UNIVERSITY OF TWENTE.

WEB-GIS AND WEB COMPUTING

OVERVIEW OF CONCEPTS AND REQUIREMENTS

ULAN TURDUKULOV

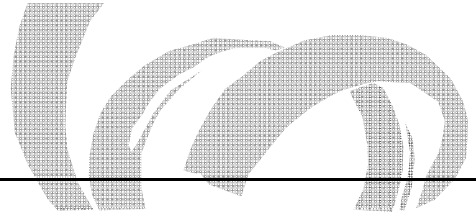
DEPARTMENT OF GEO-INFORMATION PROCESSING

FACULTY ITC

UNIVERSITY OF TWENTE

ITC FACULTY OF GEO-INFORMATION SCIENCE AND EARTH OBSERVATION

OUTLINE



In this short time you will be exposed to number of concepts/definitions such as:

- Distributed GIS systems, Internet GIS/web-GIS
- Client-Server, thin and thick clients, web services
- OGC web services
- Software architectures that allow implementation of OGC web services

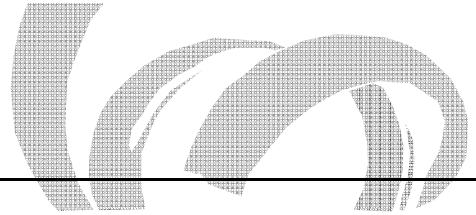
There are no web-GIS examples since you will be given number of presentations later in the day.

OVERVIEW OF REQUIREMENTS

Characteristics of Disasters:

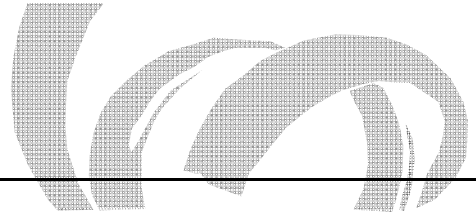
- **Large scale**, which means two or more jurisdictions (municipalities, provinces, etc.) will be involved.
- **Rapid onset**, which means the events give people no time or short time to prepare to evacuate.
- **Dynamic,** which means disasters evolve as they progress.

NEEDS IN DISASTER MANAGEMENT



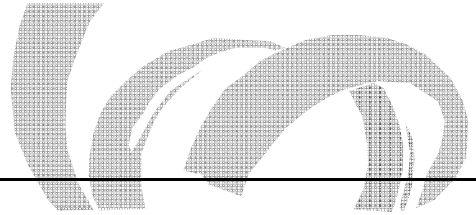
- The need for efficient and effective access to decision-making information relevant to all phases of risk and emergency management;
- The need for collaboration among the agencies at different levels;
- The need for the collaboration of different agencies that are in charge of different thematic data.

WHY WEB-GIS?



- Flexible information access / exchange
  - End-Users can be both: those that use specialized GIS software and those that use only a web browser
- Information sharing and integration
  - Access multiple Internet Map Servers at the same time– various organizations, etc.
- Real-time information update and distribution

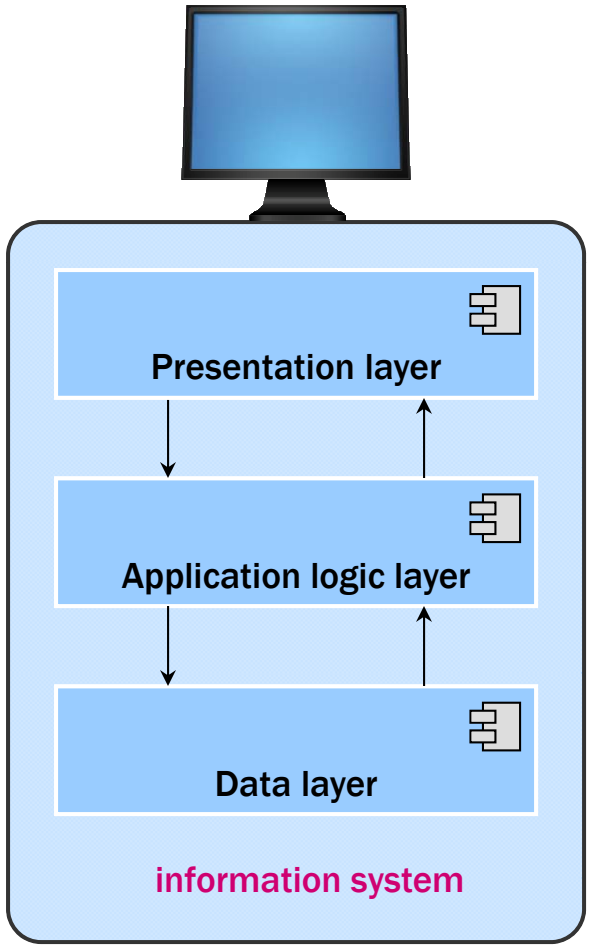
TRADITIONAL GIS



- Built on stand-alone platform.
- Hard to communicate among systems.
- Users at one end of internet/intranet cannot share information with users at the other end.

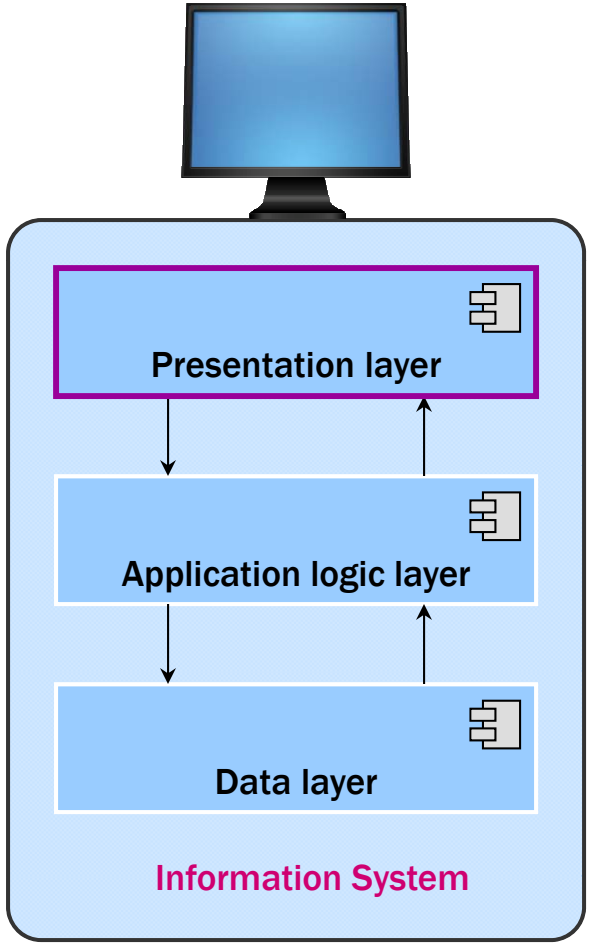
INFORMATION SYSTEM ARCHITECTURE

- Layers of an information system
  - Presentation Layer
  - Application logic layer
  - Data layer



INFORMATION SYSTEM LAYERS

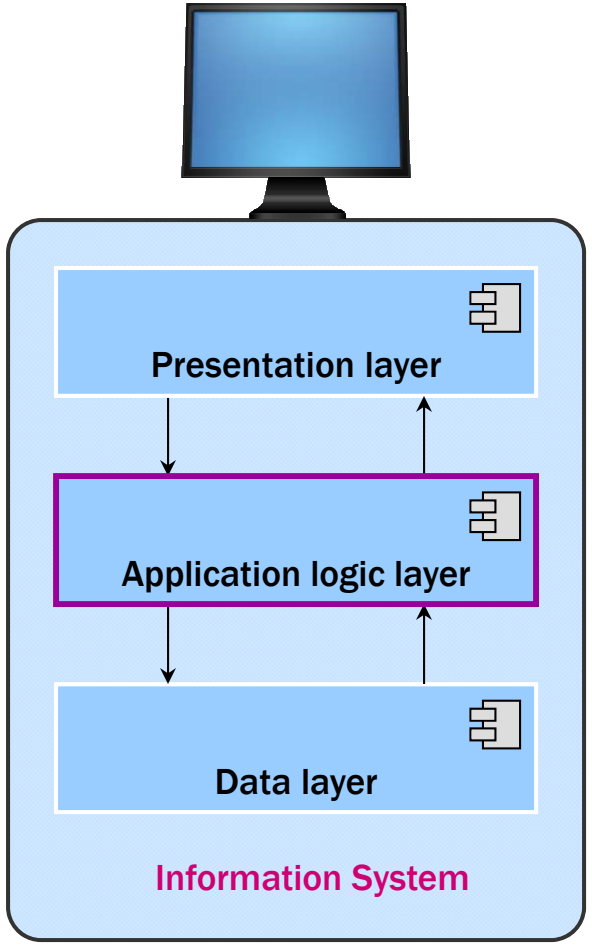
- Presentation layer
  - Realizes the (graphical) user interface
  - Renders the output and accepts user input to the application layer
  - Interface to the application layer



INFORMATION SYSTEM LAYERS

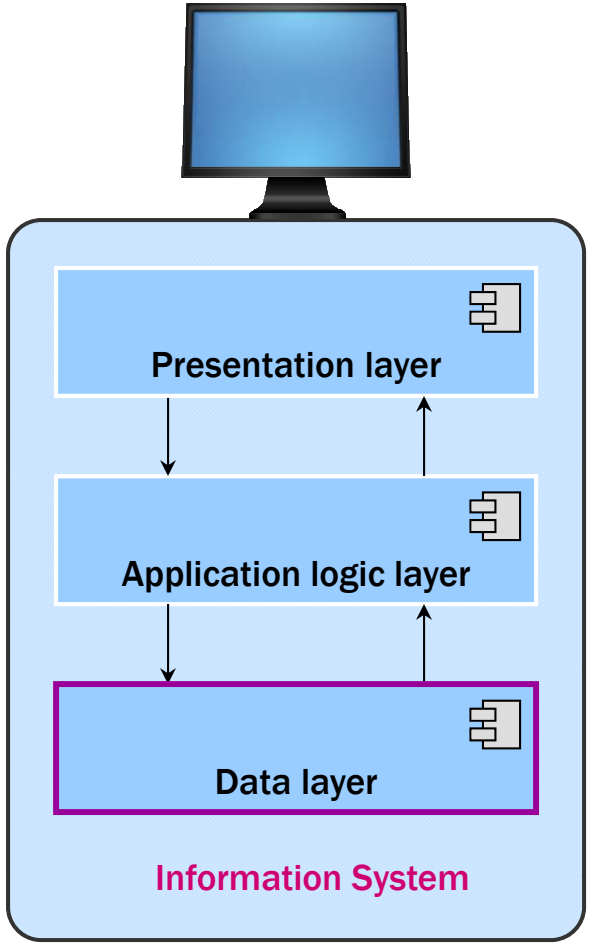
- Application logic layer
  - Offers the processing functionalities needed to derive the outputs that the system offers

