



---

# Building Service-Oriented Architecture Based Geospatial Web Portal

**Peisheng Zhao, Liping Di, Weiguo Han, Yaxing Wei, Xiaoyan Li**  
Center for Spatial Information Science and System  
George Mason University

*GeoInformatics 2008*



# Outline

---

- **SOA Overview**
- **Portal Overview**
- **Portal Architecture**
- **Portal Solution**
- **Use Case**
- **Conclusion**



# Web Service

---

- A software designed to support interoperable machine-to-machine interaction over a network (W3C).
  - *Encapsulate discrete functionalities*
  - *With an interface described in a machine-processable format*
  - *Programmatically accessible over standard Internet protocols*
  - *Loosely coupled, reusable distributed components for SOA*
  - *Assembling individual Web Services into a service chain (representing a more complicated process flow) to achieve desired results*



# Geospatial Web Service

---

- **Geospatial Data Service**

- *OGC WCS, WFS, WMS and SOS*

- **Geospatial Process Service**

- *OGC WPS: clipping, data fusion*

- *Others:*

- *GRASS (GMU): <http://geobrain.laits.gmu.edu:81/grassweb/manuals/>*

- *ADaM (UAH): <http://datamining.itsc.uah.edu/adam/>*

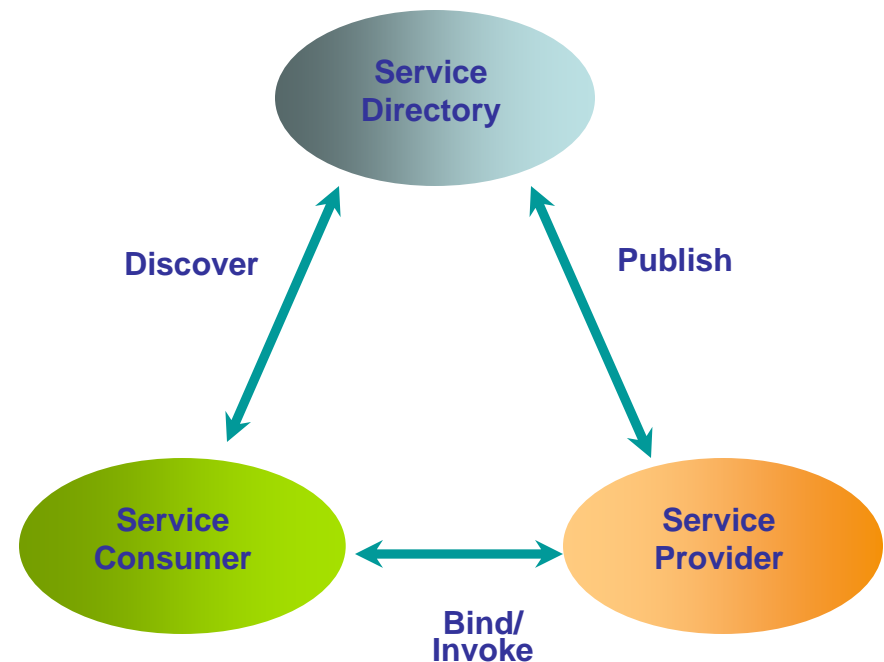
- **Geospatial Catalog Service**

- *OGC CSW*



# Service Oriented Architecture

- A way of thinking in terms of services and service based application development .
  - *A system is consisted of a collection of loosely coupled services that communicate with each other through standard interfaces.*
  - *New solutions can be created by composing together new application-specific services and existing, recombinant services.*





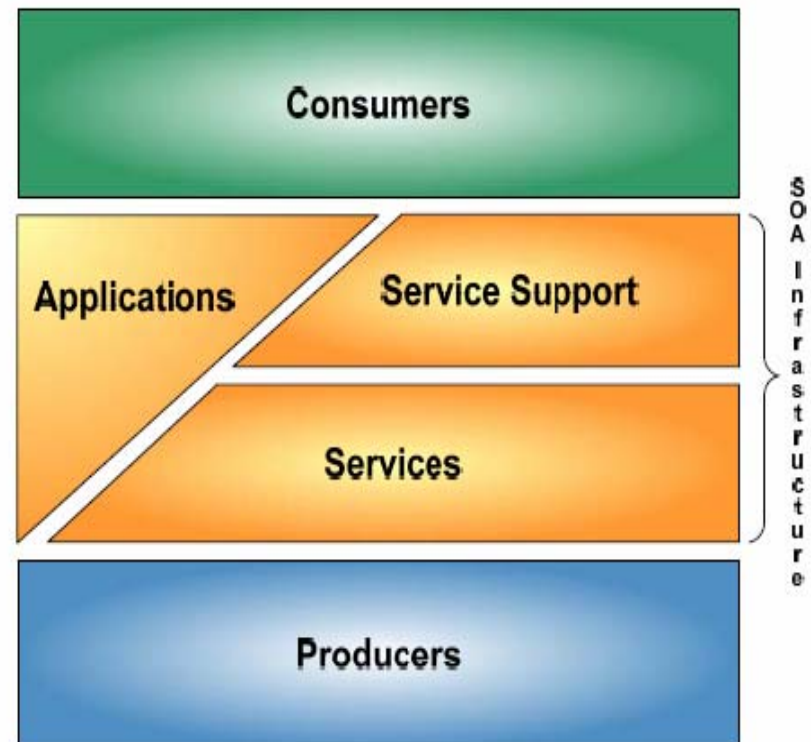
# Benefits of SOA

- **Main drivers of SOA**

- *Consistency*
- *Interoperability*
- *Orchestration*
- *Efficiency*
- *Flexibility*

