Tutorial OGC
SI.org
Dec 17-18 2013

Dr. Luis Bermudez
Director Compliance Program, OGC

Outline

1. Value of Standards
2. The OGC Overview
3. The OGC Interoperability Program (IP)
4. The OGC Standards DevelopmentProgram
5. Introduction to Distributed Computing and Web Services
6. Web Feature Service (WFS)
7. Web Feature Service (WFS) Demo
8. Geographic Modeling Language (GML)
9. WEB Map Service (WMS)
10. KML

11. Overview Sensor Web Enablement (SWE)
12. Observations and Measurements (O&M)
13. SWECommon
14. Sensor Model Language (SensorML)
15. SWE Services
16. Metadata and Catalogs
17. Web Processing Service (WPS)
18. Geospatial Architectures
19. Cloud Computing

Value of Standards

Why Standards?

"We want to have standards applied to all important interfaces . . . Being vendor-independent, vendor-neutral helps us protect our equity."

Dawn Meyerriecks, DISA, in an interview with the OpenGroup (2002)

Why Standards?

"People want the government to be transparent, so why shouldn't the technology be?"

Jim Willis, director of E-government at the Rhode Island Secretary of State office. (2002)

Why Standards?

"Standardization is one of the essential building blocks of the Information Society. There should be particular emphasis on the development and adoption of international standards ..."

The Declaration of Principles WSIS-03/GENEVA/DOC/0004 concluded in paragraph 44, from the first phase of the World Summit on the Information Society, Geneva, December 10 to 12 2003.

Why Standards?

"...integration is key. Using standard data structures and formats all data about any object of interest—any person, place or thing—can be easily shared with and accessed by anyone with the need to know..."

Letitia A. Long, Director, National Geospatial-Intelligence Agency, October 4, 2013, KMI Interview

Why Standards?

“ When you are delivering spatial web services on behalf of 20 government agencies to more than a 1000 organisations running their own spatial systems, you need standards."

Kylie Armstrong , Landgate

Why Standards?

"OGC's interface and encoding standards are an essential part of the National Spatial Data Infrastructure. They play a key role in providing technical interoperability among geospatial systems used at all levels of government..."

Ivan DeLoatch, Executive Director of the FGDC.

Why Open Standards?

They work better when they are open

Why Open Standards?

- Prevents a single, self-interested party from controlling a standard
- Lower systems and life cycle costs
- Encourage market competition: Choose based on functionality desired and avoid “lock in” to a proprietary solution
- Stimulates innovation beyond the standard by companies that seek to differentiate themselves.

Source: Open Standards, Open Source, and Open Innovation: Harnessing the Benefits of Openness, April 2006. Committee For Economic Development. www.ced.org

Why Open Standards?

"Governments like to say they can publish to OGC KML instead of Google KML ... everyone has confidence we won’t take advantage of the format or change it in a way that will harm anyone ..."

Michael Weiss-Malik, Google KML product manage

Location Interoperability

Defined as the ability of diverse data sources, systems and organizations to work together (inter-operate).

Location Interoperability

- Ease information sharing
- Promote information reuse
- Reduce duplication of effort
- Flexibility to add new capabilities
- Vendor neutral
- Saves time, reduces cost, increases market choice, protects assets and lives

Location Interoperability Challenge

Ability to access, fuse and apply diverse content when and where needed is critical to situational awareness and disaster planning/ response in cross-boundary and cross-domain settings

The OGC Overview

Open Geospatial Consortium

- Funded in 1994
- Voluntary consensus (475+ members)
- Standards organization (40+ standards)
- Leads the development of standards for geospatial and location based services.

Open Geospatial Consortium (OGC)

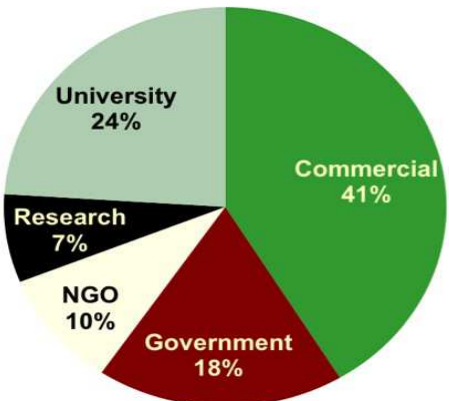
Mission: To serve as a global forum for the collaboration of developers and users of spatial data products and services, and to advance the development of international standards for geospatial interoperability.

What does the OGC as an entity provides?

- **An agreed upon consensus process**
    - to encourage effectiveness and efficiency
    - for defining, testing, documenting, and approving specifications
- **Staff knowledge, expertise and support** to work with the members to facilitate the consensus process
- A comprehensive **communications infrastructure**
- A **consensus-based forum** for conflict resolution

Members

http://www.opengeospatial.org/ogc/members/report/



Industry Members

Example Government Members US

- US DHS
- US DNI (PM/ISE)
- US EPA
- US FAA
- US NASA
- USGS / FGDC
- US NGA
- US NOAA
- Dept. of Land Conservation and Development (Oregon, USA)
- Oakridge National Lab

Example Government Members Europe

- Eurocontrol
- European Environment Agency
- European Space Agency
- EU Joint Research Centre
- UK MET
- METEO France
- BRGM (France)
- Ordnance Survey (UK)
- State Land Agencies (Germany)
- City of Vienna (Austria)

Example Government Members Other Parts of the world

- Quebec Dept. of Natural Resources (Canada)
- DOD Australia
- Geoscience Australia
- Landgate (Western Australia)
- GIS Center for Security (Abu Dhabi, UAE) Abu Dhabi Systems & Info. Center
- Dubai Municipality
- Dept. Science & Technology (India)
- Ministerio de Bienes Nacionales - Secretaria Ejecutiva SNIT (Chile)

OGC Alliance Partners

Implementations

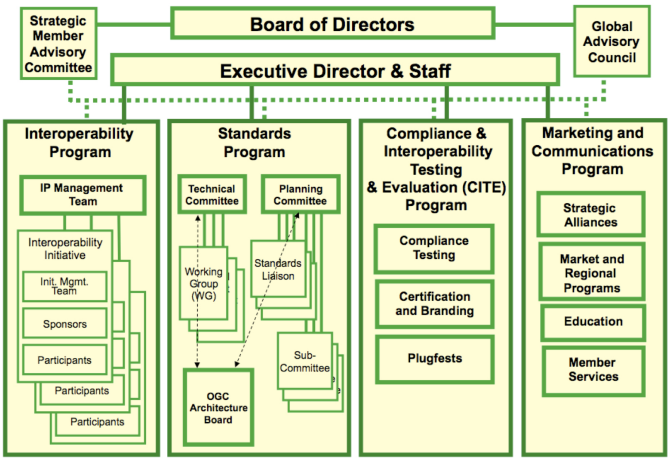
There are: 751 Implementing Products and 194 Compliant Products

The table provides a summary of implementing and compliant products. Click on a Specification to view all associated registered products. You can also click the column headers to sort the table.

Total = Number of implementations. **Comp** = Number of compliant products.

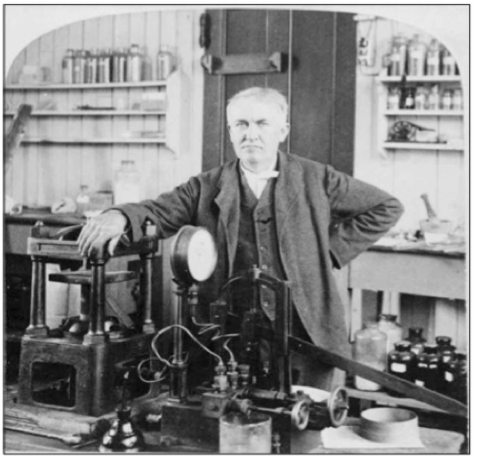
Total	Comp.	Specification / Version	Abrv / Version
462	77	Web Map Service (1.1.1)	WMS 1.1.1
293	54	Web Feature Service (1.0.0)	WFS 1.0.0
271	63	Web Map Service (WMS) Implementation Specification (1.3.0)	WMS 1.3.0
265	0	Web Map Service (1.0)	WMS 1.0
233	0	Web Map Service (1.1)	WMS 1.1
214	36	Web Feature Service (WFS) Implementation Specification (1.1.0)	WFS 1.1.0
181	34	Web Coverage Service (WCS) Implementation Specification (Corrigendum) (1.0.0)	WCS 1.0.0
161	0	Geography Markup Language (GML) Encoding Specification (3.1.1)	GML 3.1.1
146	0	Filter Encoding (1.0)	Filter 1.0
134	0	Geography Markup Language (2.1.2)	GML 2.1.2

OGC Structure



The OGC Interoperability Program (IP)

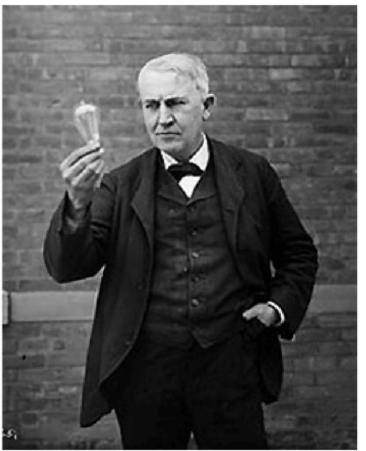
How do we know if an idea works?



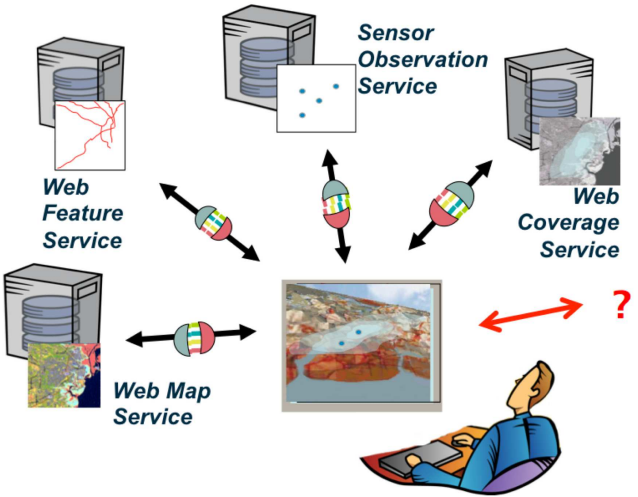
I have not failed, I've just found 10,000 ways that won't work.

Thomas Edison

Innovation about what?



Innovation about what?



Interoperability Program (IP)

- A global, innovative, hands-on rapid prototyping and testing program designed to unite users and industry in accelerating interface development and validation, and the delivery of interoperability to the market.
- 40 initiatives since 1999
- **Deliverables:**
    - Technical Documents
    - Prototype Implementations
    - Demonstrations

Benefits of Involvement in IP Initiatives

For Participants

- Early insights and skill building
- Early visibility
- Early market deployment
- Direct influence
- Broaden market reach

Benefits of Involvement in IP Initiatives

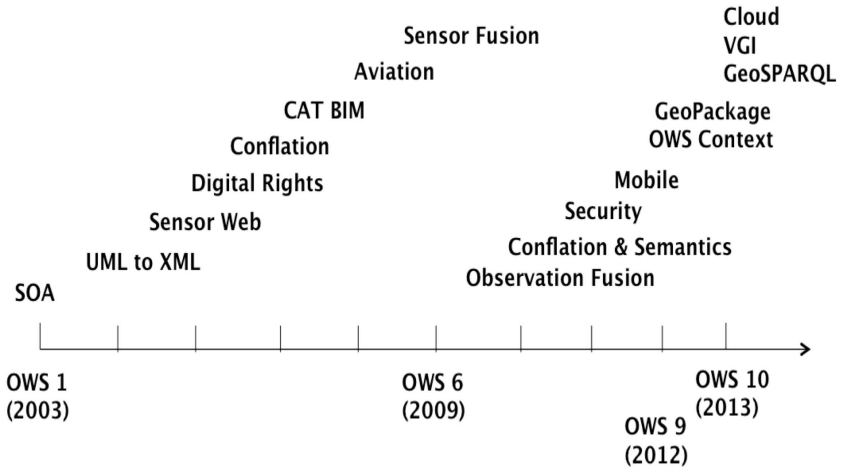
For Sponsors

- Ability to Determine Market Interest
- Accelerated process - workable interface specifications in 4-6 months
- Vendors test, validate and demonstrate interface integrity – Rapid time to market
- Leverage of other sponsor' funding to solve common/similar problems
- Significant ROI 2-3.5 overall (and as high as 25 for individual sponsors)

IP Type of activities

- **Testbeds**: provide an environment for collaborative, fast-paced, multi-vendor rapid prototyping efforts to define, design, develop, and test candidate interface and encoding specifications.
- **Pilots**: apply and test OGC standards in real world applications using standards-based commercial off-the-shelf (SCOTS) products that implement OGC standards. Pilot Projects also help organizations identify gaps to be addressed by further standards development work.
- **Interoperability Experiments**
- ...