    e.g. complex calculations, statistical analysis, sorting, …

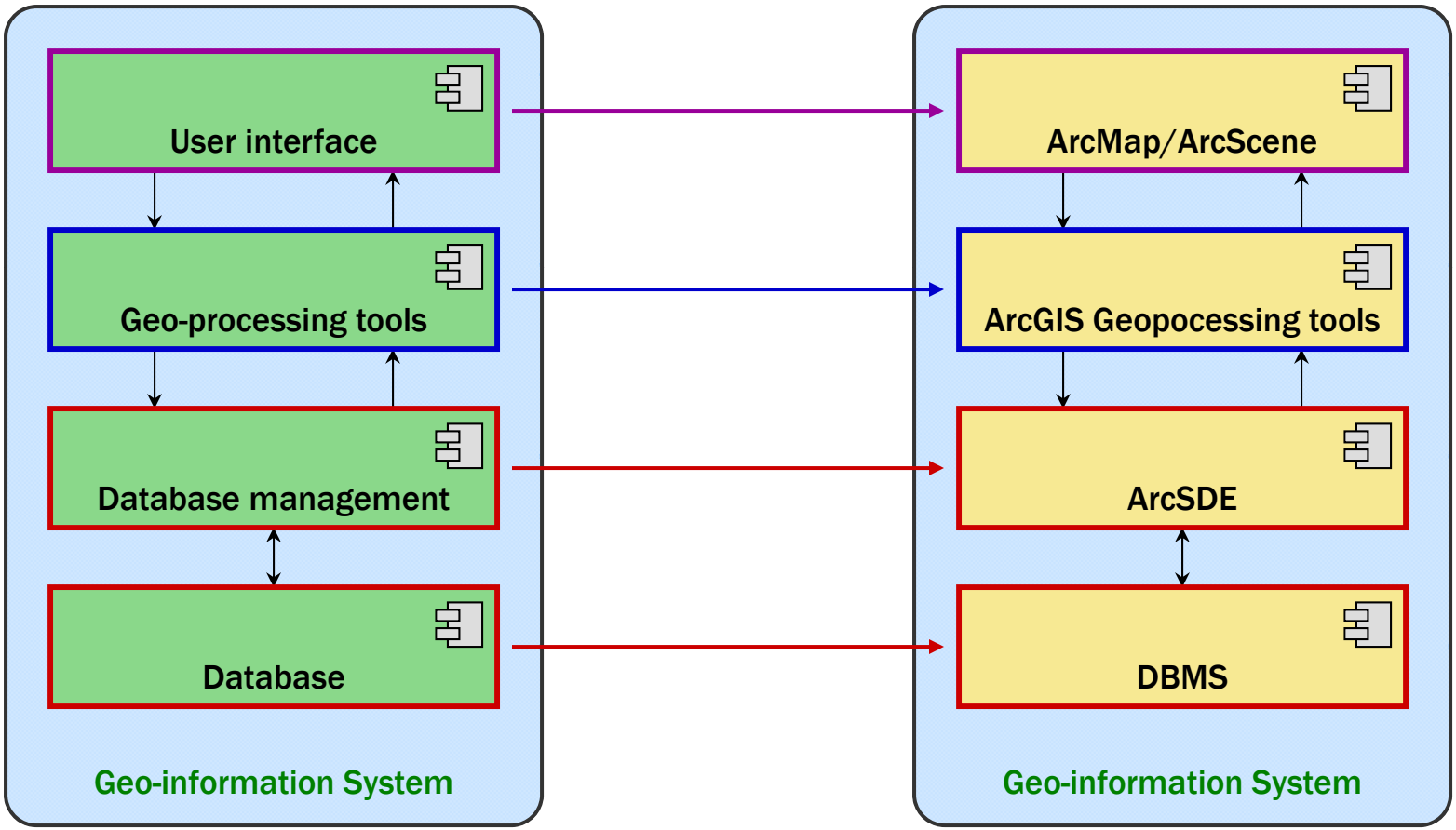


INFORMATION SYSTEM LAYERS

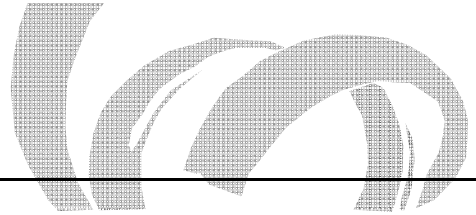
- Data layer
  - Manages and allows access to data sources



GI-APPLICATIONS



GI–ARCHITECTURES



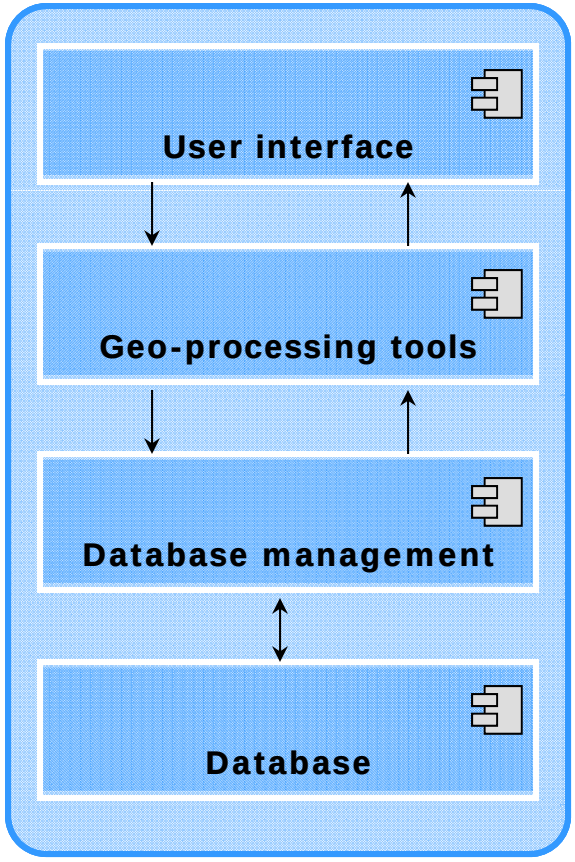
From the deployment perspective different architecture styles can be distinguished by combining the defined layers in various ways.

A single or combined layer is call “Tier”

- 1-Tier (monolithic)
- 2-Tier (client/server)
- 3-Tier
- n-Tier

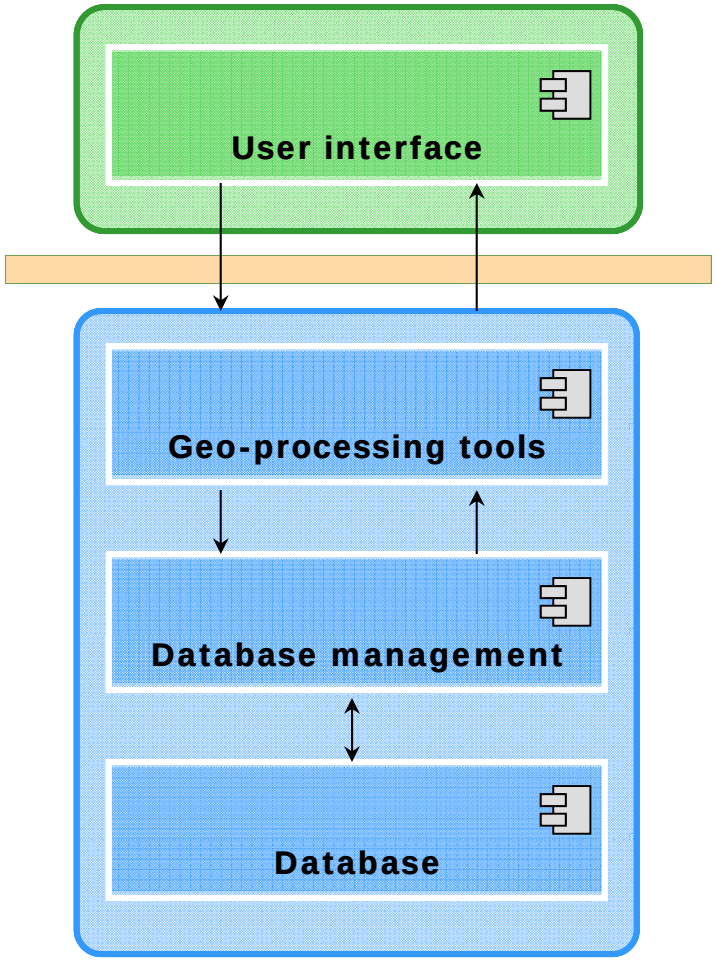
GI–ARCHITECTURES

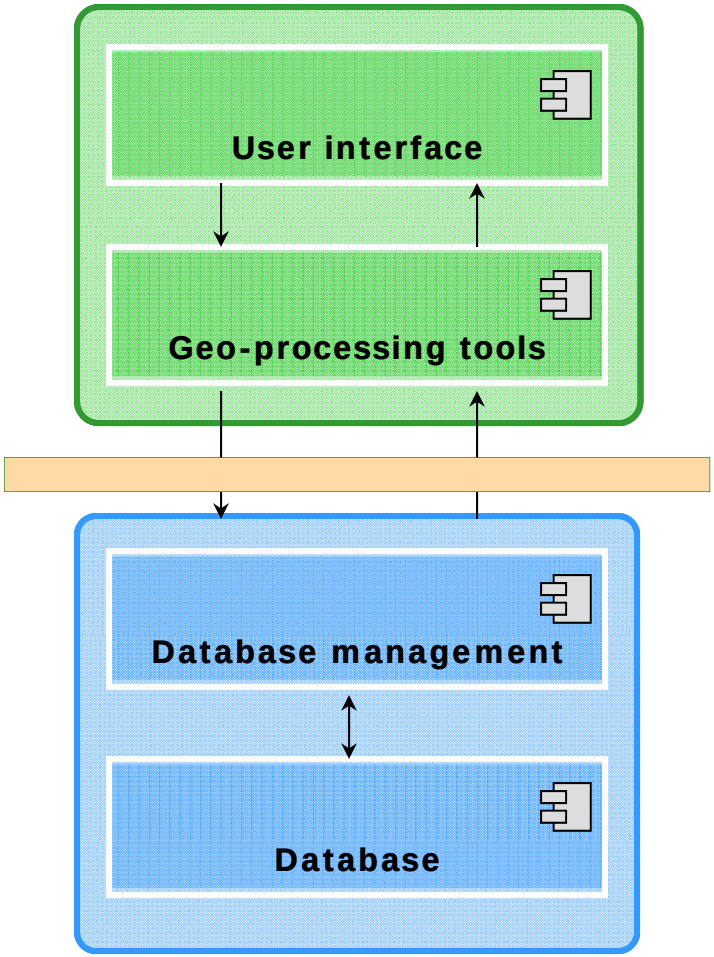
- 1-Tier (monolithic)