- **Value of SOA**

- *A simple scalable paradigm for organizing large networks of systems that require interoperability to realize the value inherent in the individual components.*





# SOA-Based Geospatial Web Portal

---

- A fully extensible portal system for discovering, retrieving, analyzing and visualizing geospatial data by bringing distributed geospatial Web services together.
  - *A single point of access to geospatial information and processes over the Web*
  - *User customization and collaboration platform by integrating and chaining user specific Web services*



# Alignment to Key SOA Principles

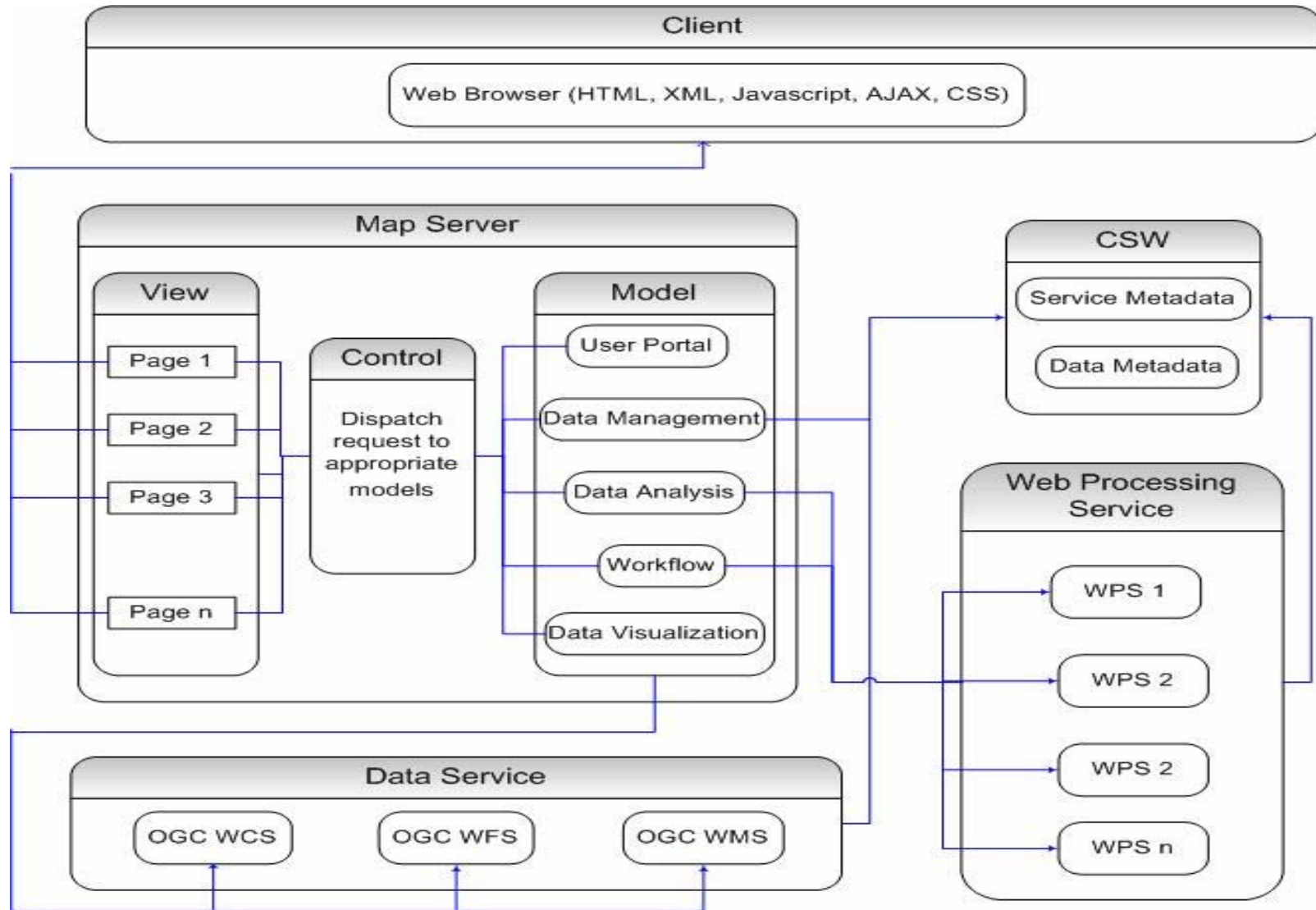
---

- **Services**
  - *All functions are provided through interoperable Web services*
- **Service Descriptions:** *WSDL, OWL-S*
- **Visibility:** *Catalog ebXML*
- **Interaction:** *SOAP, REST*
- **Real World Effect**
  - *Access, integrate and compose distributed geospatial services*
- **Execution Context:** *Asynchronous, synchronous*
- **Contract and Policy:** *user management*





