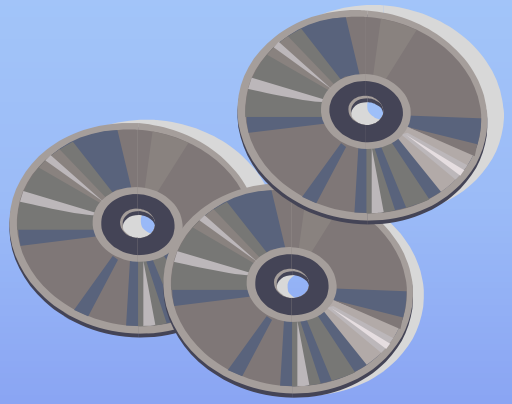
Interoperability, Geospatial Web Services, and Future Direction: Distributed GIServices (Peer-to-Peer)

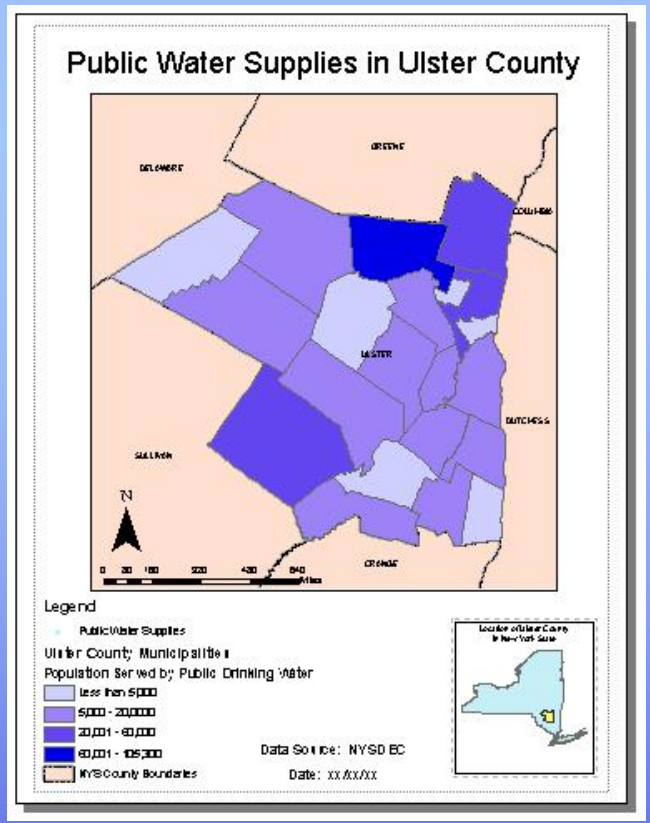
According to Dr. Ming-Hsiang Tsou

Interoperability and GIS Web Services

Data and Information

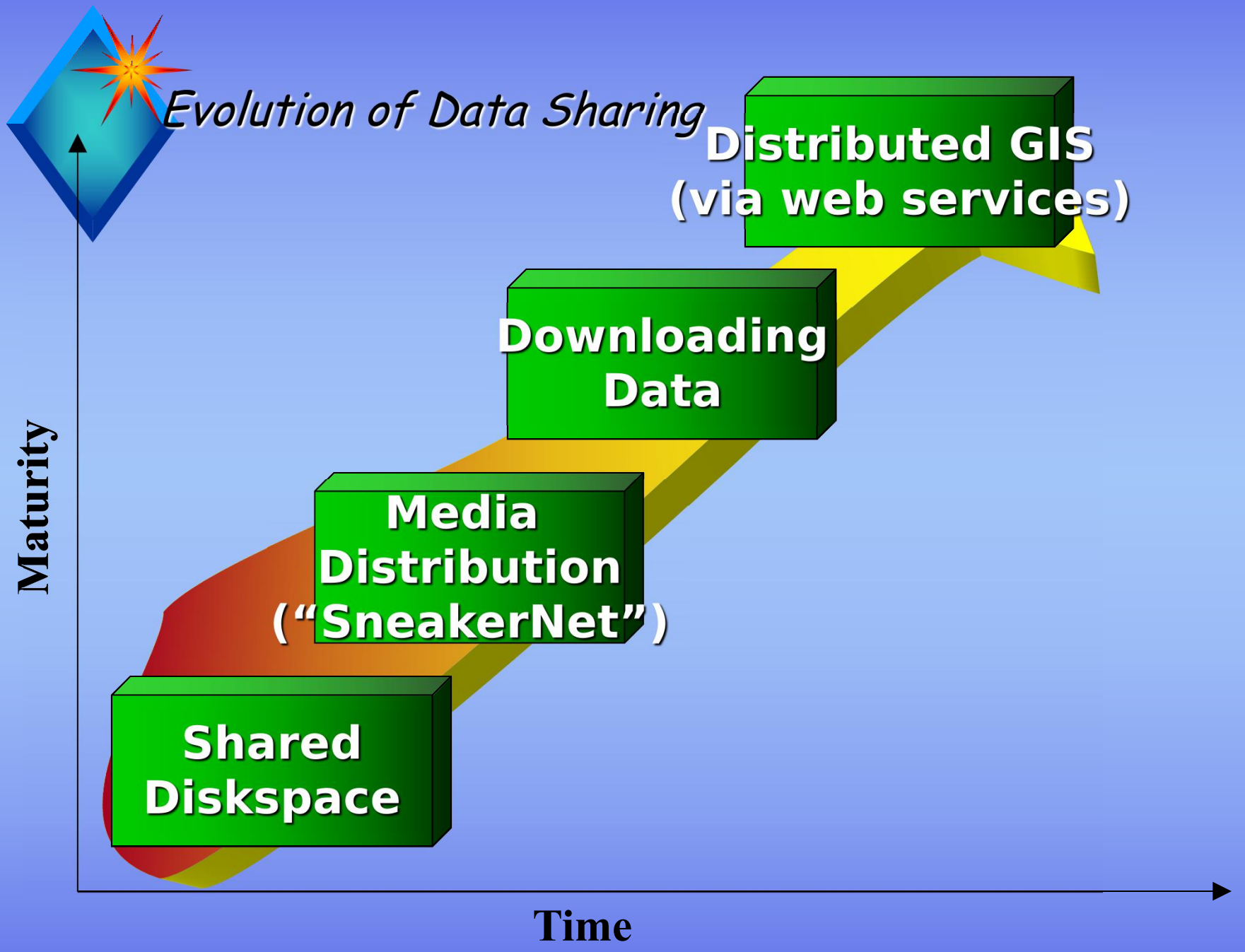
Data ≠ Information





Importance of Geographic Data

- GIS is different than most applications
- Word processing, Spreadsheets… are entirely dependent on input of YOUR data
- GIS users nearly always require reference map data (e.g., streets, boundaries) that are maintained by others
- As GIS users, we rely heavily on "external" data sources


Evolution of Data Sharing
Distributed GIS
(via web services)
Downloading
Data
Media
Distribution
("SneakerNet")
Shared
Diskspace
Maturity
Time

Issues with Previous Data Sharing Methods

- **Vintage**
- **Format**
- **Coordinate System/Projection**