GI–ARCHITECTURES

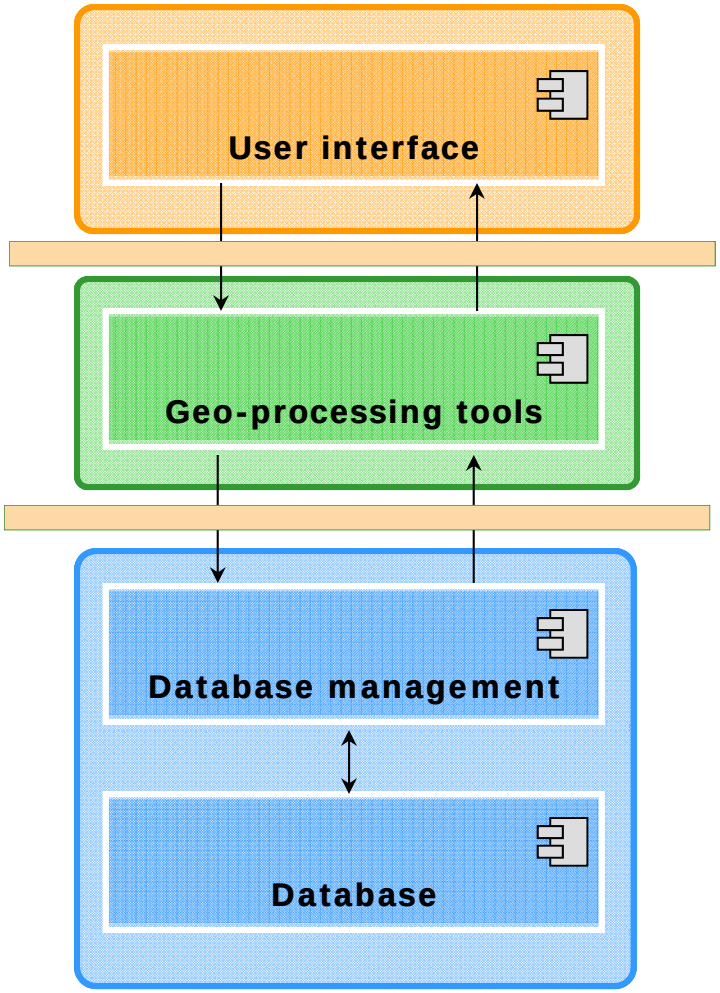
- 2-Tiers (client/server)





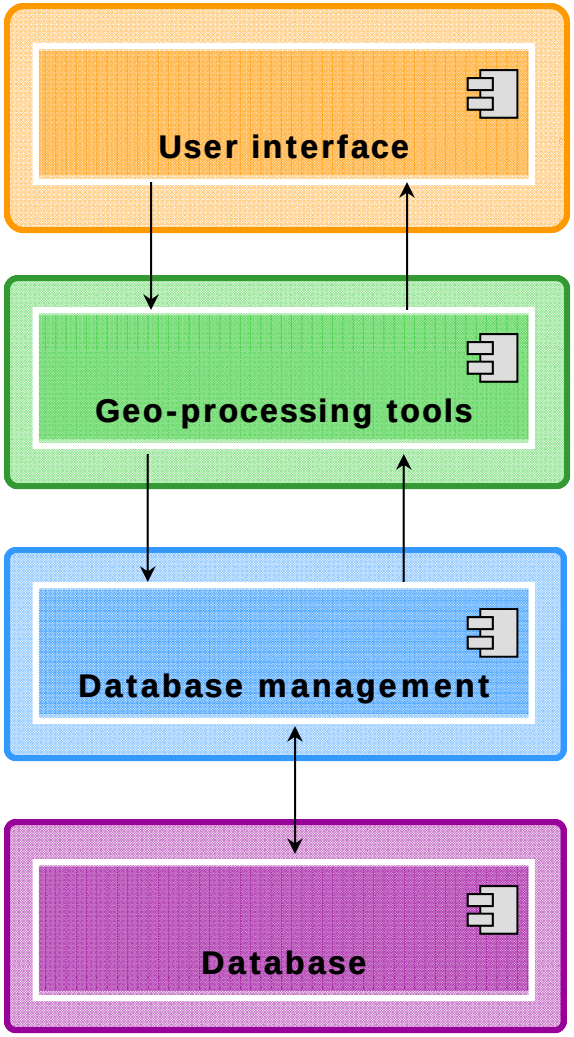
GI–ARCHITECTURES

- 3-Tiers



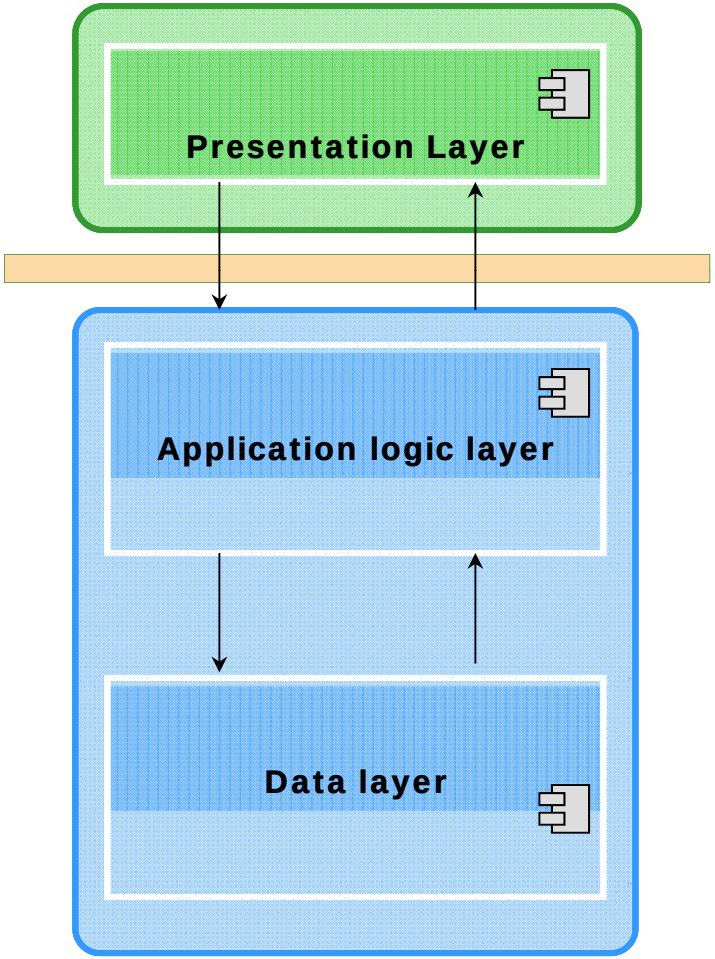
GI–ARCHITECTURES

- n-Tiers



CLIENT SERVER ARCHITECTURE

Web browser

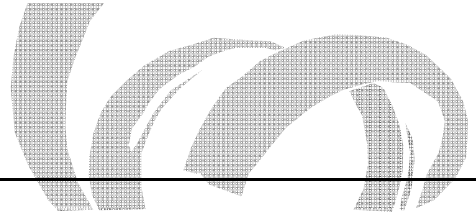


Internet

W3C standards
http, XML

Web Server

TERMINOLOGY



- Distributed GIS is geospatial technologies combined with distributed computing and the standards of Internet.

- Distributed computing is based on client-server architecture; Client – requests information, Server – responses

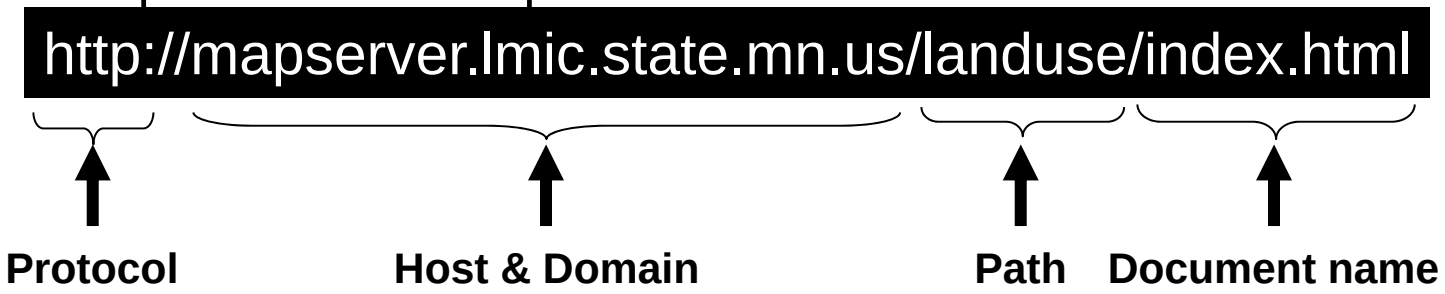
- Internet GIS/web-GIS
  - Internet refers to the network infrastructure.
  - Web-based GIS is one kind of Internet GIS
    - Based on HTTP protocol, which is one of many applications that runs on Internet (others: SMTP, FTP, Telnet…)

TRANSPORT OF MESSAGES

PROTOCOL

- Hypertext Transfer Protocol (http:// has few methods: GET,POST,etc..)
- An web-application is identified by its Uniform Resource Locator (URL)

Example-URL of a simple html-document:



HTTP REQUEST RESPONSE

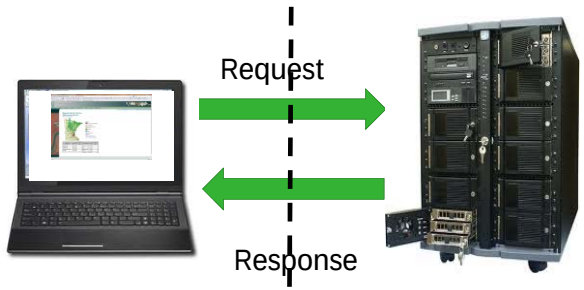
http://mapserver.lmic.state.mn.us/landuse/index.html

- Client request:

  GET /landuse/index.html HTTP/1.1

  Host: mapserver.lmic.state.mn.us

- Server response:
  - HTTP/1.1 200 OK
  - Date: Mon, 6 Dec 2010 09:38:34 GMT
  - Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
  - Content-Type: text/html; charset=UTF-8
  - ….



