



Facilitate Open Science Training for European Research





Geo-information infrastructures
for aggregation, visualization and analysis
of heterogeneous geo-spatial Open Data

Vassilios Vescoukis
v.vescoukis@cs.ntua.gr
NTUA, OKF Greece



INTRODUCTION

About this lecture...

- What it is...
 - An overview of **open technologies** involved in the development of "spatial-data-friendly" information systems
 - An presentation of software engineering / information systems concepts from the viewpoint of geo-information applications
 - A short discussion/demo of geo-spatial applications
- What it is not...
 - A software engineering / information systems class
- Needless to say...
 - Many other technologies can be used in the same context and vice versa



Introduction, recent works

- V.Vescoukis, C.Bratsas, *Open Data in Natural Hazards Management*, topic report N.2014/01, EPSI platform EU
- V.Vescoukis, *Architectures for distributed mission-critical geospatial applications: challenges and opportunities*, talk in Geomatik seminars (2014), ETH Zurich
- V.Vescoukis, *Distributed Web-GIS, Linked Open Data*, Lectures in MSc Geomatics (2014), Dept. of Civil, Environmental and Geomatics Engineering, ETH Zurich

Introduction, recent works

- V.Vescoukis, **Geo-information applications development**, Course in MSc in Computing Science (2015), University of Groningen, the Netherlands
- V.Vescoukis et al., **Geo-information infrastructures for inter-disciplinary risk analysis research**, European Security and Reliability Conference (2015), September 2015, ETH Zurich
- V.Vescoukis, **Integration of inter-disciplinary approaches in hazard management: a Geo-Information Engineering perspective**, ETH Risk Center Fall 2015 Seminars, Oct 13, 2015, ETH Zurich

THE BIG PICTURE

What is this about?

- Information systems

- An application idea makes sense if there is a **need** and the intended functionality can be **implemented**
- Data will be input to the system by its users, depending on the application specifics

- Geo-Information systems

- An application idea makes sense if there is a **need** and the intended functionality can be **implemented** **AND** the **spatial data needed** can be made available
- The availability of **data drives new application ideas**
- Need to access data from literally ANY source, in any format:
No data -> No application



Need to share data

- Why NOT share data?
 - Data is expensive to acquire and maintain
 - Proprietary formats mean more 'loyal' customers
 - Data is power
- Why share data?
 - Data enables new services
 - Data creates new knowledge
 - Data does not belong to those who collect it
 - Data is power

Open Data

- Questions
 - What is Open Data?
 - Who creates Open Data?
 - Who needs Open Data?
 - Who provides Open Data?
 - How open is "Open Data"?
 - Quality of Open Data?
- What's the big deal?
- Facts
 - Every human activity (almost!) produces digital data that is stored and processed
 - This data does not "belong" to specific entities: it is "just there"
 - Huge opportunities arise
- Cases
 - Social networking, sharing
 - Daily activities, transportation
 - Sensors, Internet of things
 - VGI, crowdsourcing, ...

Open Data

- Why is Open Data important
 - Decision making
 - Public awareness
 - Transparency
 - New government ethics
- Who benefits?
 - Everybody:
individuals and societies alike
- Who supports Open Data
 - Citizens, of course
 - Governments (some!), NGOs
 - Public and even some private entities of any size
- Who is threatened?
 - Those who want to keep the power of knowledge for their own

Geo-spatial apps and Open Data

- Geo-spatial applications
 - Technologies, standards:
SF, GML, WMS, WFS, WPS and many others
 - Data:
Images, rasters, vectors in any conceivable format
- Data, however
 - Are critical (not as in a "personal address manager" app)
 - Are expensive, hard to maintain up-to-date, non-standard, ...
- Geo-spatial apps heavily rely on geo-spatial data, which makes "aggregation, visualization and analysis" a far from trivial task

What is special about spatial?

- Geo-information systems are about managing geographical information (data)
- What distinguishes **geographical data** from any other kind of data processed today?
- Are there any **unique attributes** of software applications that process geographical data?



What is special about spatial?

14

- Data vs. spatial data
 - Spatial data is data used to describe spatial entities: roads, blocks, buildings, etc.
 - Spatial data is also data about the location of events (purchases, transactions, any activity)
 - Spatial data may also involve time (tracks, snapshots)
- Spatial data...
 - Is BIG data (and keeps getting bigger)
 - Can be complex (coordinate systems and more)
 - Is computationally intensive to process and organize



GIS and spatial data

15

- GIS is (used to be) about managing spatial data for applications such as
 - Mapping
 - Geo-statistics
 - Environmental, planning, etc.
- Early GIS software was **desktop-based** and offered a wide range of geo-spatial analysis tools for targeted application domains
- **Traditional desktop-based GIS applications cannot deal with the quantity, complexity and distributed nature of spatial data produced today**



Distributed Web-GIS

- Spatial data
 - Base maps, surface models, imagery
 - Thematic layers, roads, cities, Pals
 - Events with spatial reference and timestamp: measurements, tracks, tweets, check-ins, etc.
- Services
 - Queries and analytics on data
 - Mapping, geo-statistics, computations etc.
- Presentation
 - Web-based mapping platforms (autonomous, embedded)
 - Handheld devices, infographics, wearables.

Distributed Web-GIS

- Why “distributed”?
 - Data comes from many different sources
 - Possible services are restricted only by imagination
 - Services and computations may be offered independently of data
 - Pluralism and heterogeneity of user interfaces: classic, mobile, wearables (what’s next?)
- What makes it interesting?
 - Unlimited possibilities and business cases
 - Technical challenges raised by diversity

Distributed Web-GIS - example

18

Presentation

Desktop

Mobile (Android)

Smart watch

Logic

Environmental
data analysis

Transportation
data analysis

Geo-spatial
statistics

Data

Environmental
measurements
data

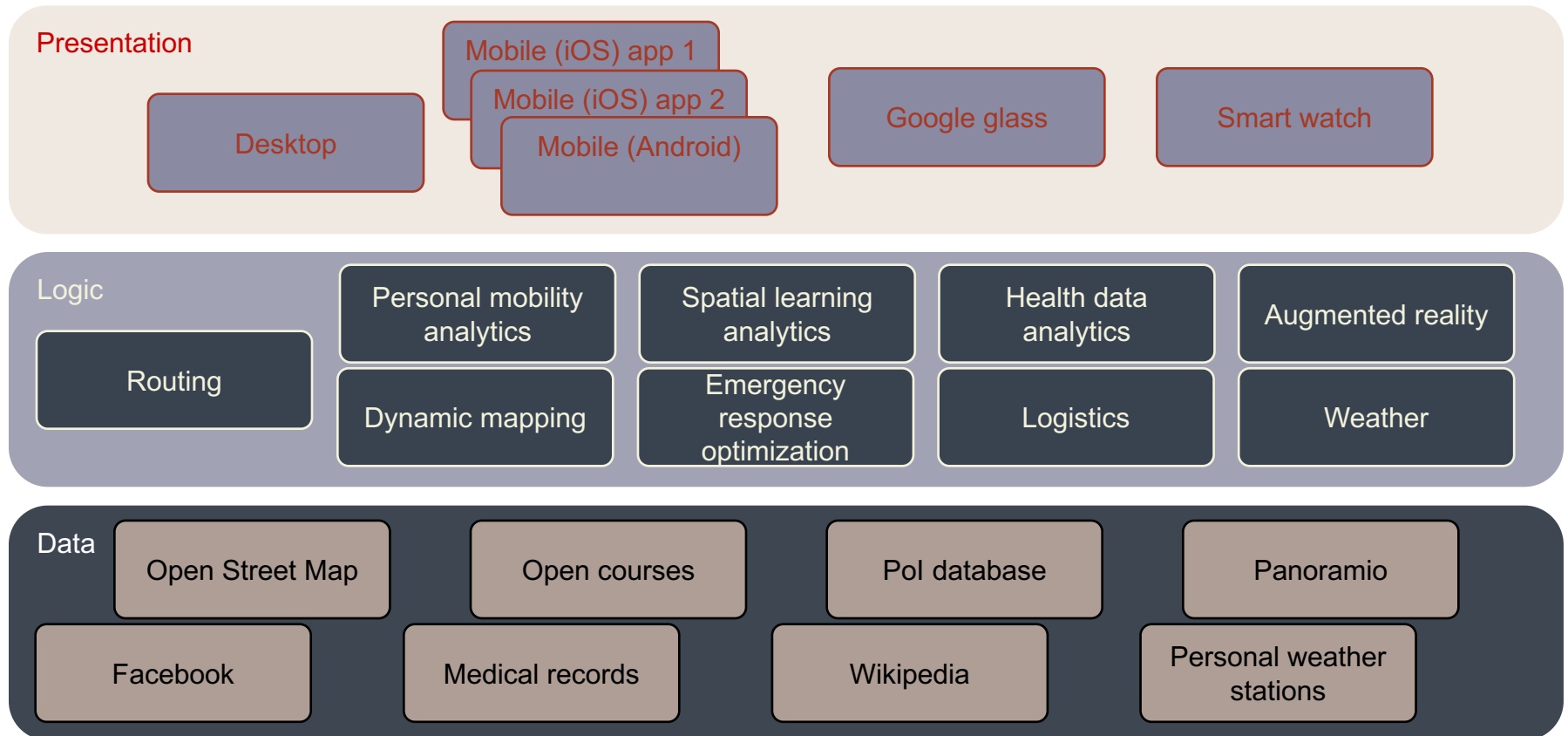
Mobility data

City and
population data

Transportation
networks data

Distributed Web-GIS - example

19

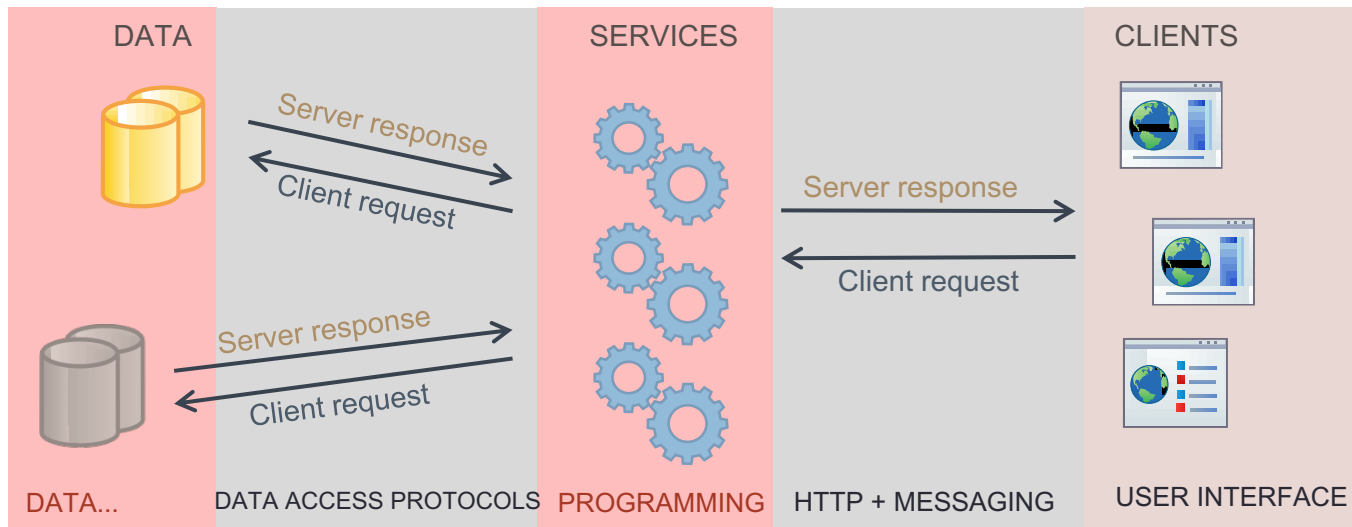


Geo-information applications, today

- It is not about "software development" anymore!
- Even multi-tier applications seem "so early 2000s"...
- **Question:** How to deal with heterogeneity of spatial data, both semantic and structural?
Answer: Accept heterogeneity, go with Open Data and make good use of Open Data technologies
- To do useful things we need **infrastructures**, not single information systems, to enable "aggregation, visualization and analysis", even mission-critical applications using Open geo-spatial Data!

FUNDAMENTALS: ARCHITECTURES AND TECHNOLOGIES

A common reference architecture



Key technologies: Data tier

- Relational databases + SQL

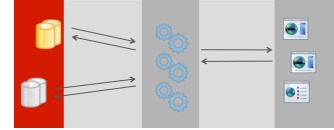
ACID: Atomicity, Consistency, Isolation, Durability

- Heavy processing requirements
- Relational schemas, data management fundamentals
- Examples: MySQL, Oracle, SQL Server, Postgres

- Non-relational databases

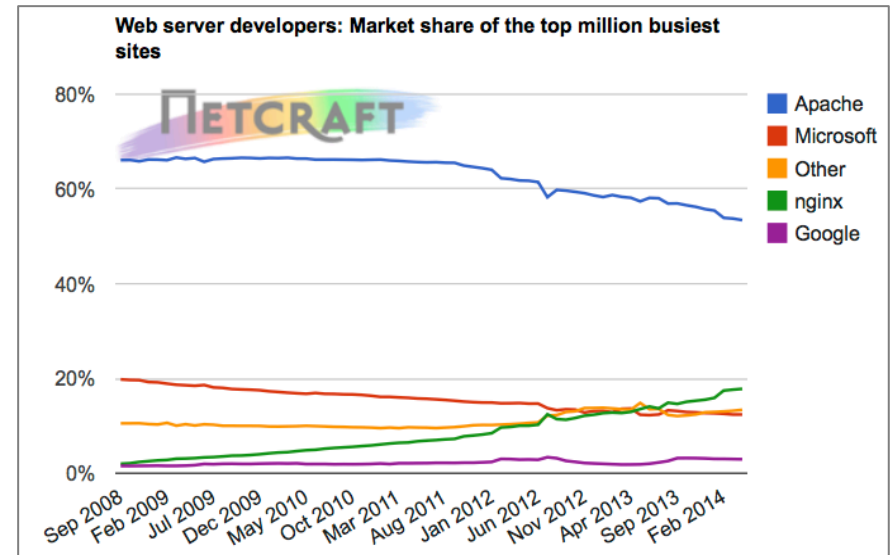
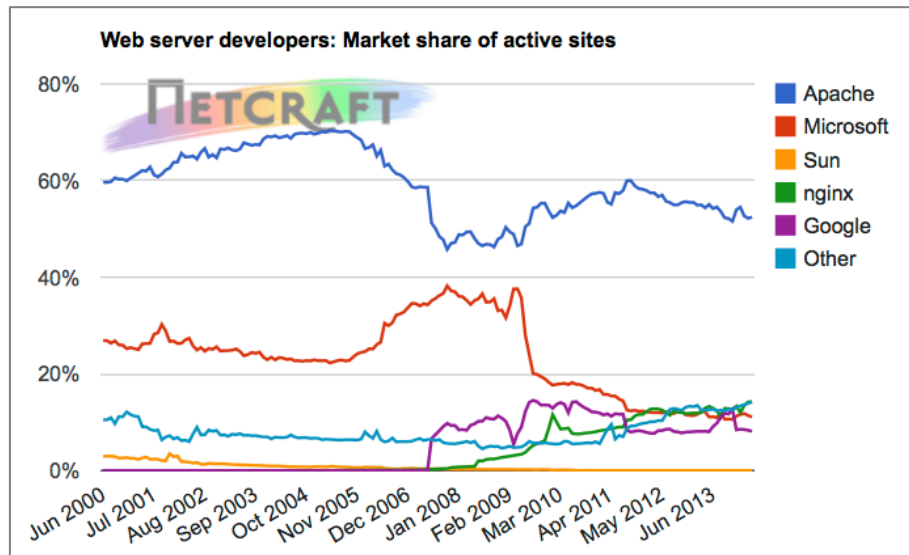
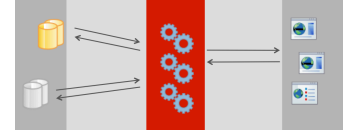
BASE: Basically Available, Soft-state, Eventually consistent

- Useful for big data manipulation
- Used by Google, Amazon, Facebook, 4-square, ...
- Examples: NoSQL, MongoDB, SPARQL



Key technologies: Service tier

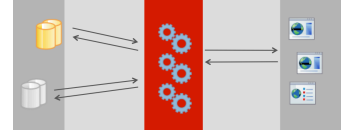
- Web servers: HTTP only
 - Apache: ~60%, Open source, multi-platform
 - IIS: ~15%, Proprietary, Windows-only



Source: netcraft.com

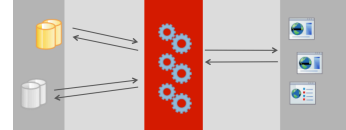
Key technologies: Service tier

- Application servers
 - Support more protocols than HTTP to implement also business logic into the server
 - Lately they lose ground from web service architectures
 - Examples: **GeoServer**, Tomcat, IBM Websphere, etc.
- Programming
 - Server-side scripting (PHP, Python, ...)
 - Service-specific descriptions (example: SLD for WMS)
 - Java Enterprise Edition, Java Server Pages
 - Microsoft ASP.NET, C#, Visual Studio tools



Server-side scripting

- How server-side scripting works
 - Request received by the client
 - Server runs locally a program (script)
 - The output of the program is HTML or any other text
 - The resulting text is sent to the client using HTTP
 - The client does not know if the received data comes from a script or is static text (stored as text file on the server)
- Scripts have local privileges on the server as needed for connecting to databases or other services



Server-side scripting

page1.html

```
<html>
<header><>This is
title</title></header>
<body> Hello world
</body> </html>
```



page1.html
page2.html
...