Testbed Activities



OWS-9 Testbed numbers

- 55 organizations
- More than 200 individuals
- 60 components (Servers, Clients ..)
- 28 reports
- $2.85 Million
- Duration: less than year

OWS-9 Testbed Sponsors

- US National Geospatial Intelligence Agency (NGA)
- US Geological Survey (USGS)
- US Army Geospatial Center (AGC)
- US Federal Aviation Administration (FAA)
- EUROCONTROL

OWS-9 Testbed Sponsors

- US National Aeronautics & Space Administration (NASA)
- UK Defence Science & Technology Laboratory (DSTL)
- Lockheed Martin Corporation
- GeoConnections/Natural Resources Canada
- GeoViqua/CREAF/European Commission (EC)

OWS-9 Testbed Participants

The OGC Standards DevelopmentProgram

It's all about coordinating people

“Interoperability seems to be about the integration of information. What it’s really about is the coordination of organizational behavior.”

David Schell, OGC Founder

Standards Development Program

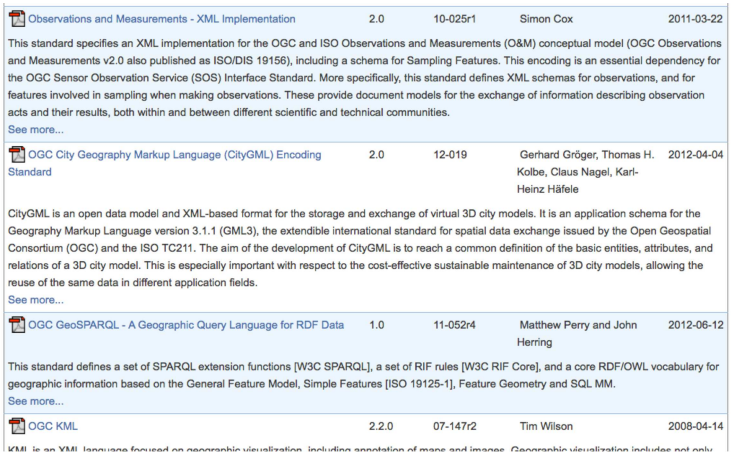
- Consensus standards process similar to other Industry consortia (World Wide Web Consortium, OMA etc.).
- 33 “core” standards
- 15 extensions/profiles

What is a standard?

- A **document**, established by **consensus** and **approved** by the OGC Membership, that provides rules and guidelines, aimed at the optimum degree of interoperability in a given context.
- **Conveys:**
    - Community requirements
    - Member requirements
    - Market trends
    - Technology trends

Standards List

http://www.opengeospatial.org/standards/is



Observations and Measurements - XML Implementation	2.0	10-025r1	Simon Cox	2011-03-22

This standard specifies an XML implementation for the OGC and ISO Observations and Measurements (O&M) conceptual model (OGC Observations and Measurements v2.0 also published as ISO/DIS 19156), including a schema for Sampling Features. This encoding is an essential dependency for the OGC Sensor Observation Service (SOS) Interface Standard. More specifically, this standard defines XML schemas for observations, and for features involved in sampling when making observations. These provide document models for the exchange of information describing observation acts and their results, both within and between different scientific and technical communities.
See more...

OGC City Geography Markup Language (CityGML) Encoding Standard	2.0	12-019	Gerhard Gröger, Thomas H. Kolbe, Claus Nagel, Karl-Heinz Häfele	2012-04-04

CityGML is an open data model and XML-based format for the storage and exchange of virtual 3D city models. It is an application schema for the Geography Markup Language version 3.1.1 (GML3), the extendible international standard for spatial data exchange issued by the Open Geospatial Consortium (OGC) and the ISO TC211. The aim of the development of CityGML is to reach a common definition of the basic entities, attributes, and relations of a 3D city model. This is especially important with respect to the cost-effective sustainable maintenance of 3D city models, allowing the reuse of the same data in different application fields.
See more...

OGC GeoSPARQL - A Geographic Query Language for RDF Data	1.0	11-052r4	Matthew Perry and John Herring	2012-06-12

This standard defines a set of SPARQL extension functions [W3C SPARQL], a set of RIF rules [W3C RIF Core], and a core RDF/OWL vocabulary for geographic information based on the General Feature Model, Simple Features [ISO 19125-1], Feature Geometry and SQL MM.
See more...

OGC KML	2.2.0	07-147r2	Tim Wilson	2008-04-14

KML is an XML language focused on geographic visualization, including annotation of maps and images. Geographic visualization includes not only

Type of Specifications

Implementation Specifications - Standards

Basis for working software; detail the interface structure between software components

Abstract Specifications

Conceptual foundation / reference model for specification development

Best Practices

Describe use of specifications

Engineering Reports

Results from the OGC Interoperability Program

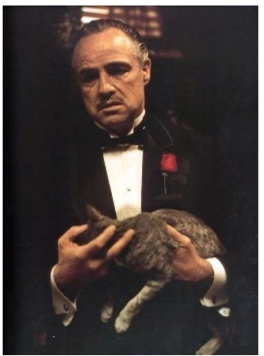
Discussion Papers

Forum for public review of concepts

What it takes

- Requires collaboration on a global basis
- Requires concensus by many organizations
- Requires give and take
- Requires certified, repeatable process

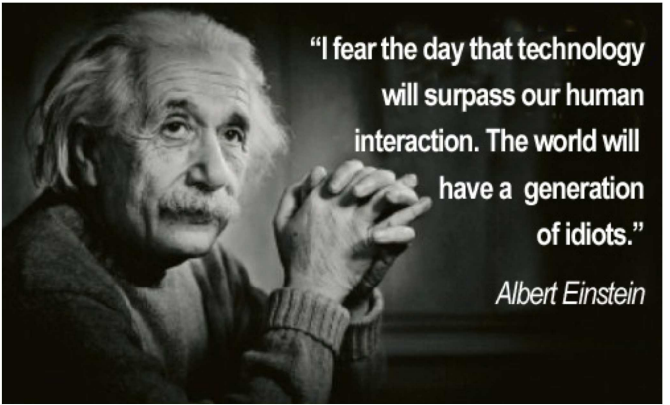
The Core of Consensus Philosophy



"Never get angry. Never make a threat. Reason with people."

Don Vito Corleone
(The Godfather)

Is Social


“I fear the day that technology will surpass our human interaction. The world will have a generation of idiots.”
Albert Einstein

Consensus decision-making

Is a group decision making process that seeks the consent of all participants. Consensus may be defined professionally as an acceptable resolution, one that can be supported, even if not the "favourite" of each individual.

- This is what the OGC standards process is all about!
- Guided by the TC Policies and Procedures
- Also documented in the TAO: A Guide for New Members, the OGC New Member Orientation, and numerous other documents.
- Also guided by the OGC Principals of Conduct

Technical Committee

- Where the formal standards development consensus discussion and approval process occurs.
- Comprised of a number of Domain Working Groups (DWGs or WGs for short) and Standards Working Groups (SWGs).
- Work guided by the Technical Committee Policies and Procedures

Domain Working Groups (DWGs)

Where Members discuss technical requirements, use cases, and issues related to the development and revision of OGC standards. The results of OGC interoperability projects are presented and discussed. Many Member presentations on the use of OGC standards. Any Member or invited guest can attend any DWG session.

Standards Working Groups (SWFs)

- Groups that work on a new (candidate) OGC standard or revision to an existing OGC standard
- Members only may participate
- Guided by the OGC Policies and Procedures
- OGC Intellectual Property policies in full effect
- Usually last from 6 months to 18 months

What is the OGC Planning Committee?

- The Planning Committee provides guidance and the management structure for the Technical Committee and the Interoperability Program.
- Members participate in OGC business planning and market focus activities
- Manages the consortium's technology release process
- Approves special negotiated memberships and committee participation.
- Must be a Principal or Strategic Member or a TC representative to the PC to participate in PC activities.

What is the OGC Architecture Board (OAB)?

- Work with the TC and the PC to insure architecture consistency of the Baseline and provide guidance to the OGC membership to insure strong life cycle management of the OGC standards baseline
- Reviews all OGC standards and revisions to standards prior to public comment.
- General technology discussions and guidance
- Evaluate candidate standards for fast track process
- Issue resolution
- Nominated and Elected by the OGC Membership

Domain Working Groups by Market

- Aviation
- Business Intelligence and Decision Support
- Defence and Intelligence
- Earth System Science (ESS)
- Energy and Utilities
- Hydrology
- LandInfra
- Law Enforcement and Public Safety (LEAPS)
- Location Services and Mass Market Domain
- Met/Oceans

Domain Working Groups by Technology

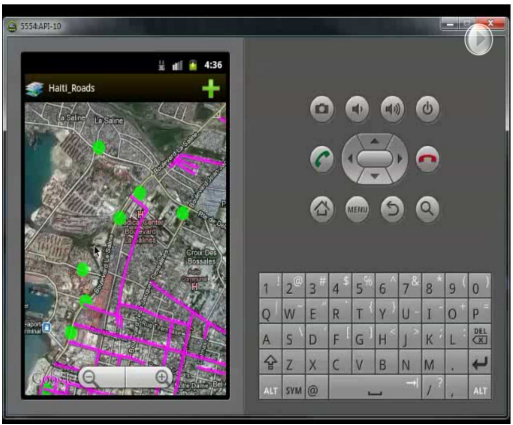
- 3D
- Geosemantics
- Security
- Sensor Web Enablement (SWE)
- Workflow

Introduction to Distributed Computing and Web Services

Publishing Geospatial Data

First Map of the World
Catal Huyuk (South East Turkey) 6200 BC

Publishing Geospatial Data



Viewing and creating data from mobile devices

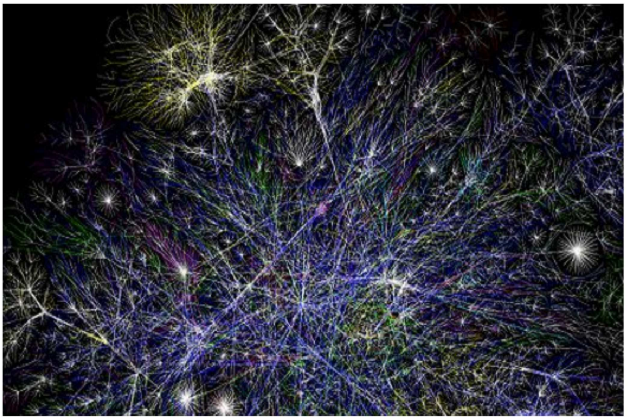
Geo Package

What changed?

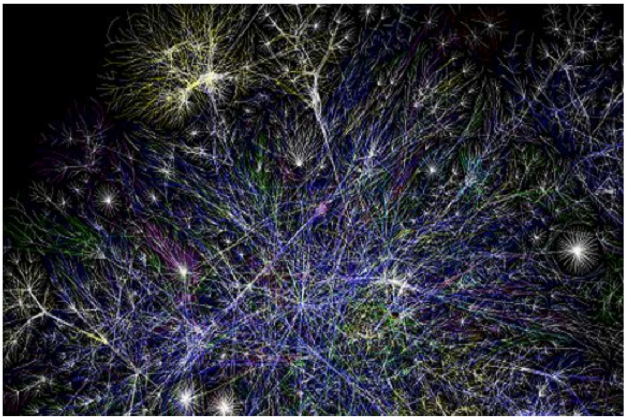
What changed?

Technology

What is this?



Internet



World Wide Web

- Enabled by Internet
- All **Information** from Computers connected around the world.

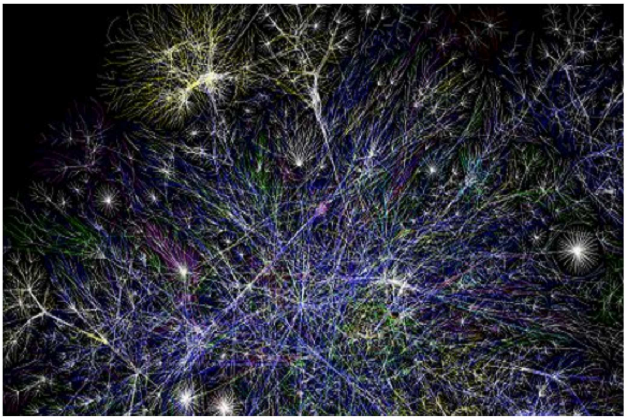
How do we connect to the Web?

- URL
- HTTP GET
- Protocols
- Encodings
- ...

Uniform Resource Locator (URL)

- http://my.umbc.edu/
- ftp://ftp.funet.fi/pub/standards/RFC/rfc959.txt
- mailto:bermud@me.com

Every node can be a URL



Protocols

Protocols

- TCP/IP
- HTTP (GET, POST)

HTTP GET is a URL

```
http://localhost:8080/geoserver/topp/ows?
service=WFS&
version=1.0.0&
request=GetCapabilities
```

HTTP has Parameters and Values

```
http://localhost:8080/geoserver/topp/ows?

service=WFS&
version=1.0.0&
request=GetCapabilities
```

HTTP POST

```
<?xml version="1.0"?>
   <wfs:GetCapabilities
      service="WFS"
      version="1.0.0"
     ...
```

Encodings

Service

Distinct part of the functionality that is provided by an entity through interfaces. (ISO 19119:2005)

Interface

Named set of operations that characterize the behaviour of an entity.