- **Media Compatibility**
- **File Size**
- **Bandwidth**
- **Metadata Distribution**

Typically requires data manipulation prior to use (reformatting, reprojecting, clipping, etc.)

Distributed GIS Model

USGS

NYS DEC

NYS CSCIC

US EPA

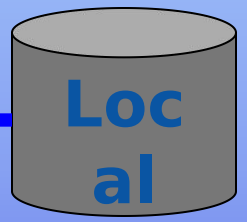
DEM

Wetlands

Orthos

Haz Sites

Real-time access to geographic data and



Local Data

Virtually Any Client

Mashup

- An application that combines data from multiple sources
- Referred to as "content aggregation"
- Combines similar types of data (e.g., maps,) from different systems/services
- Term originated from music industry when a new song was made from several existing tracks

Distributed GIS Advantages

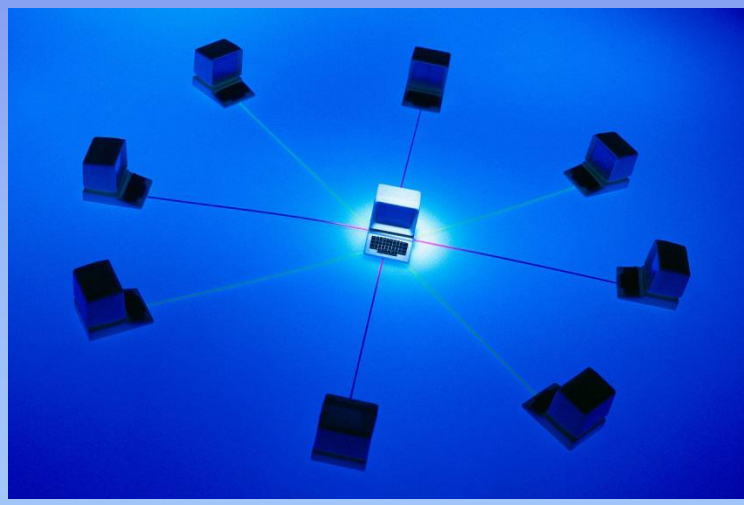
- Guarantees latest data vintage
- Format is irrelevant
- No media involved
- Size is irrelevant (only requested data needed)
- Software independent (AV, MI)
- Device independent (PC, phone)
- Saves Time!