1 <http://www.server.ch/page1.html>

2 contents of page1.html

```
<html>
<header><title>This is title</title></header>
<body>
Hello world
</body>
</html>
```



page1.php

```
<?php
$html_content= "<html>
<header><title>This is title</
title></header>
<body> Hello world
</body> </html>";
echo $html_content;
?>
```



page1.php
page2.php
...

1 <http://www.server.ch/page1.php>

2 result of the execution of page1.php

```
<html>
<header><title>This is title</title></header>
<body>
Hello world
</body>
</html>
```



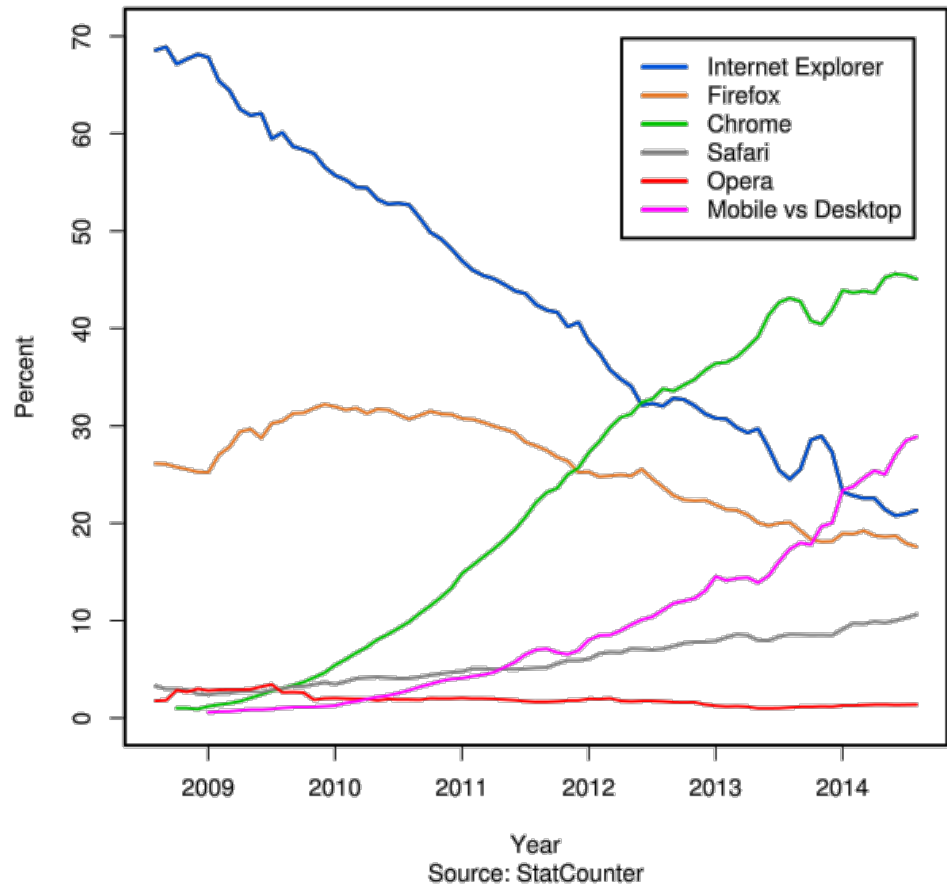
run local
service(s)

2 3
run
page1.php

Key technologies: Presentation tier

- Layout engines
 - Webkit, Mozilla Gecko
- Browser languages
 - HTML/CSS
 - Javascript, AJAX
 - Flash, Silverlight
 - ...
- Client-side scripting

Usage share of web browsers

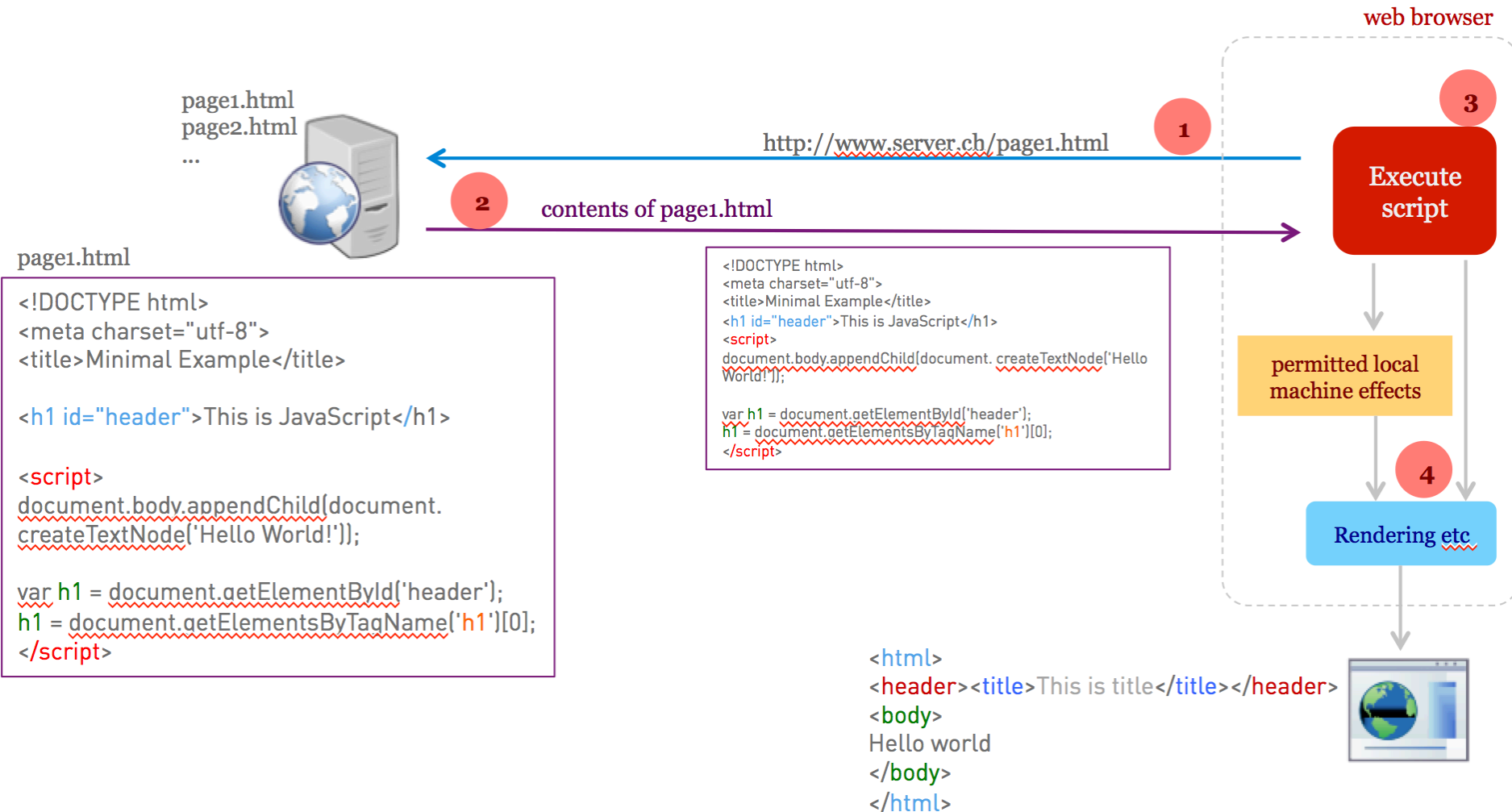


Client-side scripting

- How client-side scripting works
 - Response received by the server contains a recognizable script
 - The browser decides where to execute it (browser or operating system of the client machine)
 - The execution of the script may modify the client machine!
 - The output of the script is rendered as any other HTML content
- Security warning: scripts actually run on the client machine (yours!)

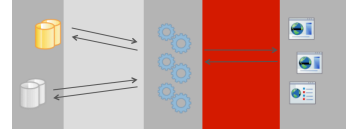


Client-side scripting (simplified)



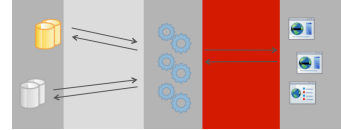
Key technologies: XML

- XML is for Extensible Markup Language
- From HTML to XML
 - HTML is used to mark up text to be displayed to users
 - XML is used to mark up data to be processed by computers
 - HTML describes both structure and appearance
 - XML describes only content, or “meaning”
 - HTML uses a fixed, unchangeable set of tags
 - In XML you make up your own tags
- Both XML and HTML come from SGML
(Standard Generalized Markup Language)



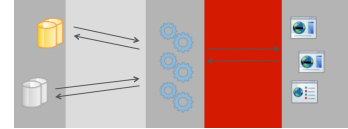
XML

- XML- Related
 - **DTD** (Document Type Definition) and **XML Schemas** are used to define legal XML tags and their attributes for particular purposes
 - **CSS** (Cascading Style Sheets) describe how to display HTML or XML in a browser
 - **XSLT** (eXtensible Stylesheet Language Transformations) and XPath are used to translate from one form of XML to another
 - **DOM** (Document Object Model), SAX (Simple API for XML), and JAXP (Java API for XML Processing) are all APIs for XML parsing



Key technologies: JSON

33



- JSON stands for JavaScript Object Notation
 - Very simple to write and parse using Javascript (it follows a subset of Javascript's syntax, anyway)
 - Efficient and simple structured data communication
 - Data types: Numbers, Strings, Booleans, Arrays, Objects, Nulls
- GeoJSON
 - Simple JSON format for geographical features
 - Objects supported: geometry (position, point, multipoint, etc), features, collections, coordinates

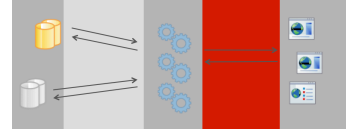
JSON

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [102.0, 0.6]
      },
      "properties": {
        "prop0": "value0"
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0,
1.0]
        ]
      },
    },
  ]
}
```

