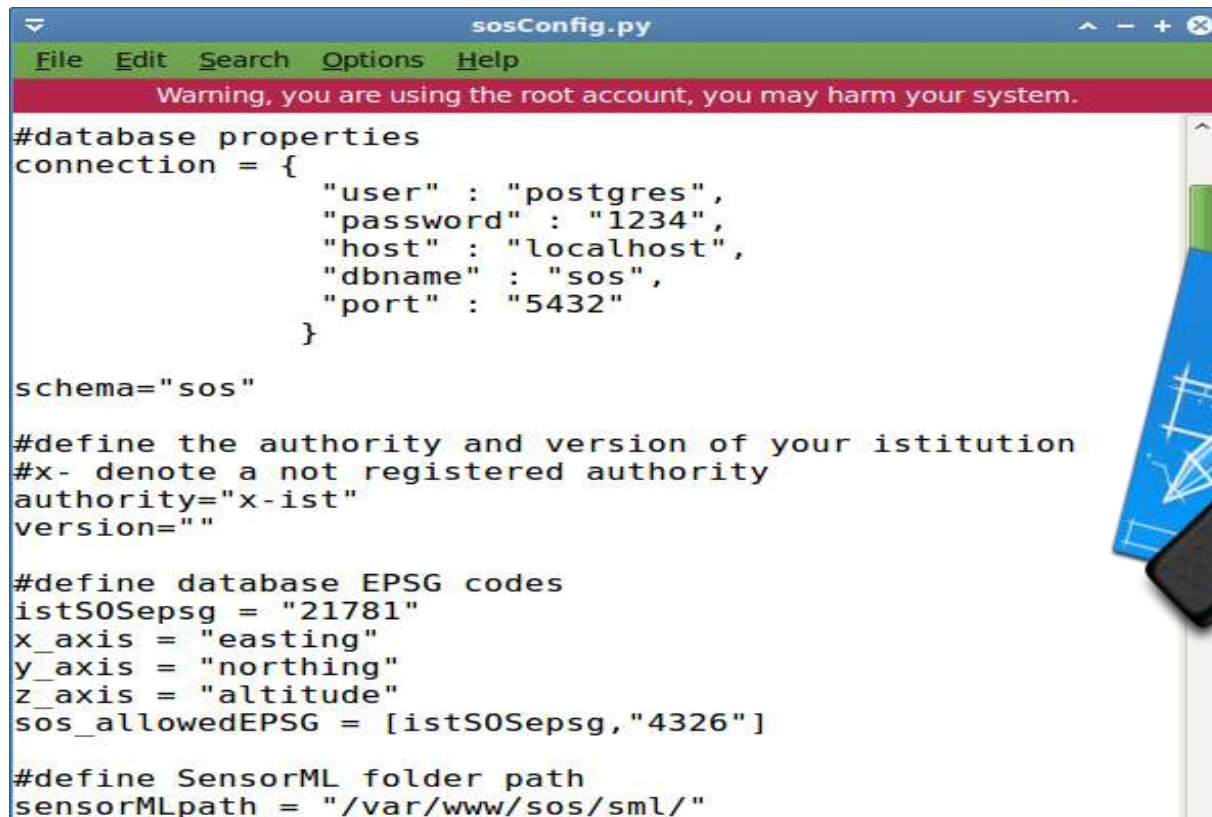
- > # copiamo i file di istSOS
- > cd .../istSOS-1.1rc2/
- > cp -R service /usr/local/sos/servizio1
- > #copiamo l'interfaccia di test
- > cp -R testClient /var/www/

Struttura di istSOS



Configuriamo istSOS

- > sudo mousepad /usr/local/sos/servizio1/sosConfig.py



```
sosConfig.py
File Edit Search Options Help
Warning, you are using the root account, you may harm your system.

#database properties
connection = {
    "user" : "postgres",
    "password" : "1234",
    "host" : "localhost",
    "dbname" : "sos",
    "port" : "5432"
}

schema="sos"

#define the authority and version of your istitution
#x- denote a not registered authority
authority="x-ist"
version=""

#define database EPSG codes
istS0Sepsg = "21781"
x_axis = "easting"
y_axis = "northing"
z_axis = "altitude"
sos_allowedEPSG = [istS0Sepsg, "4326"]

#define SensorML folder path
sensorMLpath = "/var/www/sos/sml/"
```



Configuriamo istSOS

#database properties

```
connection = {  
    "user" : "postgres",  
    "password" : "1234",  
    "host" : "localhost",  
    "dbname" : "sos",  
    "port" : "5432"  
}
```

schema="servizio1"

#define the authority and version of your institution

#x- denote a not registered authority

authority="x-ist"

version=""

Configuriamo istSOS

#define database EPSG codes

istS0Sepsg = "21781"

x_axis = "east ing"

y_axis = "north ing"

z_axis = "al ti tude"

sos_allowedEPSG = [istS0Sepsg, "4326", "900913"] # 900913 =
Google project ion

#define SensorML folder path

n.b.: la cartella deve esistere con i necessari
permessi (rw)

sensorMLpath = "/usr/local/sos/servizi o1/sml /"