Distributed GIS Disadvantages

- Dependent on server availability
- Dependent on internet availability
- Users need to be aware of web service
- Can be difficult to find

Service Oriented Architecture

- Architecture that is based on integrating "loosely coupled", interoperable services
- Loosely coupled means NOT physically bound or compiled like an EXE or DLL
- These services can be invoked and consumed remotely over a network
- Data and messages are exchanged between clients and services



What is a Web Service?

- A web service is an application service that provides one or more functions that can be remotely requested
- Each function within the service performs a specific task (e.g., generate a map)
- Clients remotely invoke a function on a web server
- Input parameters and data output are passed back and forth in XML format
- XML is exchanged using the Simple Object Access Protocol (SOAP)
- Independent of programming language and operating system

XML - eXtensible Markup Language

- XML is a tagged markup language like HTML, but is general purpose (users define tags)
- Designed to simplify storing and exchanging data
- Documents are in text with opening and closing tags surrounding data
- The rules with which an XML document must conform is referred to as a schema
- Schemas are stored as XML in a XML Schema Definition file (.XSD)

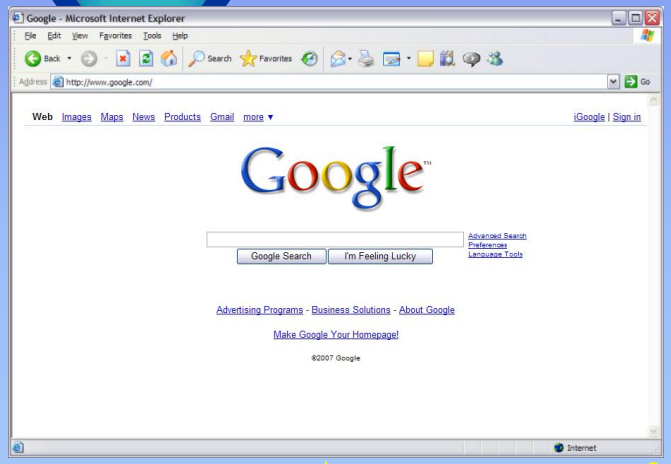
```
<person>

<gender>male</gender>
   <age>29</age>

<firstname>John</firstna
me>

<lastname>Doe</lastnam
e>
</person>
```

Gender	Age	First Name	Last Name
Male	29	John	Doe

Simple Object Access Protocol (SOAP)

- XML based protocol that allows applications to call web service functions, pass input parameters and receive results
- This is the protocol that is used with web services

Basic Web Page Operation



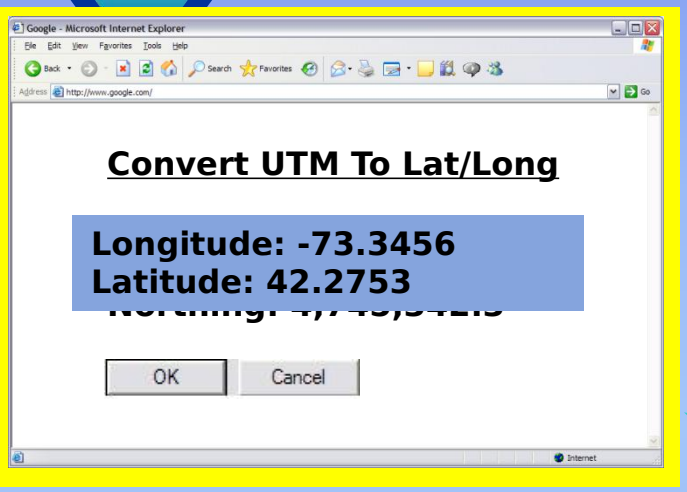
Web Server (Listening for requests)

Request

Response (HTML document)

Web Service Operation

Client



Web Server
(Listening for requests)