```
  "properties": {
    "prop1": 0.0,
    "prop0": "value0"
  },
  {
    "type": "Feature",
    "geometry": {
      "type": "Polygon",
      "coordinates": [
        [
          [100.0, 0.0], [101.0, 0.0], [101.0, 1.0], [100.0, 1.0],
          [100.0, 0.0]
        ]
      ]
    },
    "properties": {
      "prop1": {
        "this": "that"
      },
      "prop0": "value0"
    }
  }
]
```

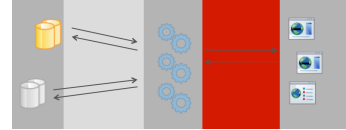
Key technologies: XML-RPC

- Concept
 - Mechanism for calling functions across a network (RPC is for Remote Procedure Call)
 - XML for messaging, HTTP for communication between computers
- XML-RPC structure
 - Data types for requests and responses
 - Basic and complex data types (arrays, structs)
 - Request: HTTP post with method name and parameters
 - Response: HTTP response with return values



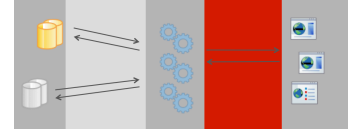
XML-RPC

- Use an XML-RPC library to make function calls
 - Any programming language
 - Apache XML-RPC supports Java
- Generic workflow
 - Develop the function(s) to be called
 - Create the server
 - Register the function(s) to the server (RPC handler(s))
 - Start the server



Key technologies: SOAP

- Simple Object Access Protocol
 - Yet another protocol specification for remote execution of methods over XML and HTTP (other protocols possible)
 - Platform- and language- independent
 - Developed by Microsoft and others (IBM, too)
- SOAP message structure
 - Envelope
 - Header (optional)
 - Body
 - No DTD or processing instructions



SPATIAL DATA: WHAT THIS IS ALL ABOUT...

Spatial data management

- **Representation of spatial data:**

All forms of spatial data must be storable in the system

- Boundaries, roads, blocks, buildings, paths, points, ...

- **Spatial queries:**

Queries with spatial reference or context

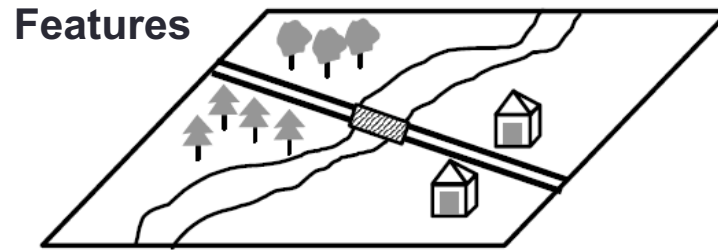
- Which properties lie next to a main road?

- **Integration with thematic data:**

Combine thematic and spatial data in an adequate form

- Which properties in corporate ownership lie next to rivers?
- What types of vegetation exist in country X?

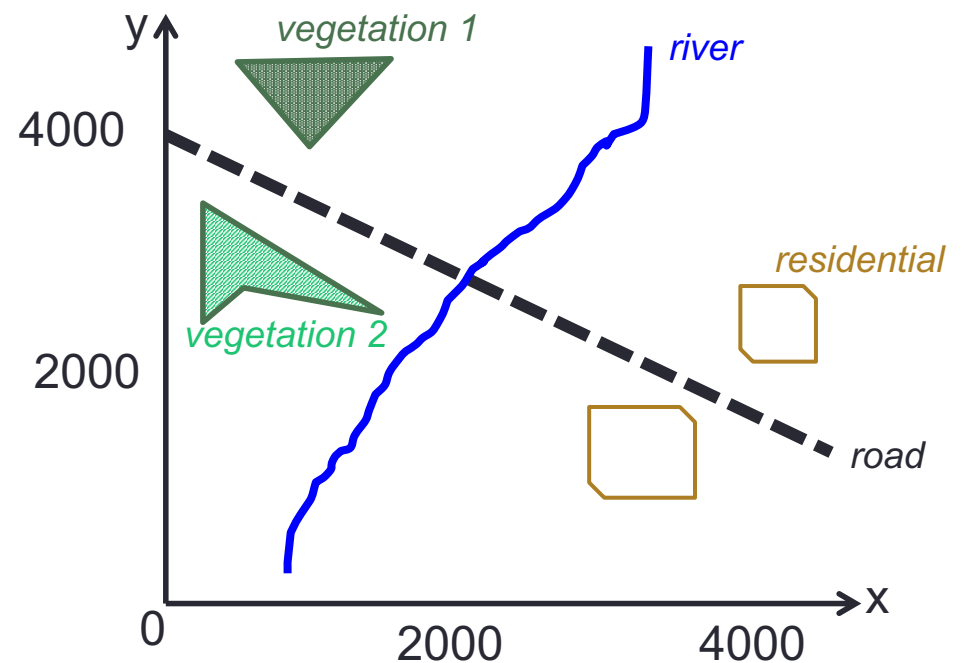
Representation of spatial data



Raster

t1		t1		rv		
R	t1			rv		
t2	R		rv			
t2	t2	R	rv			
		rv	R			
		rv		R	h	
	rv				R	
	rv				h	R
	rv					

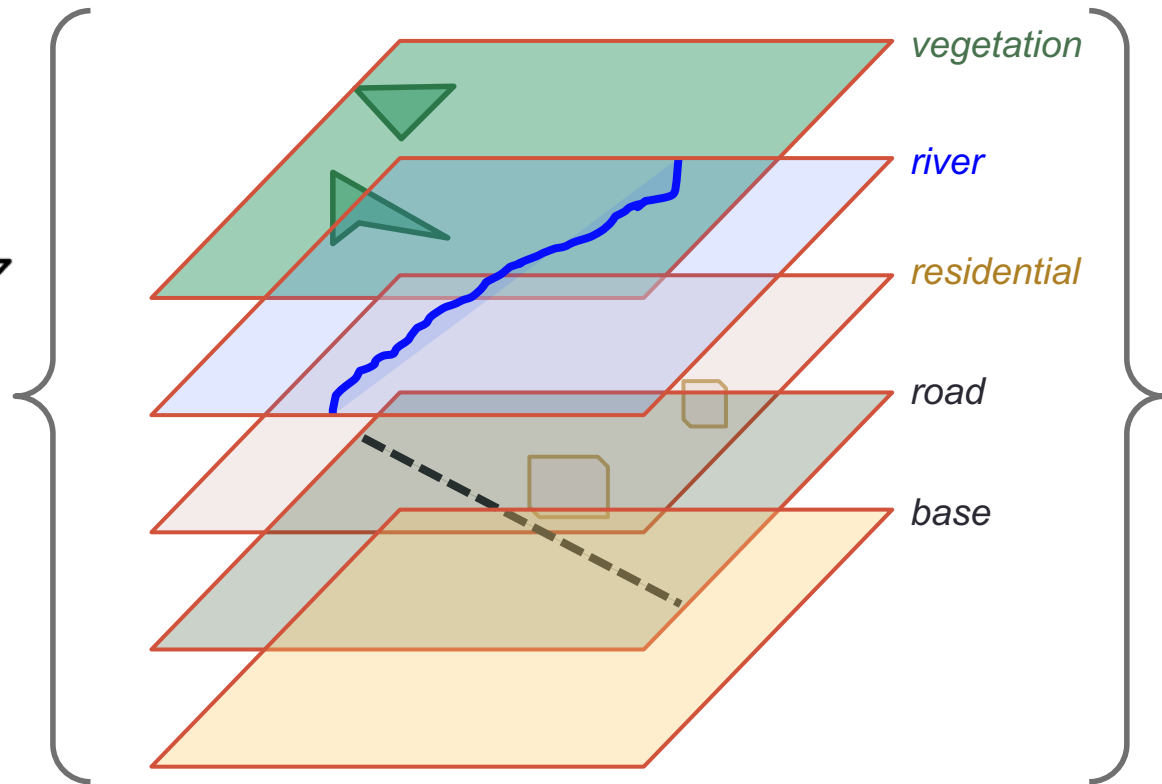
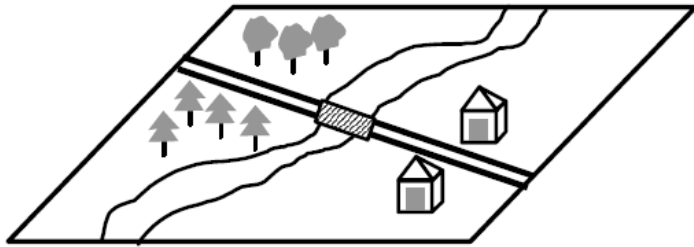
Vector



Representation of spatial data

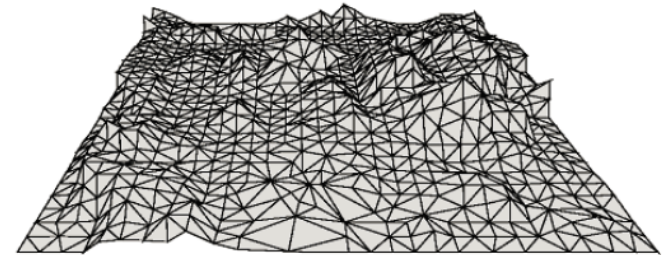
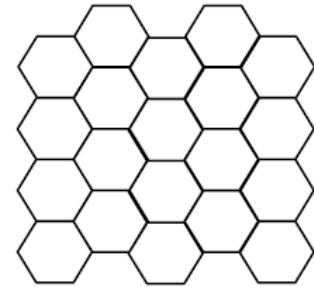
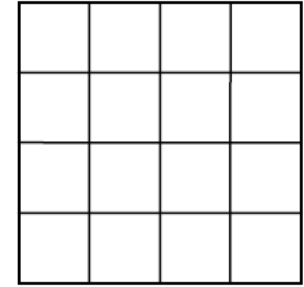
- Layers

- a layer contains spatial data under a thematic abstraction
- thematic abstractions are usually application-dependent



Rasters

- Space is represented as a mosaic
 - Non-overlapping cells
 - Each cell contains information about some attribute
 - Multiple-dimensions are possible
 - An image is a raster with optical information
- Cells' shape may be
 - Canonical (grid)
 - TIN



Spatial data management

- Thematic data can be managed on the basis of existing data models (e.g. relational model).
- Spatial data may be integrated in different ways:
 - by employing a DBMS data model
 - by extending a given DBMS data model