#define the http address of the service

```
serviceUrl = {  
    "get" : "http: //l ocal host/sos/srv1",  
    "post" : "http: //l ocal host/sos/srv1"  
}
```

Configuriamo istSOS

#identify the service

serviceDefinition={

```
"title" : "Tutorial",  
"abstract" : "servizio 1",  
"keywords" : ["SOS", "IST", "SUPSI"],  
"fees" : "NONE",  
"accessConstraints" : "NONE"  
}
```

Configuriamo istSOS

#i nformati ons on servi ce provi der

```
servi ceProvi der={  
  "provi derName" : "I sti tuto Sci enze del l a Terra",  
  "provi derSi te" : "http: //i stgeo. i st. supsi . ch",  
  "servi ceContact" : {  
    "i ndi vi dual Name" : "Massi mi l i ano Cannata",  
    "posi ti onName" : "Geomati ca",  
    "contactI nfo" : {  
      "voi ce" : "6214",  
      "fax" : "6200",  
      "del i veryPoi nt" : "Vi a Trevano",  
      "ci ty" : "Canobbi o",  
      "admi ni strati veArea" : "Ti ci no",  
      "postal Code" : "6952",  
      "country" : "Swi tzerl and",  
      "emai l " : "i nfo@supsi . ch",  
    }  
  }  
}
```

Configuriamo apache

- Configuriamo mod_wsgi per eseguire il servizio:
in /etc/apache2/sites-enabled/000-default
aggiungiamo le seguenti righe:

```
WSGI_ScriptAlias /sos/srv1 /usr/local/sos/servizi01/sos.py
<Directory "/usr/local/sos/servizi01">
    AllowOverride None
    Options None
    Order deny,allow
    Allow from all
</Directory>
```