Operations

- Specification of a transformation or query that an object may be called to execute. It has a name and a list of parameters.
- For example GetFeature of WFS
- Is defined by a set of protocols (e.g. HTTP GET)

Service

http://en.wikipedia.org/wiki/File:Roomba_original.jpg

Interfaces


Room Selection Interface
iRobot
Roomba
Robotic FloorVac
S
M
L
Select Room Size to Start / Stop
http://en.wikipedia.org/wiki/File:Roomba_original.jpg
Electric Interface

In an OGC Web Service

- Service = WFS 2.0
- Interface = Basic or Transaction
- Operations for Basic = DescribeFeature and GetFeature
- Protocol = GetFeature uses HTTP Get

Let's Try - Setup

To view HTTP requests, we can use Firefox:

- Open: Tools / Web Developer / Web Console
- Select Net

Let's Try - Test

- Open any page
- Open a wrong URL
- Inspect the console
- Click on example petition: "Get http:// ...."

What Should I get back if I ask a server for a "Toluca"

- A Map of Toluca
- A video
- A picture
- The history
- List of Hotels
- ???

There is a need to design specialized operations

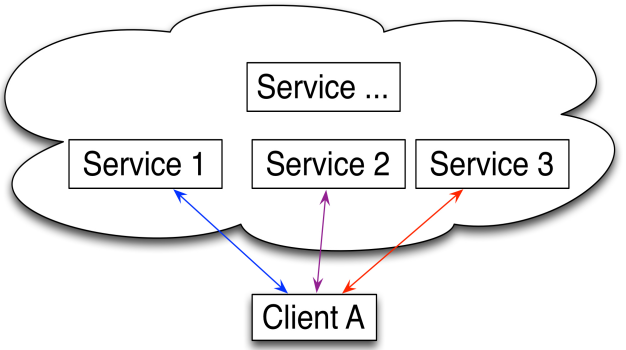
For example, for restaurants services:

- Find me all restaurants 5 km from here
- Find me the restaurant with the best guacamole
- Add my review to restaurant X: " Meat very good, service terrible"
- Reserve the restaurant X for two for today at 8 PM and bring 25 red flowers to the table

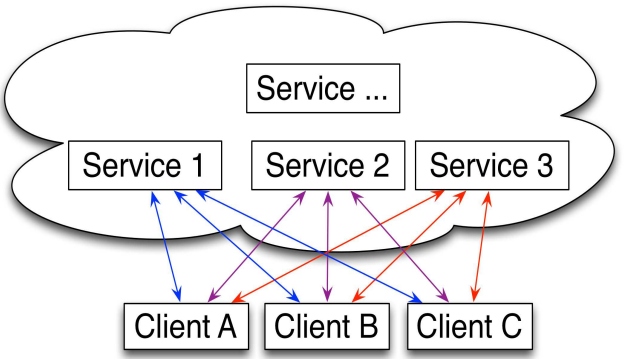
How to get a map?

- getMap
- getImage
- get2dmap
- getlocation
- ...

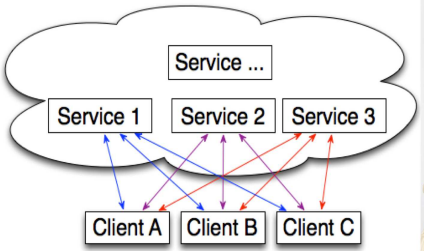
Heterogeneous Services



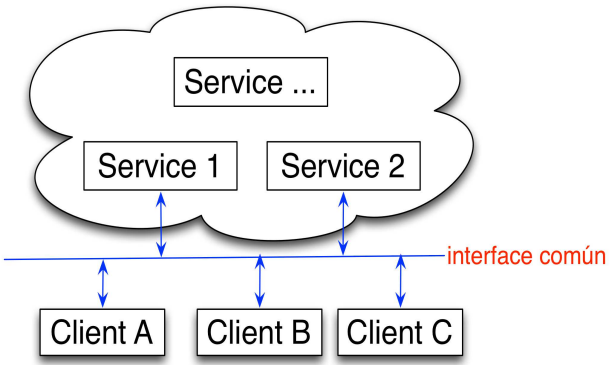
Heterogeneous Services



Lack of agreement looks bad



Agreement of Interfaces- Great!



Web Feature Service (WFS)

Web Feature Service (WFS)

- Service (Protocol)
- Do the following with Geographic Features:
    - publish
    - access
    - manipulate

WFS Operations

- *getCapabilities* - summary of the service
- *DescribeFeatureType* - structure of the feature types
- *GetFeature* - get the feature instance
- *Transaction* - create, update and delete geographic features
- *Lock Feature* - Protects feature record when updating it.

Geographic Modeling Language (GML)

Feature

Feature has Properties

Geospatial Feature

Generalization

Class = Feature Type = Tree

Properties of Tree:

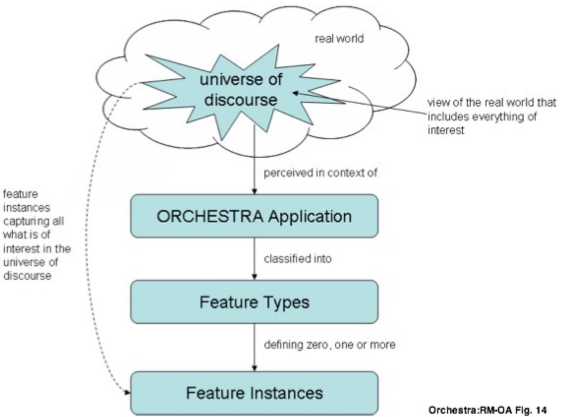
- Height
- Location

All my Trees will have a height and location.

Feature Instance

My favorite tree in Cartagena is the one in front of the hotel Colombia.
Height = 5 mt
Location: Latitude, Longitude
10.407793,-75.551262

Features Modeling



Orchestra:RM-OA Fig. 14

Geographic Markup Language (GML)

```
<Bridge>
  <span>100</span>
  <height>200</height>
  <gml:centerLineOf>
    <gml:LineString>
       <gml:pos>100 200</gml:pos>
       <gml:pos>200 200</gml:pos>
    </gml:LineString>
  </gml:centerLineOf>
</Bridge>
```

Feature Instance of Type Bridge

```
<Bridge>
  <span>100</span>
  <height>200</height>
  <gml:centerLineOf>
    <gml:LineString>
       <gml:pos>100 200</gml:pos>
       <gml:pos>200 200</gml:pos>
    </gml:LineString>
  </gml:centerLineOf>
</Bridge>
```

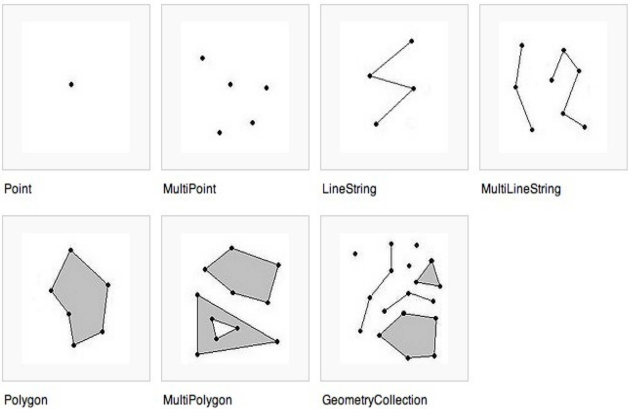
Feature Types are defined in GML

```
<xs:complexType name="BridgeType">
     <xs:complexContent>
          <xs:extension base="gml:AbstractFeatureType">
              <xs:sequence>
              <xs:element name="span"  type="xs:integer"/>
              <xs:element name="height" type="xs:integer"/>
                 <xs:element ref="gml:centerLineOf"/>
               </xs:sequence>
            </xs:extension>
     </xs:complexContent>
 </xs:complexType>
```

GML Provides

- Defines an abstract feature model
- Provides XML Schemas
- Provides primitives:
    - Geometry
    - Feature Types
    - Coordinate Reference Systems

GML Geometries



How can radio station be modeled?

- Point - when looking at a country map
- Polygon - location of the building
- Multipolygon - Area of Transmition

Communities develop their own Application Schemas

- AIXM – Aviation
- CAAML – Canadian avalanches
- CityGML – 3D city models
- CSML – Climate Science Modelling
- DAFIF – Defense aviation
- GeoSciML - Geoscience
- ...

WEB Map Service (WMS)

This tutorial provides a practical introduction to OGC Web Map Service (WMS) Interface standard.

Goals

- Understand what WMS can be used for
- Understand WMS requests and best practices

Web Map Service (WMS)

The latest version of WMS is 1.3.0 [#ogc-06-042].

A WMS Server:

- Provides information about what maps a service can produce
- Produces a Map
- Answers queries about content of a Map

WMS Usage

- Produce a Map
- Answer queries about content of the map

WMS Operations

- GetCapabilities
- GetMap
- GetFeatureInfo

WMS GetCapabilities Request

Request example:

```
http://ri.opengeospatial.org:8680/degree-wms-130/services?
service=WMS&
version=1.3.0&
request=GetCapabilities
```

Link to getCapabilities

WMS GetCapabilities Response

- How to invoke GetMap
- Types of exceptions
- List of layers

WMS GetMap Request

Request example:

```
http://ri.opengeospatial.org:8680/degree-wms-130/services?
service=WMS&
version=1.3.0&
request=GetMap&
format=image/png&
width=300&
height=300&
crs=EPSG:4326&
layers=cite:BasicPolygons,cite:Forests&
STYLES=default,default
BBOX=-2,-1,2,6&
```

Link to GetMap

WMS GetFeatureInfo Request

Request example:

```
http://ri.opengeospatial.org:8680/degree-wms-130/services?
service=WMS&
version=1.3.0&
request=getfeatureinfo&
format=text/html&
crs=EPSG:4326&
layers=cite:BasicPolygons&
query_layers=cite:BasicPolygons&
BBOX=-2,-1,2,6&
width=300&
height=300&
i=1&
j=1
```

Link to GetFeatureInfo

KML

This tutorial provides an introduction to KML

KML is used for

- Annotate the Earth
- Specify icons and labels to identify locations on the surface of the planet
- Create different camera positions to define unique views for KML features
- Define image overlays to attach to the ground or screen
- Define styles to specify KML feature appearance

KML is used for

- Write HTML descriptions of KML features, including hyperlinks and embedded images
- Organize KML features into hierarchies
- Locate and update retrieved KML documents from local or remote network locations
- Define the location and orientation of textured 3D objects

KML Example

```
<kml xmlns="http://www.opengis.net/kml/2.2">
    <Document>
       <name>Document.kml</name> <open>1</open>
       <Style id="exampleStyleDocument">
           <LabelStyle>
           <color>ff0000cc</color>
           </LabelStyle>
         </Style>
```

KML Example

```
<Placemark>
    <name>Document Feature 1</name>
    <styleUrl>#exampleStyleDocument</styleUrl>
    <Point>
       <coordinates>-122.371,37.816,0</coordinates>
    </Point>
</Placemark>
```

KML Example

```
      <Placemark>
          <name>Document Feature 2</name>
          <styleUrl>#exampleStyleDocument</styleUrl>
          <Point>
             <coordinates>-122.370,37.817,0</coordinates>
          </Point>
      </Placemark>
  </Document>
</kml>
```

Overview Sensor Web Enablement (SWE)

SWE definition

Set of OGC standards that work together to better:

- discover
- access
- control
- use sensor data.

SWE scope

Let's call all these **Sensor Systems** or *Systems*

- Detectors
- Sensors
- Sensor Networks
- Platforms

SWE motivation

- Systems are disconnected
- Systems are heterogeneous
- Systems produce massive amount of data

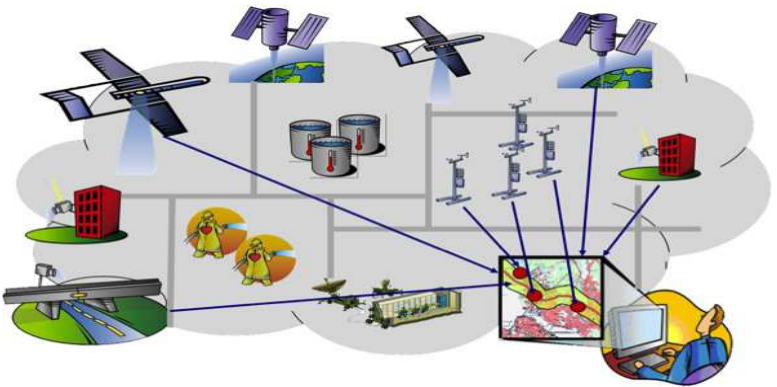
Systems are disconnected

Systems are heterogeneous

Systems produce lots of data

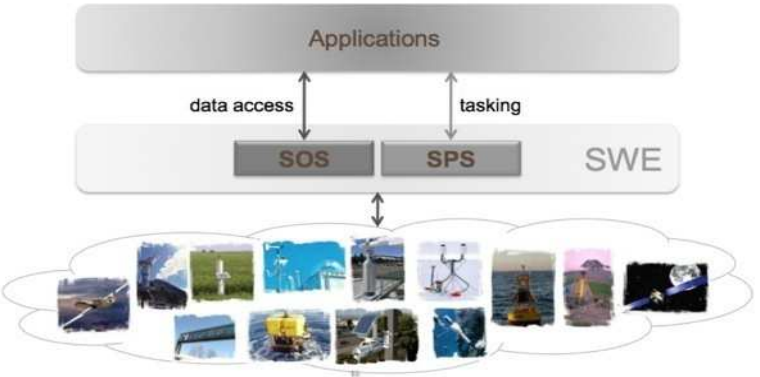
640 terabytes of operational data on just one Atlantic crossing