 - by separate management in a specialized storage system

Spatial data management

- Geographical Information Systems (GIS):
Visualize and analyze spatial data
 - Search (location, address, ...)
 - Analyze (buffer, overlay, ...)
 - Terrain attributes (slope, features, ...)
 - Measurements (distance, area, perimeter, ...)
 - Can use data from a SDBMS
- Spatial Databases (SDBMS)
 - Efficiently manage large volumes of spatial data
 - Spatial indexing and search (query) optimization
 - No mapping tools

Spatial data management

- Create **new spatial data types**, which are implemented as abstract data types (ADT) within the DBMS.
 - Usage in analogy to primitive SQL data types
 - Spatial index structures can be used within the query process
 - Spatial operators and predicates may be evaluated using special algorithms within the DBMS
- Optimize **indexing and query processing** for efficiency and speed due to the geometric nature of data

OGC

- The Open Geospatial Consortium (OGC)
 - non-commercial organization, consisting of authorities, companies and universities.
 - OGC offers standards for spatial data and services

OGC Simple Features

- The OGC Simple Feature Specification for SQL:
 - Describes a set of geometry data types for SQL based on the OGC geometry model
 - Describes a set of SQL operations on these types
- Characteristics:
 - A "feature" is an abstraction of a phenomenon of the real world ("geo-object"), stored as a dataset in a feature table
 - Modeling of the geometry of spatial objects:
 - Only 0-2 dimensional objects
 - Only linear interpolation between points
 - No explicit representation of topology

Geo-spatial data sharing practices

- Today there are a number of approaches, sometimes called design patterns, for accomplishing geospatial data exchange between dissimilar systems
 - **File based approach:** geographic data is encoded in a structured file format, for batch transfer or download
 - **Application programming interface (API) approach:** geographic data is exchanged as needed between software applications running locally (not on a network)
 - **Web services approach:** geographic data is accessed and exchanged over networks and the Internet between software components, using HTTP and other web-based protocols



Standards and interoperability

- Standards (in general)
 - Are needed to achieve interoperability
 - Specify interfaces that different vendors should use
 - Need to be agreed upon (not easy!)
- Data standards
 - Specify a conception of the spatial world (vocabularies, hierarchies, attributes, ...)
- Web service standards:
 - specify format of HTTP requests and responses: what parameters, names of parameters, type of value for parameters, type of results, security, ...

Main standards bodies

- **OGC** (Open Geospatial Consortium)
- **ISO/TC 211** (International Organization for Standardization, Technical Committee 211)
 - Covering digital geographic information and geomatics.
- **W3C** (World Wide Web Consortium)
 - Address issues of incompatibility in Web technology by different vendors

Geography Markup Language