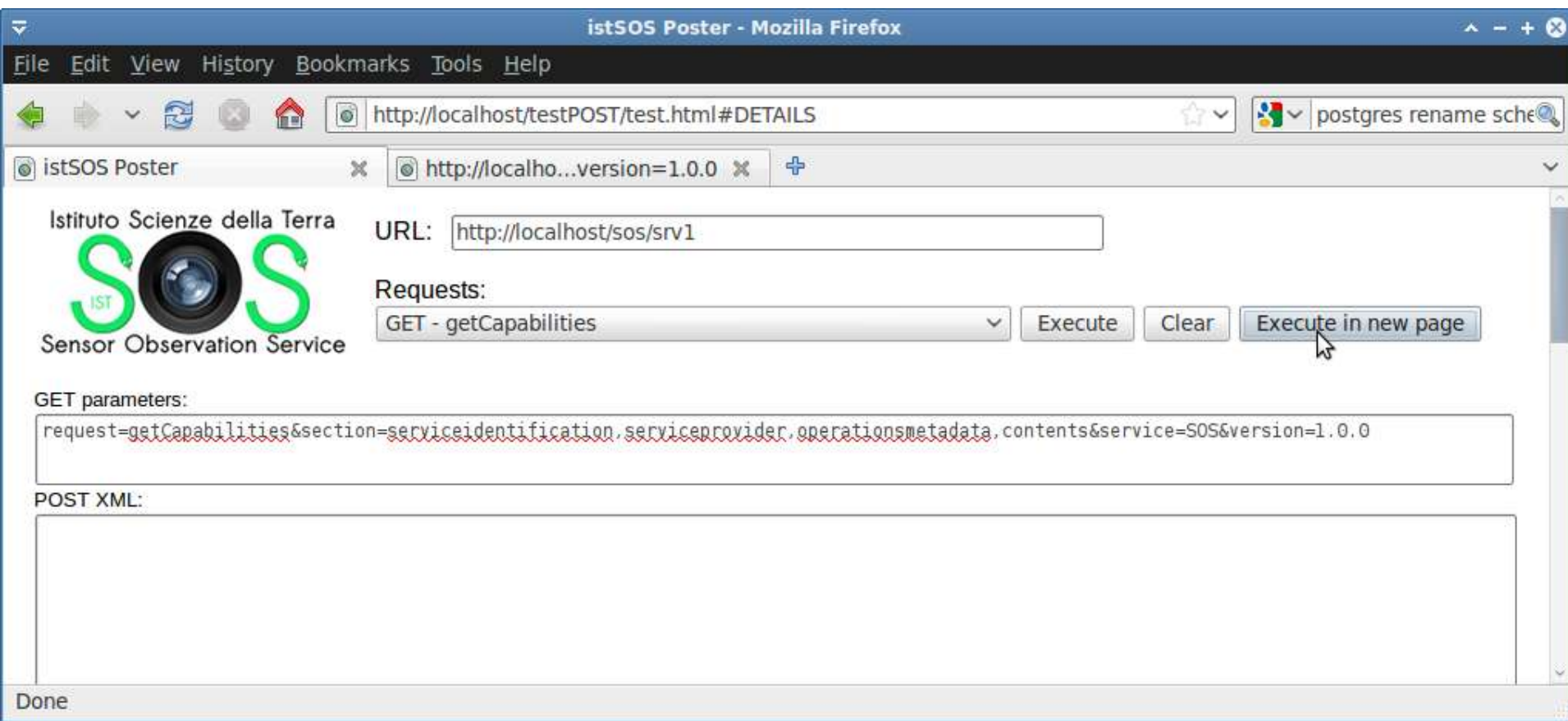


Configuriamo apache

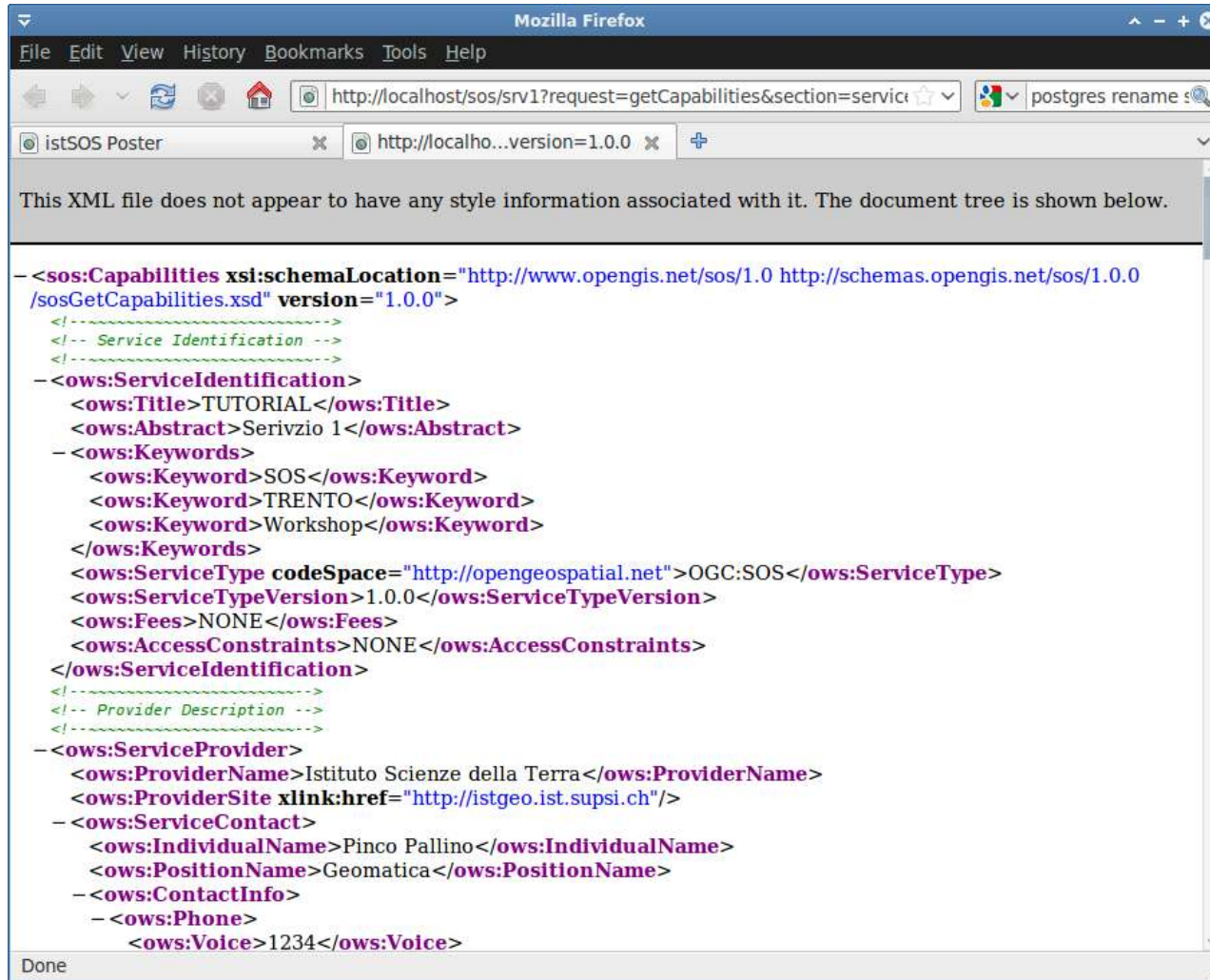
- > # Definiamo il proprietario e i permessi
- > `chown www-data:www-data -R /usr/local/sos`
- > `sudo 755 -R /usr/local/sos`
- > # Riavviamo Apache
- > `sudo /etc/init.d/apache restart`