1. Creates XML Request

2. Request sent to web service

3. Receives request and parses XML

4. Calls the function

5. Creates response XML with results

6. Response sent to client

7. Receives response and parses

Sample Web Service Request

```
POST /TerraService2.asmx HTTP/1.1
Host: terraserver.microsoft.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://terraservice-usa.com/ConvertUtmPtToLonLatPt"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ConvertUtmPtToLonLatPt xmlns="http://terraservice-usa.com/">
            <utm>
                <Zone>18</Zone>
                <X>623456.3</X>
                <Y>4745342.3</Y>
            </utm>
        </ConvertUtmPtToLonLatPt>
    </soap:Body>
</soap:Envelope>
```

Sample Web Service Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ConvertUtmPtToLonLatPtResponse xmlns="http://terraservice-
    usa.com/">
         <ConvertUtmPtToLonLatPtResult>
            <Lon>-73.3456</Lon>
            <Lat>42.2753</Lat>
         </ConvertUtmPtToLonLatPtResult>
      </ConvertUtmPtToLonLatPtResponse>
   </soap:Body>
</soap:Envelope>
```

Web Services Description Language (WSDL)

- An XML document that publishes a list of the functions available within a web service as well as their definitions

```
<s:element name="ConvertUtmPtToLonLatPt">
   <s:complexType>
      <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name=“zone"
type=“int" />
         <s:element minOccurs="1" maxOccurs="1" name=“x"
type=“float" />
         <s:element minOccurs="1" maxOccurs="1" name=“y"
type=“float" />
       </s:sequence>
    </s:complexType>
</s:element>
<s:element name="ConvertUtmPtToLonLatPtResponse">
```

Types of Geospatial Web Services

- Map/Data Oriented Services
  - Map Services (Image)
  - Feature Services (Vector)
  - Coverage Services (Grid)

- Task Oriented Services
  - Geocoding Services
  - Routing Services
  - Geoprocessing Services

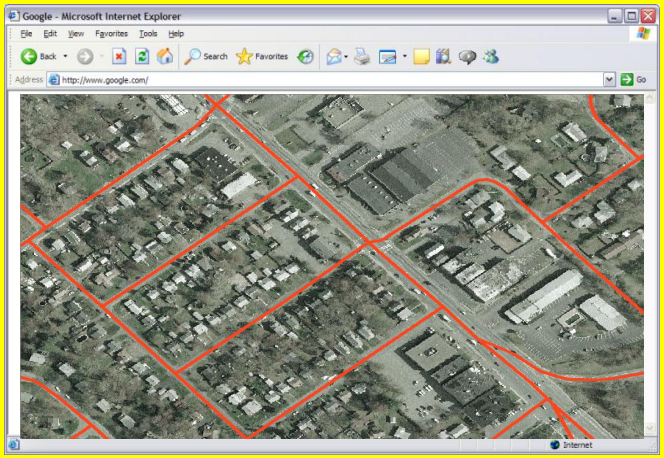
Map Services

- Client requests a map from a server for a specified geographic extent
- Map Server renders the requested map internally and converts to an image file (e.g., JPG, PNG, GIF) and returns the map image to the client for display
- Fast map display is possible

99% of current activity is Map/Image Serving

Map Service Operation

Client



Web Server/Map Server

1. Creates XML Request

2. Request sent to web service

3. Receives request and parses XML

4. Calls the map server to request map

5. Map Server generates map as image file

6. Creates response XML with image file

7. Response sent to client

8. Receives response, parses to extract image and updates map

Cached Map Services

- Maps are pre-rendered and stored (cached) as a series of small tiles at several predefined scales
- The requested scale is identified and the tiles that cover the extent are sent to the client and assembled
- Can result in MUCH faster display, however, it is limited to preset scales, there is no layer control and tiles must be regenerated when any layer changes
- Cached based sites/products include: Google Maps, Yahoo Maps, Live Search Maps, MapQuest, and ArcGIS Server

Proprietary Map Service Products

- Most commercial GIS server products and search sites deliver maps and geospatial functionality using a proprietary architecture/API
- These products include the following:
  - ArcIMS (ESRI)
  - ArcGIS Server (ESRI)
  - ArcWeb Services (ESRI)
  - MapXtreme (MapInfo)