THIN AND THICK CLEINT

Depending on the location where the actual information processing takes place, two kinds of task distribution strategies can be distinguished:

1. **Thin Clients**: processing is done on the server side
2. **Thick (or Fat) Clients**: processing is done on the client side

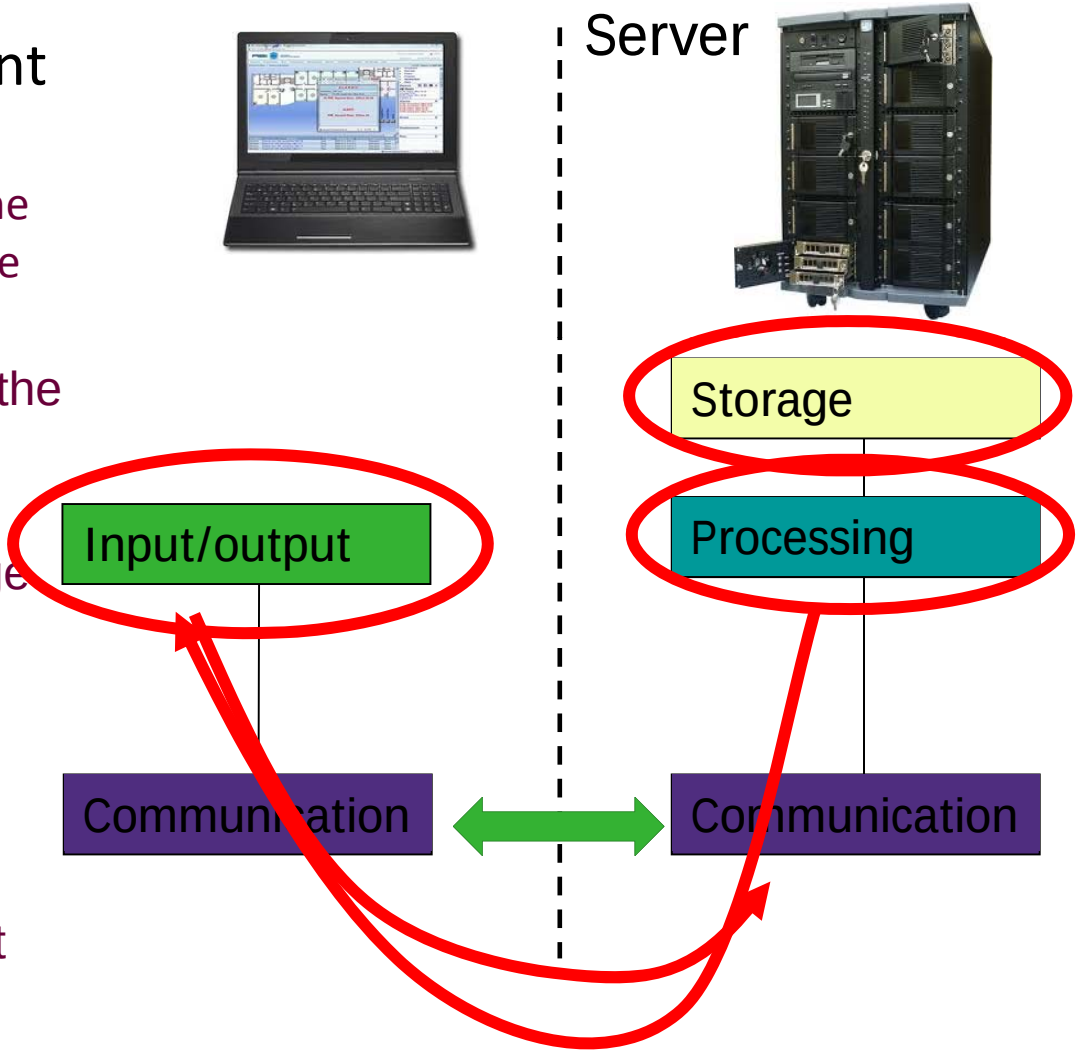
Example:

- Imagine a server/client application, that offers the user the functionality to request a rectified satellite image.
- How can this organized in a client/server environment?

CLIENT/SERVER TASK DISTRIBUTION

Thin client

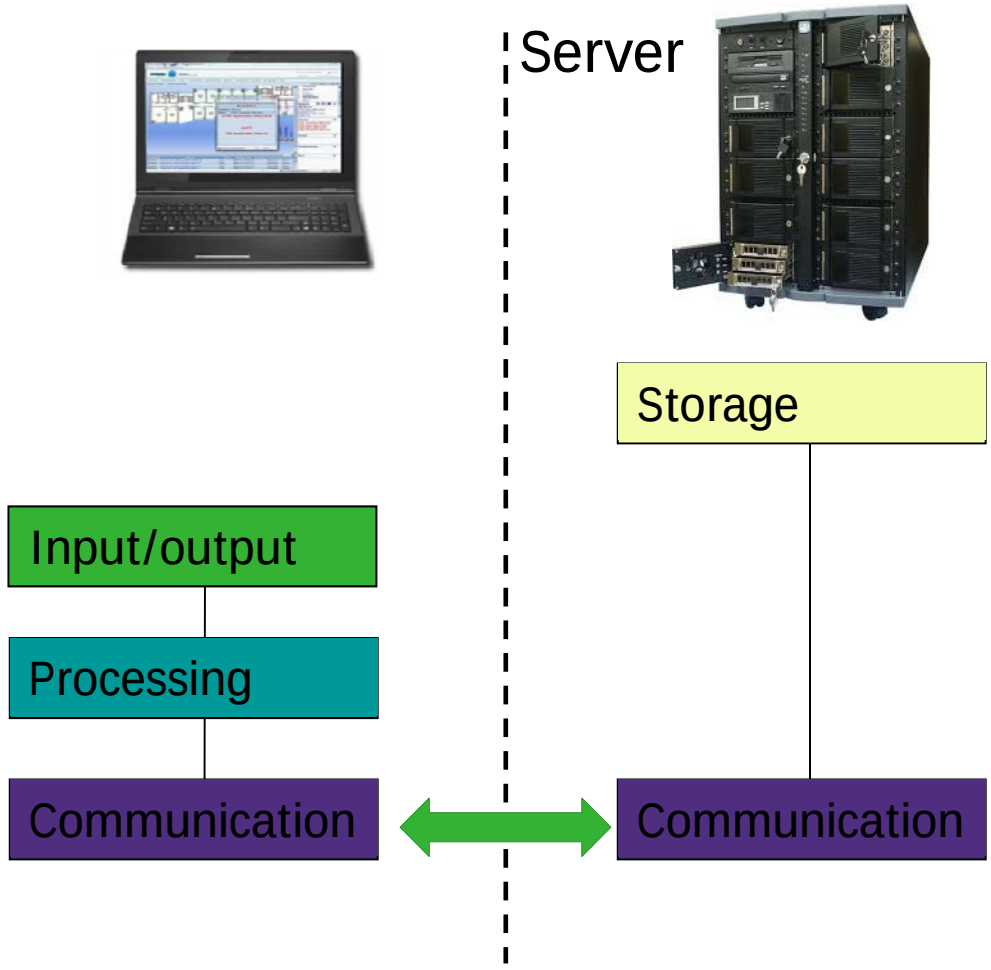
1. Client offers just a user interface to formulate the request (e.g. to select the image to be rectified)
2. The request is sent to the server
3. Requested data is retrieved from a storage device (here: an non-rectified image)
4. The retrieved data is processed (here: rectified)
5. Processed data is sent back to the client



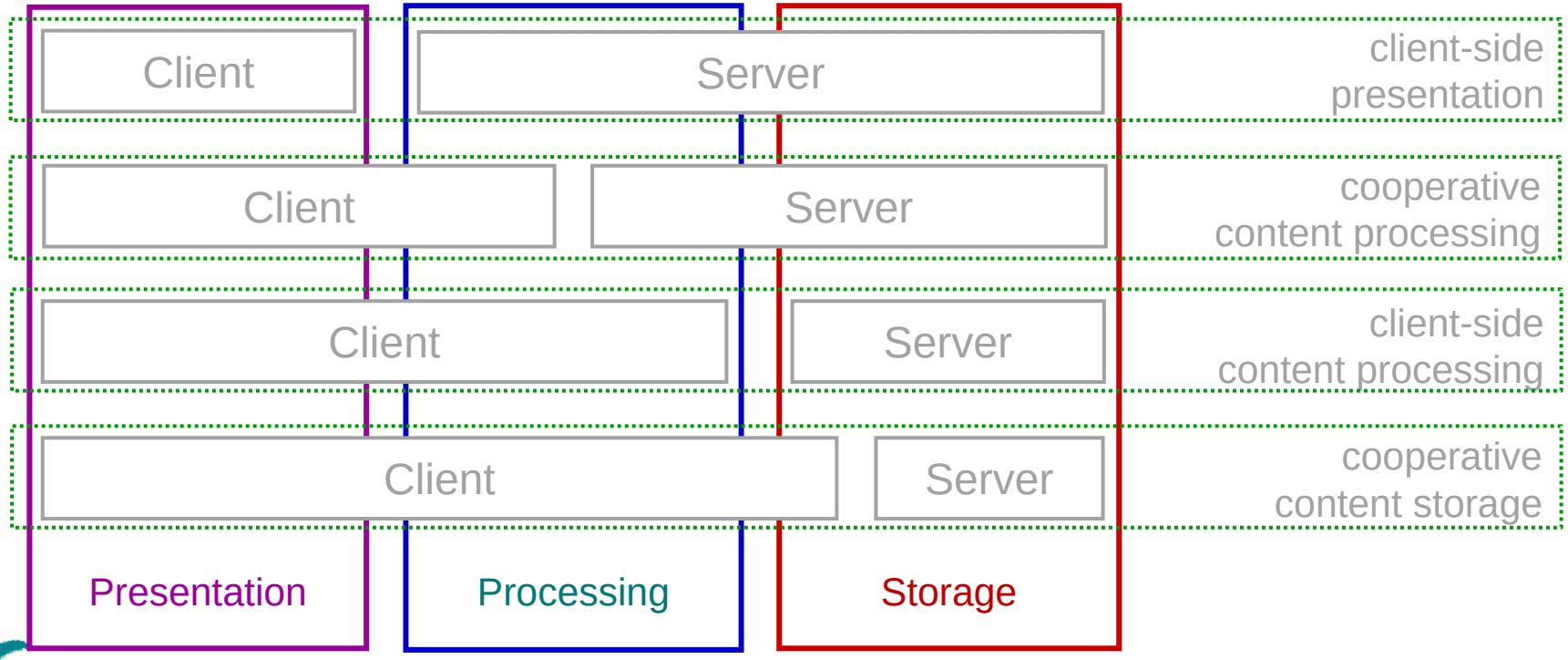
CLIENT/SERVER TASK DISTRIBUTION

Thick client

- Client provides processing capabilities itself (here: rectification would be performed at the client machine)
- Different from thin clients a fat client has to be equipped with more sophisticated software