Verifica del servizio

- <http://localhost/testClient/>



La risposta



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Service Identification -->
<ows:ServiceIdentification>
  <ows:Title>TUTORIAL</ows:Title>
  <ows:Abstract>Servizio 1</ows:Abstract>
  <ows:Keywords>
    <ows:Keyword>SOS</ows:Keyword>
    <ows:Keyword>TRENTO</ows:Keyword>
    <ows:Keyword>Workshop</ows:Keyword>
  </ows:Keywords>
  <ows:ServiceType codeSpace="http://opengeospatial.net">OGC:SOS</ows:ServiceType>
  <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
  <ows:Fees>NONE</ows:Fees>
  <ows:AccessConstraints>NONE</ows:AccessConstraints>
</ows:ServiceIdentification>
<!-- Provider Description -->
<ows:ServiceProvider>
  <ows:ProviderName>Istituto Scienze della Terra</ows:ProviderName>
  <ows:ProviderSite xlink:href="http://istgeo.ist.supsi.ch"/>
  <ows:ServiceContact>
    <ows:IndividualName>Pinco Pallino</ows:IndividualName>
    <ows:PositionName>Geomatica</ows:PositionName>
  </ows:ServiceContact>
  <ows:Phone>
    <ows:Voice>1234</ows:Voice>
  </ows:Phone>
</ows:ServiceProvider>
</ows:ProviderDescription>
</ows:Provider>
</ows:Service>
</ows:Capabilities>
```

Usiamo il servizio

Scuola universitaria professionale
della Svizzera italiana

SUPSI

Istituto Scienze della Terra



Sensor Observation Service

RegisterSensor: richiesta

- Funziona solo con il metodo POST

```
<sos:RegisterSensor
xsi:schemaLocation="http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
service="SOS" version="1.0.0">
  <SensorDescription> <!-- è il SensorML che descrive il sensore -->
    <System> .... </System>
  </SensorDescription>
  <ObservationTemplate> <!-- è O&M che descrive i dati rilevati -->
    <om:Observation> .... </om:Observation>
  </ObservationTemplate>
</sos:RegisterSensor>
```

RegisterSensor: richiesta

- L'ObservationTemplate (O&M)

```
<om:Observation >
```

```
  <om:procedure > ... </om:procedure >
```

```
  <sa:samplingTime>... </sa:samplingTime>
```

```
  <om:observedProperty>...</om:observedProperty>
```

```
  <om:featureOfInterest>...</om:featureOfInterest>
```

```
  <om:result>
```

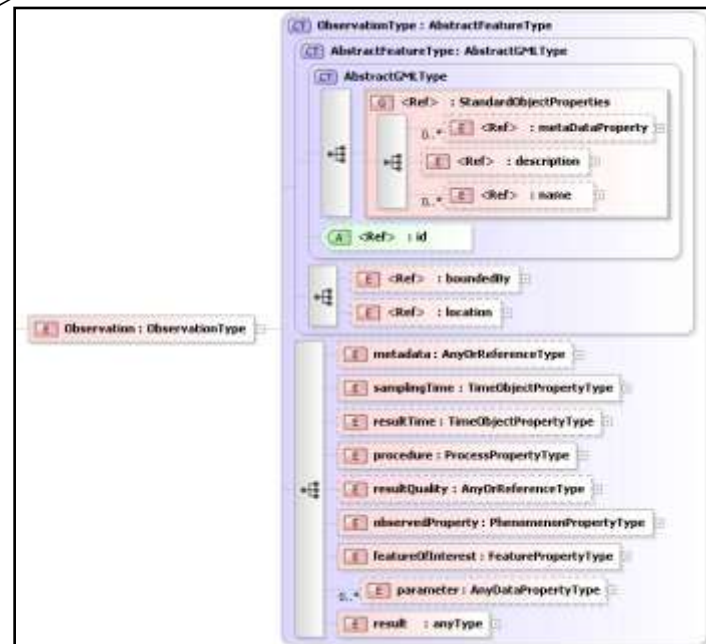
```
    <swe:DataArray>...<swe:DataArray>
```

```
    <!-- oppure swe:SimpleDataRecord -->
```

```
  </om:result>
```

```
</ om:Observation >
```

- Vedi schema:



<http://www.liquid-technologies.com/LibraryDocs/OpenGis/om/1.0.0/>

RegisterSensor: richiesta

- Il SensorDescription (SensorML)

<System>

<gml:description>text description </gml:description>

<gml:name>my sensor name</gml:name>

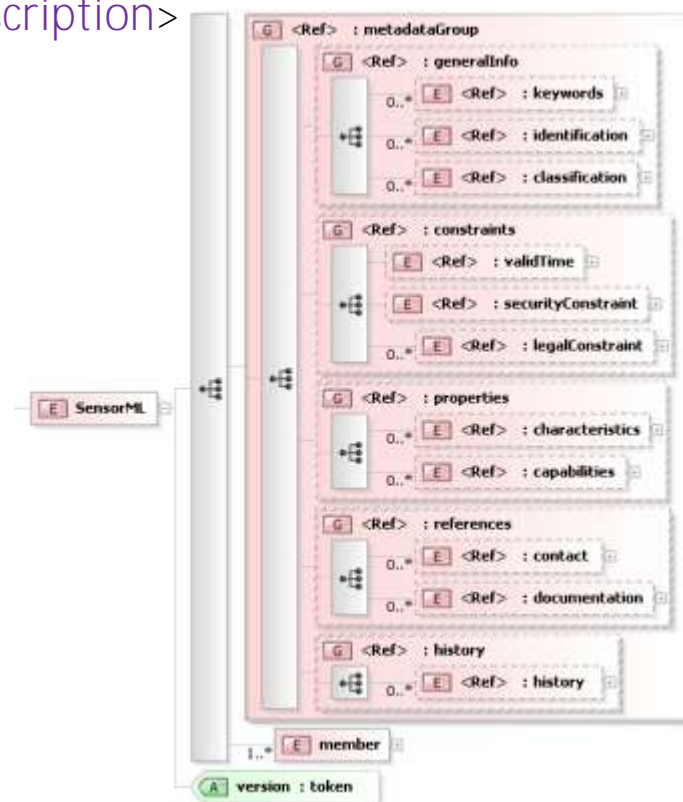
< keywords>...< /keywords>

<identification>...</identification>

<classification>...</classification>

<components>...</components>

</System>



RegisterSensor: risposta

- Restituisce l'ID assegnato al sensore registrato: un codice identificativo da usare per inserire osservazioni..