- Geography Markup Language (**GML**) has its roots in decades-old geo-data exchange standards in the US, developed to solve the problem of packaging geospatial data in a file format independent of any GIS vendor's software
- XML-based encoding standard for geographic information
- Defines an XML schema for geographic entities
- GML objects can represent features, geometries, topologies, coordinates, observations, styles, values and more



GML and XML

- Because GML is based on XML, it leverages a wealth of standards, tools and practices for data exchange being developed by several consortia around the world
- Standard XML technologies exist...
 - for encoding and data modeling (DTD, RDF and XSD)
 - for linking and associating resources (Xlink)
 - for selecting and pointing (XPath, Xpointer)
 - for transforming content (XSLT)
 - for graphical rendering (SVG, VML, X3D)

GML vs Simple Features

- GML is an XML representation of geometrical entities ("features", collections)
 - It is about communicating data over the web
 - Also, about communicating meta-data
- Simple Features is a standard for adding geometrical data types in databases
 - Definition, Constraints
 - Operations on geometries, topological relationships, spatial operations

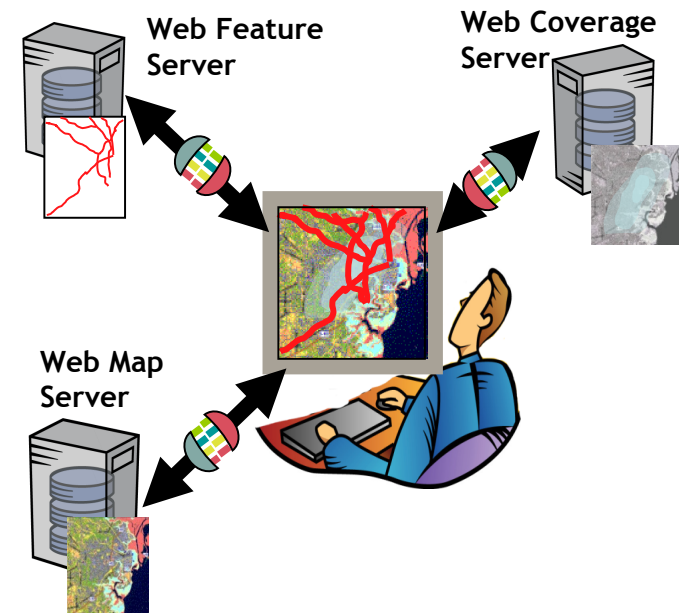
SERVICE LAYER OPEN STANDARDS

OGC Web Services

- Geospatial Web Services from OGC
 - Map Service: Map = $f(\text{semantics, map extent, scale, ...})$
 - > OGC Web Map Service (**WMS**)
 - Data Access- / Download- / Feature- / Coverage- Service:
Geospatial Data = $f(\text{filter criteria})$
 - > OGC Web Feature Service (**WFS**) for vector data (Features)
 - > OGC Web Coverage Service (WCS) for fields
 - Geocoding Service: Point = $f(\text{postal address})$
 - > OGC OpenLS
 - Catalogue Service: Metadata = $f(\text{filter criteria})$
 - > OGC Catalog Service Web (CSW)
- More at www.opengeospatial.org/standards

OGC Web Standards

- Enable the geo-spatial web
 - Web Map Service (WMS)
 - Web Map Tile Service (WMTS)
 - Web Feature Service (WFS)
 - Web Processing Service (WPS)
 - Web Coverage Service (WCS)
 - Catalogue (CSW)
 - Geography Markup Language (GML)
 - KML
 - Others...



OGC Web services

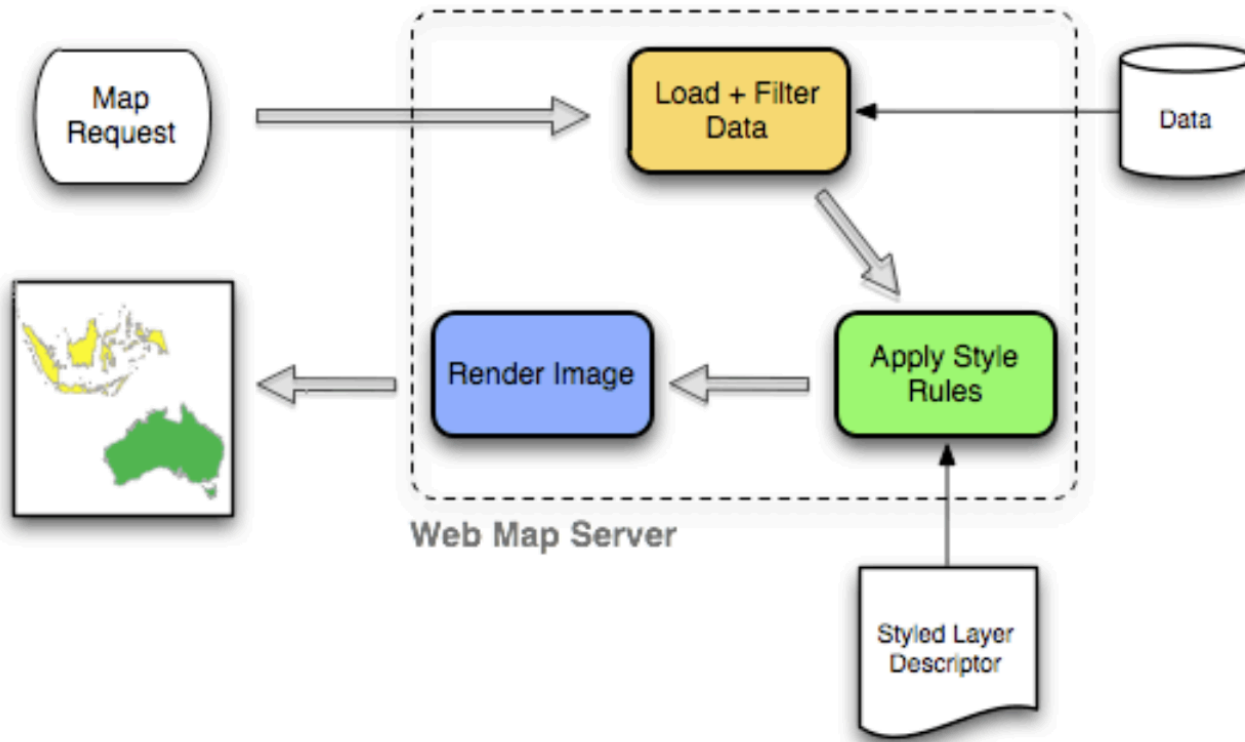
- Map services (WMS, WMTS, WCS)
 - Offer maps for use in your application
- Feature services (WFS, CSW)
 - Offer spatial data
- Processing services (WPS)
 - Provide a framework for spatial data processing over the web
- Enabling technologies for mash-ups and new value-added services

OGC Web Services

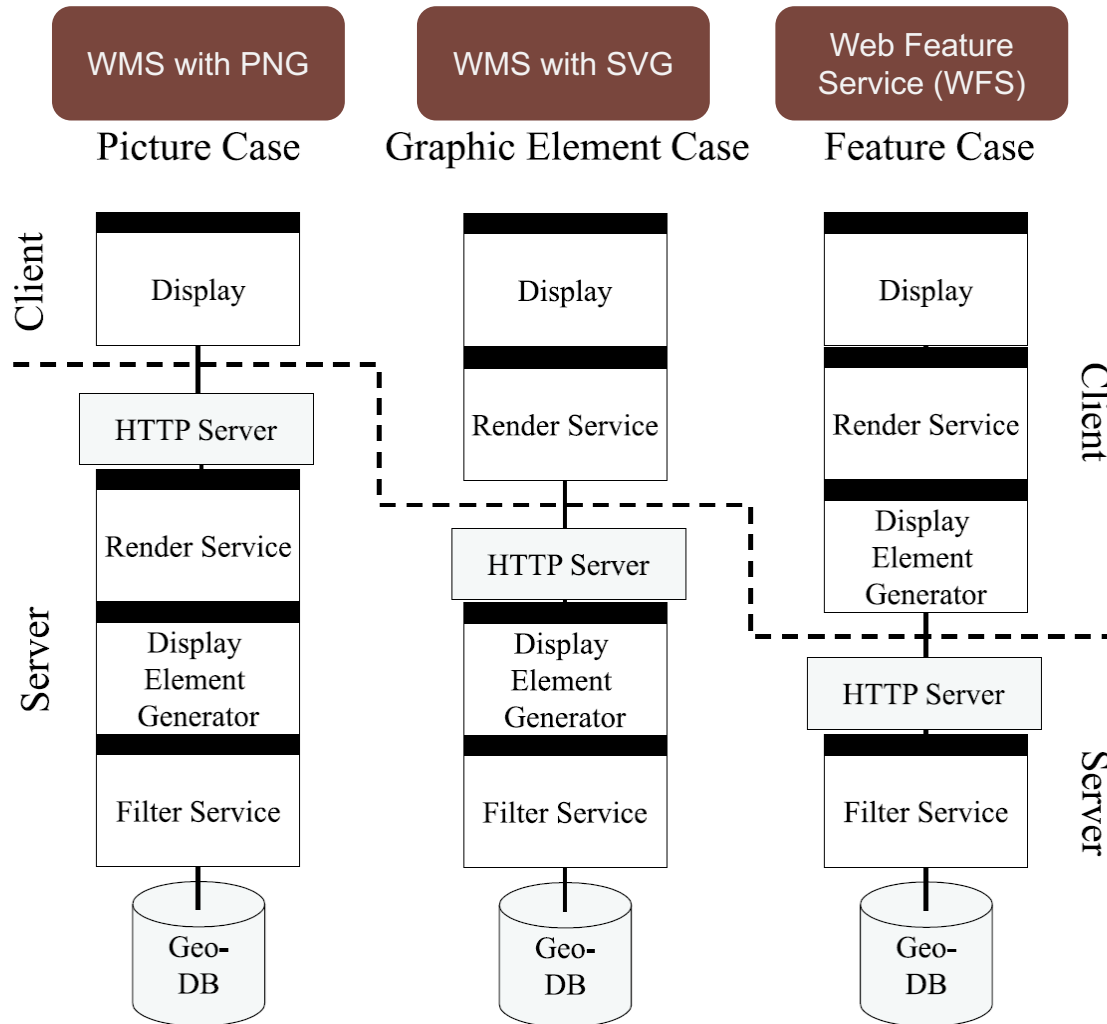
- Interface concept
 - Get the **capabilities** of the service:
returns information about what a specific implementation of the service can do
 - Get **info** on some entity/property:
returns the attributes of entities offered by the service
 - **Run** the service:
accepts parameters and returns the result of the service

OGC WMS - Web Map Service

- OGC & ISO standard for requesting & serving maps over the Internet in pictorial format (PNG, GIF, JPEG).



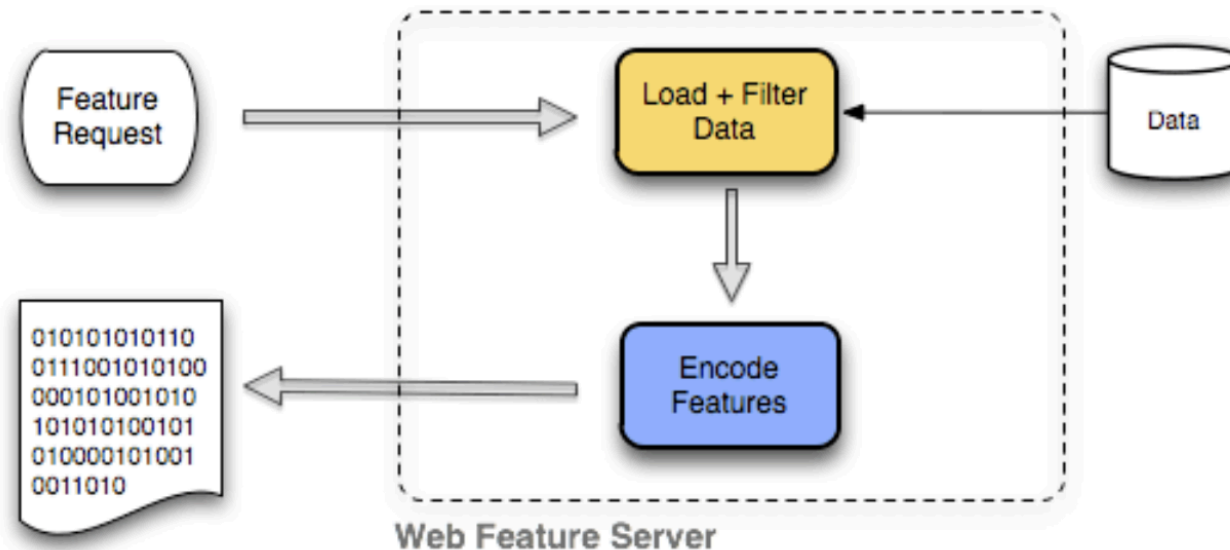
WMS System Architecture



[Image source: OGC 2000
modified by A. Donaubauber]

OGC WFS - Web Feature Service