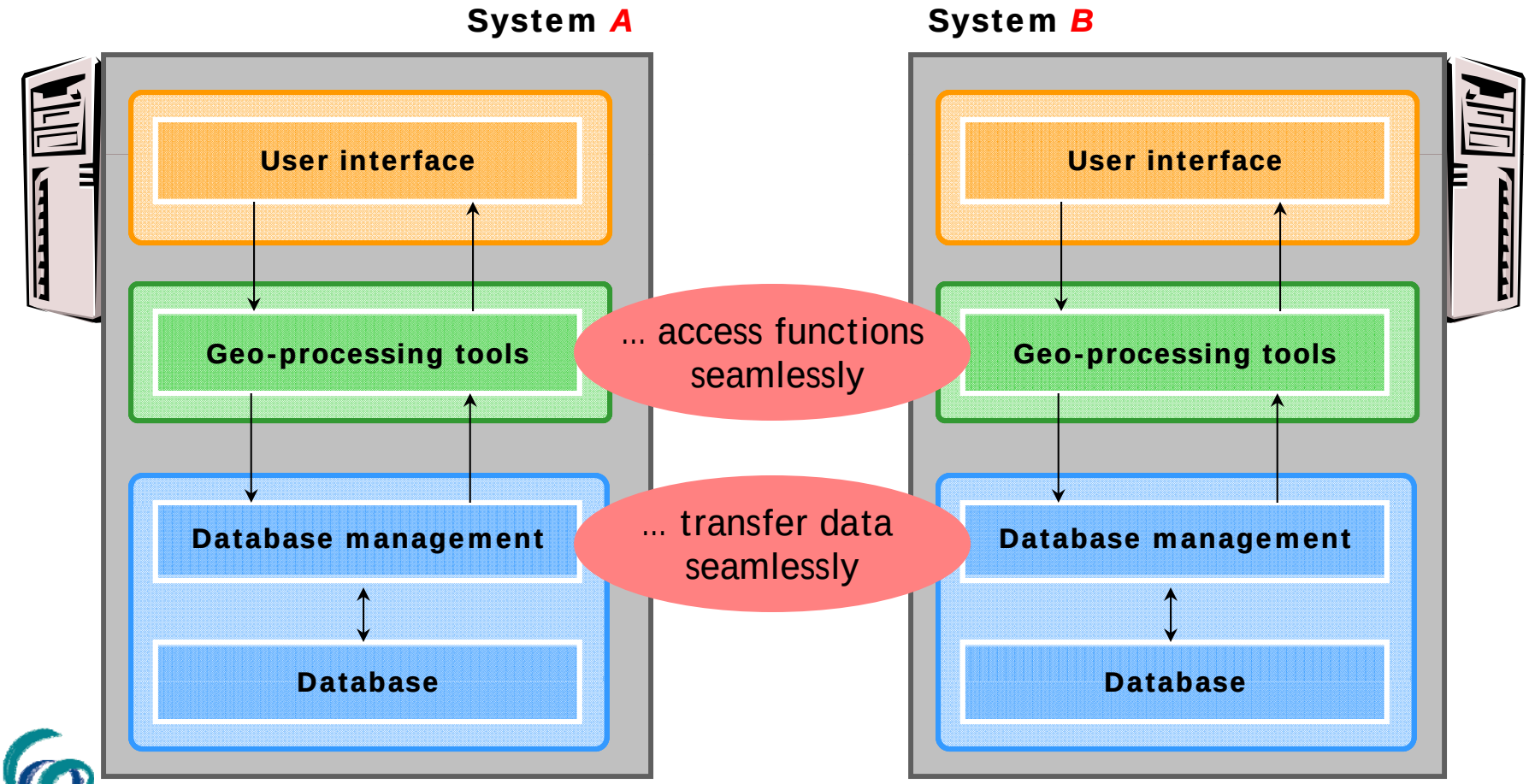


GI–ARCHITECTURES (*HOW?)*



INTEROPERABILITY

Two information systems are interoperable, if they are able to …



INTEROPERABILITY

BUT HOW TO …

1. make data seamlessly transferable & accessible?

   - Encode data in a standardized, platform & application independent manner

2. access distributed functionality seamlessly?

   - Specify and set up an infrastructure of interoperable (software) services, which encapsulate functionality and make it accessible via well specified interfaces

INTEROPERABILITY

BUT HOW TO …

1. make data seamlessly transferable & accessible?

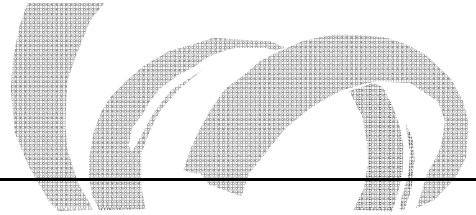
XML

2. access distributed functionality seamlessly?

Web Services

THE *EXTENSIBLE MARKUP LANGUAGE*

WHAT IS XML?



- (Markup-) language for the platform independent interchange of structured content on the Web.
- Provides a way to encode both structure and data
- Open Standard
  - Not defined by a single company
  - Supported by W3C (GML by OGC)
- Intended to store the structure and relationship of content in a readily parse-able format

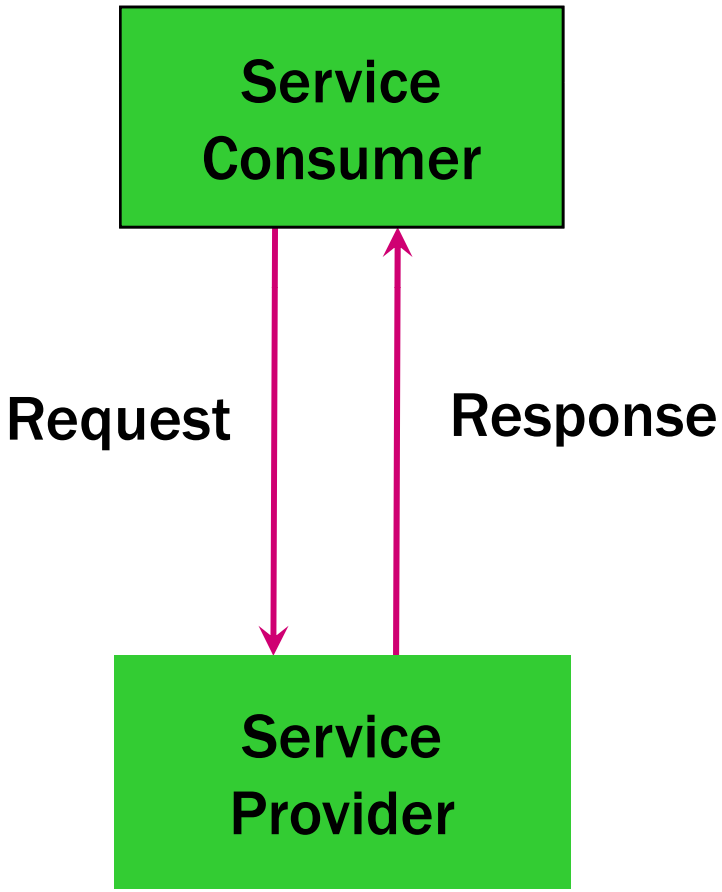
XML BASICS

```
<?xml version="1.0"?>
<WeatherForecast date="2/10/2007" unit="C">
   <city>
      <name>Frankfurt</name>
      <temperature>
         <min>07</min>
         <max>13</max>
      </temperature>
   </city>
   <city>
      <name>London</name>
      <temperature>
         <min>12</min>
         <max>17</max>
      </temperature>
   </city>
   <city>

   </city>
</WeatherForecast>
```

WEB SERVICE

- A service is a program that communicates by exchanging messages (XML over HTTP)



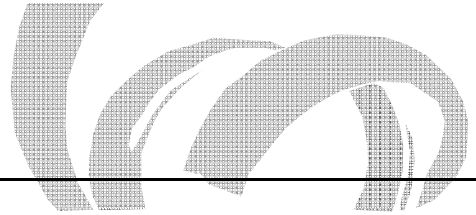
GEO-SERVICES

Some examples:

- A geo-name service
- A geo-referencing service
- A weather data service
- A route service
- A national atlas service

- Google maps / Google earth
- …

A SERVICE EXAMPLE: *GOOGLE MAPS*

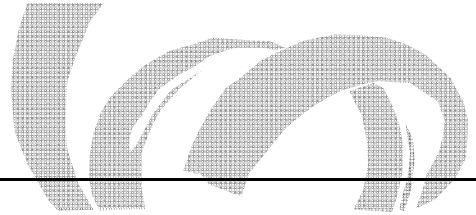


HTTP Request

- To access the Google Maps API geocoder we send a request to: http://maps.google.com/maps/geo? with the following parameters in the URI:
  - q -- The address that you want to geocode.
  - key -- Your API key.
  - output -- The format of the output. Options: xml, kml, csv, json

    http://maps.google.com/maps/geo?**q**=99+Hengelosestraat,+Enschede,+NL&**output**=xml&**key**=abcdefg.....

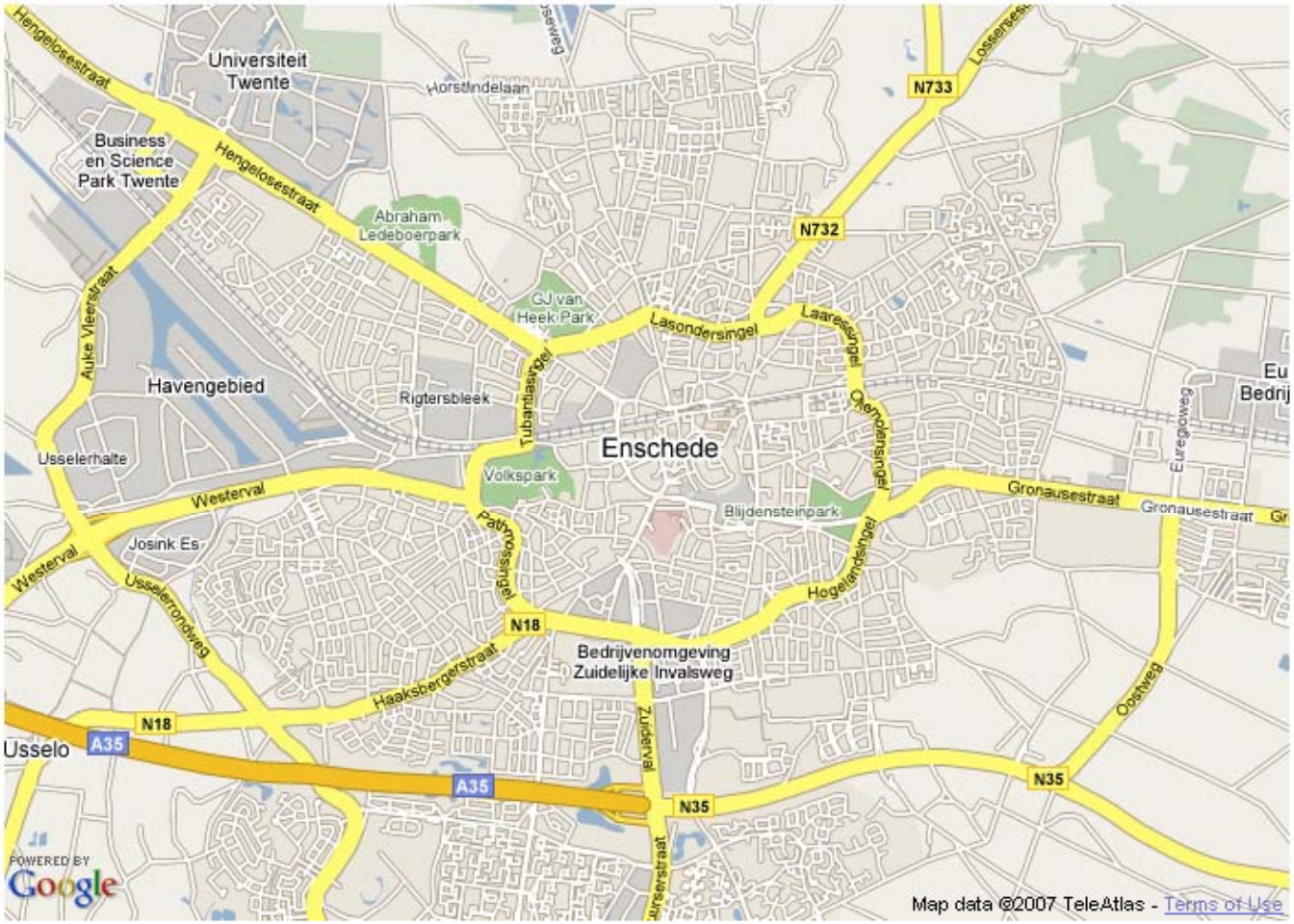
A SERVICE EXAMPLE: *GOOGLE MAPS*



```
<?xml version="1.0" encoding="UTF-8" ?>
- <kml xmlns="http://earth.google.com/kml/2.0">
  - <Response>
      <name>99 Hengelosestraat, Enschede, NL</name>
    - <Status>
        <code>200</code>
        <request>geocode</request>
      </Status>
    - <Placemark>
        <address>Hengelosestraat 99, 7514 Enschede, Enschede (Overijssel), Netherlands</address>
      - <AddressDetails Accuracy="8" xmlns="urn:oasis:names:tc:ciq:xsdschema:xAL:2.0">
        - <Country>
            <CountryNameCode>NL</CountryNameCode>
          - <AdministrativeArea>
              <AdministrativeAreaName>Overijssel</AdministrativeAreaName>
            - <Locality>
                <LocalityName>Enschede</LocalityName>
              - <DependentLocality>
                  <DependentLocalityName>Enschede</DependentLocalityName>
                - <Thoroughfare>
                    <ThoroughfareName>Hengelosestraat 99</ThoroughfareName>
                  </Thoroughfare>
                - <PostalCode>
                    <PostalCodeNumber>7514</PostalCodeNumber>
                  </PostalCode>
                </DependentLocality>
              </Locality>
            </AdministrativeArea>
          </Country>
        </AddressDetails>
      - <Point>
          <coordinates>6.890647,52.223300,0</coordinates>
        </Point>
      </Placemark>
    </Response>
</kml>
```