```
<sos:RegisterSensorResponse  
  xsi:schemaLocation="http://schemas.opengis.net/sos/1.0.0/sosAll.xsd">  
    <AssignedSensorId>  
        urn:ogc:object:sensor:x-ist::5b65a6226e03999afe0f711305f57943  
    </AssignedSensorId>  
</sos:RegisterSensorResponse>
```

- *Attenzione è l'unica volta che si vede il SensorID !!!*

"temporary" offering

- Secondo l'approccio di istSOS i sensori registrati sono assegnati ad un particolare offering "temporary".
- Su questi è possibile effettuare inserimento di osservazioni ma non sono disponibili per la consultazione fino a quando non vengono registrati dall'amministratore ad un altro offering

assegnamo l'offering voluto

The screenshot displays the pgAdmin III interface with three windows open, illustrating database relationships. A green arrow points from the 'id_off' column in the 'istso.offerings' table to the 'id_off_prc' column in the 'istso.procedures' table, indicating a foreign key relationship.

istso.offerings Table:

	name_off	desc_off	expiration	active_off	id_off
	character	text	timestamp	boolean	[PK] serial
1	temporary	temporary offering to hold s		TRUE	1
*					

istso.procedures Table:

	name_prc	desc_prc	stime_prc	etime_prc	id_prc	id_tru_fk	time_res
	character	text	timestamp	timestamp	[PK] serial	integer	integer
1	thermo1		2010-02-08		3	1	10
2	LimnoGPS1		2010-02-09		4	1	10
*							

istso.off_pro Table:

	id_off_prc	id_off_fk	id_prc_fk
	[PK] integer	integer	integer
1	3	1	3
2	4	1	4
*			

The interface also shows a sidebar with a tree view of database objects, including 'observed_pro', 'off_pro', 'offerings', 'positions', 'proc_obs', 'procedures', 'quality_index', 'time_res_unit', 'uoms', 'Trigger Function', 'Views (0)', and 'public'. The bottom status bar indicates 'Retrieving Table details... Done.' and '0.06 secs'.

DescribeSensor: richiesta

- Metodo POST

```
<sos:describeSensor  
  service="SOS"  
  outputFormat="text/xml;subtype='sensorML/1.0.0'">  
  <procedure>thermo1</procedure>  
</sos:describeSensor>
```

- Metodo GET

```
http://localhost/sos/srv1?  
request=DescribeSensor&  
procedure=thermo1&  
outputFormat=text/xml;subtype='sensorML/1.0.0'&  
service=SOS&  
version=1.0.0
```

DescribeSensor: risposta

- Documento SensorML

```
<SensorML>
  <member>
    <System>
      <gml:description>text description </gml:description>
      <gml:name>my sensor name</gml:name>
      < keywords>...< /keywords>
      <identification>...</identification>
      <classification>...</classification>
      <components>...</components>
    </System>
  </member>
</SensorML>
```

InsertObservation: richiesta

- Funziona solo con il metodo POST

```
<sos:InsertObservation
xmlns.... service="SOS" version="1.0.0" >
  <AssignedSensorId>urn:ogc:object:sensor:x-ist::a4f</AssignedSensorId>
  <!-- quello restituito dal RegisterSensor -->
  <om:Observation>...<om:Observation>
</sos:InsertObservation>
```

InsertObservation: risposta

- Inserimento di un'osservazione singola

```
<sos:InsertObservationResponse>  
  <sos:AssignedObservationId>10293</ sos:AssignedObservationId >  
</sos:InsertObservationResponse >
```

- Inserimento di osservazioni multiple

```
<sos:InsertObservationResponse>  
  <sos:AssignedObservationId>2293@2294@2295</ sos:AssignedObservationId >  
</sos:InsertObservationResponse >
```

GetCapabilities: richiesta

- Metodo POST

```
<sos:GetCapabilities xmlns... version="1.0.0" service="SOS">  
  <section>serviceidentification</section>  
  <section>serviceprovider</section>  
  <section>operationsmetadata</section>  
  <section>contents</section>  
</sos:GetCapabilities>
```

- Metodo GET

```
http://localhost/sos/srv1?  
request=GetCapabilities&  
section=serviceidentification,serviceprovider,operationsmetadata,contents&  
service=SOS&  
version=1.0.0
```


GetCapabilities: risposta

- Vediamolo insieme...