- OGC Web service standard for reading & writing geographic features in vector format.



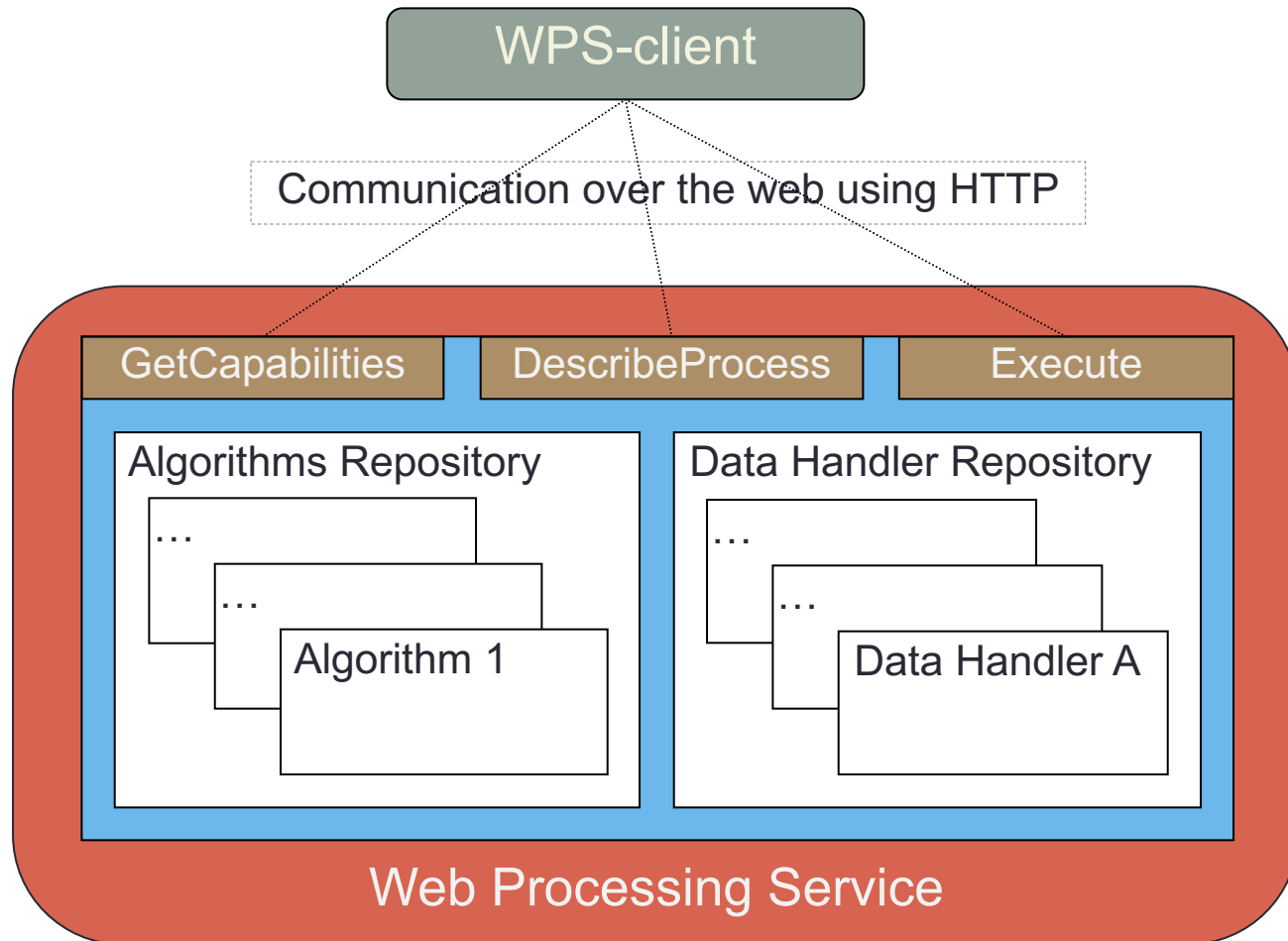
Other web services

- WPS - Web Processing Service
 - Provides rules for standardizing inputs & outputs for geospatial processing services.
 - GetCapabilities, DescribeProcess, Execute
- SWE - Sensor Web Enablement
 - Standards enable users to discover & access sensor data of a sensor Web or sensor network
 - Sensor Observation Service (SOS), Sensor Alert Service (SAS), etc.

WPS - Web Processing Service

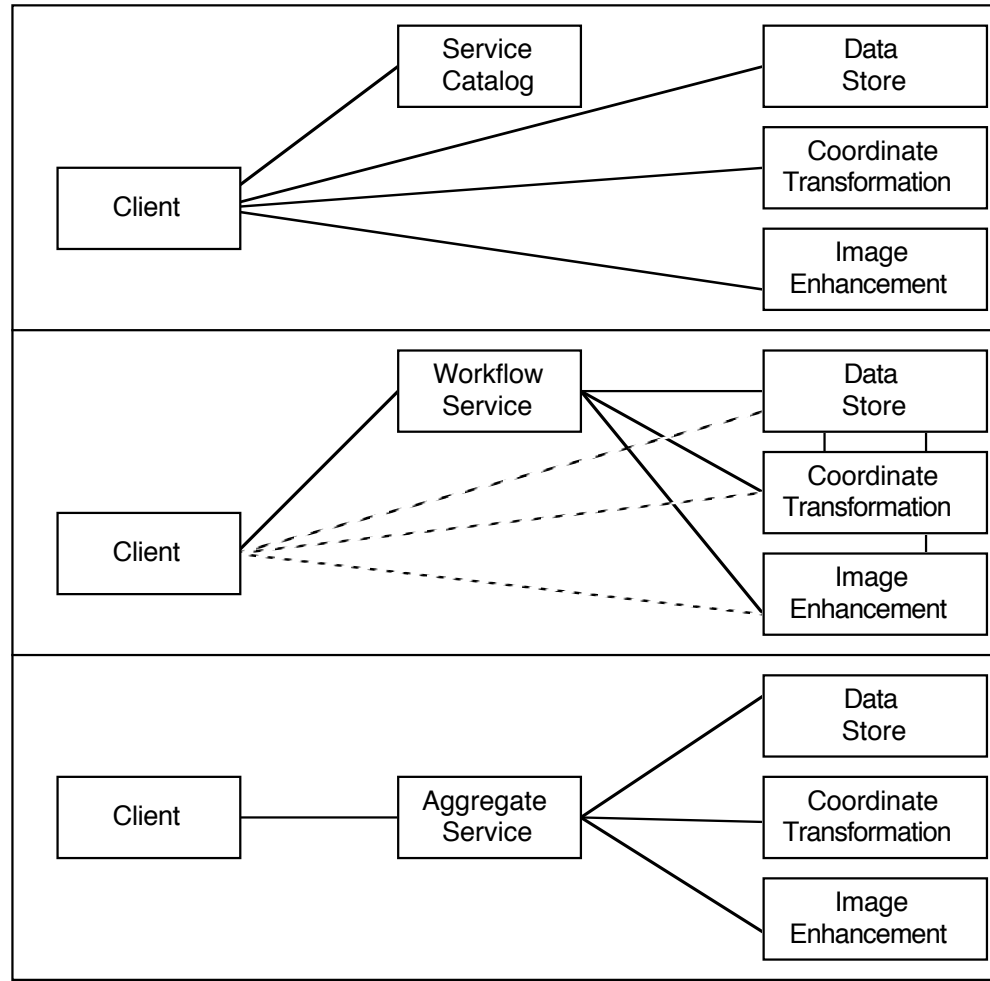
- Standardized interface
- Facilitates the publishing of geospatial processes
- Discovery of and binding to those processes by clients
- Process:
Any algorithm, calculation or model that operates on spatially referenced data and gives any data type, including spatial data, as a result

Web Processing Service



Credit: Open Geospatial Consortium

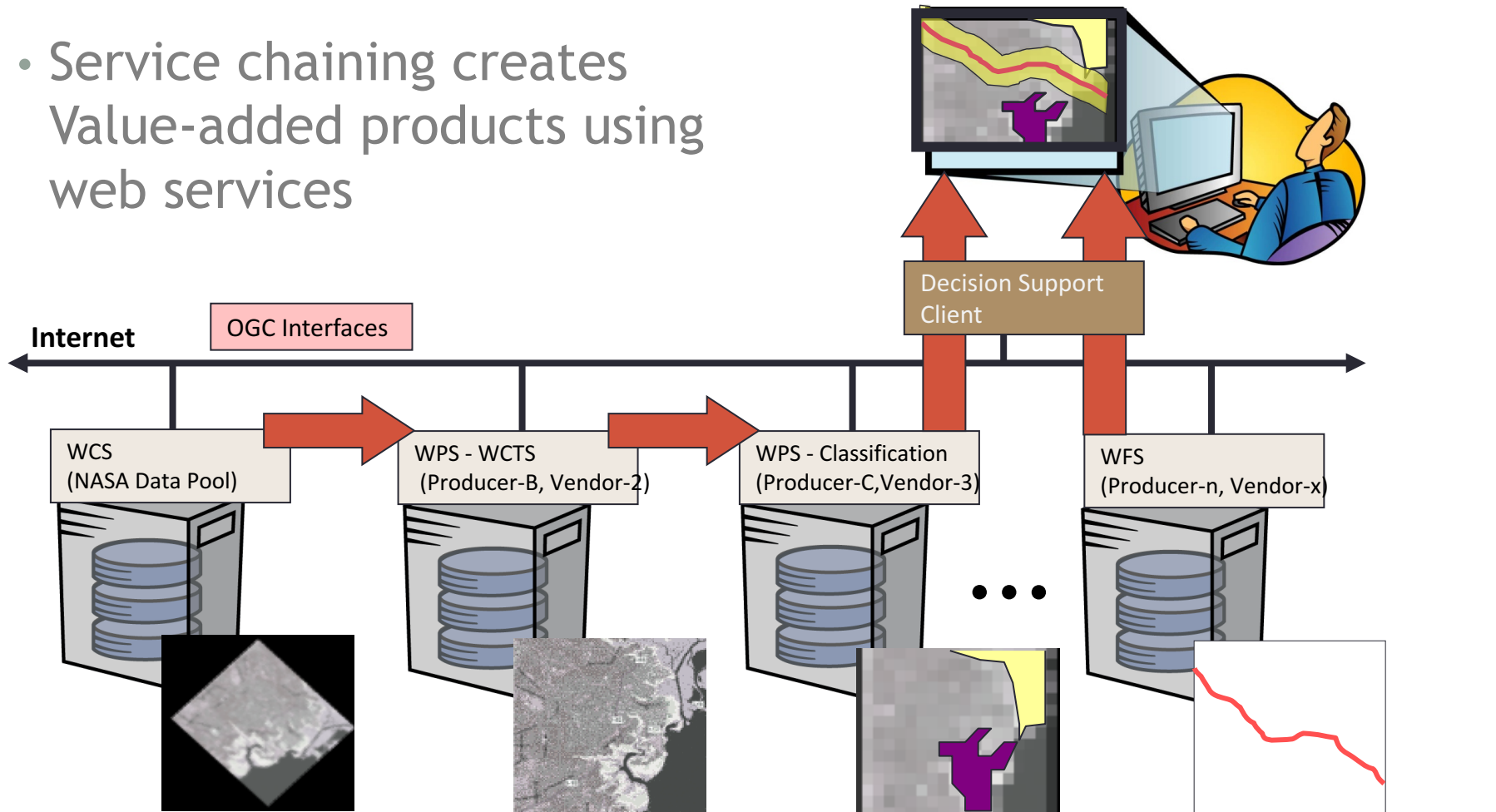
Design patterns for Service Chaining



Credit: Open Geospatial Consortium

Workflow example

- Service chaining creates Value-added products using web services



Credit: Open Geospatial Consortium

Value added and considerations

- Unlimited applications are possible
- Spatial data and service sharing
- Democracy, redefined
- However...
 - It takes some (initial only?) data therapy
 - Quality and validation considerations
 - Privacy issues
 - What about mission-critical applications?
- A shift of mentality is required



STANDARDS-BASED TOOLS FOR GEOSPATIAL APP DEVELOPMENT: GEOSERVER

What is GeoServer

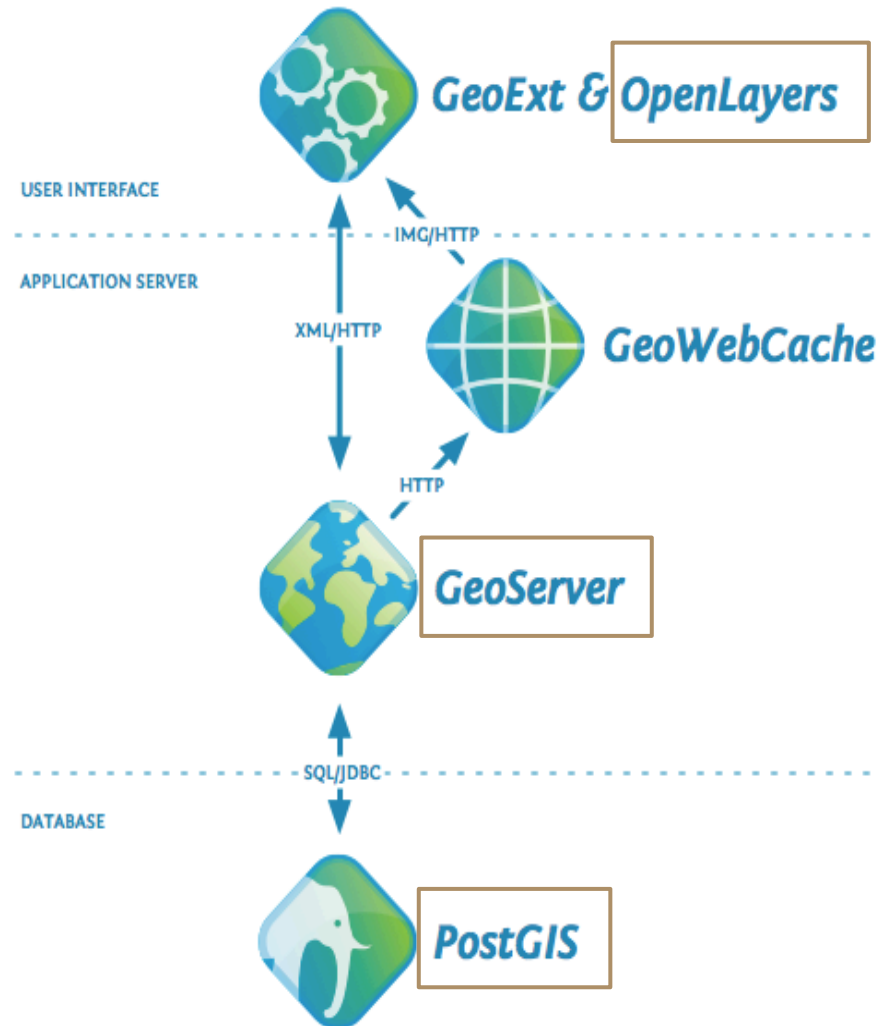
- *"GeoServer is a powerful map and feature server for sharing, analyzing, and editing geospatial data from spatial data sources using open standards"*
 - Support for many back-end data formats (ArcSDE, Oracle Spatial, DB2, MS SQL Server, Shapefile, GeoTIFF, etc.)
 - Multiple output formats (Esri Shapefiles, KML, GML, GeoJSON, PNG, JPEG, TIFF, SVG, PDF, GeoRSS)
 - Fully-featured web administration interface and REST API for easy configuration
 - Configurable role-based security subsystem Java J2EE application works with Jetty, Tomcat, JBoss, and others

Source: boundlessgeo.com

What is GeoServer



Part of OpenGeo Suite



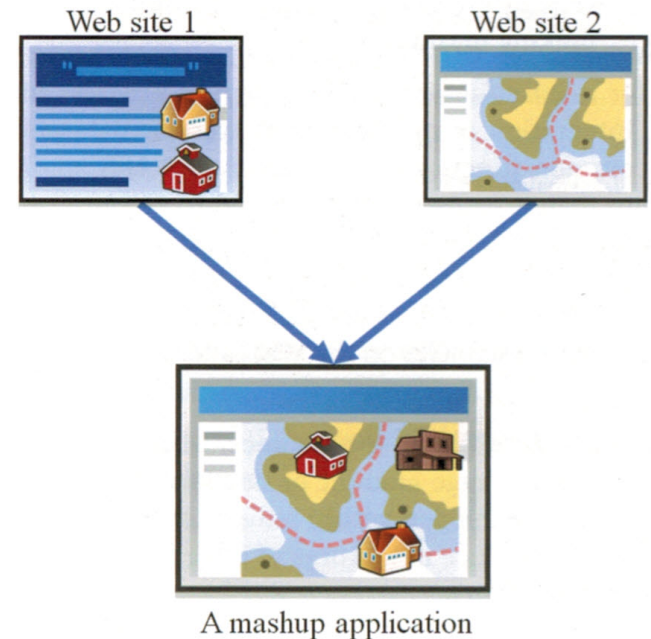
APPLICATION FAMILY: GEO-MASHUPS

What is a 'mashup'?

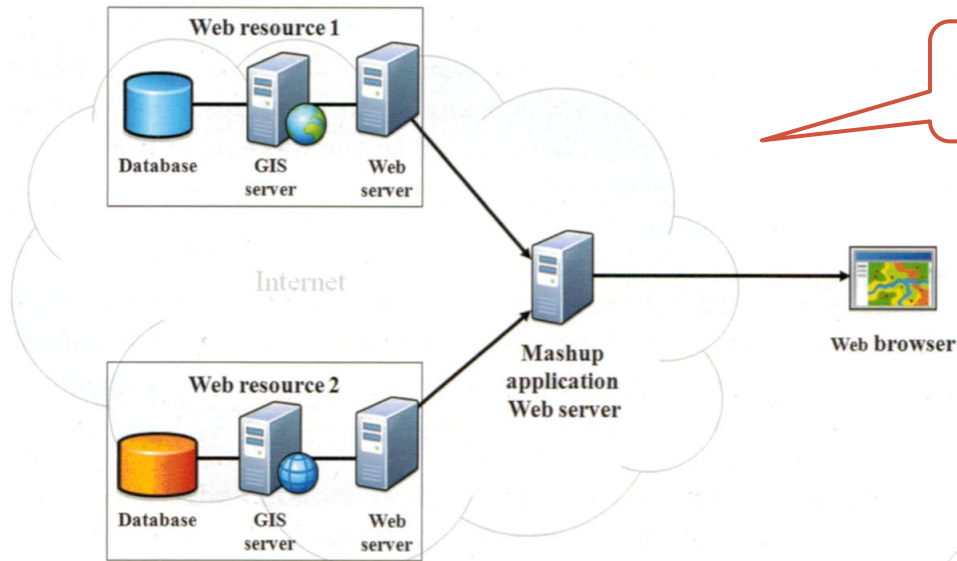
- Mashup: a Web page or application that dynamically combines contents or functions from multiple Web sites.
 - Live linkage to its sources!
- Geomashup: a mashup where at least one of the contents/functions is georeferenced.
 - Integrating multiple data sources based on common geographic location.
 - Topological (e.g., flood boundaries with city boundaries) & graphic overlays.

What can be 'remixed'?

- Maps, web services, web pages, blogs, photos, videos.
 - Housing Maps
 - www.housingmaps.com
 - Craigslist & Google Maps
 - Crime Mapping
 - www.crimemapping.com
 - Crime activity by neighborhood
 - Transportation services
 - Real-time traffic+weather information
 - Road network
 - Routing service

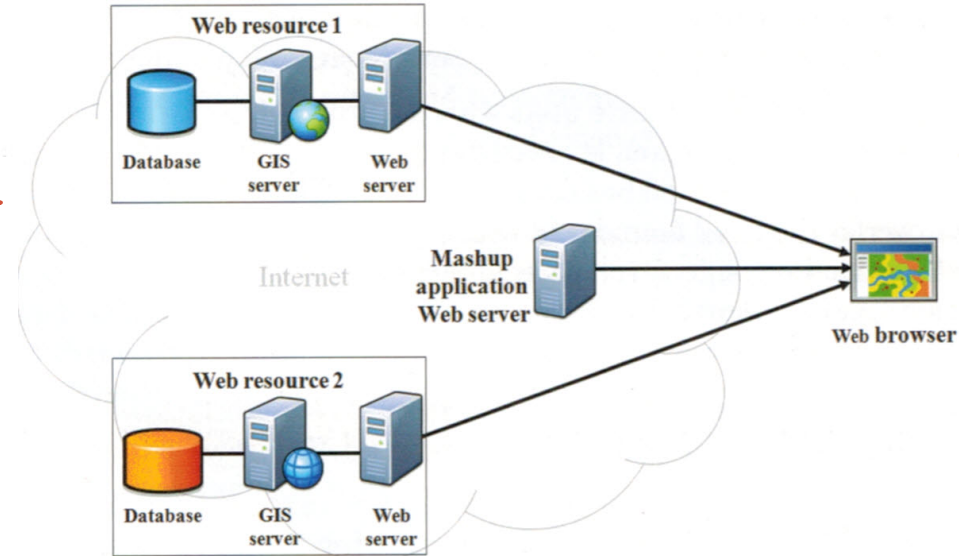


Design patterns for mashups



Mashups: browser-side architecture