User connected to sensor systems



SWE requirements

SWE solution



SWE encoding standards

- Observations and Measurements (O&M)
- SensorML
- SWECommon
- PUCK

SWE interface standards

- Sensor Observation Service
- Sensor Planning Service
- PUCK

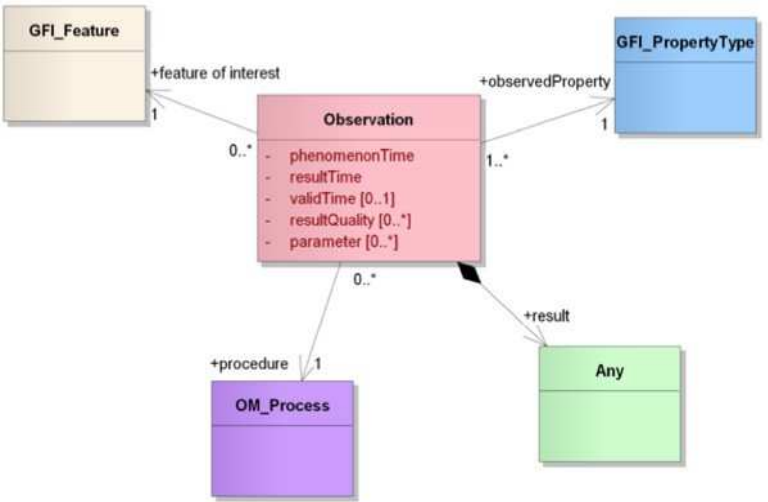
Observations and Measurements (O&M)

This tutorial provides an introduction to the OGC Encoding Standard Observations and Meaurementes

O&M

- An observation is an *event* that
- estimates an *observed property*
- of a *feature of interest*,
- using a *procedure*, and
- generating a *result*

O&M UML


GFI_Feature
+feature of interest
1
0..*
Observation
- phenomenonTime
- resultTime
- validTime [0..1]
- resultQuality [0..*]
- parameter [0..*]
+observedProperty
1
1..*
GFI_PropertyType
0..*
+procedure
1
OM_Process
+result
Any

O&M XML Example

Lets walk through an example from Geonovum

O&M XML Header

```
<?xml version="1.0" encoding="windows-1250"?>
     <om:Measurement gml:id="obsTest"
        xmlns:om="http://www.opengis.net/om/1.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xmlns:gml="http://www.opengis.net/gml"
          xsi:schemaLocation="http://www.opengis.net/om/1.0
          ../extensions/observationSpecialization_override.xsd">
```

O&M XML Time

```
<om:samplingTime>
  <gml:TimeInstant>
    <gml:timePosition>2008-10-14T00:09:53+02:00</gml:timePosition>
  </gml:TimeInstant>
</om:samplingTime>
```

O&M XML Procedure

```
<om:procedure
   xlink:href=
         "urn:ogc:object:feature:OSIRIS-HWS:
            a1ef1094-c201-4f9f-8f2e-0ff97bf65f03"/>
```

O&M XML Observed Property

```
<om:observedProperty
   xlink:href="urn:x-ogc:def:property:OGC::RelativeHumidity"/>
```

O&M XML Feature of Interest

```
   <om:featureOfInterest>
         <sa:SamplingPoint
      gml:id=
         "urn:ogc:object:feature:OSIRIS-HWS:
            a1ef1094-c201-4f9f-8f2e-0ff97bf65f03"
      xsi:type="ns:SamplingPointType"
         xmlns:ns="http://www.opengis.net/sampling/1.0"
      xmlns:gml="http://www.opengis.net/gml">
    <gml:name>roof of the ifgi</gml:name>
    <sa:sampledFeature
      xlink:href="urn:ogc:object:feature:
         OSIRIS-HWS:hygrometera1ef1094-c201-4f9f-8f2e-0ff97bf65f03"/>
    <sa:position>
      <gml:Point srsName="urn:ogc:def:crs:EPSG:4326">
        <gml:pos>52.07349 9.42125</gml:pos>
      </gml:Point>
    </sa:position>
  </sa:SamplingPoint>
</om:featureOfInterest>
```

O&M XML Observed Property

```
   <om:result uom="%"
      xlink:href="urn:x-ogc:def:uom:OGC:percent" >41
   </om:result>
</om:Measurement>
```

SWECommon

SWECommon provides

- primitive data types (boolean, categories, text, quantities ..)
- aggregate data types (records, arrays, matrices)
- specialized data types (curves, time dependent)
- Structure to encode quality information
- Structure to provide semantic annotations

SWECommon relation with other encoding standards

- In SensorML inputs and outputs are express via SWECommon
- In O&M the result can be expressed with SWECommon

SWECommon XML Example

Namespace declaration:

```
<?xml version="1.0" encoding="UTF-8"?>
<swe:DataStream id="EXAMPLE_01"
   xmlns:swe="http://www.opengis.net/swe/2.0"
   xmlns:gml="http://www.opengis.net/gml/3.2"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xsi:schemaLocation=
         "http://www.opengis.net/swe/2.0 http://schemas.opengis.net/sweCommon/2.0/swe.xsd">
```

SWECommon XML Example

Element Count:

```
<swe:elementCount>
   <swe:Count>
      <swe:value>10</swe:value>
   </swe:Count>
</swe:elementCount>
```

SWECommon XML Example

Data Record and definition of first field (Time):

```
<swe:elementType name="weather_data">
   <swe:DataRecord>
      <!-- -->
      <swe:field name="time">
         <swe:Time definition=
               "http://www.opengis.net/def/property/OGC/0/SamplingTime">
            <swe:label>Sampling Time</swe:label>
            <swe:uom xlink:href=
               "http://www.opengis.net/def/uom/ISO-8601/0/Gregorian"/>
         </swe:Time>
      </swe:field>
```

SWECommon XML Example

Definition of second field (Temperature):

```
<swe:field name="temperature">
   <swe:Quantity definition=
         "http://mmisw.org/ont/cf/parameter/air_temperature">
      <swe:label>Air Temperature</swe:label>
      <swe:uom xlink:href="Cel"/>
      <swe:constraint>
         <swe:AllowedValues>
            <swe:value>1</swe:value>
            <swe:value>2</swe:value>
                     <swe:value>3</swe:value>
            <swe:interval>-50 +50</swe:interval>
            <swe:significantFigures>2</swe:significantFigures>
         </swe:AllowedValues>
```

```
      </swe:constraint>
   </swe:Quantity>
</swe:field>
```

SWECommon XML Example

Definition of third field (Pressure):

```
<swe:field name="pressure">
   <swe:Quantity definition=
         "http://mmisw.org/ont/cf/parameter/air_pressure">
      <swe:label>Atmospheric Pressure</swe:label>
      <swe:quality>
         <swe:Quantity definition=
               "http://sweet.jpl.nasa.gov/2.0/sciUncertainty.owl#Accuracy">
            <swe:uom code="%"/>
            <swe:value>10</swe:value>
         </swe:Quantity>
      </swe:quality>
      <swe:uom code="mbar"/>
   </swe:Quantity>
</swe:field>
```

SWECommon XML Example

Definition of fourth and fifth fields (Wind Vector):

```
      <swe:field name="windSpeed">
         <swe:Quantity definition=
                  "http://mmisw.org/ont/cf/parameter/wind_speed">
            <swe:uom code="km/h"/>
         </swe:Quantity>
      </swe:field>
      <!-- -->
      <swe:field name="windDirection">
         <swe:Quantity definition=
                "http://mmisw.org/ont/cf/parameter/wind_to_direction">
            <swe:uom code="deg"/>
         </swe:Quantity>
      </swe:field>
      <!-- -->
   </swe:DataRecord>
</swe:elementType>
```

SWECommon XML Example

Definition of the encoding and the data:

```
   <swe:encoding>
      <swe:TextEncoding tokenSeparator=","
            blockSeparator="&#32;"
            decimalSeparator="."/>
   </swe:encoding>
   <swe:values>
      2009-01-01T10:00:25Z,25.3,1098,5,56
      2009-01-01T10:00:35Z,25.4,1098,15,59
      2009-01-01T10:00:45Z,25.4,1098,12,42
      ...
   </swe:values>
</swe:DataStream>
```

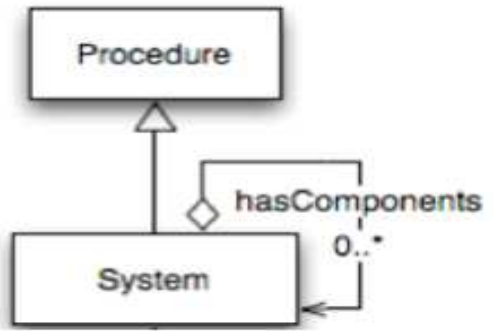
Sensor Model Language (SensorML)

SensorML For Sensor Systems

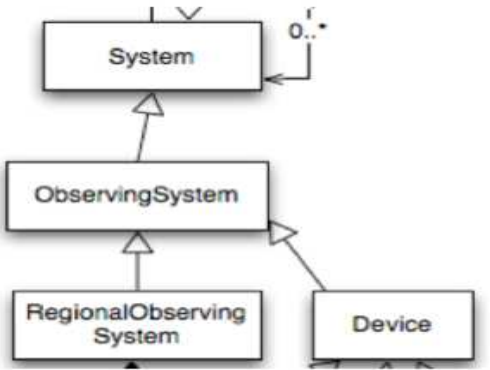
Model to describe Sensor Systems:

- Detector
- Sensor
- Platform
- Observatories

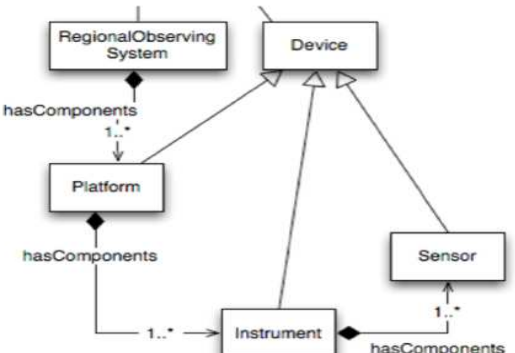
SensorML can define systems



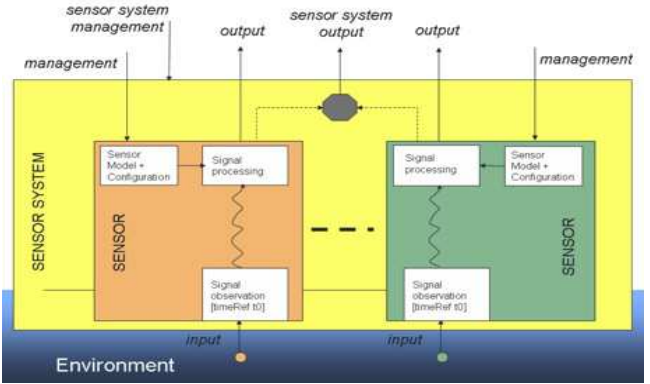
Some types of environmental systems



Instrument composition



SensorML Instrument System



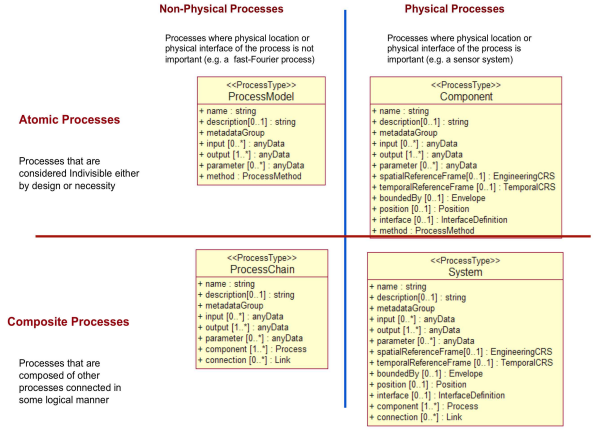
Sensor Web Enablement Architecture, OGC-06-021r4

SensorML can describe a process

Model to describe processes

- inputs
- output
- service chaining

SensorML Processes



SensorML XML Example

Lets walk through a SensorML example from Geonovum

SensorML XML Keywords

```
<keywords>
  <KeywordList>
    <keyword>weather station</keyword>
    <keyword>precipitation</keyword>
    <keyword>wind speed</keyword>
    <keyword>temperature</keyword>
  </KeywordList>
</keywords>
```

SensorML XML Identification

```
<identification>
  <IdentifierList>
    <identifier name="uniqueID">
      <Term definition="urn:ogc:def:identifier:OGC:uniqueID">
        <value>urn:ogc:object:feature:Sensor:IFGI:weatherStation123</value>
      </Term>
    </identifier>
    <identifier name="longName">
      <Term definition="urn:ogc:def:identifier:OGC:1.0:longName">
        <value>OSIRIS weather station 123 on top of the IfGI building</value>
      </Term>
    </identifier>
    <identifier name="shortName">
      <Term definition="urn:ogc:def:identifier:OGC:1.0:shortName">
        <value>OSIRIS Weather Station 123</value>
      </Term>
    </identifier>
  </IdentifierList>
</identification>
```

SensorML XML Classification

```
<classification>
  <ClassifierList>
    <classifier name="intendedApplication">
      <Term definition="urn:ogc:def:classifier:OGC:1.0:application">
        <value>weather</value>
      </Term>
    </classifier>
  </ClassifierList>
</classification>
```