HTTP Response

A SERVICE EXAMPLE: *GOOGLE MAPS*

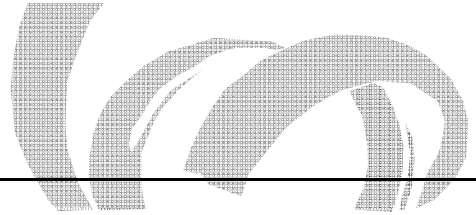


WHAT IS THE OGC?

- Open Geospatial Consortium, Inc.® (OGC)
  - Not-for-profit, international voluntary consensus standards organization founded in 1994
  - 260+ industry, government, and university members

OGC®
Making location count.
www.opengeospatial.org

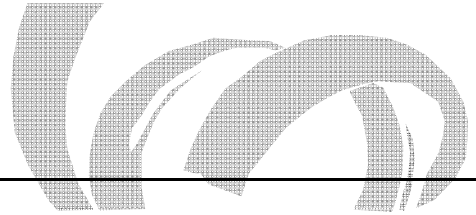
BASE STANDARDS SET



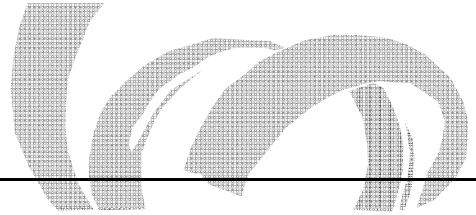
- Catalog services
  - Catalogue Service (CS-W)
  - Geo Digital Rights Management (GeoDRM)

- Resource Services
  - Web Map Service (WMS) → *presentation service*
  - Web Feature Service (WFS, WFS-T) → *data service*
  - Web Coverage Service (WCS) → *data service*
  - Web Processing service (WPS…)
  - Geo Processing Workflow (GPW…)
  - Sensor Web Enablement (SWE…)

PRESENTATION SERVICE



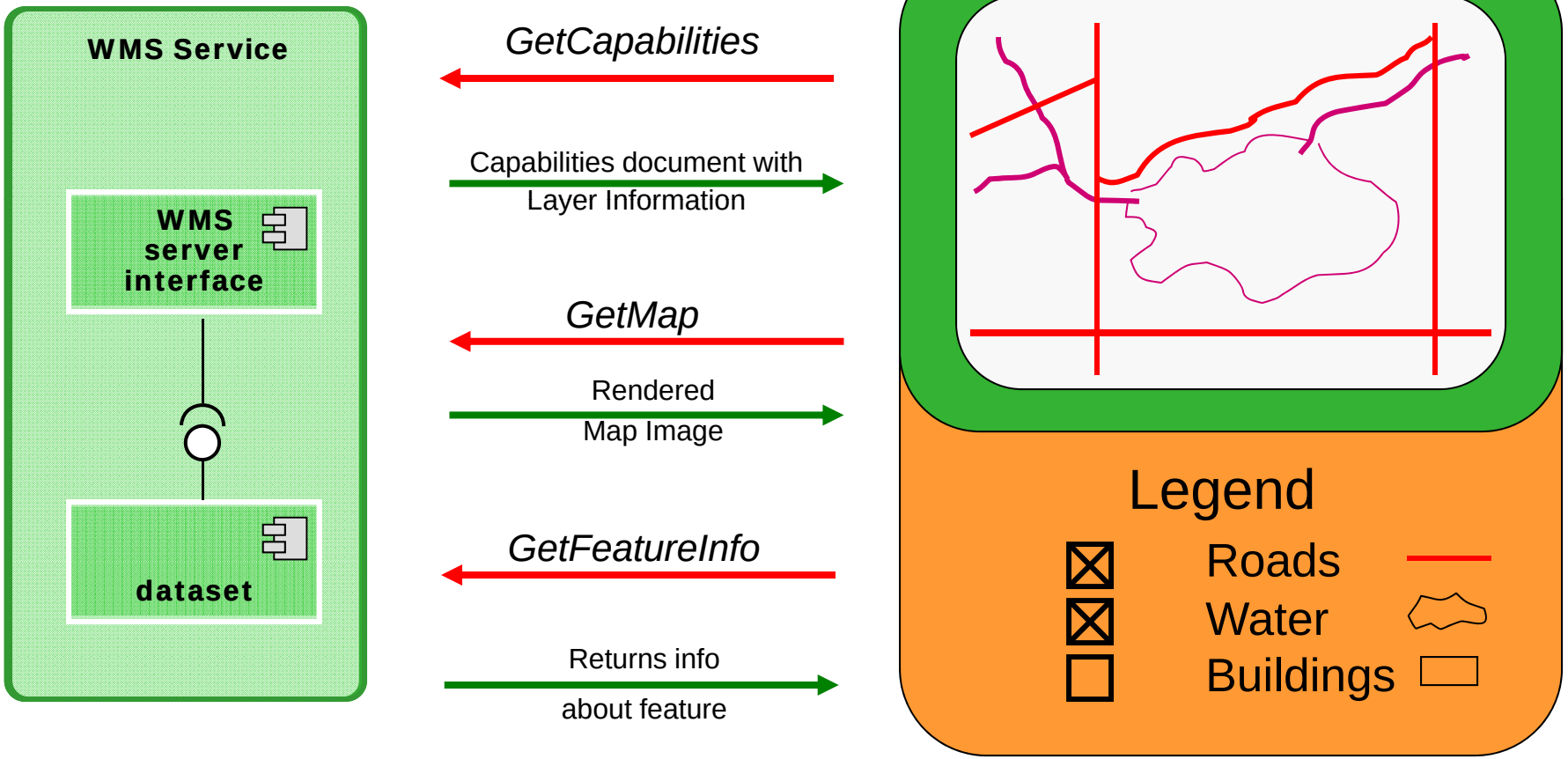
- Web Map Service (WMS)
  - Standardized interface for the creation of superimposed map-like views of geographic information
  - Cascadable, meaning that one WMS can act as a ‘gateway’ to other services
  - WMS is, as of today, the most mature and widest adopted OWS specification



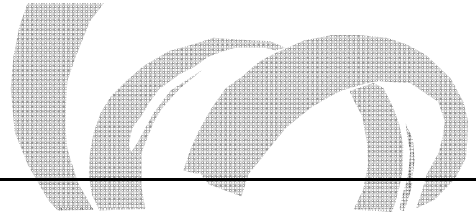
Operations

- GetCapabilities
  - Returns server-level metadata, description of services and content, acceptable request parameters
- GetMap
  - Returns map image whose geospatial and dimensional parameters are well-defined
- GetFeatureInfo
  - Returns information about particular features shown on map (optional)

WMS INTERFACE: *OPERATION*

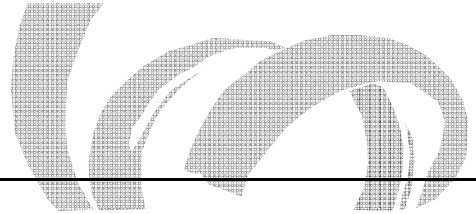


WEB MAP SERVICE (WMS)



- Parameters
  - Layers
  - Styles (possibly)
  - Bounding Box
  - Projection or geographic coordinate reference system
  - Desired output format
  - Output size (width, height)
  - Background transparency or color

SERVICE EXAMPLE: *WMS*

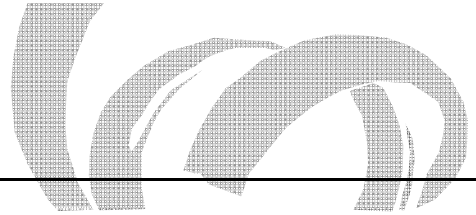


HTTP Request

- The following is the structure of a *getcapabilities* request
  - address -- http://geoserver.itc.nl/cgi-bin/mapserv.exe?
  - service = WMS
  - request = getcapabilities
  - version = 1.1.1
  - map = D:/Inetpub/geoserver/mapserver/config.map

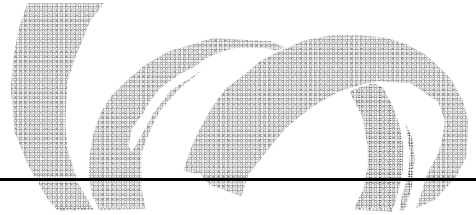
http://geoserver.itc.nl/cgi-bin/mapserv.exe?
map=D:/Inetpub/geoserver/mapserver/config.map&
SERVICE=WMS&
VERSION=1.1.1&
REQUEST=GetCapabilities

WEB FEATURE SERVICE (WFS)