# Architecture





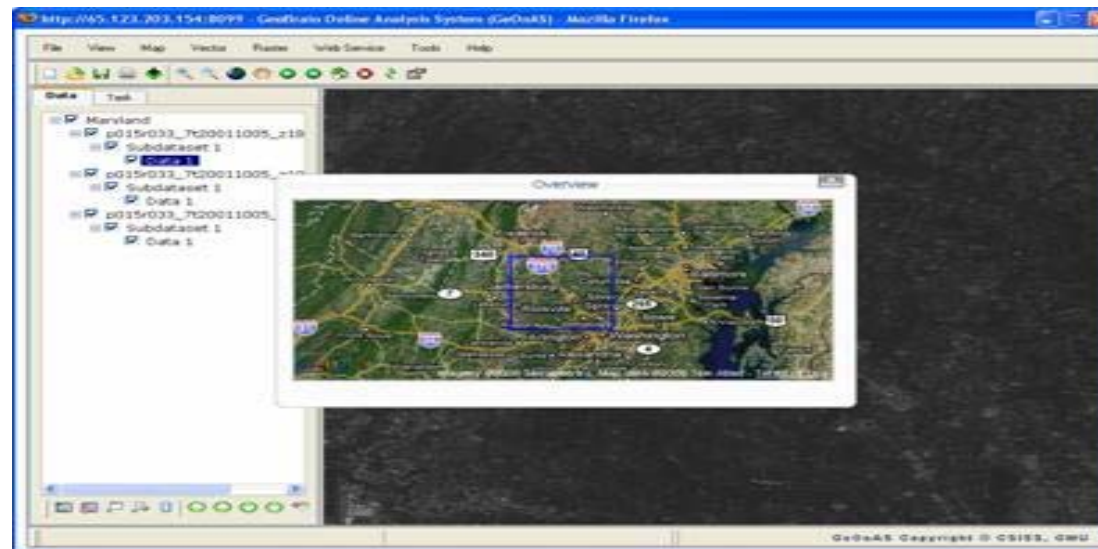
# Consumer

- **Web Browser**

- *Platform and OS neutral user interface: Firefox, IE...*
- *Javascript: AJAX*

- **Service Calling a Service**

- *A single outer service calling multiple portal services for consolidating information.*



<http://geobrain.laits.gmu.edu:81/OnAS/>



# Applications

---

- **User Portal**

- *Each user is able to store the current state of the portal in an OGC Web Map Context (WMC) document that can be imported again later to restore the portals state. The WMC allows users to build their own maps and systems by including list of all preferred data layers and relevant processing services.*

- **Data Management**

- *Each user can discover data from catalog service, retrieval geospatial data from remote service and store them on the map server temporarily with a network accessible point. All meta-information of the saved data is encoded in XML for further use.*

- **Data Analysis**

- *Each user is able to select or integrate a preferred processing service to do data analysis. This model provides users an easy way to set a processing service and get the service outputs.*

- **Workflow**

- *If an analysis task is too complex to be performed by an individual service, this model allows user to build a chain of services to perform the task.*

- **Data Visualization**

- *Users can set up their own preference on how to display data, such as overlay sequence, data subsetting and image palette. A set of different rendering services are provided for the different purposes of data visualization.*



# Services

---

- **Data Service**

- *Provides client access to customized vector and raster data. The customization can be spatial and temporal subsetting, or data format and projection transformation.*

- **Processing Service**

- *provides client access to pre-programmed calculations and/or computation models that operate on spatially referenced data. The calculation can be as simple as subtracting one set of spatially referenced numbers from another, or as complicated as a global climate change model.*



# Service Implementation

---

- **From scratch:**

- *Start with the WSDL design using UML or other visual design tools*
- *Decide the message styles (either RPC encoded/literal or document encoded/literal)*
- *Develop the service program*
- *End with the publication and testing of the Web services in a SOAP container*

- **Wrapping existing system**

- *Create of adapters that make the legacy system compatible with the Web services.*
  - *Create an HTTP adapter to convert the input and output between SOAP and HTTP messages*
  - *Develop a Java adapter to call the C/C++ component through JNI (Java Native Interface)*



# Implement OGC Data Services

---

- **Analyze Geospatial Data**
  - *Purpose*
  - *Data Characteristics*
- **Determine Proper Interfaces**
  - *OGC Services*
- **Develop Software**
  - *Based on Open Source: MapServer, GeoServer*
  - *By Own: CGI, ASP, Servlet*



# Implement GRASS Services

---

- **GRASS**

- *GRASS is an open source GIS software with over 350 programs and tools for raster and vector data analysis.*