SensorML XML Valid Time

```
<validTime>
  <gml:TimePeriod>
    <gml:beginPosition>2009-01-15</gml:beginPosition>
    <gml:endPosition>2009-01-20</gml:endPosition>
  </gml:TimePeriod>
</validTime>
```

SensorML XML Capabilities

```
<capabilities>
  <swe:DataRecord definition="urn:ogc:def:property:capabilities">
    <swe:field name="status">
      <swe:Text definition="urn:ogc:def:property:OGC:1.0:status">
        <gml:description>The operating status of the system.</gml:description>
        <!-- station is active -->
        <swe:value>active</swe:value>
      </swe:Text>
    </swe:field>
    <!-- Area that is observed by the station. In this case is insitu
     It matches the location of the station.  -->
    <swe:field name="observedBBOX">
      <swe:Envelope definition="urn:ogc:def:property:OGC:1.0:observedBBOX">
        <swe:lowerCorner>
```

```
<swe:Vector>
  <swe:coordinate name="easting">
    <swe:Quantity axisID="x">
```

```
            <swe:uom code="m"/>
            <swe:value>2592308.332</swe:value>
          </swe:Quantity>
        </swe:coordinate>
        <swe:coordinate name="northing">
          <swe:Quantity axisID="y">
            <swe:uom code="m"/>
            <swe:value>5659592.542</swe:value>
          </swe:Quantity>
        </swe:coordinate>
      </swe:Vector>
    </swe:lowerCorner>
    <swe:upperCorner>
      <swe:Vector>
        <swe:coordinate name="easting">
          <swe:Quantity axisID="x">
            <swe:uom code="m"/>
```

```
              <swe:value>2592308.332</swe:value>
            </swe:Quantity>
          </swe:coordinate>
          <swe:coordinate name="northing">
            <swe:Quantity axisID="y">
              <swe:uom code="m"/>
              <swe:value>5659592.542</swe:value>
            </swe:Quantity>
          </swe:coordinate>
        </swe:Vector>
      </swe:upperCorner>
    </swe:Envelope>
  </swe:field>
 </swe:DataRecord>
</capabilities>
```

SensorML XML Contact

```
<contact>
  <ResponsibleParty gml:id="WWU_IfGI_weather_station_contact">
    <organizationName>Westfälische Wilhelms-Universität Münster - Sensor Web and Simulation Lab</organizationName>
    <contactInfo>
      <address>
        <electronicMailAddress>swsl-ifgi@listserv.uni-muenster.de</electronicMailAddress>
      </address>
    </contactInfo>
  </ResponsibleParty>
</contact>
```

SensorML Position

```
<position name="systemPosition">
  <swe:Position referenceFrame="urn:ogc:def:crs:EPSG:6.14:31466">
    <swe:location>
      <swe:Vector gml:id="SYSTEM_LOCATION">
        <swe:coordinate name="easting">
          <swe:Quantity axisID="x">
            <swe:uom code="m"/>
            <swe:value>2592308.332</swe:value>
          </swe:Quantity>
        </swe:coordinate>
        <swe:coordinate name="northing">
          <swe:Quantity axisID="y">
            <swe:uom code="m"/>
            <swe:value>5659592.542</swe:value>
```

```
          </swe:Quantity>
        </swe:coordinate>
        <swe:coordinate name="altitude">
          <swe:Quantity axisID="z">
            <swe:uom code="m"/>
            <swe:value>297.0</swe:value>
          </swe:Quantity>
        </swe:coordinate>
      </swe:Vector>
    </swe:location>
  </swe:Position>
</position>
```

SensorML Inputs

```
<inputs>
  <InputList>
    <input name="precipitation">
      <swe:ObservableProperty definition="urn:ogc:def:property:OGC:1.0:precipitation"/>
    </input>
    <input name="wind">
      <swe:ObservableProperty definition="urn:ogc:def:property:OGC:1.0:wind"/>
    </input>
    <input name="atmosphericTemperature">
      <swe:ObservableProperty definition="urn:ogc:def:property:OGC:1.0:temperature"/>
    </input>
  </InputList>
</inputs>
```

SensorML Outputs

```
<outputs>
  <OutputList>
    <output name="precipitation">
      <swe:Quantity definition="urn:ogc:def:property:OGC:1.0:precipitation">
        <swe:uom code="mm"/>
      </swe:Quantity>
    </output>
    <output name="windDirection">
      <swe:Quantity definition="urn:ogc:def:property:OGC:1.0:windDirection">
        <swe:uom code="deg"/>
      </swe:Quantity>
    </output>
    <output name="windSpeed">
      <swe:Quantity definition="urn:ogc:def:property:OGC:1.0:windSpeed">
```

```
        <swe:uom code="m/s"/>
      </swe:Quantity>
    </output>
```

```
    <output name="temperature">
      <swe:Quantity definition="urn:ogc:def:property:OGC:1.0:temperature">
        <swe:uom code="Cel"/>
      </swe:Quantity>
    </output>
  </OutputList>
</outputs>
```

SensorML Components

```
<components>
  <ComponentList>
    <component name="rainGauge" xlink:href="http://mySensorMLregistry.com?object=98765"/>
    <component name="anemoneter" xlink:href="http://mySensorMLregistry.com?object=33333"/>
    <component name="thermometer">
      <Component>
        ...
      </Component>
    </component>
  </ComponentList>
</components>
```

SensorML Component

```
<!-- similar to System, Contact and Position Information are inherited -->
<Component>
  <keywords>
    ...
  </keywords>
  <identification>
    <IdentifierList>
      <identifier name="uniqueID">
        <Term definition="urn:ogc:def:identifier:OGC:uniqueID">
          <value>urn:ogc:object:feature:Sensor:IFGI:thermometer123</value>
        </Term>
      </identifier>
      <identifier name="longName">
        <Term definition="urn:ogc:def:identifier:OGC:1.0:longName">
```

```
          <value>OSIRIS Thermometer at weather station 123</value>
        </Term>
      </identifier>
      <identifier name="shortName">
        <Term definition="urn:ogc:def:identifier:OGC:1.0:shortName">
```

```
        <value>OSIRIS Thermometer 123</value>
      </Term>
    </identifier>
  </IdentifierList>
</identification>
<classification>
  <ClassifierList>
    <classifier name="sensorType">
      <Term definition="urn:ogc:def:classifier:OGC:1.0:sensorType">
        <value>thermometer</value>
      </Term>
    </classifier>
  </ClassifierList>
</classification>
<capabilities>
  <swe:DataRecord definition="urn:ogc:def:property:capabilities">
    <swe:field name="status">
```

```
      <swe:Text definition="urn:ogc:def:property:OGC:1.0:status">
        <gml:description>The operating status of the system.</gml:description>
        <swe:value>active</swe:value>
      </swe:Text>
    </swe:field>
```

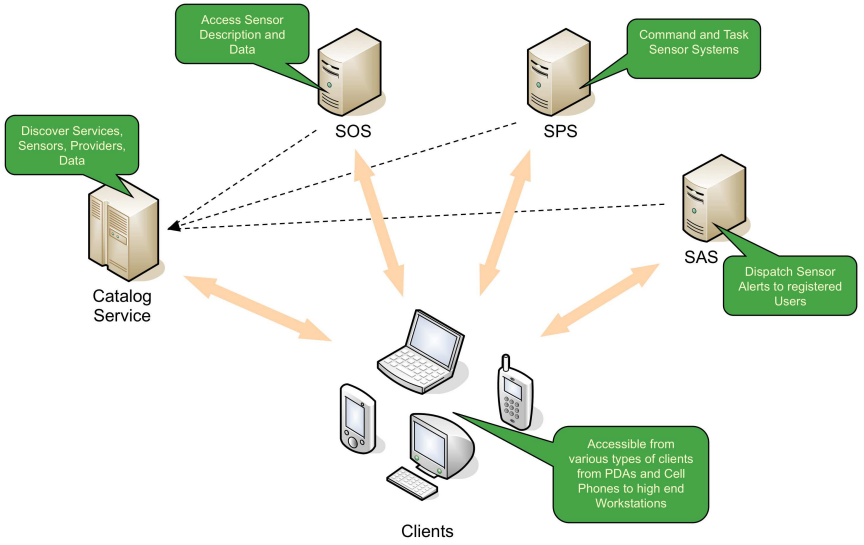
```
    </swe:DataRecord>
  </capabilities>
  <inputs>
    <InputList>
      <input name="atmosphericTemperature">
        <swe:ObservableProperty definition="urn:ogc:def:property:OGC:1.0:temperature"/>
      </input>
    </InputList>
  </inputs>
  <outputs>
    <OutputList>
      <output name="temperature">
        <swe:Quantity definition="urn:ogc:def:property:OGC:1.0:temperature">
          <gml:groupName codeSpace="ObservationOffering"> Weather </gml:groupName>
          <swe:uom code="Cel"/>
        </swe:Quantity>
      </output>
    </OutputList>
  </outputs>
</Component>
```

SWE Services

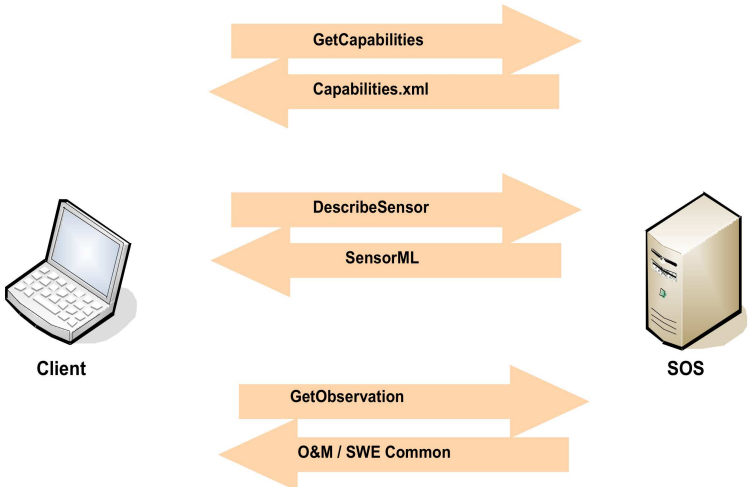
SWE Services

- **Sensor Observation Service** – Access observations and descriptions for sensor systems
- **Sensor Alert Service** – Subscribe to alerts based upon sensor observations
- **Sensor Planning Service** – Request collection feasibility and task sensor system for desired observations
- **Web Notification Service** – Manage message dialogue between client services for long duration (asynchronous) processes
- **Sensor Registries** – Discover sensors and sensor observations

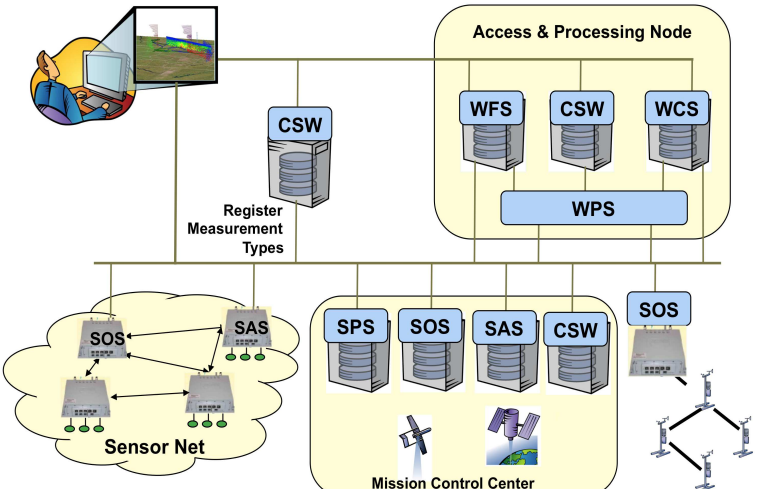
SWE Services


Access Sensor Description and Data
SOS
Command and Task Sensor Systems
SPS
Discover Services, Sensors, Providers, Data
Catalog Service
SAS
Dispatch Sensor Alerts to registered Users
Accessible from various types of clients from PDAs and Cell Phones to high end Workstations
Clients