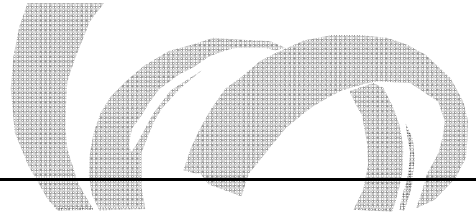
- Standardized interface specification for accessing vector spatial data
  - The database used to store the features is opaque to client applications and their only view of the data is through the WFS interface
- WFS output is encoded in GML
- WFS uses a standardized query language *(filter encoding specification)*

WEB FEATURE SERVICE (WFS)



- WFS Operations:
  - GetCapabilities
  - DescribeFeature Type
  - GetFeature
- WFS-T
  - Insert a Feature
  - Update a Feature
  - Delete a Feature
  - Create a new feature

WFS EXAMPLE REQUESTS



```
WFS DescribeFeatureType request:

--> http://sigma.openplans.org/geoserver/ows?
      SERVICE=wfs&
      REQUEST=DescribeFeatureType&
      VERSION=1.0.0&
      TYPENAME=topp:countries
```

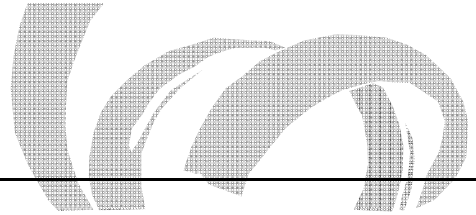
```
WFS GetFeature requests:

--> http://sigma.openplans.org/geoserver/ows?
      SERVICE=wfs&
      REQUEST=GetFeature&
      VERSION=1.0.0&
      TYPENAME=countries


--> http://sigma.openplans.org/geoserver/ows?
      SERVICE=wfs&
      REQUEST=GetFeature&
      VERSION=1.0.0&
      TYPENAME=countries&
      PROPERTYNAME=country_name&
      MAXFEATURES=5


--> http://sigma.openplans.org/geoserver/ows?
      SERVICE=wfs&
      REQUEST=GetFeature&
      VERSION=1.0.0&
      FEATUREID=countries.147709
```

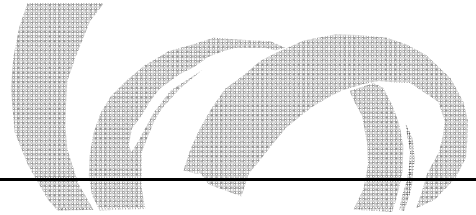
WEB PROCESSING SERVICE (WPS)



- Standardized interface that provides client access to pre-programmed calculations and/or computation models that operate on spatially referenced data.
- The data required by the service can be delivered across a network, or available at the server.
- The calculation can be simple as, e.g.:
  - Determining the difference in influenza cases between two different seasons,

  or complex as, e.g.:
  - A global climate change model.

WEB PROCESSING SERVICE (WPS)



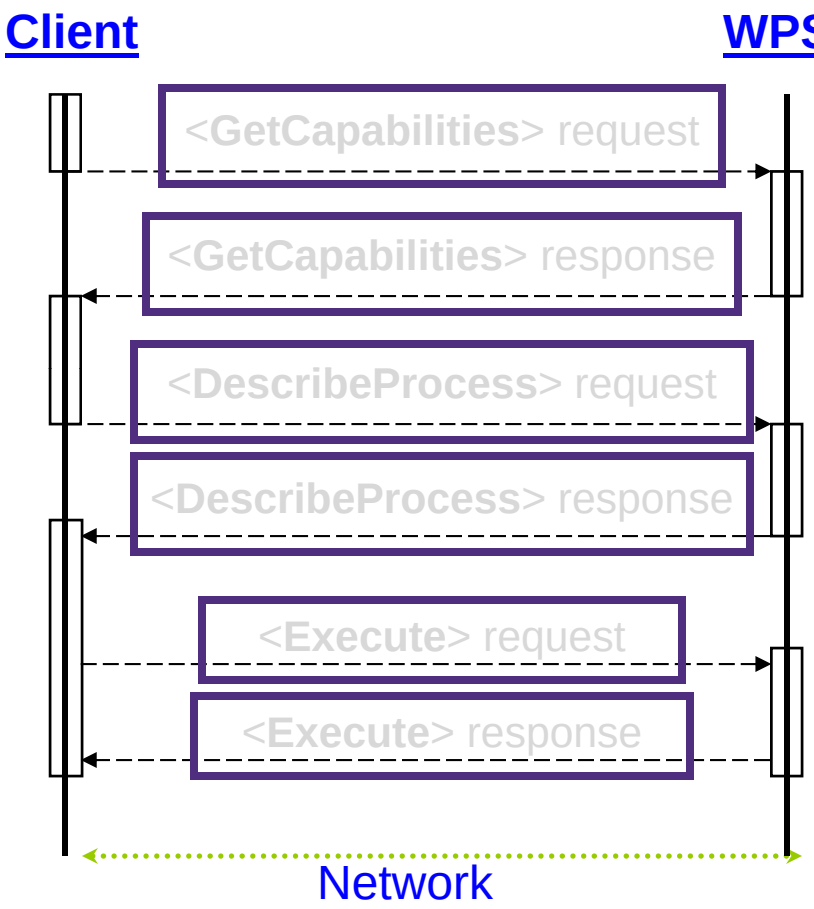
Operations

- GetCapabilities: allows requests for service metadata documents that describe the abilities of the specific server
  - Service description
  - Access description
  - Brief process descriptions

- DescribeProcess: allows requests for detailed information about one or more process(es)
  - Full process description