- **GRASS Services**

- *Atomic services based on GRASS API. However, each command in GRASS API is tightly couple with each other that is contradict with the Web service design, and some of them have no explicit physical meaning unless they are in use combined with others.*
  - *Services on top of a large GRASS script which represents a complex physical model. For example, “create\_dummy\_location”, “r\_in\_gdal”, “r\_mapcalc” and “r\_out\_png” are used in composing a NDVI service.*



# Service Support – Catalog Service

---

- **Directory Role**

- *Providers advertise the availability of their resources, and consumers can then query the metadata to discover and run-time access them.*

- **EB/RIM Information Model**

- *Specifies formally how domain objects are organized, constrained and interpreted based on domain conceptual structure.*

- **OGC CSW -- Standard Interfaces**

- *GetCapabilities, describeType, getRecord...*

<http://geobrain.laits.gmu.edu/csw/discovery/>





# Service Registration and Discovery

### Service Registration

\* Required Fields

---

#### Basic Information

Name \*:  ⓘ

Description:

Version:

Keywords:

---

#### Addresses

Service Address \*:  ⓘ

WSDL Address:  ⓘ

---

#### Service Classification ⓘ

LAITS\_VDP:

OGC:

ISO19119:

GCMD:



# Data Registration and Discovery

http://127.0.0.1:8080 - Select Data to Load - Mozilla Firefox

Catalog   Spatial   Temporal   Others

Choose Catalog Services:

GMU-LAITS CSW  
 NASA ECHO

Tips:


1. You can choose any or both catalogs from which to search data.
2. GMU-CSISS CSW is a local catalog maintained by GMU CSISS. It catalogs global Landsat TM/ETM data, some MODIS and ASTER data, and other remote sensing data.
3. NASA ECHO is an external catalog service. It may take several minutes to fulfill your request.

a

http://127.0.0.1:8080 - Select Data to Load - Mozilla Firefox

Catalog   Spatial   Temporal   Others

Ignore Bounding Box Conditions



Choose Location:

State:

County, City, etc:

Or

(Note: Please input country name to define BBox.)

Or Input BBox:

CRS:

Northern:

Western:   Eastern:

Southern:

b

http://127.0.0.1:8080 - Select Data to Load - Mozilla Firefox

Catalog   Spatial   Temporal   Others

Choose Data and Time:

ProductTime:

CollectionRange: BeginDateTime:  EndDateTime:

Tips:

1. **ProductTime:** Date and time when a specific granule was produced by a PGE.
2. **CollectionRange:** Date and time when the temporal coverage period being described began and ended.
3. Datetime should be input in the format as "2003-08-20 23:34:45". You can use calendar tools to choose a datetime.
4. Operator "><" means "between". If you choose it, please input the corresponding value as "LowerBoundaryDateTime & UpperBoundaryDateTime". The "&" between those two datetime values is a delimiter.

c

http://127.0.0.1:8080 - Select Data to Load - Mozilla Firefox

Catalog   Spatial   Temporal   Others

Other Attributes:

Collection:

SensorShortName:

InstrumentShortName:

PlatformShortName:

TopicKeyword:

DisciplineKeyword:

TermKeyword:

VariableKeyword:

ParameterKeyword:

ProcessingLevel:

archiveCenter:

Day/Night Flag:

d

itic



# Integrate New Services

---

- **Allows users to add their own services to build a customized portal.**
- **More powerful as more users are involved.**
  - *Register a service with its WSDL in catalog service.*
  - *Select a data that is for analysis.*
  - *Discover a service that can handle the data selected in step 2.*
  - *Select a proper operation from the operation list*
  - *Input the parameter values of the selected operation.*
  - *Invoke the selected service and add the results into the portal*



# Service Support -- Workflow Engine

- **BPELPower:** <http://data.laits.gmu.edu:9180/bpelasync>

**Business Process Execution Language (BPEL) for Web Services**  
**Web Service Definition Language (WSDL)**

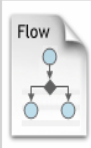


**Workflow Engine - 100% supports BPEL and WSDL**

**BPEL Process Manager - BPELPower v3.0(beta)**

Enhanced with WS-Addressing for Aynchronony.

Supports: WS-BPEL (1.1), WSDL 1.1, SOAP (1.1, 1.2), WS-Addressing (2003/03/03,2004/04/08, 2005/05/08)