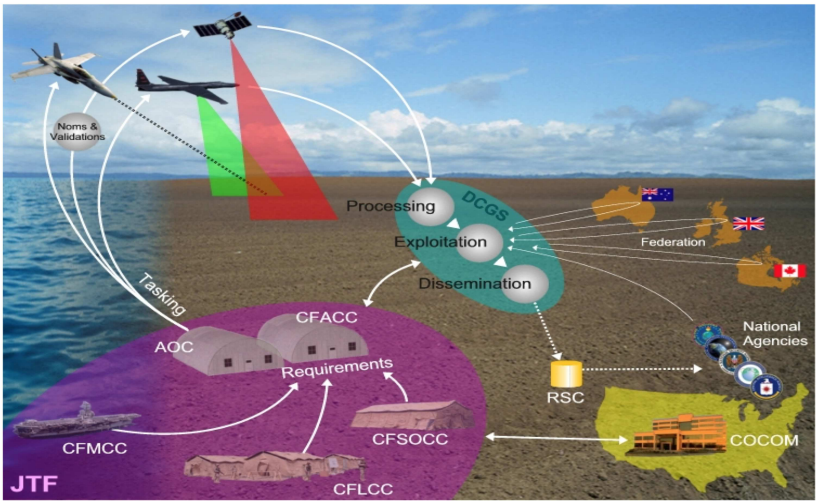
SOS Operations



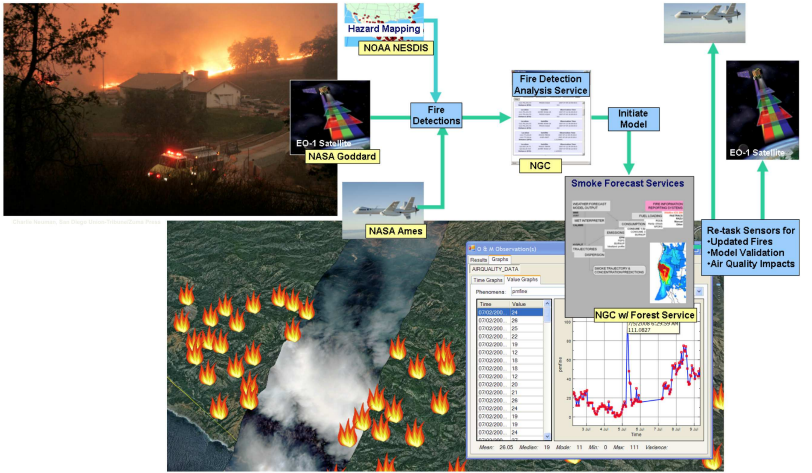
SWE and Geo-processing Workflow



SWE in Defense



SWE in Air Quality



Metadata and Catalogs

- Metadata
- Search
- Catalog
- FGDC
- ISO
- Profiles
- CSW Operations
- OpenSearch

Resource

- A thing
- Anything which is worth uniquely identifying (over the Web)
- Can be data

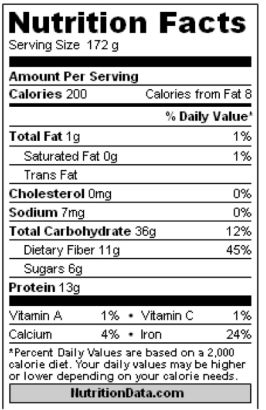
Metadata

- Data about a resource
- Data about data

Metadata

This is the metadata for this.

What is Missing ?



Nutrition Facts

Serving Size 172 g

Amount Per Serving

Calories 200 — Calories from Fat 8

	% Daily Value*
Total Fat 1g	1%
Saturated Fat 0g	1%
Trans Fat	
Cholesterol 0mg	0%
Sodium 7mg	0%
Total Carbohydrate 36g	12%
Dietary Fiber 11g	45%
Sugars 6g	
Protein 13g	

Vitamin A	1%	• Vitamin C	1%
Calcium	4%	• Iron	24%

*Percent Daily Values are based on a 2,000 calorie diet. Your daily values may be higher or lower depending on your calorie needs.

NutritionData.com

What is Missing ?

Metadata - Who

- Who collected the data?
- Who processed the data?
- Who wrote the metadata?
- Who to contact for questions?
- Who to contact to order?
- Who owns the data?

Metadata - What

- What are the data about?
- What project were they collected under?
- What are the constraints on their use?
- What is the quality?
- What are appropriate uses?
- What parameters were measured?
- What format are the data in?

Metadata - Why

- Why were the data collected?

Metadata - Why

- Why were the data collected?

Where

- Where were the data collected?
- Where were the data processed?
- Where are the data located?

When

- When were the data collected?
- When were the data processed?

How

- How were the data collected?
- How were the data processed?
- How do I access the data?
- How do I order the data?
- How much do the data cost?
- How was the quality assessed?

Metadata requires update

1980	2005
British Honduras	Belize
West Germany	Germany

Metadata Value

- Organizations: captures the knowhow of an organization
- Developers: help share reliable information
- Users: helps discover data

Search

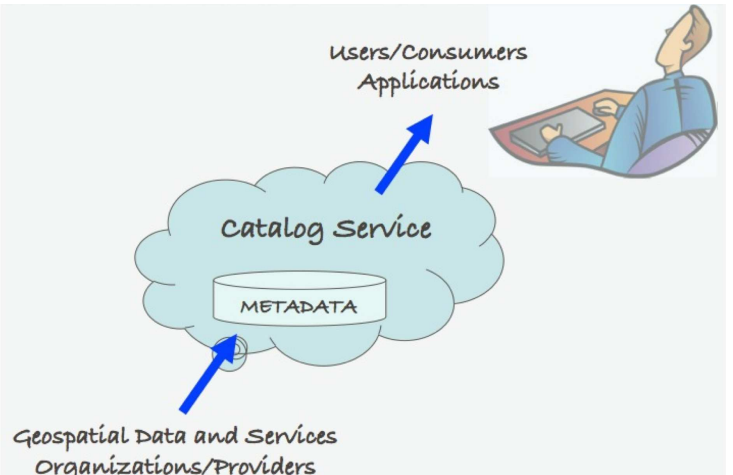
Discovery & evaluation of resources through (summary) metadata

Catalog

- Organized, detailed, descriptive list of items
- arranged systematically (so they can be found)

Catalog

Catalog Service



Registry

System for maintaining a register or authoritative list of names / values / types / relationships (so they can be referenced)

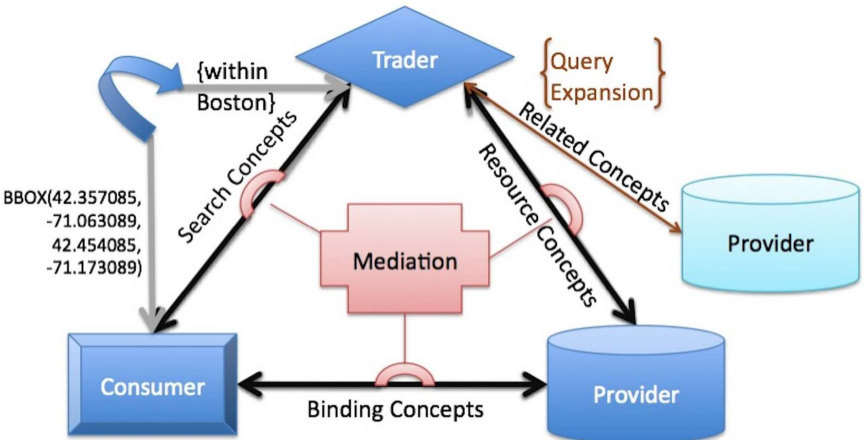
Repository or Archive

Place for storage of resources (so they can be retrieved)

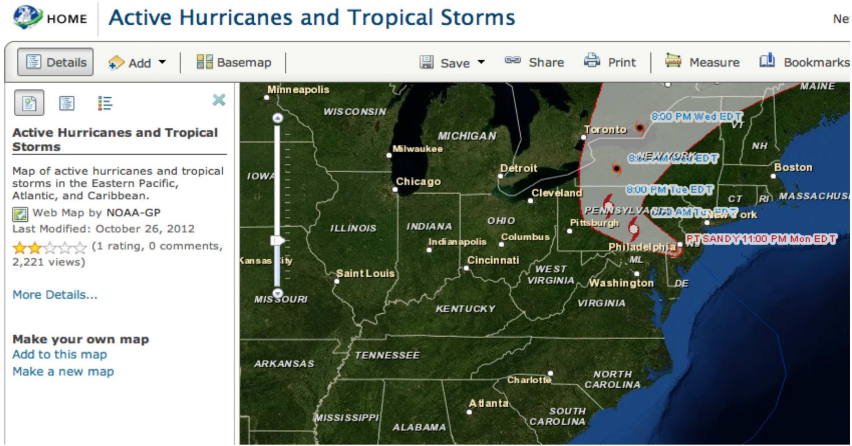
Trader

- Intermediary in a service oriented architecture
- Connects providers with consumers)

Discovery using SOA



Example: Geospatial Platform



Metadata for Hurricane Map

Active Hurricanes and Tropical Storms

Map of active hurricanes and tropical storms in the Eastern Pacific, Atlantic, and Caribbean.

Web Map by NOAA-GP
Last Modified: October 26, 2012
★★☆☆☆ (1 rating, 2,222 views)
Sign In to rate this item.

Open

Description

This map contains the following layers:

Watches, Warnings and Advisories
A warning is issued when a hazardous weather or hydrologic event is occurring, imminent or likely. A warning means weather conditions pose a threat to life or property.
People in the path of the storm need to take protective action.

A watch is used when the risk of a hazardous weather or hydrologic event has increased significantly, but its occurrence, location or timing is still uncertain.

Comments (0)

Metadata for Hurricane Map

Access and Use Constraints

> The services in this map are not operational and therefore may not be available 24x7.

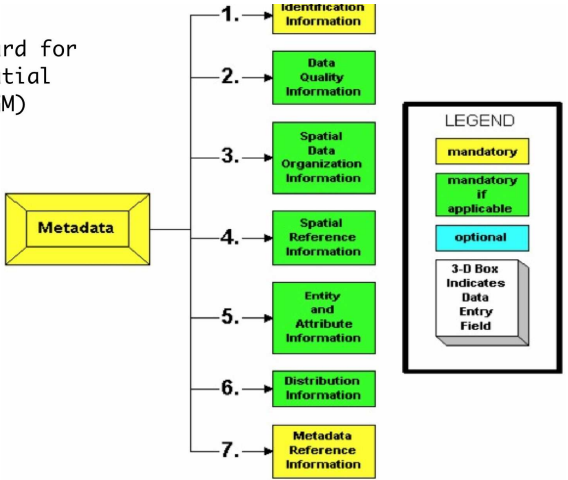
Map Contents

> Atlantic and Caribbean:
> http://gis.srh.noaa.gov/ArcGIS/rest/services/AtStormViewer/MapServer
> Eastern Pacific:
> http://gis.srh.noaa.gov/ArcGIS/rest/services/EpStormViewer/MapServer
> Imagery with Labels:
> http://services.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer
> Imagery with Labels:
> http://services.arcgisonline.com/ArcGIS/rest/services/Reference/World_Boundaries_and_Places/

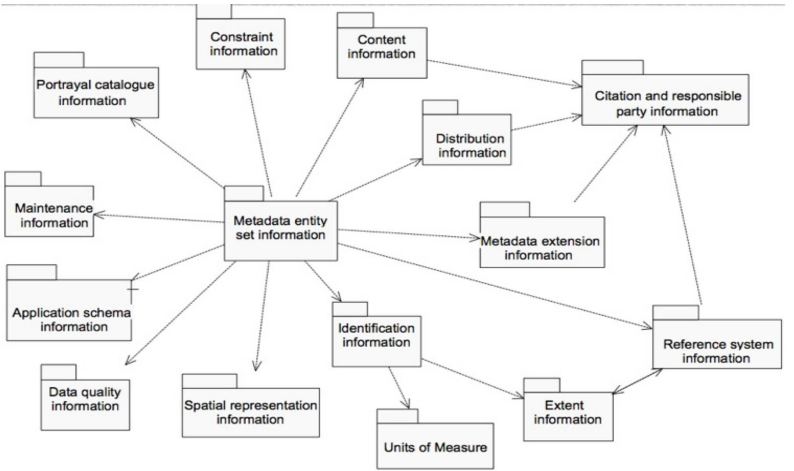
Tags	hurricane forecast, tropical storms
Credits	National Weather Service, National Hurricane Center
Size	184 KB
Extent	Left: -95.05 Right: -50.8 Top: 44.73 Bottom: 16.44

FGDC

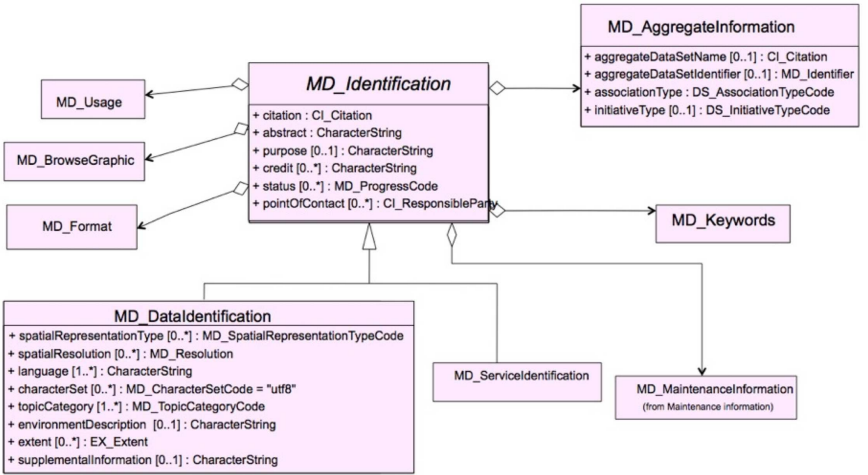
Content Standard for
Digital Geospatial
Metadata (CSDGM)