  - MapGuide (Autodesk)
  - Geomedia WebMap (Intergraph)
  - MapPoint
  - Google Maps (Google)
  - Live Search Maps (Microsoft)
  - Mapquest
  - Yahoo Maps
- However, most of the above products can publish open services (e.g, WMS) that meet open standards
- Additionally there are many open source GIS products (e.g., MapServer) which are collaborative efforts and do not use proprietary architectures… these are typically based on open standards

Disadvantage of Proprietary Services

- In most cases, only applications from the same vendor can be used to consume these services
  - For example, MapInfo cannot consume a proprietary ArcIMS service, while ArcView cannot consume a proprietary MapXtreme service
- The proprietary API limits the number of developers that have experience in providing these solutions
- Limits the sharing of geospatial data and functionality between users/organizations

OGC and Open Web Mapping

- The Open Geospatial Consortium (OGC) was founded in 1994
- An international consortium of 346 organizations that lead the development of open standards and specifications to enable the interoperability of geospatial and location based services
- They have developed a series of open standards for geospatial web services such as:
  - Geography Markup Language (GML)
  - Web Mapping Services (WMS)
  - Web Feature Services (WFS)
  - Web Coverage Services (WCS)
- Referrred to as "OpenGIS®" standards

OGC®
Open Geospatial Consortium, Inc.

"Making location count"

Web Mapping Services (WMS)

- WMS is the OGC specification for implementing a map (image) service
- Does not allow for cached map services
- WMS provides 3 functions for clients to access functionality
- Two functions are mandatory to be implemented, the third function is optional
- This results in two general "types" of WMS Services
  - Basic
  - Queryable

Web Mapping Services (WMS)

Basic WMS Service

Every WMS service MUST support the following two functions:

- GetCapabilities – Describes the capabilities and available layers
- GetMap – Generates a map image based on input parameters

Queryable WMS Service

The following function is optional for WMS services:

- GetFeatureInfo – Gets the attributes for a specified feature

What can be done with a WMS service?

- Each client application (e.g., ArcView, MapInfo, Google Earth, etc.) handles WMS services differently
- However, generally the following functions are available:
  - Map Display and navigation
  - Zoom to layer
  - Layer Control
  - Identify feature (only with Queryable Map services)
- Anything else is not available
  - No feature selection
  - No geoprocessing
  - No attribute table/browser display
  - Menus/Buttons are greyed out and disabled

Geography Markup Language (GML) and Web Feature Services (WFS)

Geography Markup Language (GML)

- GML is the OGC specification for encoding the location and attributes of geographic features in an XML format
- GML is the data transport for Web Feature Services (WFS)
- GML separates presentation from content
- The FULL GML specification is enormous and extremely complex to implement in it's entirety (over 600 pgs)
- OGC created subsets of GML called "Profiles"
- The most popular is the "Simple Features Profile" (GML-SF), which focuses on points, lines and polygons
- GML can also be used as a file format to exchange layers
- Most desktop GIS software have the ability to import GML-SF files

GML Examples