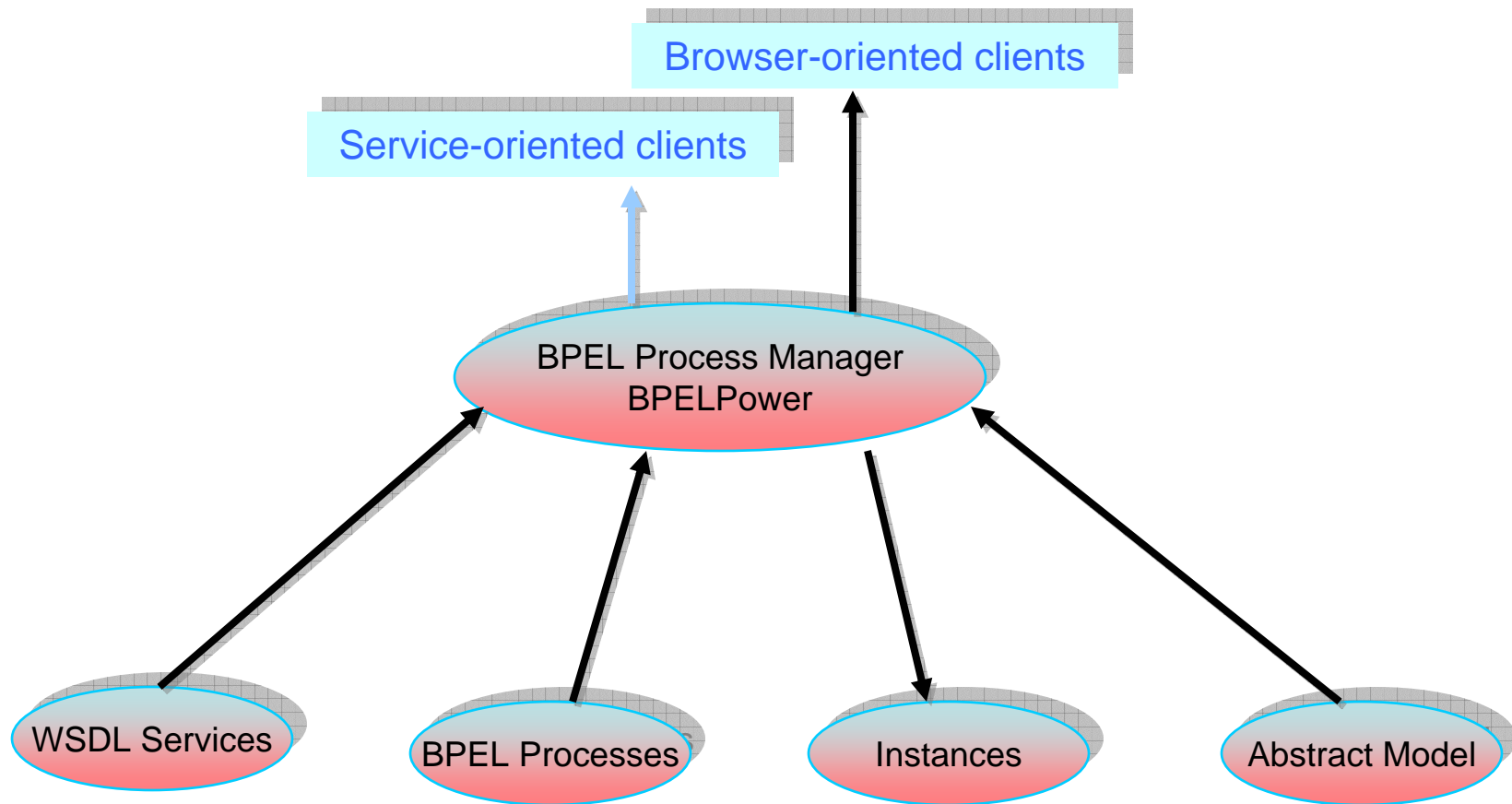
What do you want to do today? Please [Run](#) the admin client.

Copyright © 2004,2005,2006,2007 CSISS/LAITS BPELPower. All rights reserved. [About CSI](#)



# BPELPower Architecture





# BPELPower – Easy Orchestration

---

- **BPEL Process manager**
  - *Integrates services into collaborative and transactional business processes, working within a Service Oriented Architecture (SOA) as an orchestration engine*
- **Based on the mainstream standards**
  - *BPEL, WSDL, WSIF, Xalan, Xerces, UDDI, AXIS, SOAP, JNDI, J2EE (servlets/EJBs/JSPs), Jetspeed (Portlets) and JMX. It runs on top of popular application servers, such as Tomcat, J2EE, JBoss, Weblogic and WebSphere.*
- **“Deploy it”.**
  - *WSDL-based web services and BEPLE-based web services chain can be deployed in BPELPower, where their validations are checked.*
- **“Try it”.**
  - *WSDL-based web services and BEPLE-based web services chain can be executed in BPELPower dynamically. Different invocations (e.g., HTTP POST/GET, SOAP document/rpc, etc.) are well supported.*



# BPELPower -- Web User Interface

BPEL Process Manager - BPELPower - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://geobrain.laits.gmu.edu:8098/bpel

Web Model Designer Web Services Orchestration: ... BPEL Process Manager - B...

**BPELPower** BPEL Process Manager

WSDL Services BPEL Processes Instances LBPL Activities

### Deployed WSDL Services

Service Name	Service Name
<a href="#">CurrencyExchangeService</a>	<a href="#">EMBLNucleotideSequenceWebService</a>
<a href="#">GMU-NGA-WCS</a>	<a href="#">GMU-WICS</a>
<a href="#">Grass_General_CommandsService</a>	<a href="#">Grass_Imagery_CommandsService</a>
<a href="#">Grass_Raster3D_CommandsService</a>	<a href="#">Grass_Raster_CommandsService</a>
<a href="#">Grass_Raster_InOut_CommandsService</a>	<a href="#">Grass_Vector_CommandsService</a>
<a href="#">Grass_Vector_InOut_CommandsService</a>	<a href="#">GridSlopeAspectService</a>
<a href="#">GridSlopeService</a>	<a href="#">GridWCSService Laits</a>
<a href="#">GridWCSService No</a>	<a href="#">GridWCSService</a>

[Deploy New Services](#) [Undeploy Services](#)

Logged to domain: **default** BPELPower v2.0.