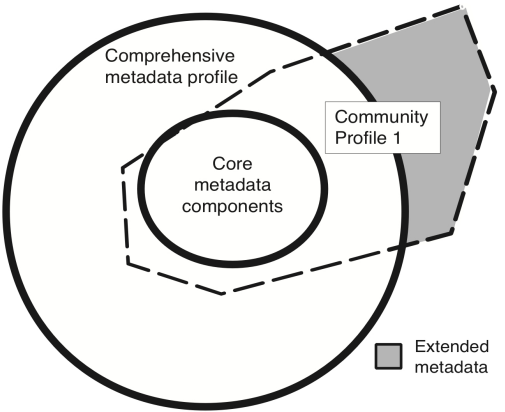
ISO 19115 Geographic Information



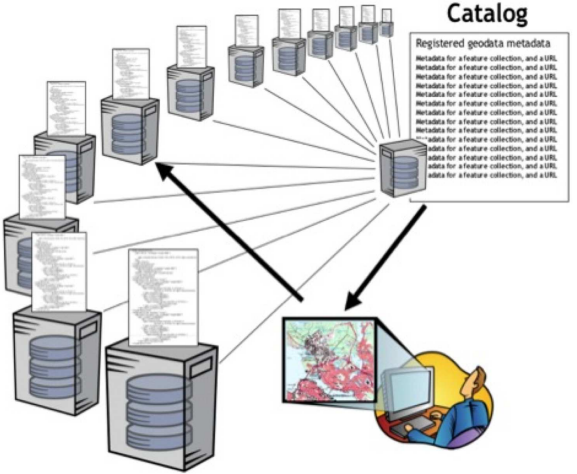
ISO 19115 Geographic Information



Profiles


Comprehensive
metadata profile
Community
Profile 1
Core
metadata
components
Extended
metadata

Catalog Service



Catalog Services

- CSW
- ISO 19119 Metadata Profile
- Z39.50 Profile
- OASIS ebRIM Profile
- OpenSearch

GEOSS Registry

Example Services

GI CAT

PYCSW

ESRI GeoPortal

CSW Operations

- GetCapabilities
- DescribeRecord
- GetRecordById
- GetRecords

CSW GetCapabilities

Request example:

```
http://ec2-174-129-9-172.compute-1.amazonaws.com/gi-cat-RI/services/cswiso?
service=CSW&
version=2.0.2&
request=GetCapabilities
```

Link

CSW DescribeRecord

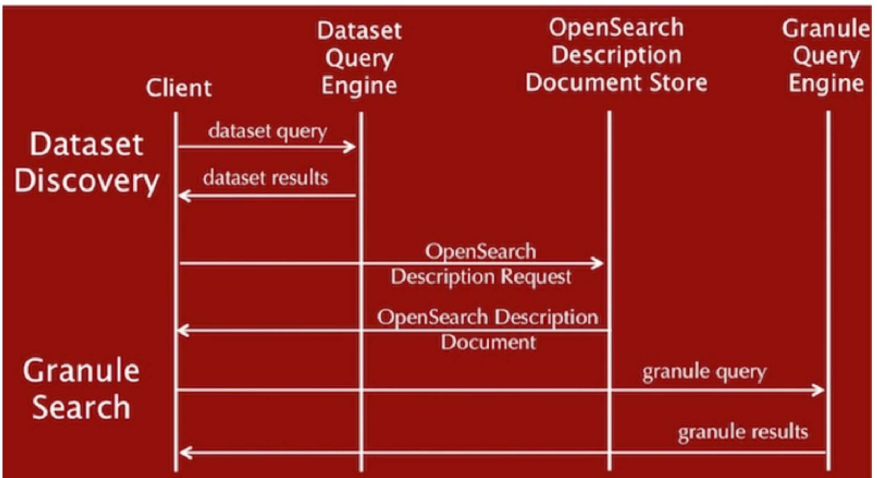
Request example:

```
http://ec2-174-129-9-172.compute-1.amazonaws.com/gi-cat-RI/services/cswiso?
service=CSW&
version=2.0.2&
request=DescribeRecord
```

Link

CSW Get Record

OpenSearch


Client
Dataset Query Engine
OpenSearch Description Document Store
Granule Query Engine
Dataset Discovery
dataset query
dataset results
OpenSearch Description Request
OpenSearch Description Document
Granule Search
granule query
granule results

OpenSearch

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription
     xmlns="http://a9.com/-/spec/opensearch/1.1/">
  <ShortName>Mirador Dataset Search</ShortName>
  <Description>Use Mirador Dataset Search to obtain a
  list of Earth Science Data Sets</Description>
  <Tags>Mirador Dataset Search</Tags>
  <Contact>mirador-disc@listserv.gsfc.nasa.gov</Contact>
  ...
```

OpenSearch

```
<Url type="application/atom+xml"
        template="http://mirador.gsfc.nasa.gov/cgi-bin/
           mirador/collectionlist.pl?
        keyword={searchTerms}&
        page=1&
        count={count}&
        osLocation={geo:box}&
        startTime={time:start}&
        endTime={time:end}&
        format=atom"/>
```

OpenSearchGeo

```
http://example.com/?
q=pizza&
bbox=-111.032,42.943,-119.856,43.039&
format=rss
```

OpenSearchGeo

```
http://example.com/?
q=pizza&
lat=43.25&lon=-123.45
&radius=10000&
format=rss
```

OpenSearchGeo