- Execute: allows a client to run a specified process
  - Process execution

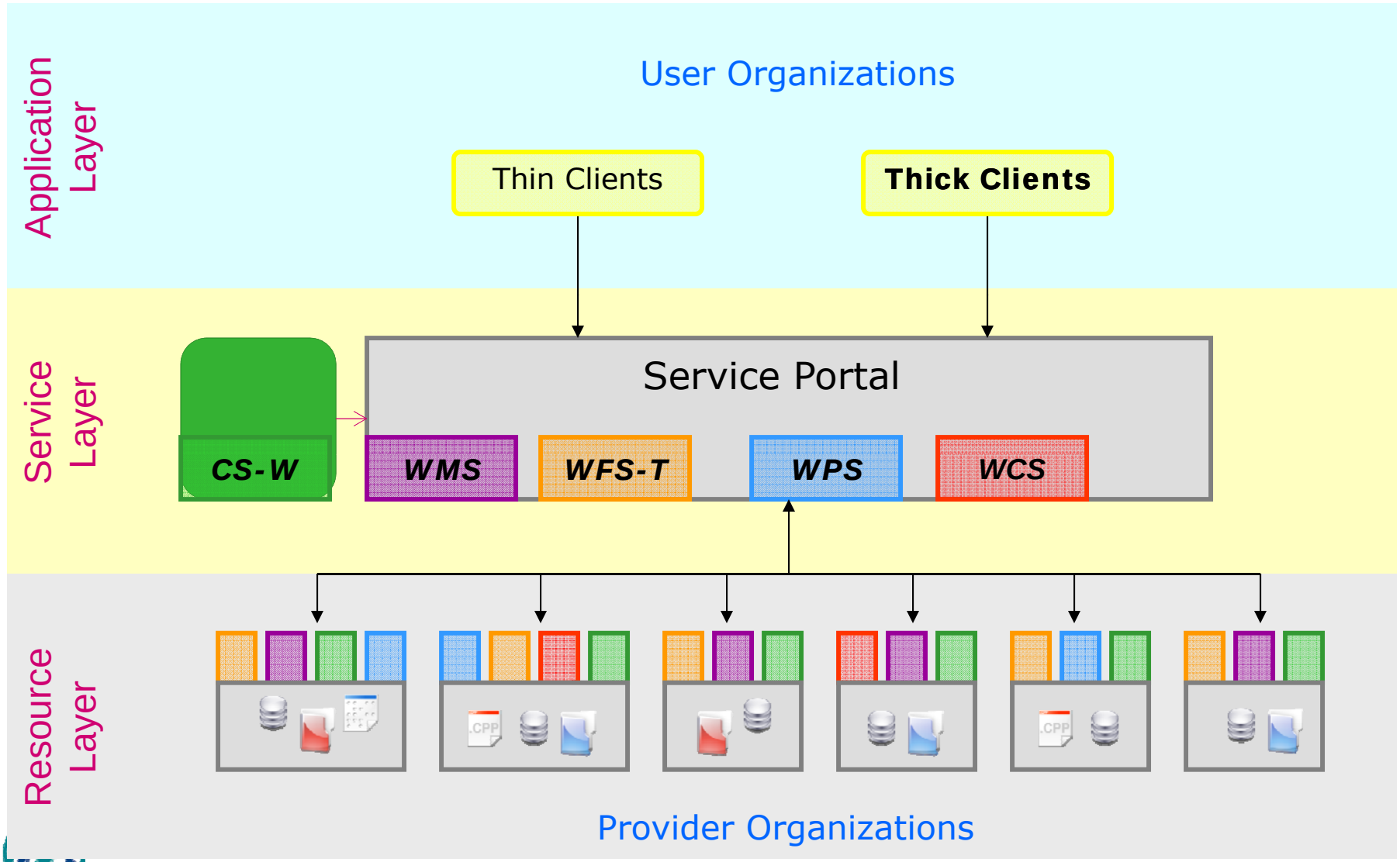
WPS IN ACTION - BUFFERING



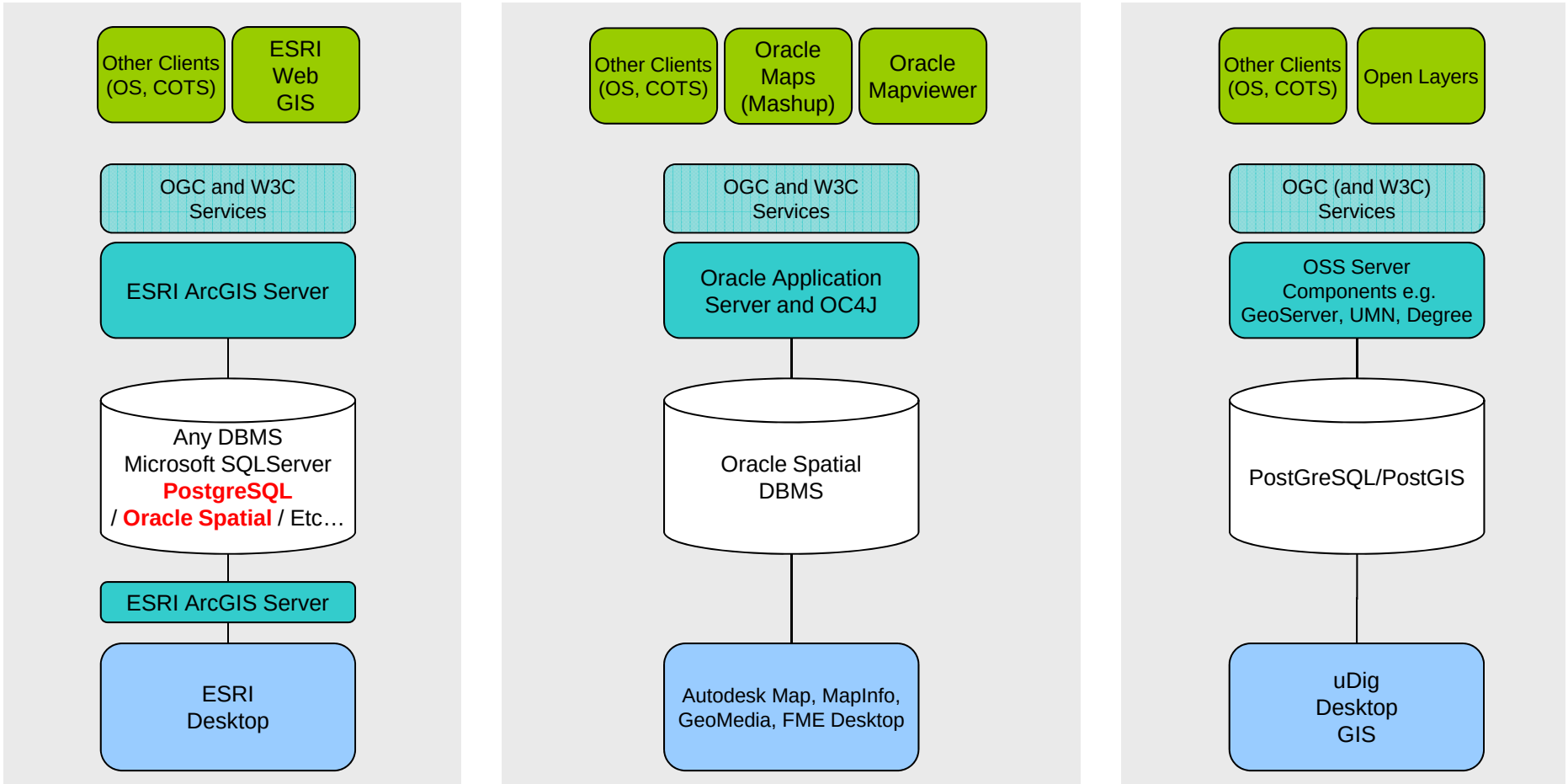

```
<ProcessDescriptions>
<ows:Identifier>Buffer</ows:Identifier> …
  <DataInputs>
  <Input>
    <ows:Identifier>InputGeometry</ows:Identifier>
    <ComplexData defaultSchema="schemaNS"/>
    <MinimumOccurs>1</MinimumOccurs>
  </Input>
  <Input>
    <ows:Identifier>BufferDistance</ows:Identifier> …
  <LiteralValue />
  <ows:DataType ows:reference="xs:int"/> …
  <ProcessOutputs>
   <Output>
     <ows:Identifier>BufferResult</ows:Identifier>
     <ComplexOutput defaultSchema="schemaNS"/> …
</ProcessDescriptions>
 </Capabilities>
```

Based on: P. Schut's presentation at the OGC TC Meeting in Huntsville 2006: *"WPS RFC responses"*

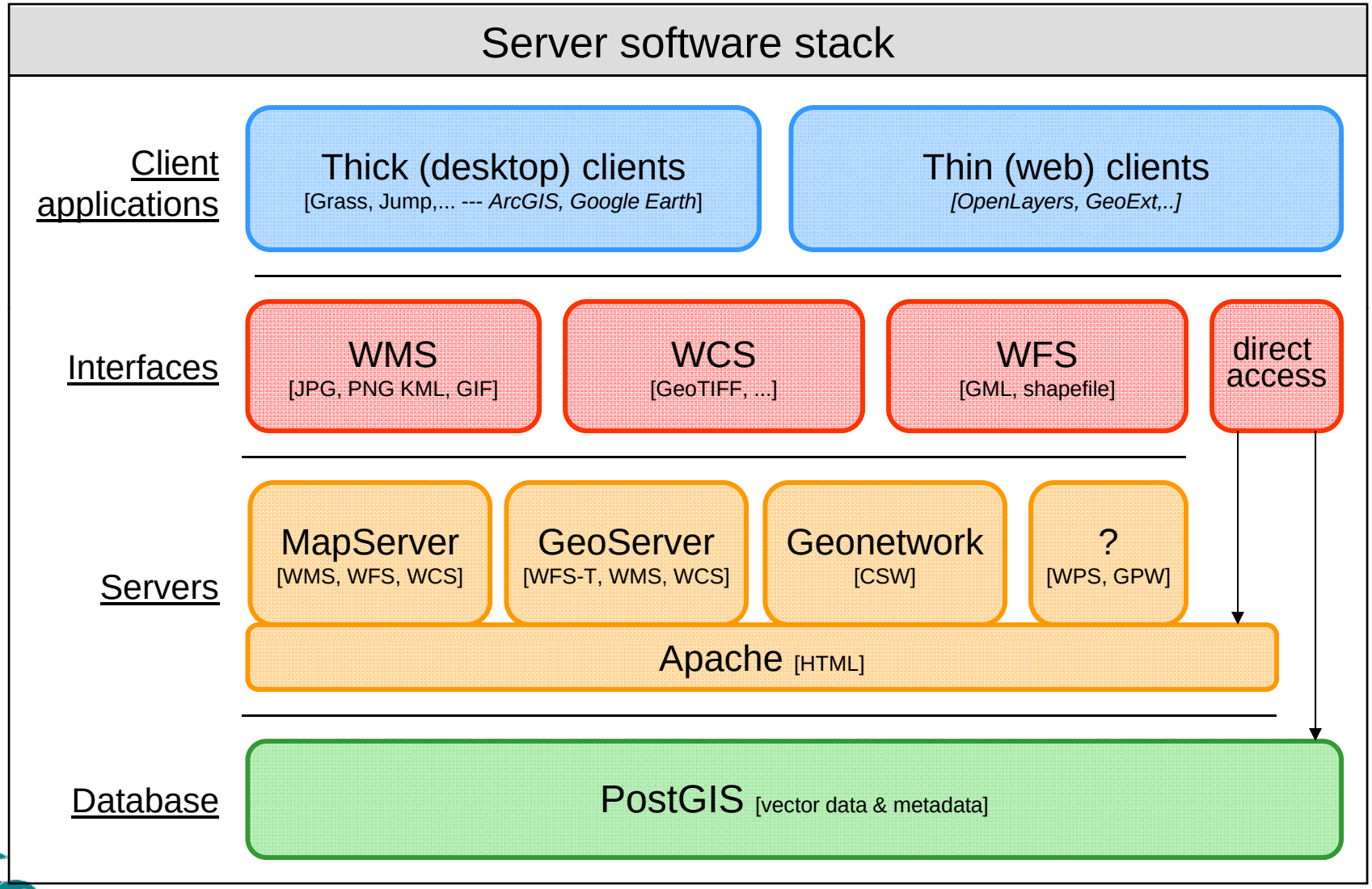
GEO-SERVICES EXTENDED ARCHITECTURE



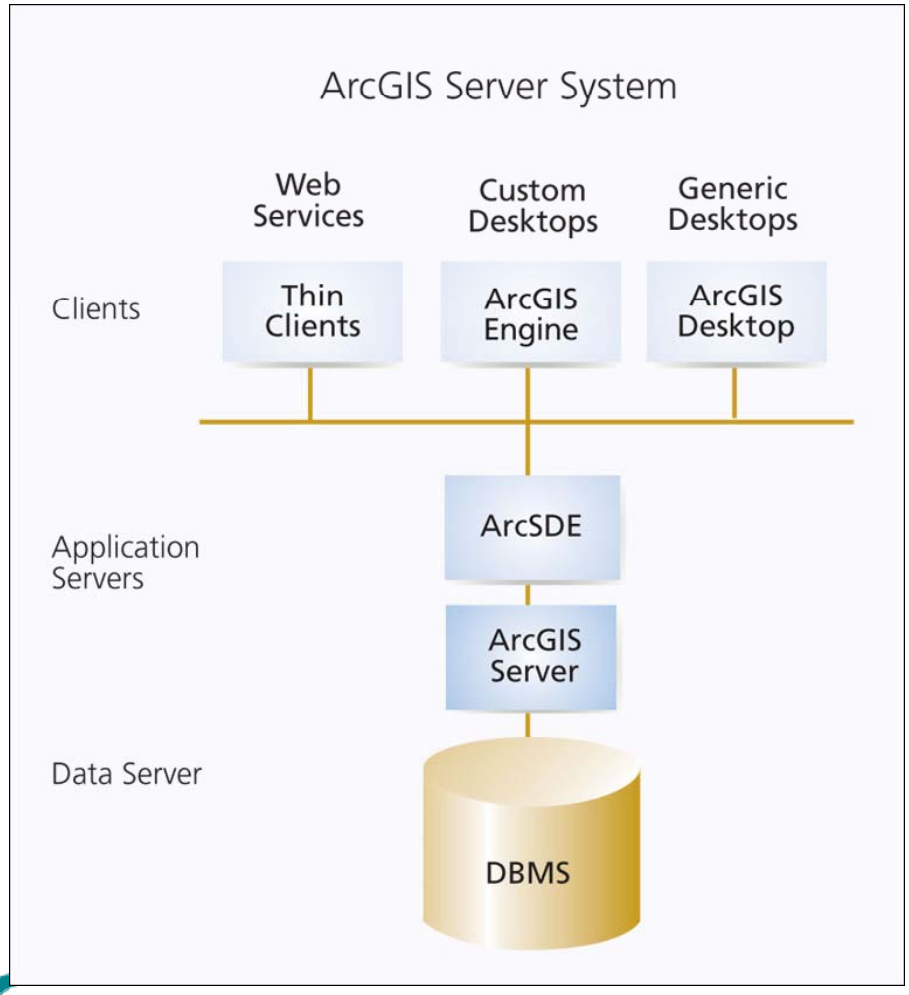
SOFTWARE COMPONENTS



SDI ARCHITECTURES BASED ON OSS



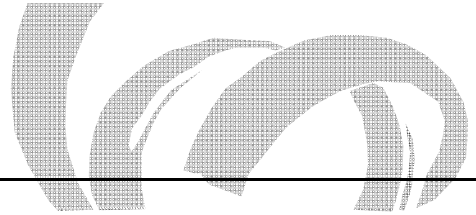
SDI ARCHITECTURES BASED ON ESRI STACK



- OGC publishing capabilities in ArcGIS Server:

	WMS	**WFS**	**WCS**
Map services	X	X	X
Geodata services		X	X
Image services	X		X

SUMMARY



In this short time you have been exposed to:

- Distributed GIS systems
- Internet GIS/web-GIS
- HTTP, XML
- Client-Server, thin and thick clients, web services
- OGC web services
  - WMS, WFS(-T), WCS, WPS…
- Software architectures that allow implementation of those services