# BPELPower – Advanced Capabilities

---

- **XML schema capabilities**
  - *GML: more complex*
- **Asynchronous support**
  - *WS-Addressing*
  - *Asynchronous service*
  - *Asynchronous workflow*
- **RESTful support**
  - *REST (Representational State Transfer)*
- **Advanced security support**
  - *Transport level security*
  - *Message level security*

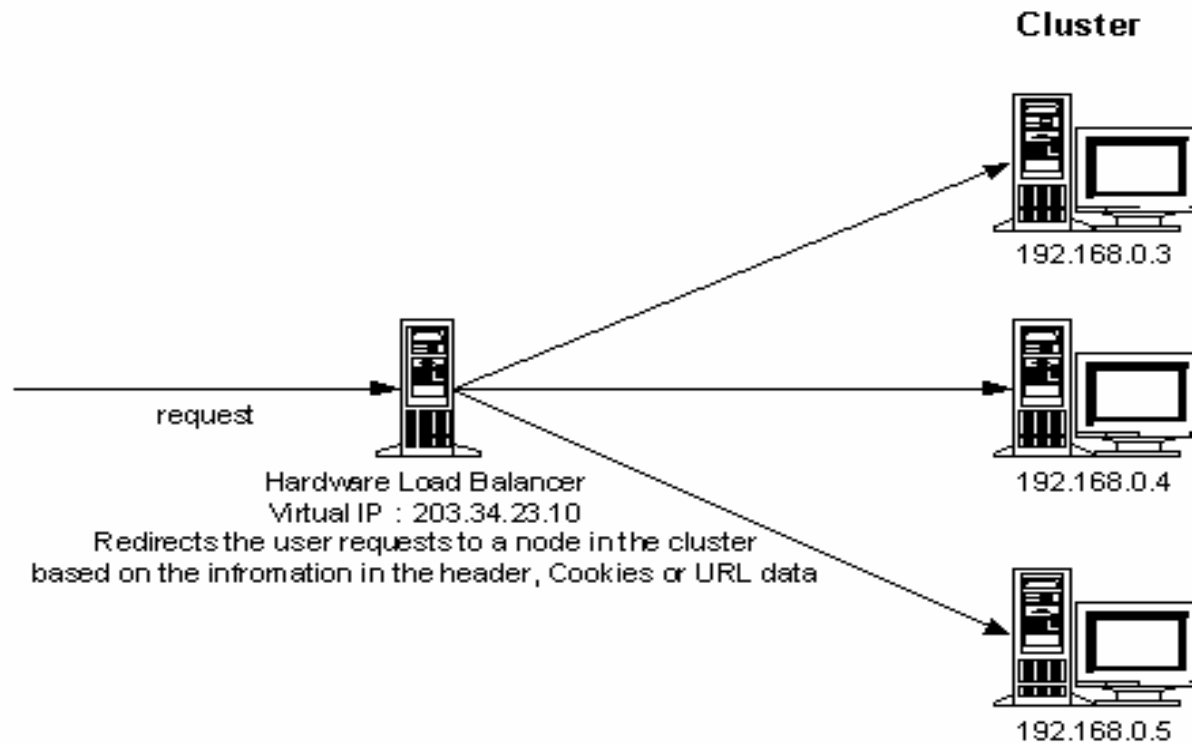




# Service Support – Load Balancing

- **Service load balancing -- server cluster**

– *a group of independent and redundant servers that are managed as a single system for higher availability, easier manageability, and greater scalability.*





# Use Case – Stream Extraction Service (1)

---

- **Stream Extraction**

- *GRASS Hydrology-based Stream Extraction*

- *Based on flow direction along steepest descent and using some threshold criteria to separate channels and hillslope*
- *Tend to produce results that are spatially uniform, not correctly reflecting the spatial variability in stream dissection patterns.*

- *Enhanced Morphology-based Stream Extraction*

- *Delineates the stream networks as parts of the raster having a U-like morphology, which is represented as positive curvature derived from DEM.*