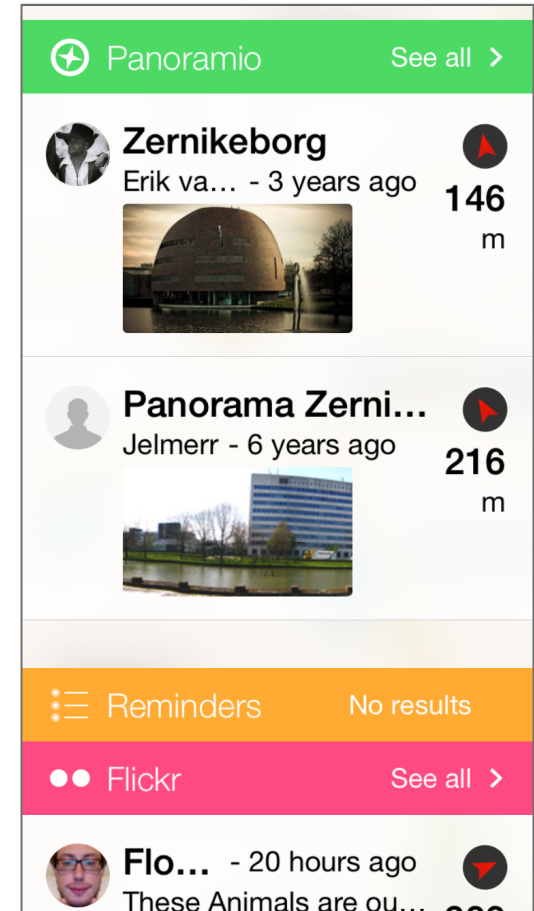
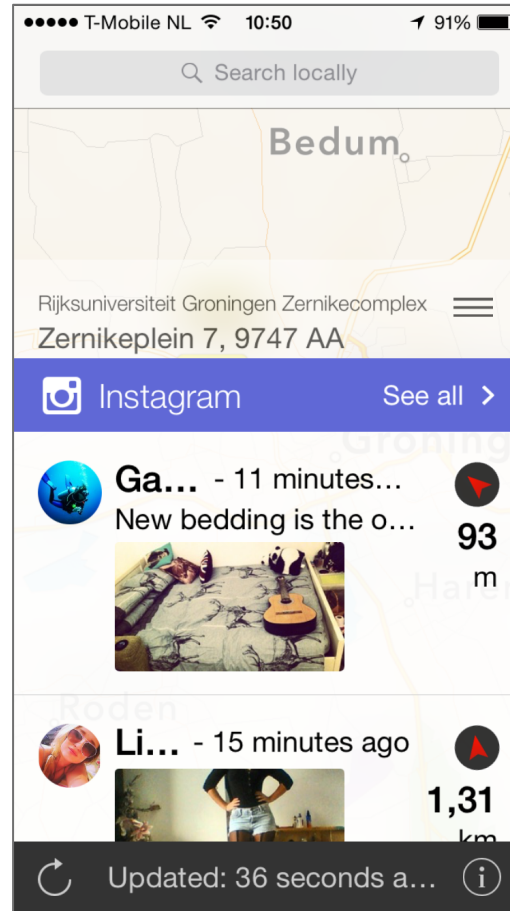
- Maps provided via JavaScript
- Users can view sources
- Google, etc. officially released their mapping capabilities via a JS API.



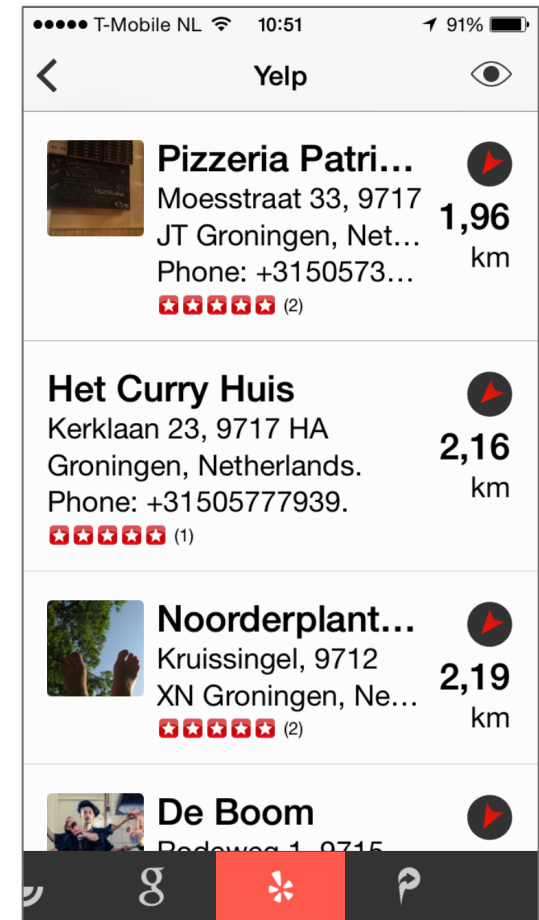
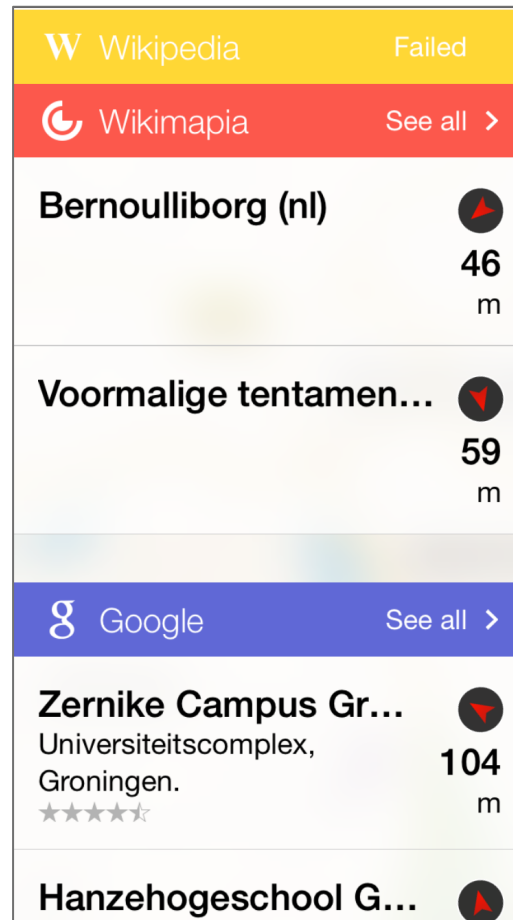
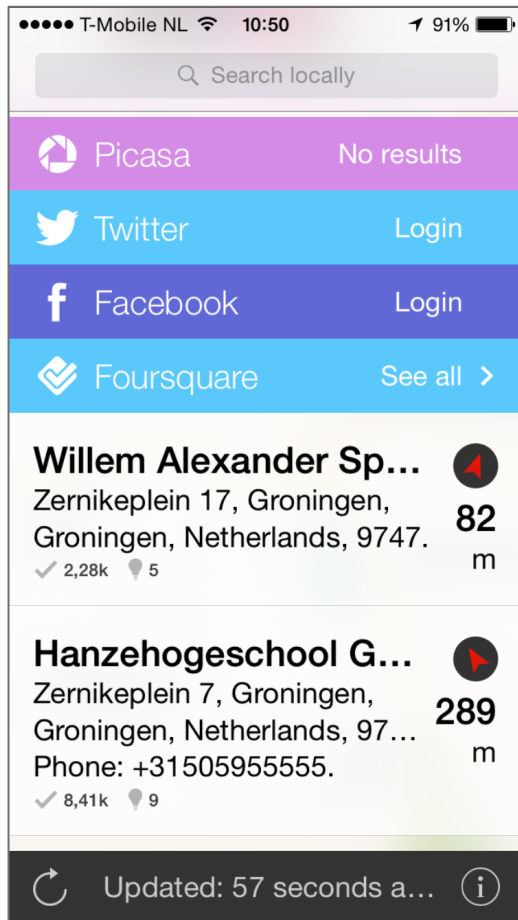
Geo-mashup application design

- We need to integrate: **WMS, WFS**
 - Basemaps, data setup (and license?) maps + data from some provider
 - Operational layers develop software to implement the user experience and respond to user actions, e.g., mouse click on a map, a form, etc.
 - Tools **WPS** develop (or interface to) software to implement business logic, analytical functions, etc.
- Mashups promote development of public participation GIS

Geo-mashup example: localscope



Geo-mashup example: localscope



SENSOR WEB ENABLEMENT: GEO-INFORMATION APPS EVERYWHERE

What is SWE?

- Sensor Web Enablement (SWE) is a set of OGC standards that enable developers to make all types of sensors, transducers and sensor data repositories discoverable, accessible and useable via the Web

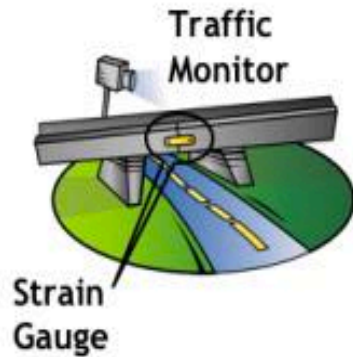
(source: opengeospatial.org)

- A sensor Web is a Web-accessible network of sensors and archived sensor data that can be discovered and accessed using standard protocols and APIs.'

(Botts et al. 2006)

- **SWE is about monitoring and controlling Objects, Phenomena and Processes through Web-enabled Sensors**

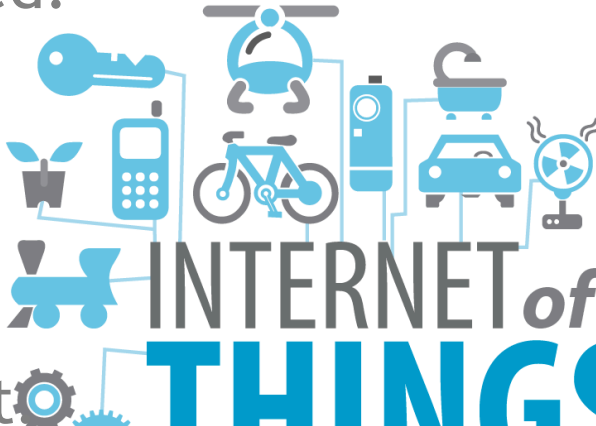
What is SWE?

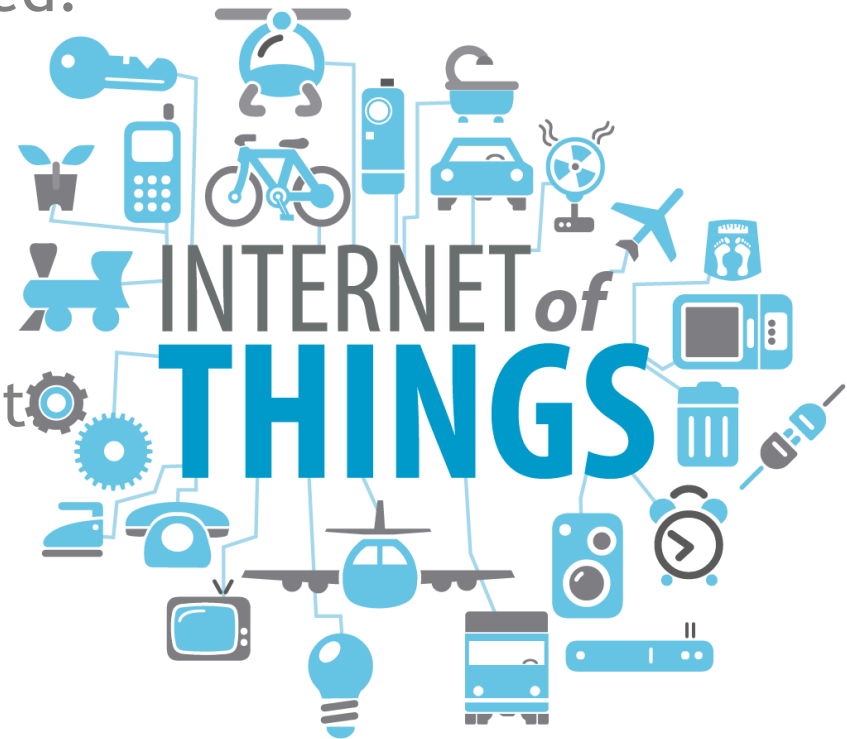


- All sensors reporting position
- All connected to the web
- All with metadata registered
- All readable remotely
- Some controllable remotely



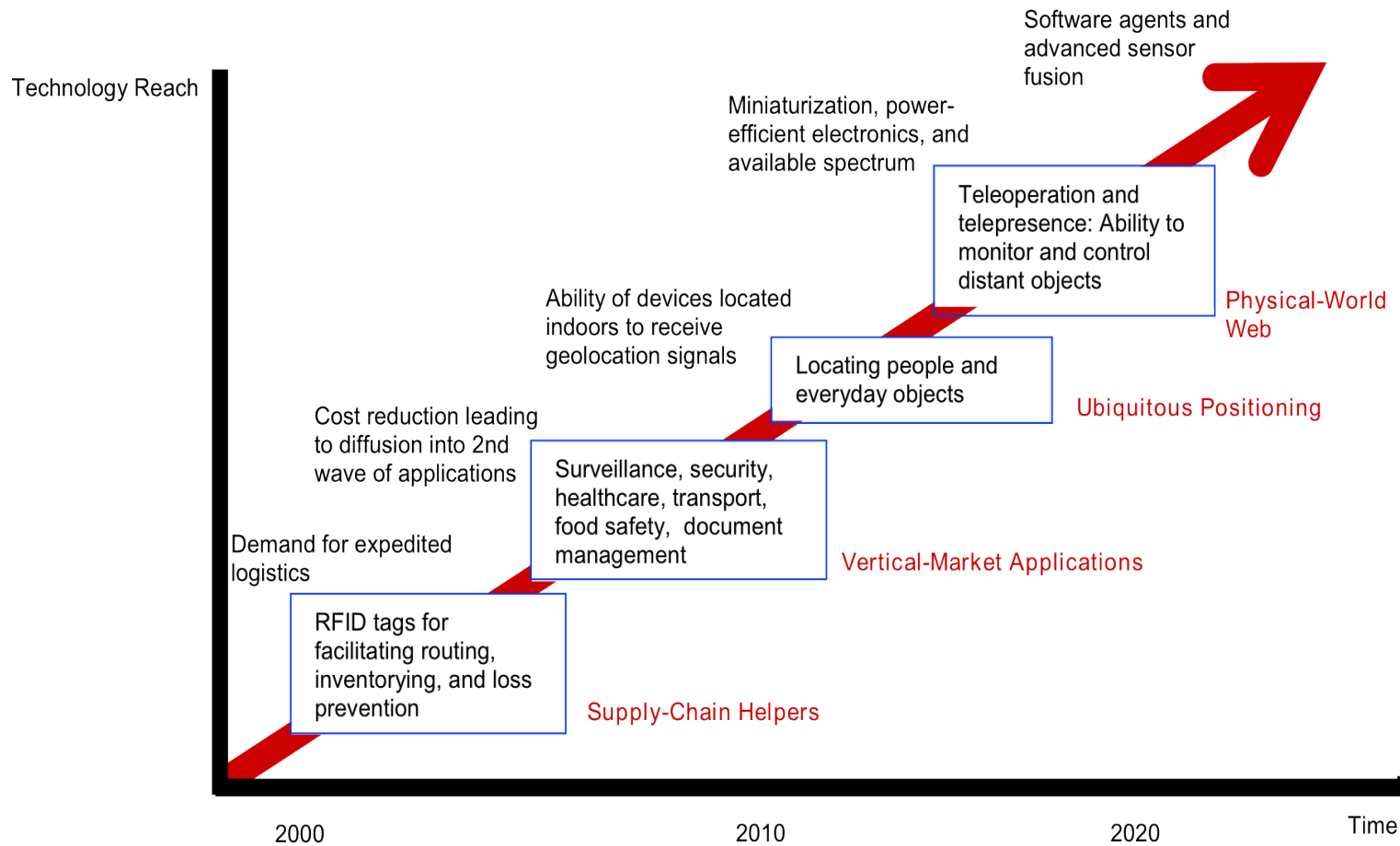
Internet of Things

- Real-world objects (lights, cars, packages, etc.) are **interlinked & connected** to the Internet.
 - Location & status can be tracked.
 - Network **intelligent** enough to self-organize information.
 - **Automatically respond** to context, circumstances, or events from the environment
- 



Internet of things

TECHNOLOGY ROADMAP: THE INTERNET OF THINGS



Source: SRI Consulting Business Intelligence

The vision behind SWE

- Quickly **discover sensors** and sensor data (secure or public) based on location, observables, quality, ability to task, etc.
- **Obtain sensor information** in a standard encoding that is understandable by my software and enables assessment and processing without a-priori knowledge.
- Readily **access sensor observations** in a common manner, and in a form specific to my needs.
- **Task sensors**, when possible, to meet my specific needs.
- **Subscribe to and receive alerts** when a sensor measures a particular phenomenon.

Why SWE?

- Enable interoperability not only within communities but between traditionally disparate communities.
 - different sensor types: in-situ vs. remote sensors, video, models
 - different disciplines: science, defense, intelligence, emergency management, utilities, etc.
 - different sciences: ocean, atmosphere, land, bio, signal processing, etc.
 - different agencies: government, commercial, private, Joe Public