```
<sos:Capabilities xmlns... >
```

```
  <ows:ServiceIdentification>...</ows:ServiceIdentification >
```

```
  <ows:ServiceProvider>...</ows:ServiceProvider>
```

```
  <ows:OperationsMetadata>...</ows:OperationsMetadata >
```

```
  <ows:Contents>...</ows:Contents >
```

```
</sos:Capabilities>
```

GetObservation: richiesta

- Metodo GET

http://localhost/sos/srv1?

service=SOS&

version=1.0.0&

request=GetObservation&

offering=temporary&

procedure=thermo1&

eventTime=2010-02-10T16:00:00+01/2010-02-10T17:00:00+01&

observedProperty=temperature&

responseFormat=text/xml;subtype='sensorML/1.0.0'

- Metodo POST

```
<sos:GetObservation xmlns... service="SOS" version='1.0.0'>
```

```
<offering>temporary</offering>
```

```
<procedure>thermo1</procedure>
```

```
<eventTime>
```

```
<gml:TimePeriod>
```

```
<gml:beginPosition>2010-02-10T16:00:00+01</gml:beginPosition>
```

```
<gml:endPosition>2010-02-10T17:00:00+01</gml:endPosition>
```

```
</gml:TimePeriod>
```

```
</eventTime>
```

```
<responseFormat>text/xml;subtype='sensorML/1.0.0'</responseFormat>
```

```
<observedProperty>urn:ogc:def:property:x-ist::rainfall</observedProperty>
```

```
</sos:GetObservation>
```

GetObservation: risposta

```
<om:ObservationCollection xmlns... >
<gml:name>temporary</gml:name>
<gml:description>temporary offering</gml:description>
<om:member>
  <om:Observation>
    <om:procedure xlink:href="thermo1"/>
    <om:samplingTime>
      <gml:TimePeriod>.....</gml:TimePeriod>
    </om:samplingTime>
    <om:observedProperty>
      <swe:CompositPhenomenon id="comp_194" dimension="2">... ..</swe:CompositPhenomenon>
    </om:observedProperty>
    <om:featureOfInterest xlink:href="VIGANELLO">
      <gml:Point srsName="EPSG:21781">....</gml:Point>
    </om:featureOfInterest>
    <om:result>
      <swe:DataArray>.....</swe:DataArray>
    </om:result>
  </om:Observation>
</om:member>
</om:ObservationCollection>
```

GetFeatureOfInterest: richiesta

- Metodo POST

```
<sos:getfeatureOfinterest xmlns... service="SOS"
outputFormat="text/xml;subtype='sensorML/1.0.0'">
  <featureOfInterest>LUGANO</featureOfInterest>
  <srsName>4326</srsName>
</sos:getfeatureOfinterest>
```

- Metodo GET

```
http://localhost/sos/srv1?
request=GetFeatureOfInterest&
featureOfInterest=LUGANO&
srsName=4326&
service=SOS&
version=1.0.0
```

GetFeatureOfInterest: risposta

- Vediamolo insieme...

```
<sa:SamplingPoint gml:id="LUGANO" xmlns...>
```

```
<gml:name> LUGANO</gml:name>
```

```
<!-- Lista di tutte le procedure che sono associate alla featureOfInterest richiesta -->
```

```
<sa:relatedObservation><om:Observation>...</om:Observation></sa:relatedObservation>
```

```
<sa:relatedObservation><om:Observation>...</om:Observation></sa:relatedObservation>
```