## Use Case – Stream Extraction Service (2)

---

- **Implementation**

- *Morphology-based Algorithm*

- *Mainly written in C++, and partly in FORTRAN*

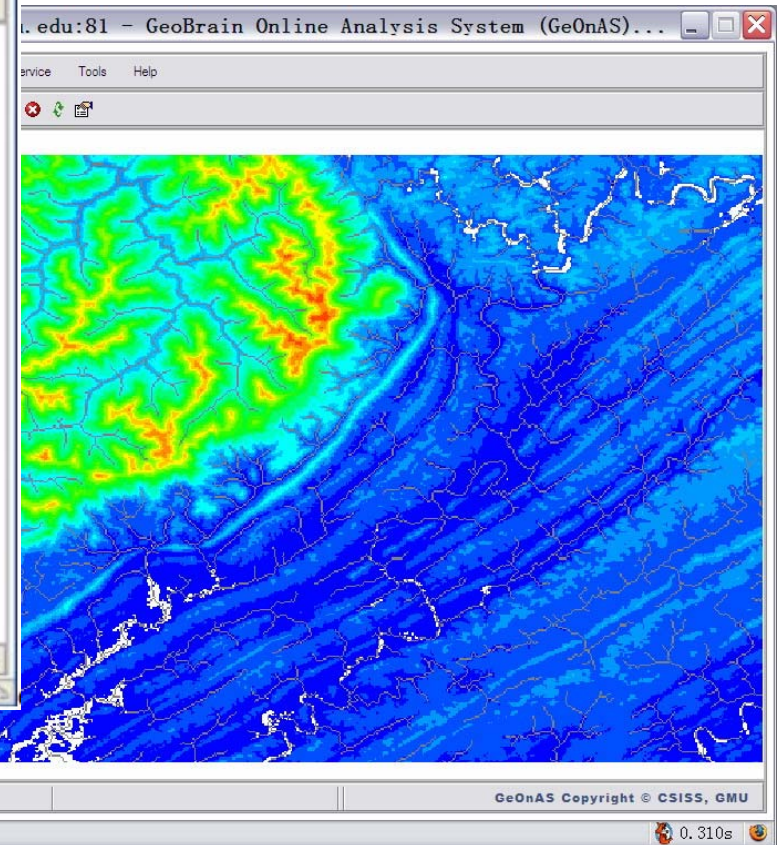
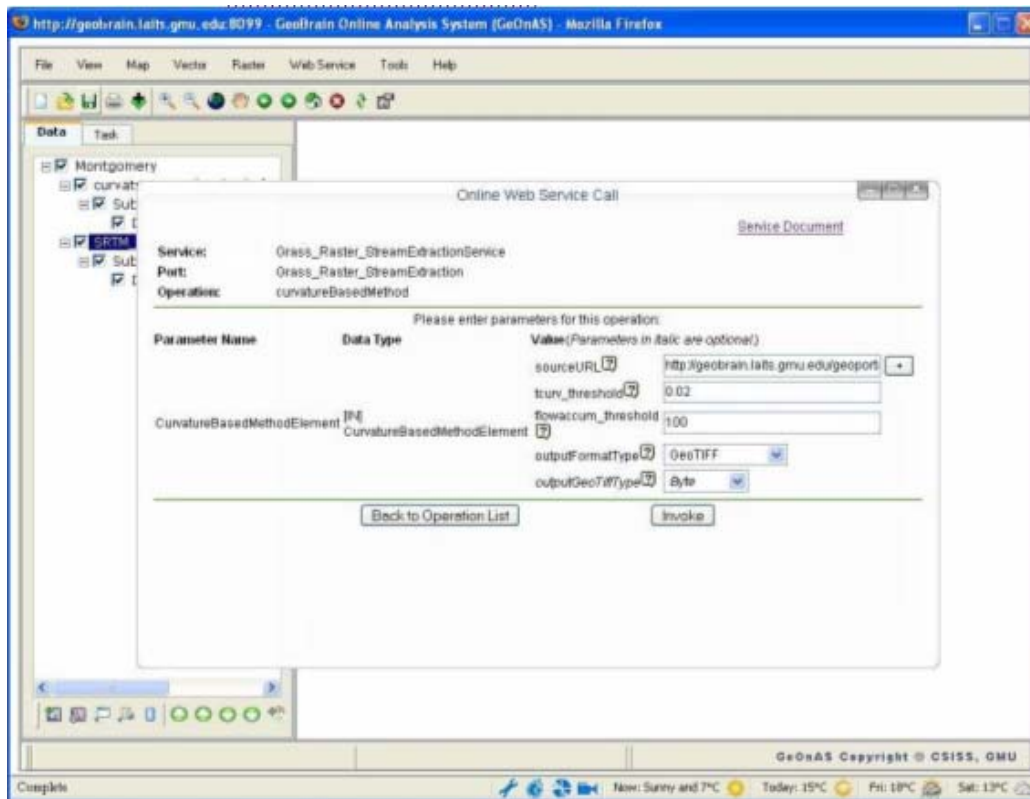
- *Web Service*

- *Written in Java, which calls some basic programs including pit filling corrections, flow direction computation, flow accumulation computation, and some utility functions ported from GRASS*
- *Described in WSDL.*
- *Registered in catalog service*