```
http://example.com/?
q=pizza&
l=boston&
format=rss
```

Example Tool: GeoNetwork

Credits

- NOAA NCDDC Metadata training materials

Web Processing Service (WPS)

This tutorial provides an introduction to the OGC Web processing Service (WPS) Interface Standard.

Geoprocessing

A GIS operation used to manipulate GIS data.

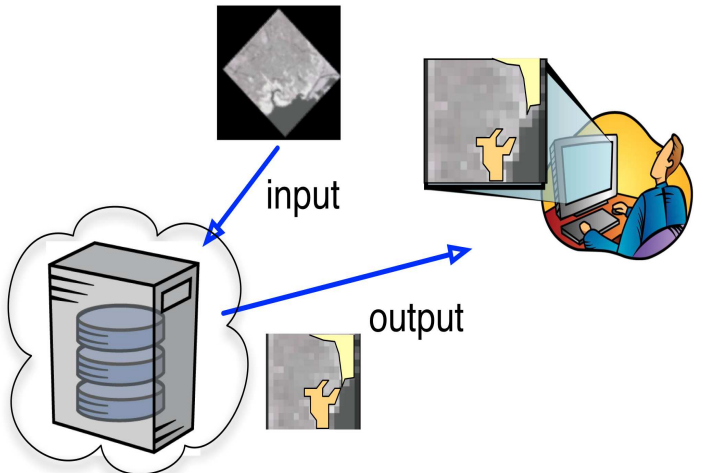
Examples?

Data Processing

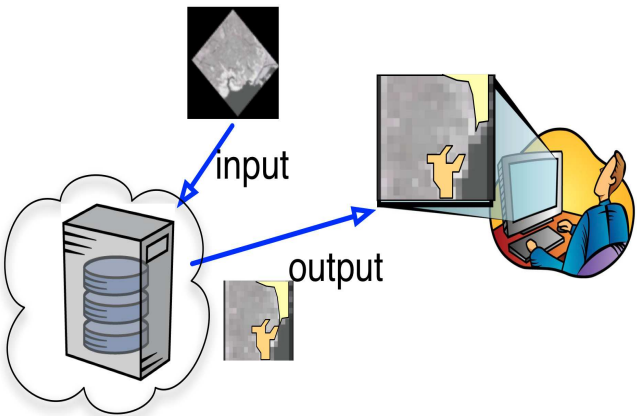
Types of Operations

- Feature Overlay
- Feature Selection
- Topology processing
- Raster processing
- Data Conversion
- Feature Analysis

Web Processing Service

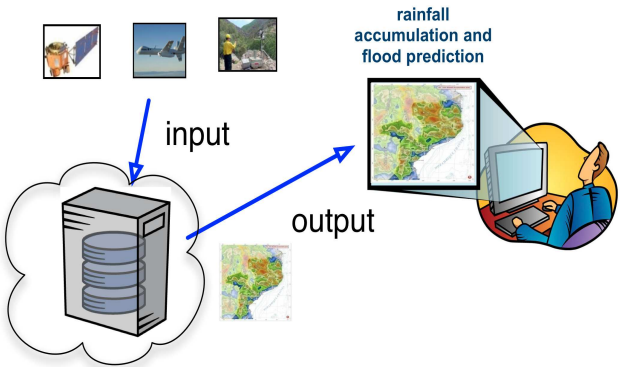


Classification



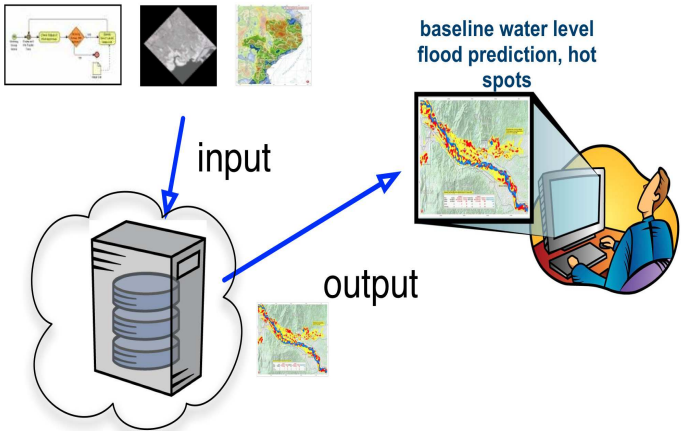
Web Processing Service (WPS) - Classification

Model Run



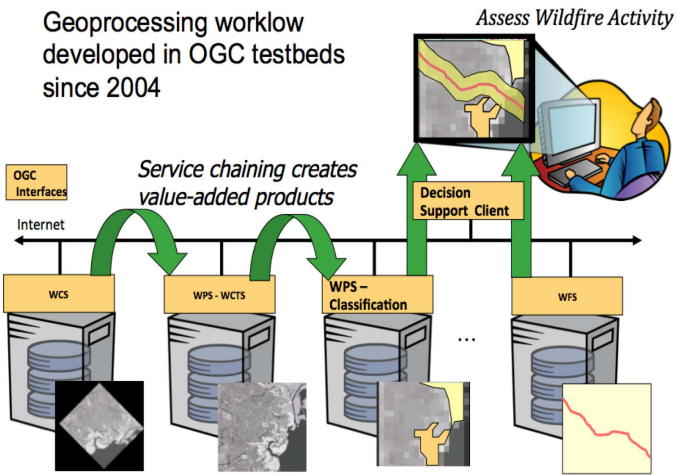
Web Processing Service (WPS) - Model Running

Workflows

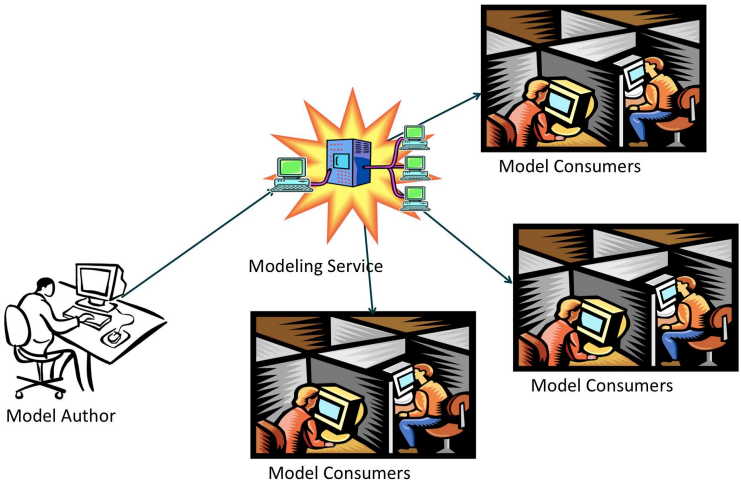


Web Processing Service (WPS) - Workflows

Service Chaining



Reusability



Operations

GetCapabilities

returns information about the service

DescribeProcess

returns details of a specific process including its inputs and outputs

Execute

returns, for a specific process, the output(s) of a process

Parameters DescribeProcess

- service
- request
- describeProcess
- identifier

Input

- URL
- GML
- Image
- Data

Output

- URL
- GML
- Image
- Another service

Open Layers Demo

http://openlayers.org/dev/examples/wps.html

- inspect process types
- run a buffer operation

52North Demo

http://geoprocessing.demo.52north.org:8080/wps/test.html

- run the default buffer operation
- what can I input to the process?

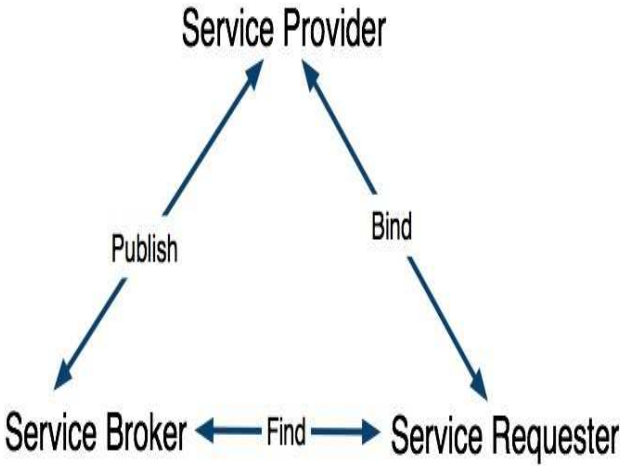
52North Demo

Use another feature collection as input.

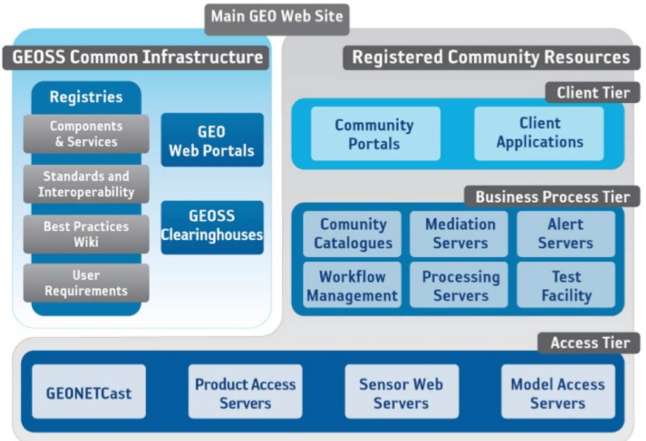
For example this getFeature request

Geospatial Architectures

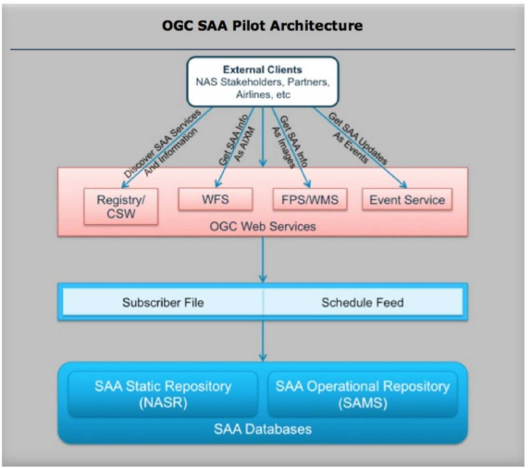
Service Oriented Architectures



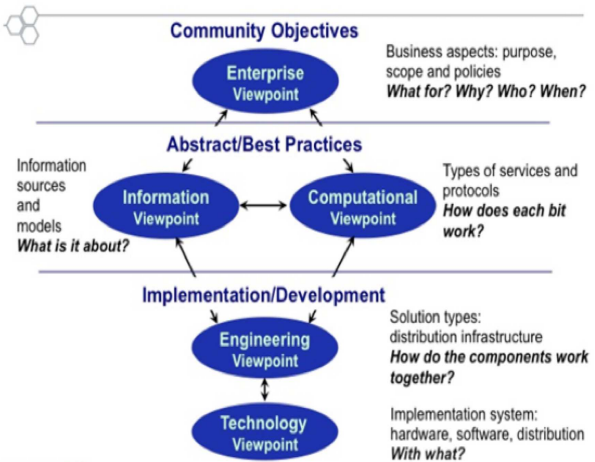
Complex Information Systems



Aviation SAA Pilot



Designing - RM ODP



Enterprise Viewpoint

- stake holders viewpoint
- why the effort in being undertaken
- scope
- objectives
- **defined by: Use Cases**

Information Viewpoint

- **Conceptual models for**
    - Data
    - Metadata
- **Defines Encoding, For example:**
    - WaterML
    - GML
    - SensorML

Computational Viewpoint

Interfaces among service consumers and providers. For example:

- WFS
- WMS
- SOS

Engineering Viewpoint

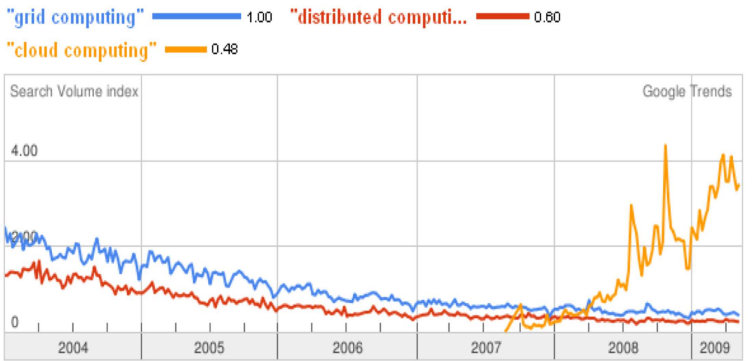
- Describes Architecture. Which components are needed?
- Explains how services related to each other.
- Services are linked via the interfaces listed in the computational viewpoint.

Other References

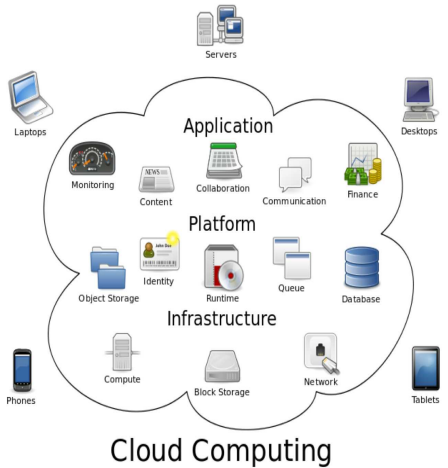
- OGC 08-062r7 - OGC Reference Model (2.1).
- OGC 02-112 - Topic 12 - The OpenGIS Service Architecture
- OGC 07-097 - Reference Model for the ORCHESTRA Architecture
- OGC 10-028r1 - GIGAS Methodology for comparative analysis of information and data management systems
- OGC 11-013r6 - OGC Engineering Report: Water Information Services Concept Development Study
- OGC 11-055 - OGC SAA Pilot Study Engineering Report

Cloud Computing

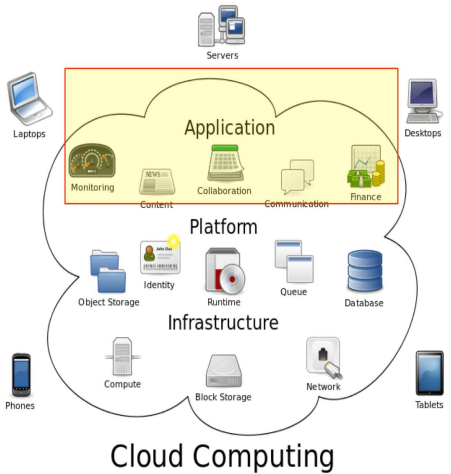
Trend


"grid computing" 1.00
"distributed computi... 0.80
"cloud computing" 0.48
Search Volume index
Google Trends
4.00
2.00
0
2004
2005
2006
2007
2008
2009

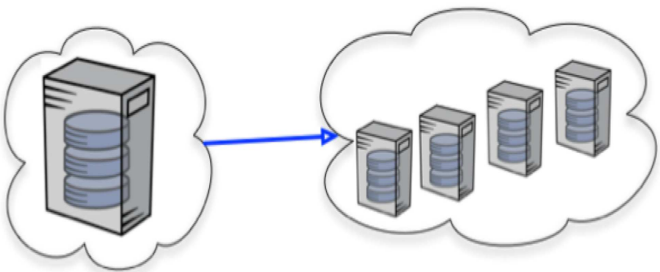
Cloud Computing



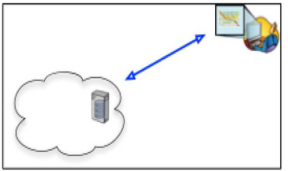
Software as a Service

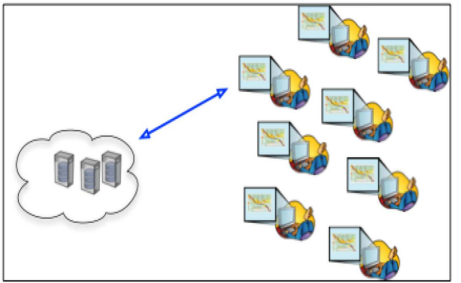


Scalability

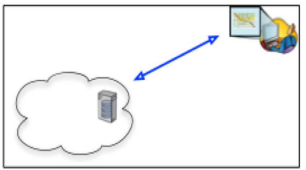


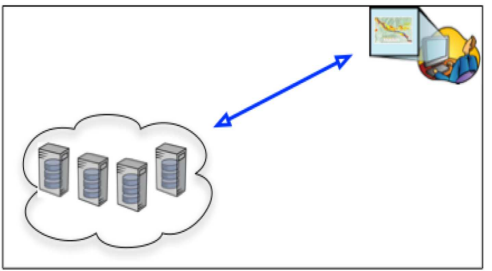
Multiple Requests



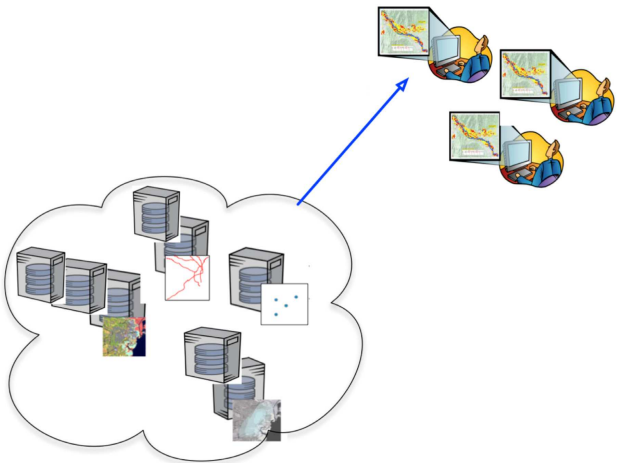


Process Intensive





OGC Services in the Cloud



Comparison Platform as Services

A COMPARISON OF PLATFORM AS A SERVICE (PAAS) CLOUDS WITH A DETAILED REFERENCE TO SECURITY AND GEOPROCESSING SERVICES

Byron Ludwig* and Serena Coetzee

Department of Computer Science, University or Pretoria, Pretoria, 0002, South Africa - byronludwig@gmail.com, scoetzee@cs.up.ac.za

Commission IV, WG IV/5

KEY WORDS: platform as a service, PaaS, geoprocessing, cloud computing, distributed computing, service level agreement, SLA, security

ABSTRACT:

Cloud computing is an emerging computing paradigm aimed at running services over the internet to provide scalability and flexibility. The advantages in using the cloud for start-up and small businesses that lack infrastructure have been shown to far outweigh the disadvantages. Cloud platform services, also known as Platform as a Service (PaaS), provide a computing platform or solution stack on which software can be developed for later deployment in a cloud. However, there are a number of security challenges because users of the cloud have to rely on third party companies to provide confidentiality, integrity and availability. Geoprocessing is the manipulation of geographic information, ranging from simple feature overlays and geocoding to raster processing and advanced climate modelling. The Open Geospatial Consortium's (OGC) Web Processing Service (WPS) defines a standardized interface that facilitates the publishing of geospatial processes. Parallelization and distribution of geoprocessing services

Monitoring Based Grid Computing

A SOA based debris flow monitoring system

Author(s): Lan-Kun Chung
Geographic Inf. Syst. Res. Center, Feng Chia Univ., Taichung, Taiwan

ABSTRACT

Taiwan is located at the collision boundary of the Philippine sea plate and the Eurasian plate. The mountain terrain is precipitous and the region, on the whole, is characterized by fragile rocks and frequent seismic activity. In addition, the concentrated torrential rainfall brought by typhoons cause extensive disasters, debris flow, the most serious disaster caused by torrential rainfall, lead to very heavy casualties in recent years. There are 17 fixed debris flow monitoring stations and 2 mobile stations deployed in Taiwan. However, the whole architecture was designed in late 2000 and implemented by traditional and proprietary methodologies. Hence, several interoperability issues have been unveiled in the recent years when the needs of interoperability increased. In this study, we propose a whole new and open standards based debris flow monitoring architecture following the service oriented architecture (SOA) paradigm. Relevant open geospatial consortium (OGC) standards (for example Web processing service, WPS specification and sensor web enablement, SWE technologies) and advancements from grid computing where lead into the proposed architecture. The use of open standards and distributed computing technologies in the proposed architecture enables heterogeneous resources (data, processing and computing power) interoperability. This study also implements an OGC WPS grid processing profile that was developed in the OGC Web services, Phase 6 (OWS-6) initiative of the OGC interoperability program.

Geoprocessing in Hybrid Clouds

Geoprocessing in Hybrid Clouds

Theodor Foerster, Bastian Baranski, Bastian Schäffer & Kristof Lange

Institute for Geoinformatics, University of Münster, Germany

{theodor.foerster; bastian.baranski;schaeffer; kristof.lange}@uni-muenster.de

Abstract. Meeting specific Quality of Service parameters in distributed architectures is one of the key requirements to build an operational infrastructure. This applies especially to SDIs, which offer geoprocessing functionality. This paper describes Hybrid Clouds as a means to meet these requirements in a efficient way by scaling the processing base load on internal (Private Cloud) and peak loads to external (Public Cloud) Cloud Computing infrastructure. The paper describes an architecture for Hybrid Clouds and a scenario performing image processing at a data center by the means of a Hybrid Cloud.

WMS in Google App Engine

GIS in the cloud: implementing a web map service on Google App Engine

Author: J. D. Blower University of Reading, United Kingdom

Many producers of geographic information are now disseminating their data using open web service protocols, notably those published by the Open Geospatial Consortium. There are many challenges inherent in running robust and reliable services at reasonable cost. Cloud computing provides a new kind of scalable infrastructure that could address many of these challenges. In this study we implement a Web Map Service for raster imagery within the Google App Engine environment. We discuss the challenges of developing GIS applications within this framework and the performance characteristics of the implementation. Results show that the application scales well to multiple simultaneous users and performance will be adequate for many applications, although concerns remain over issues such as latency spikes. We discuss the feasibility of implementing services within the free usage quotas of Google App Engine and the possibility of extending the approaches in this paper to other GIS applications.