```
<sa:position>
```

```
<gml:Point > ... </gml:Point>
```

```
</sa:position>
```

```
</sa:SamplingPoint>
```

peculiarità di istSOS

Scuola universitaria professionale
della Svizzera italiana

SUPSI

Istituto Scienze della Terra



Sensor Observation Service

istSOS

- Quello qui presentato è un'implementazione degli standard SOS così come definiti dall'OGC nel documento "OGC 06-009r6" e come interpretati dall'Istituto Scienze della Terra.

Open Geospatial Consortium Inc.

Date: 2007-10-26

Reference number of this document: OGC 06-009r6

Version: 1.0

Category: OpenGIS® Implementation Standard

Editors: Arthur Na (IRIS Corp.), Mark Priest (3eIT)

Punti di discussione aperti

- Inserimento con DataArray (tutto o niente?)
- featureOfInterest (1-N?)
- Phenomenon/parameter: istSOS e gli urn
- Errori nella definizione degli schemi (?):
elementi ricorsivi

getObservation

1. Nella risposta di un `getObservation` il tempo è espresso nello stesso fuso orario usato nell'`eventTime` della richiesta (in caso manchi viene usato di default l'`UTC`)
2. Se nella richiesta `getObservation` non è fornito l'`eventTime` is requested only the **latest available observation** is returned



getObservation

3. Il parametro result (filtero sui valori, tipo CQL) **non ancora supportato!** :- (
4. Parametro no standard
aggragateInterval e aggregateFunction
che permettono di aggregare on-the-fly i valori osservati

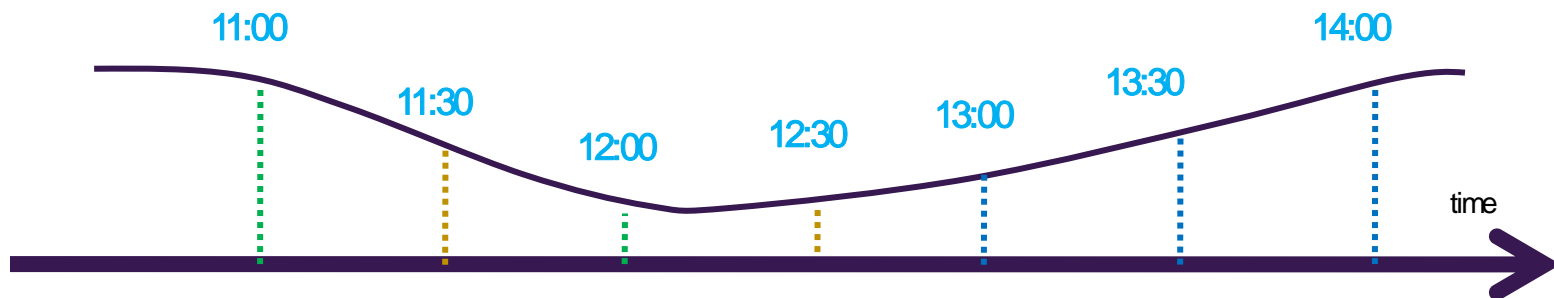


regular time series

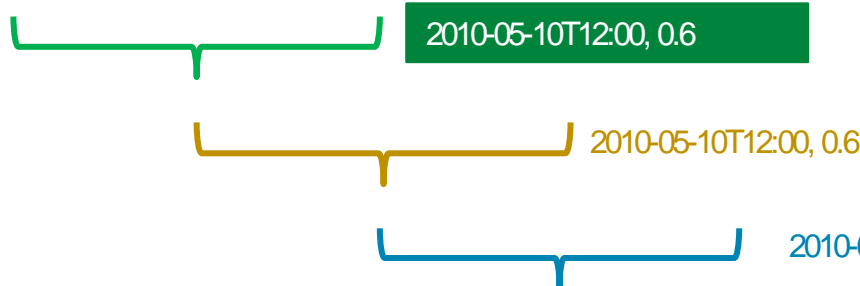
- istSOS usa intervalli temporali “in dietro” e aperti verso il basso con frequenza definita dall'elemento `<gml:timeInterval>`

`<gml:timeInterval> PT1H </gml:timeInterval>`

`] -----]` = open lower bound



?



getObservation

5. responseFormat suporta anche i formati **text/csv** e **application/json**

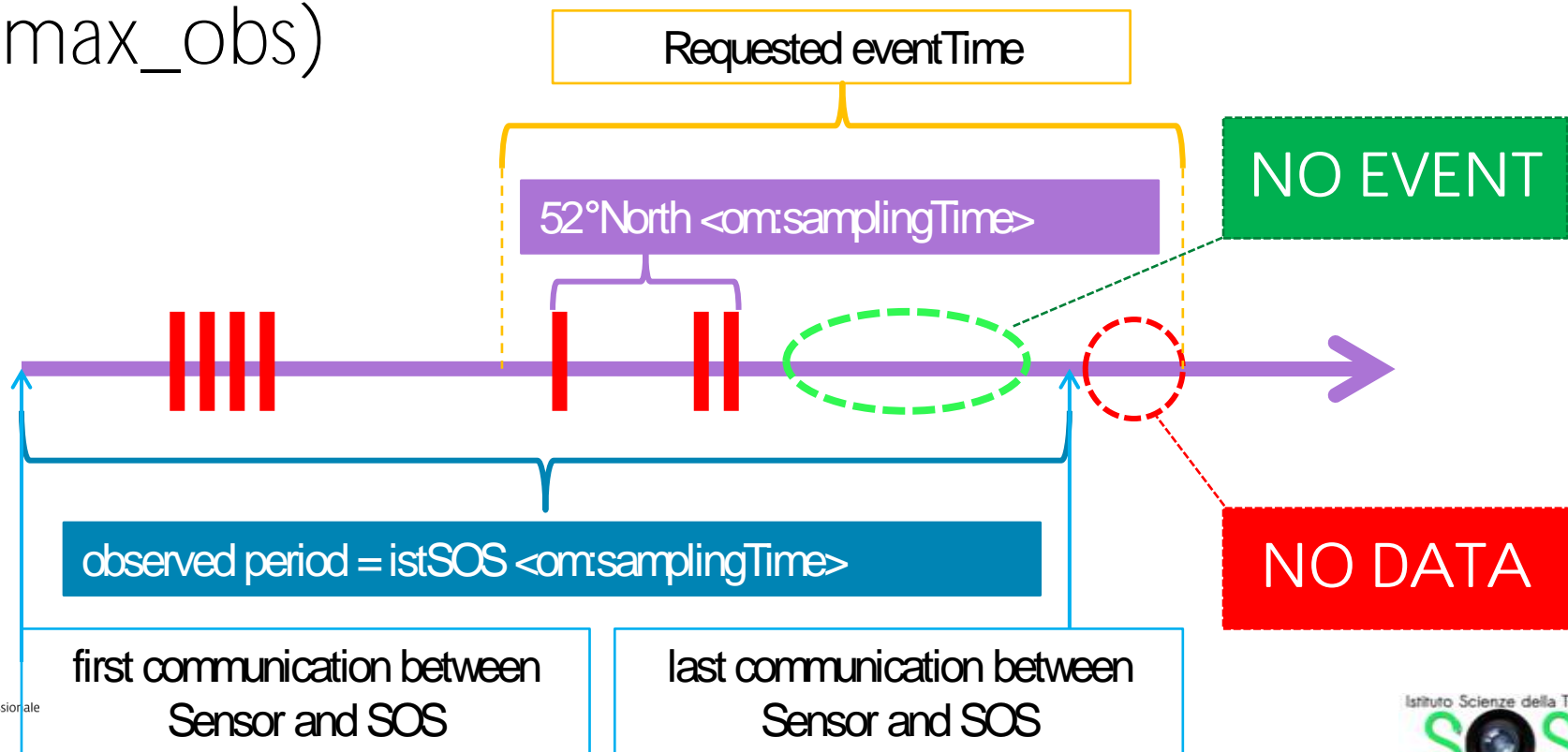


6. Supporto di serie temporali irregolari (osservazioni ad evento) e discrimina tra "no data" e "no event"!



Irregular time series

- istSOS usa `<om:samplingTime>` per fornire il periodo di osservazione, non l'intervallo delle osservazioni restituite (`min_obs-max_obs`)



registerSensor

1. Riconoscimento automatico del tipo di sensore (fixpoint or mobilepoint) in funzione della presenza di x,y,z nell' observationTemplate fornito.