# Use Case – Stream Extraction Service (3)

- **Invocation**





# Use Case – Image GeoReferencable Workflow (1)

---

- **Scenario**

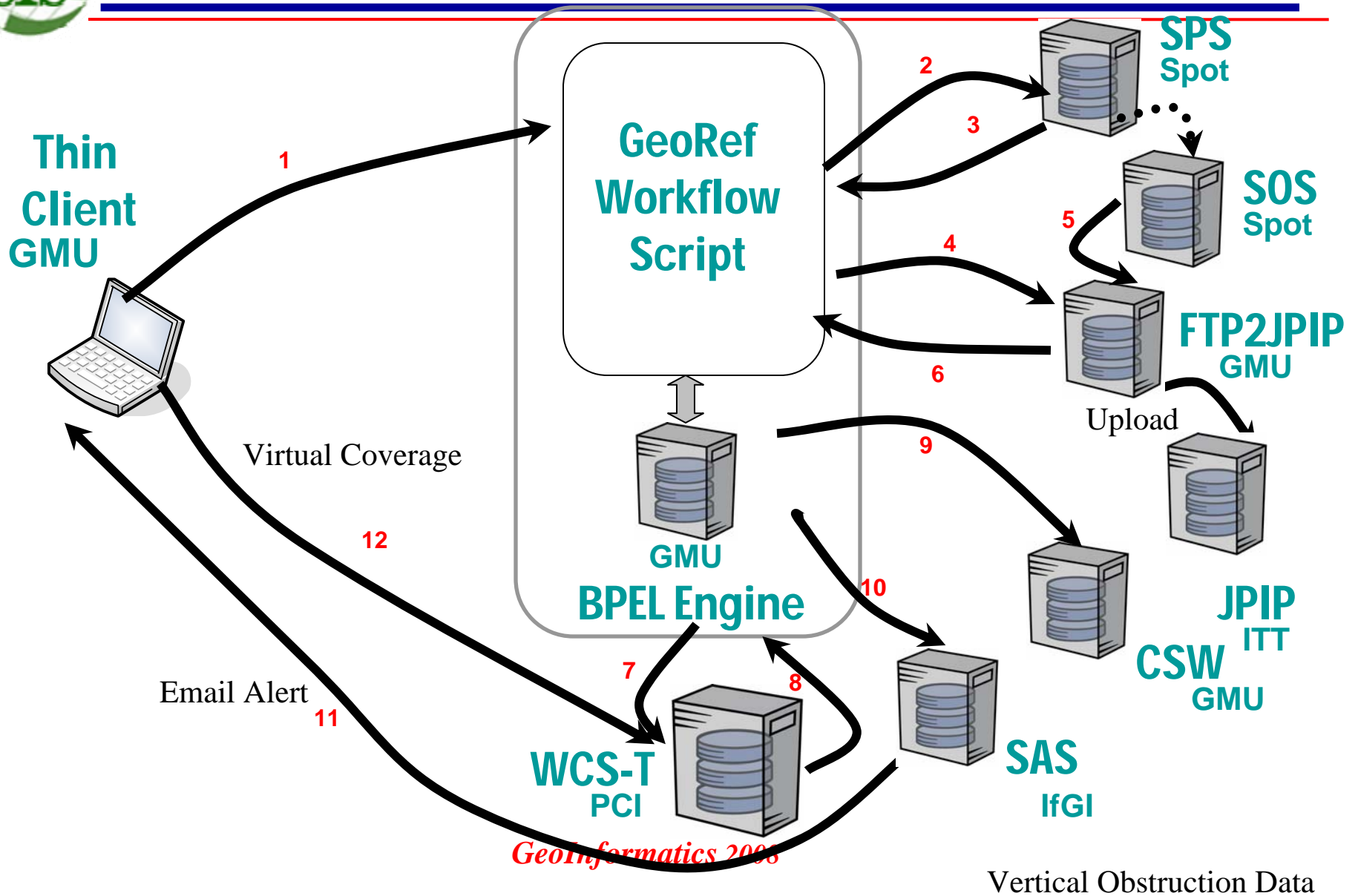
- *Establishment of a standardized means to allow the user to interactively access a subset pixels from a coverage service stored in the compressed domain (JPEG2000) and preserve the image relationship with the associated 'sensor model' parameters such that precise geopositioning capabilities can be realized in a dynamic, interactive, networked environment.*

- **Participants**





# Use Case – Image GeoReferencable Workflow (2)





## Use Case – Image GeoReferencable Workflow (3)

---

1. *Develop the workflow using Oracle BPEL Designer*
2. *Deployment the workflow in BPEL engine*
3. *Register workflow in CSW*
4. *Invoke the Workflow via a standard Web service using portal*
5. *Receive the email to know data available*
6. *Query CSW and request virtual JPIP coverage through WCS.*
7. *Show the virtual coverage in the client application*
8. *Export the virtual coverage into Google Earth*



# Conclusion

---

- **Online Data Analysis**
  - *Discovery → Analysis → Visualization*
- **Interoperability**
  - *Web Services*
    - *Catalog service, OGC data services, WPSs...*
- **Expansibility**
  - *Integrate new services*
    - *Easily register, discover and invoke new services*





# Thank You

Contact: [pzhao@gmu.edu](mailto:pzhao@gmu.edu)

*GeoInformatics 2008*