What are the benefits of SWE?

- Sensor system agnostic => virtually any sensor or modeling system can be supported
- Net-Centric, SOA-based
 - Distributed architecture allows independent development of services but enables on-the-fly connectivity between resources
- Semantically tied
 - Relies on online dictionaries and ontologies for semantics
 - Key to interoperability



Benefits of SWE cont'd

- Traceability
 - observation lineage
 - quality of measurement support
- Implementation flexibility
 - wrap existing capabilities and sensors
 - implement services and processing where it makes sense (e.g., near sensors, closer to user, or in-between)
 - scalable from single, simple sensor to large sensor collections

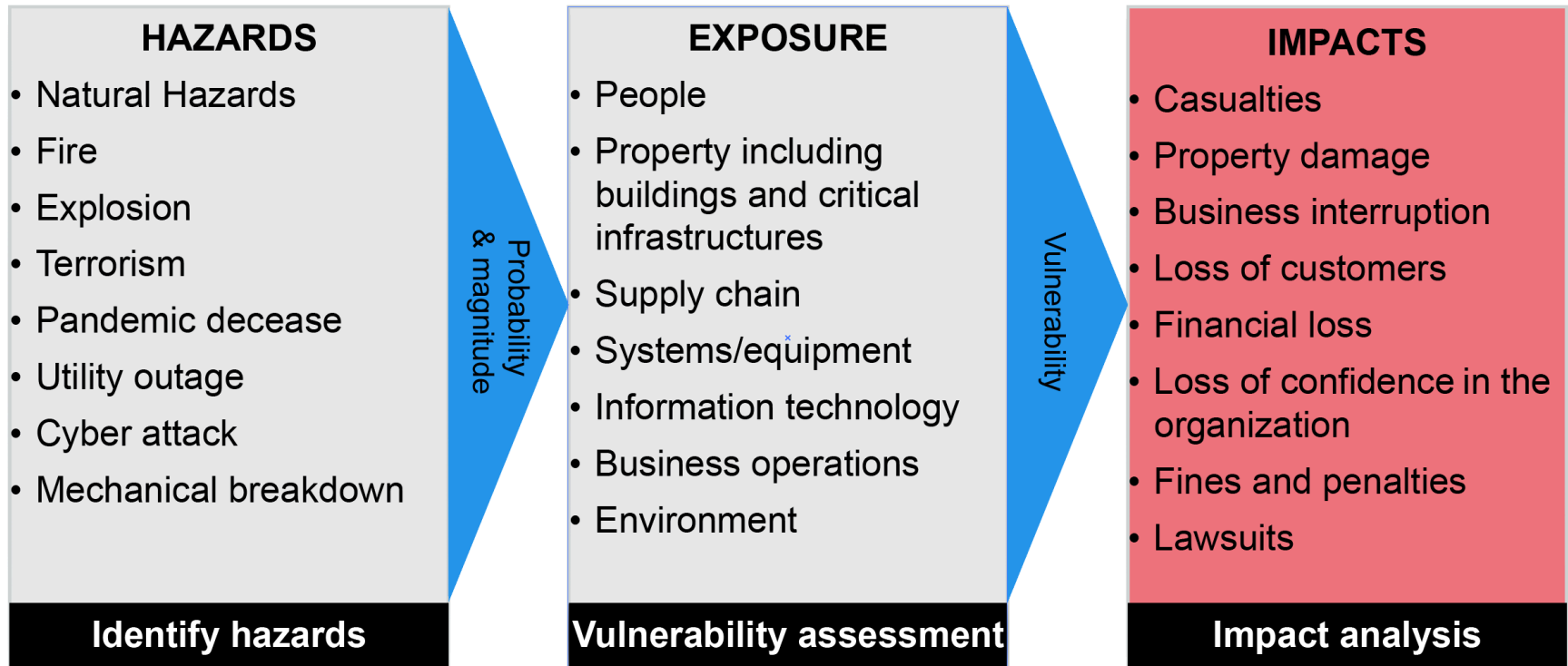
SWE related standards

- There are several adopted or working OGC standards
 - **Observations & Measurements (O&M)** - The general models and XML encodings for observations and measurements.
 - **Sensor Observation Service (SOS)** - Open interface for a web service to obtain observations and sensor and platform descriptions from one or more sensors.
 - **Sensor Model Language (SensorML)** - Standard models and XML Schema for describing the processes within sensor and observation processing systems.
 - **Sensor Planning Service (SPS)** - An open interface for a web service by which a client can 1) determine the feasibility of collecting data from sensors and 2) submit collection requests.

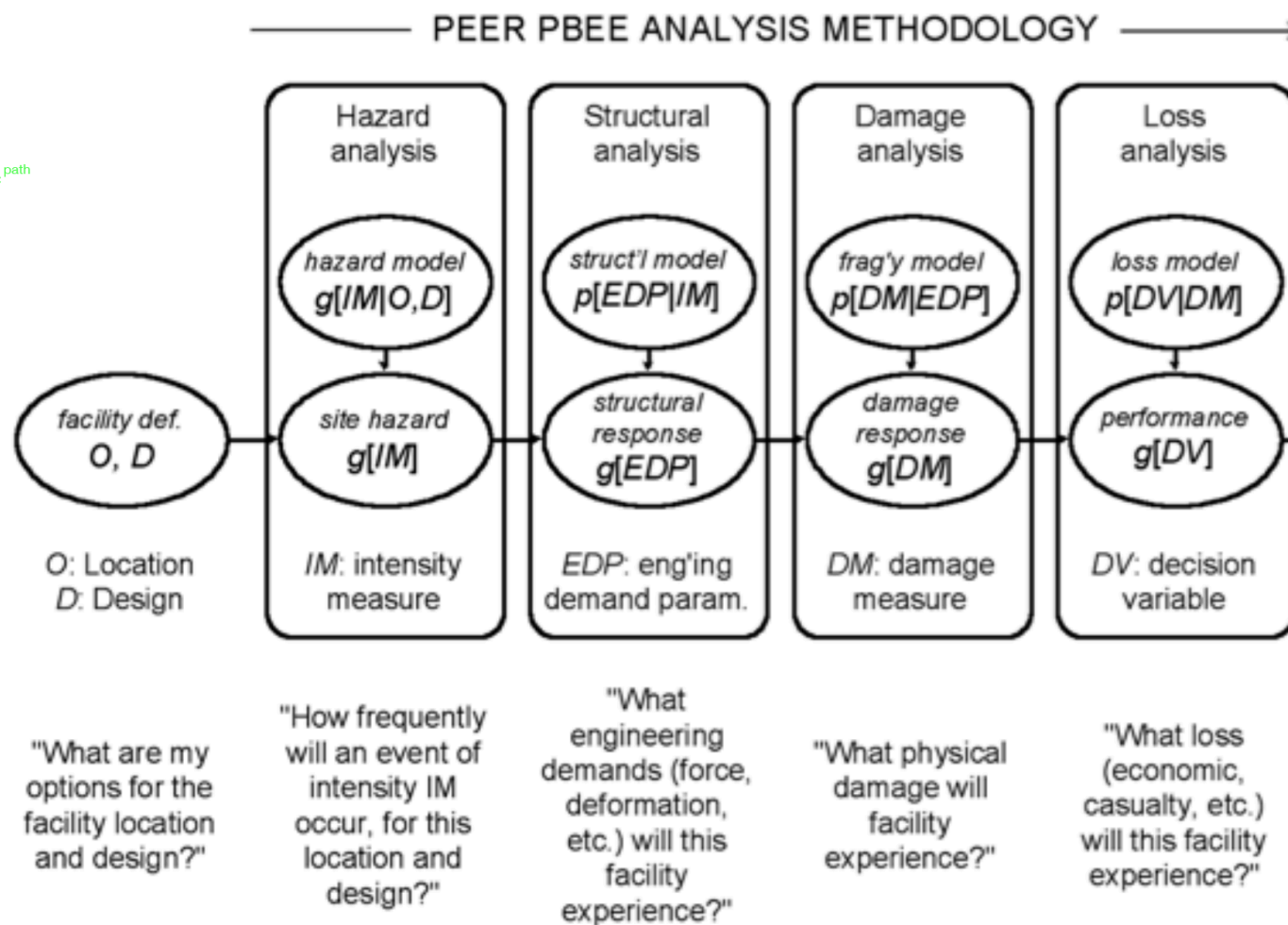
SWE related standards (cont'd)

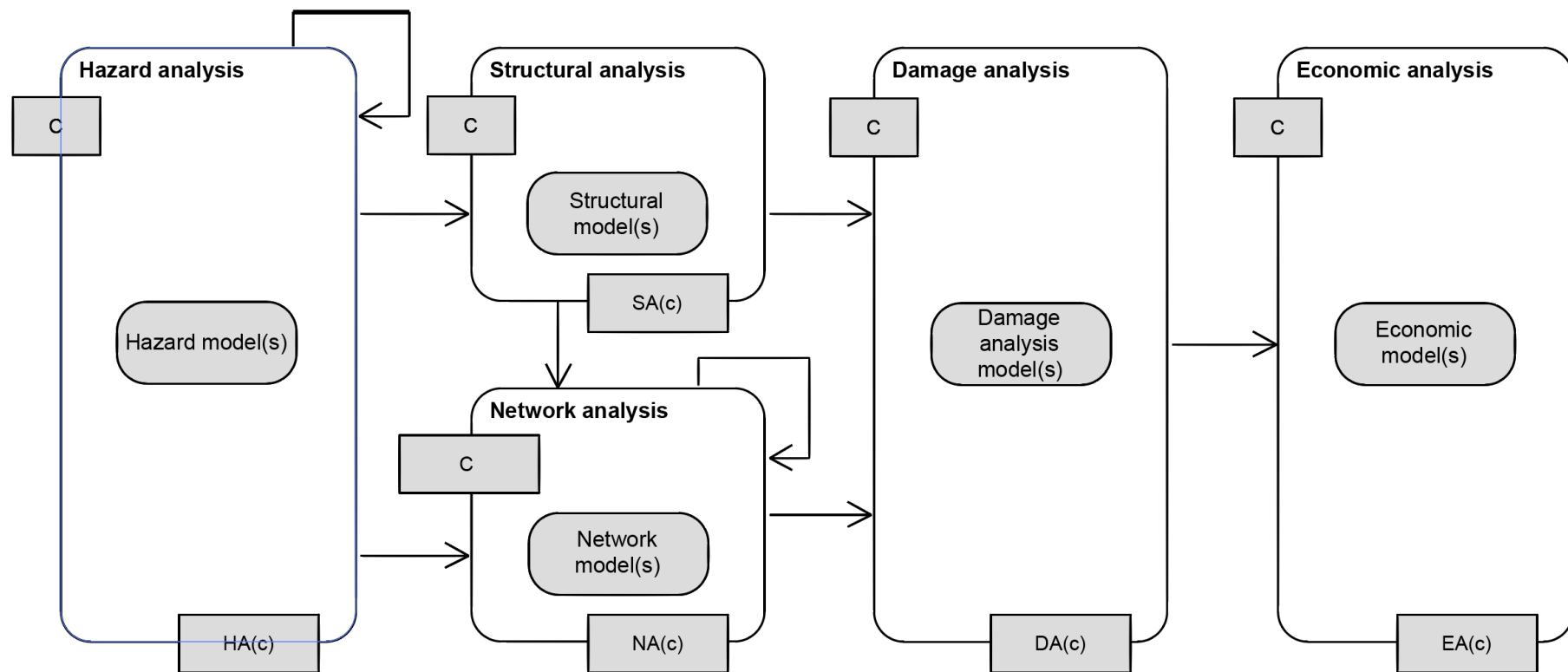
- **PUCK Protocol Standard** - Defines a protocol to retrieve a SensorML description, sensor "driver" code, and other information from the device itself, thus enabling automatic sensor installation, configuration and operation.
- **SWE Common Data Model** - Defines low-level data models for exchanging sensor related data between nodes of the OGC® Sensor Web Enablement (SWE) framework.
- **SWE Service Model** - Defines data types for common use across OGC Sensor Web Enablement (SWE) services.

HOW OPEN GEOSPATIAL STANDARDS ENABLE INTER-DISCIPLINARY COLLABORATION IN RISK-RELATED RESEARCH



path



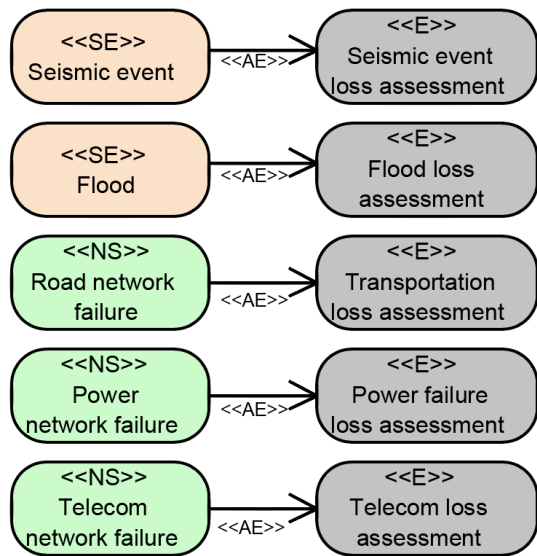


How will hazard(s) evolve and possibly trigger other hazards, assuming a specific disaster context hypothesis (c)?

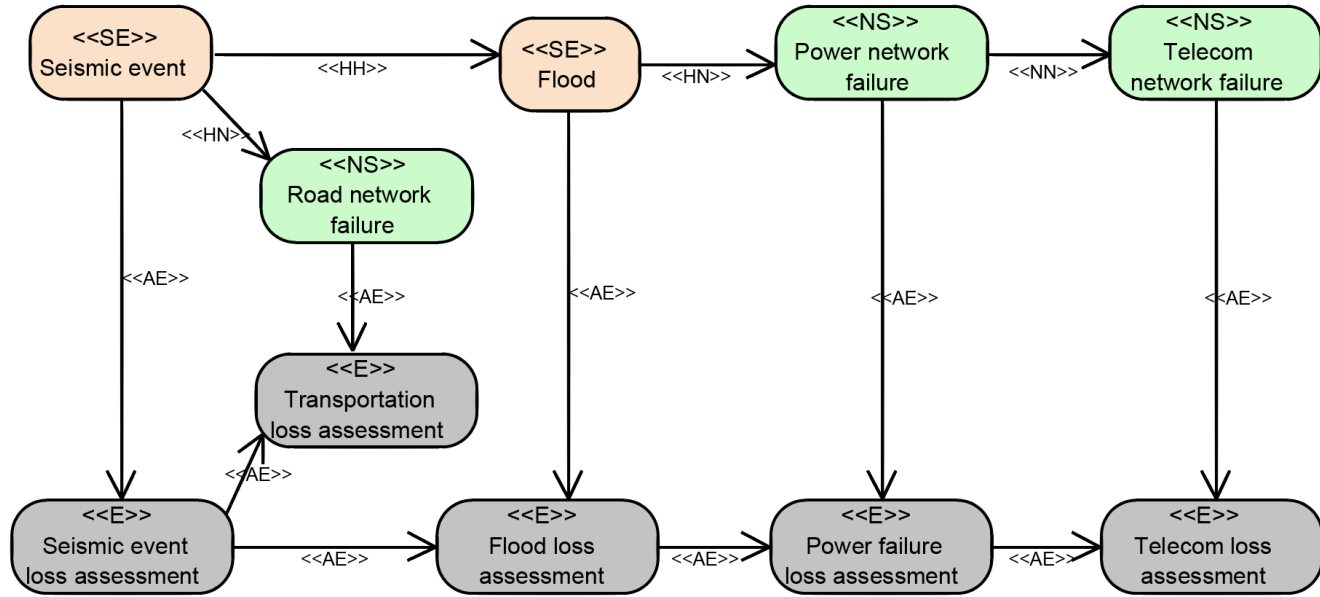
How will structures and networks be affected by hazards and possibly by structure-to-network and network-to-network cascading effects?

What damages will structures and network experience and how will they be distributed over the physical space?

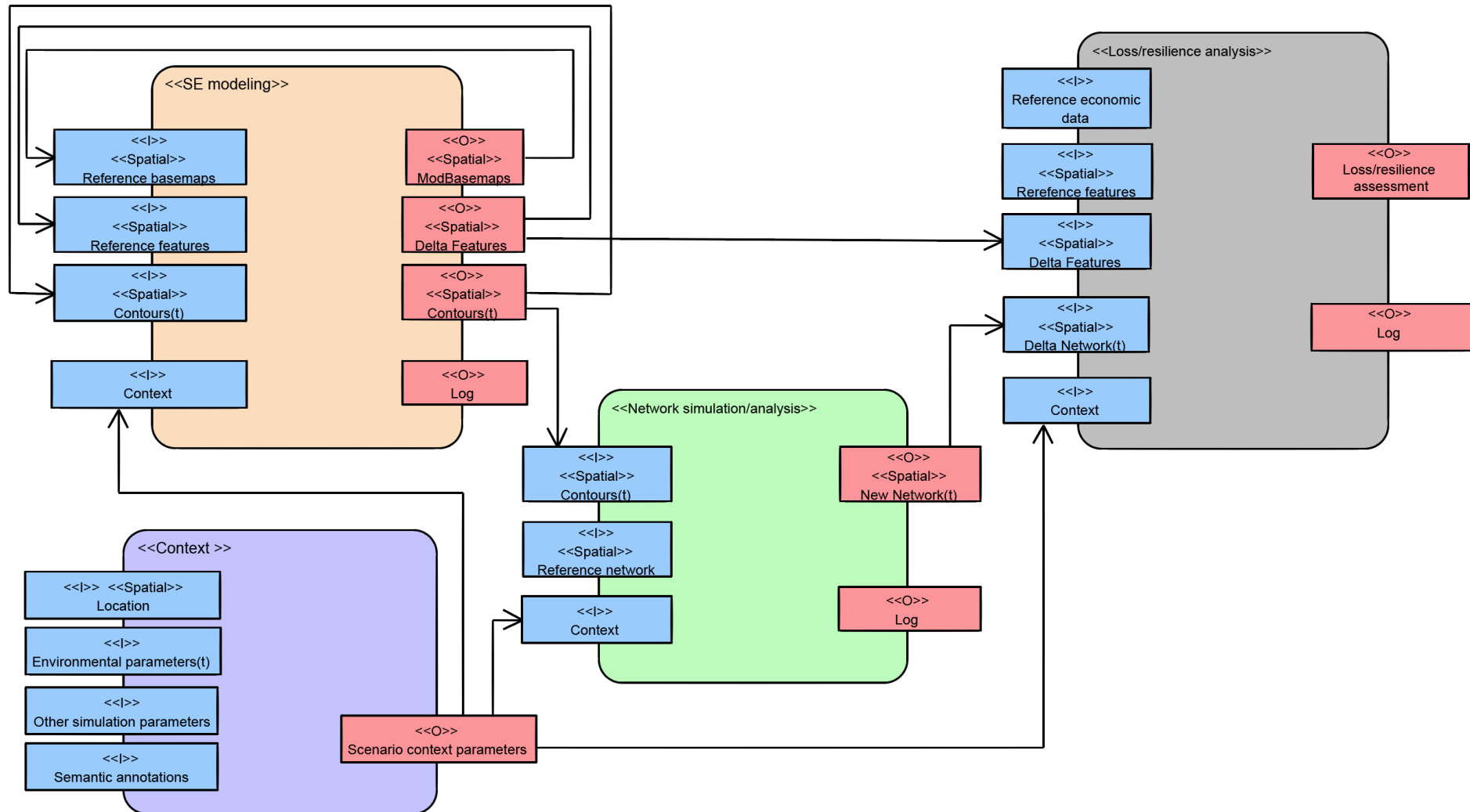
What will be the economic damages (infrastructure, business and social recovery) that will be experienced?

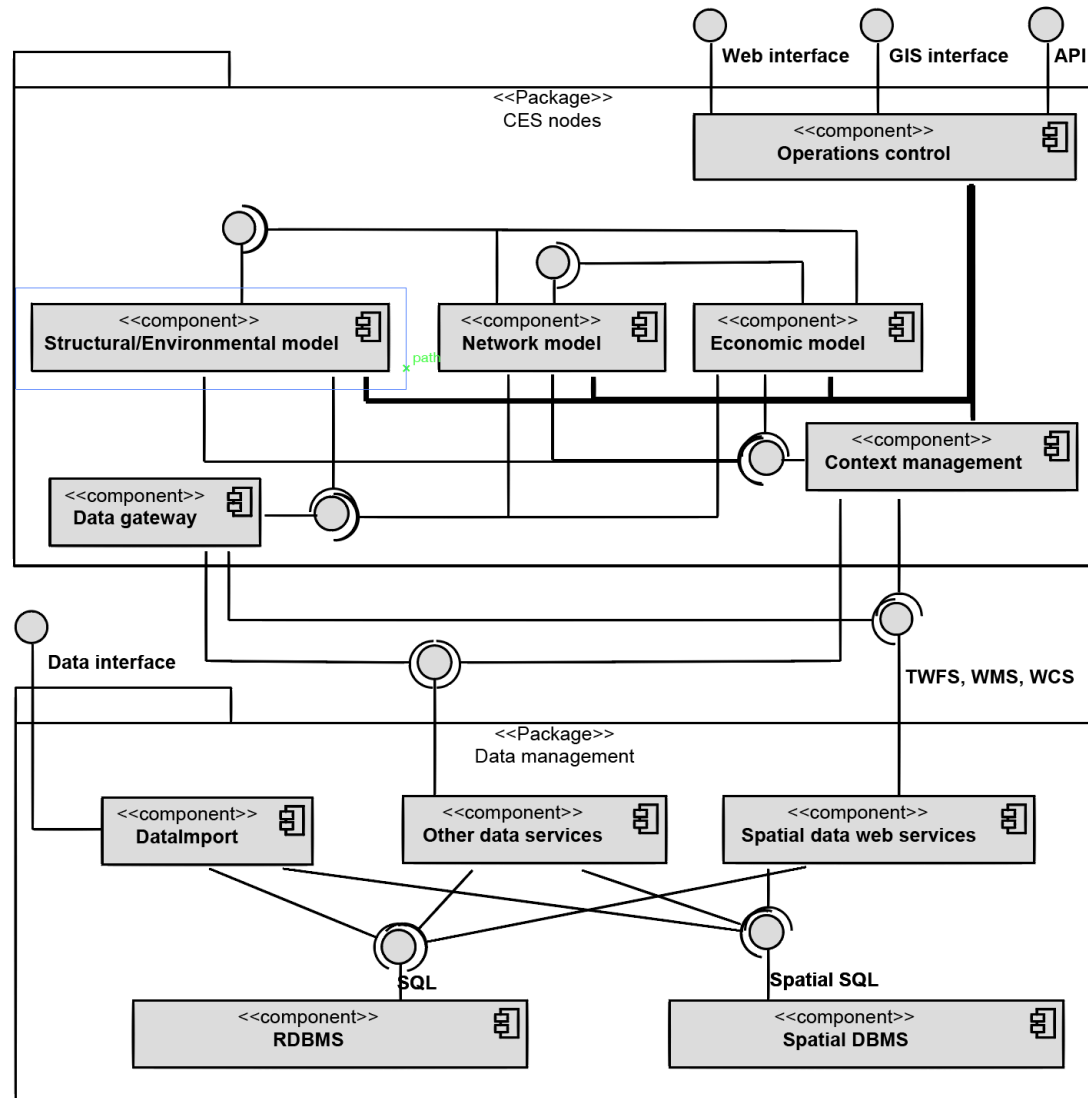


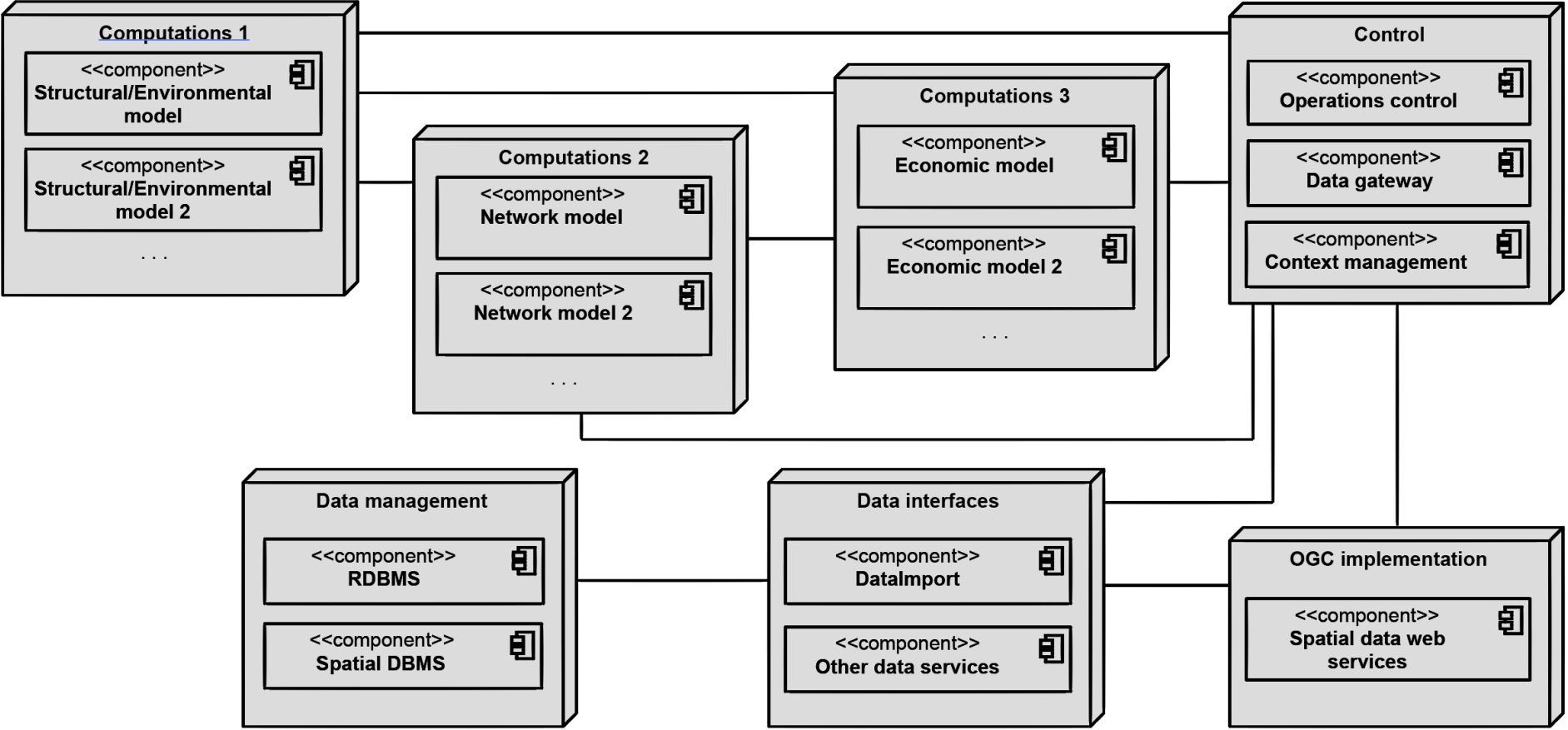
(a)

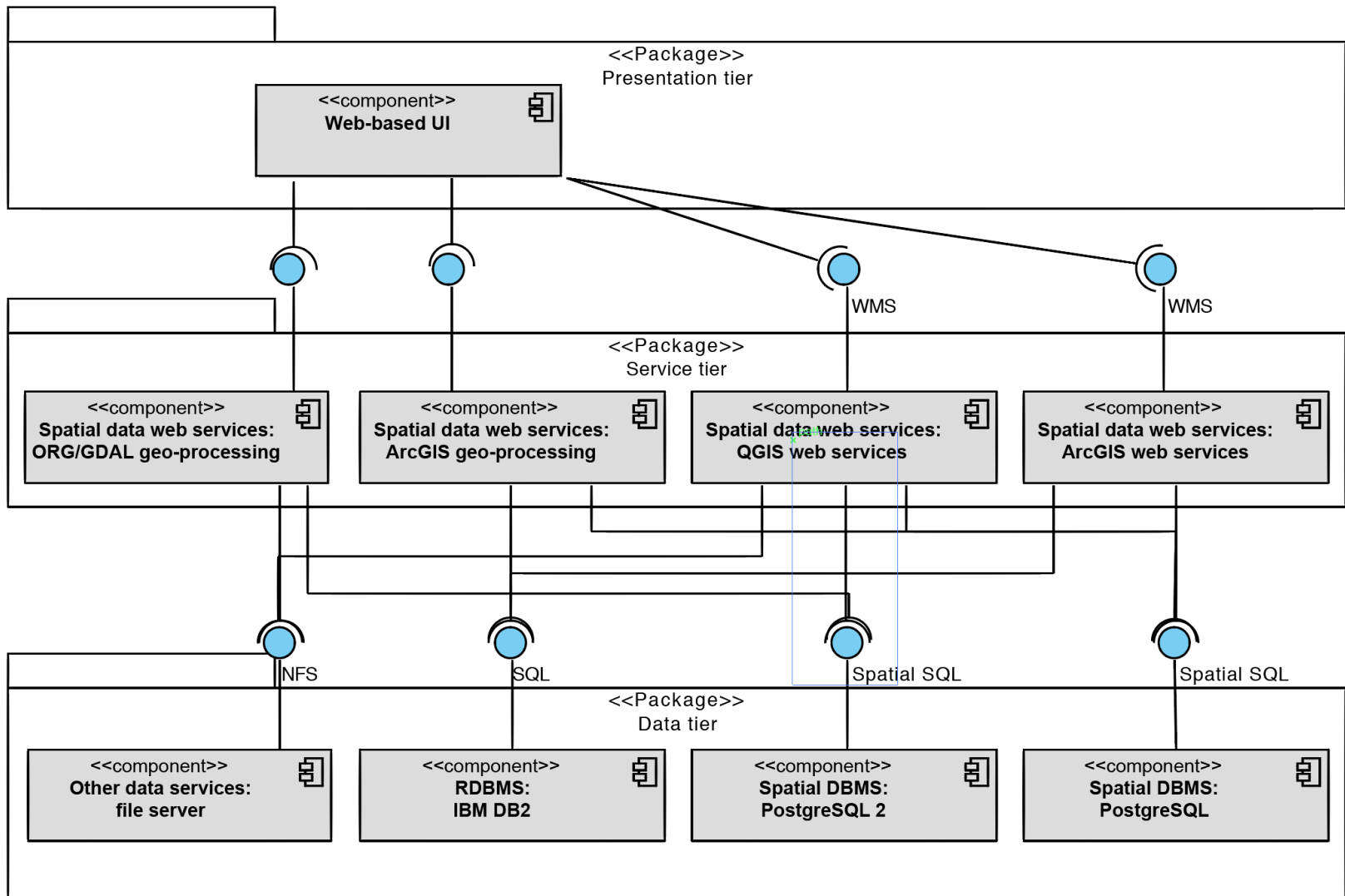


(b)









More on this topic...

Join the ETH Risk Center Fall 2015 Seminars,
on Tue, Oct 13, 2015 in Zurich, to attend my talk

Integration of inter-disciplinary approaches in hazard
management: a Geo-Information Engineering perspective