2. Tutte le nuove procedure sono registrate in un offering temporaneo ("temporary")

insertObservation

1. Permette di inserire valori multipli e restituisce un identificatore composto dalla concatenazione degli id delle osservazioni tramite il carattere "**@**". (1@2@3@4@5..)
2. **Time-value vincolo:** una procedura ha una proprietà osservata con un solo valore per istante
3. In caso di errore nessuna osservazione è registrata ed un'eccezione viene lanciata.

insertObservation

4. Il periodo di osservazione è aggiornato automaticamente dopo ogni nuovo inserimento di osservazioni
5. forceInsert: parametro non standard che permette di sovrascrivere dati già presenti nel servizio; se specificato con valore True l'insertObservation sostituisce tutti i valori compresi nell'<om:samplingTime> con quelli presenti nella richiesta (se forniti)

UpdateSensorDescription

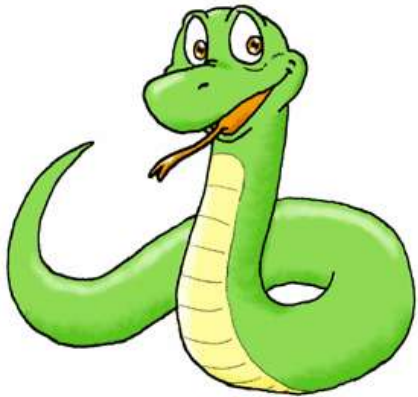
- **NON standard request**: allows for submission of a new sensor description that substitutes the current one
- this is to account for **historical changes of instruments** or particular maintenance tasks

virtual process

- istSOS permette di definire delle procedure virtuali estendendo una classe di base virtualProcess che ha un metodo per caricare I dati di una procedura classica sulla base del filtro fornito nella richiesta.
- I dati potrebbero risiedere in qualsiasi posto, basta creare una procedura che utilizzando il filtro carica I dati in un array e li restituisce !!

virtual process

FILENAME == PROCEDURE NAME in reserved FOLDER



```
from istSOS.responders.GOresponse import VirtualProcess
import datetime, decimal
```

```
class istvp(VirtualProcess):
    def __init__(self,filter,pgdb):
        VirtualProcess.__init__(self,filter,pgdb)

    #SET THE INPUTS
    self.h = self.setSOSobservationVar("A_BRB","riverheight")
```

```
def execute(self):
    data_out=[]
    for rec in self.h:
        newdata = rec[1]*0.25 + 124
        data_out.append( rec[0], newdata)
    return data_out
```

...Provate!o ...

<http://sites.google.com/site/istsosproject>

...ed unitevi allo sviluppo di istSOS

Scuola universitaria professionale
della Svizzera italiana

SUPSI

Istituto Scienze della Terra



Sensor Observation Service