```
<gml:Polygon>
    <gml:outerBoundaryIs>
      <gml:LinearRing>
          <gml:coordinates>0,0 100,0 100,100 0,100 0,0</gml:coordinates>
      </gml:LinearRing>
    </gml:outerBoundaryIs>
</gml:Polygon>
```

```
<gml:LineString gml:id="21" srsName="urn:ogc:def:crs:EPSG:6.6:4326">
    <gml:coordinates>45.67, 88.56 55.56,89.44</gml:coordinates>
</gml:LineString >
```

Feature Services

- Client requests features from one or more layers within a specified geographic extent
- Geometry and attributes of features are sent to client
- Requires a "thicker" client to receive the features and render/process locally
- Significantly less performance than map/image services when streaming data
- However, once received by the client application, many typical GIS operations are available (unlike a map service)
- Several vendor products offer feature services (e.g., ArcIMS)

Web Feature Services (WFS)

- WFS is the OGC specification for implementing a feature service
- GML is used to provide the feature data to the client
- WFS provides 5 functions for clients to access functionality
- Three (3) functions must be implemented, two (2) are optional
- This results in two general "types" of WFS Services
  - Basic
  - Transaction

Web Features Services (WFS)

Basic WFS Service

Every WFS service MUST support the following functions:

- GetCapabilities – Describes the capabilities and available layers
- GetFeature – Gets and returns the geometry and attributes for all features within a specified bounding box
- DescribeFeatureType – Returns a description of a layer (feature type, field definitions, etc.)

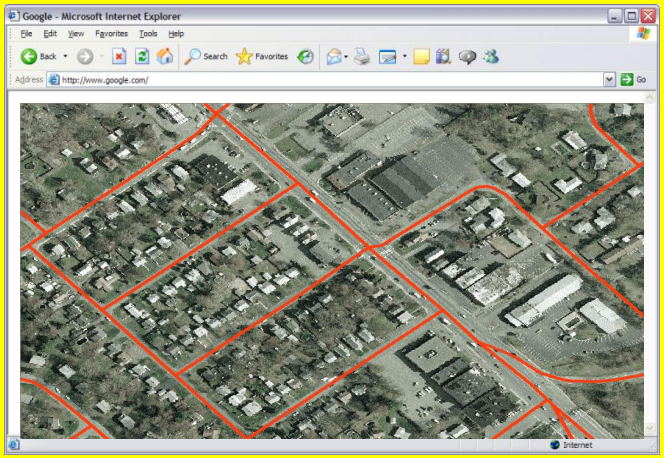
Transactional WFS Service

The following functions are *optional* for providing editing within WFS services:

- Transaction – Allows specified features to be modified as an editing transaction (allows insert, update and deleting of features)
- LockFeature – Creates a lock on the specified features (for locking during an editing transaction)

Feature Service Operation

Client



1. Creates XML Request

2. Request sent to web service

Web Server/Map Server

3. Receives request and parses XML

4. Calls the function to request features

5. Map Server selects features and attributes from database and generates GML

6. Creates response with GML embedded

7. Response sent to client

8. Receives response and parses to extract GML

9. Renders features on map display

Sample WFS Request and Response

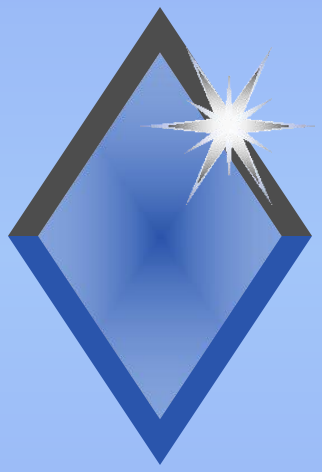
```
<?xml version="1.0" ?>
<GetFeature version="1.0.0" service="WFS" handle="Example Query"
xmlns="http://www.opengis.net/wfs" xmlns:ogc="http://www.opengis.net/ogc"
     xmlns:gml="http://www.opengis.net/gml" xmlns:myns="http://www.someserver.com/myns"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd">
<Query typeName="myns:ROADS">
     <ogc:PropertyName>myns:PATH</ogc:PropertyName>
     <ogc:PropertyName>myns:LANES</ogc:PropertyName>
     <ogc:PropertyName>myns:SURFACETYPE</ogc:PropertyName>
     <ogc:Filter>
          <ogc:Within>
               <ogc:PropertyName>myns:PATH</ogc:PropertyName>
               <gml:Box>
                    <gml:coordinates>50,40 100,60</gml:coordinates>
               </gml:Box>
          </ogc:Within>
</ogc:Filter>
</GetFeature>
```

```
<gml:featureMember>
     <ROADS fid="ROADS.100">
          <PATH>
               <gml:LineString gid="1"
                    SrsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
                    <gml:coordinates>10,10 10,11 10,12 10,13</gml:coordinates>
               </gml:LineString>
                 </PATH>
                 <SURFACE_TYPE>ASPHALT</SURFACE_TYPE>
                 <NLANES>4</NLANES>
     </ROADS>
</gml:featureMember>
```

WMS vs WFS

	WMS	WFS
# of services available	√	
*Speed of data retrieval**	√	
Thick Client Required	No	Yes
Dependability	√	
Functionality		√

***** - Once features have been retrieved, subsequent operations can be faster with**



KML and Google Earth

What is KML?

- KML (Keyhole Markup Language) is an XML-based markup language for locating and visualizing features on a 2D or 3D digital map/surface (e.g., Google Earth/Maps)
- Originally developed by Keyhole, Inc. who was acquired by Google in 2004
- KML is currently under review by OGC as a new standard
- Features specified in the KML schema include:
  - Placemarks
  - Images
  - Polygons
  - 3D models
  - Textual Annotation

What is KML? (cont'd)

- Unlike GML, KML includes tags and attributes that allow the user to describe how the feature should be rendered and visualized on the digital map
- KML uses geographic coordinates (lat/long) in WGS84 for its coordinate reference system
- Each feature is located in 3D space, using one or more x,y,z coordinates
- In addition to Google, many other vendors now utilize KML such as:
  - ArcGIS Explorer
  - Live Search Maps
  - Microsoft Virtual Earth

- Users share locations of events and features
- KMZ is a compressed (zipped) KML file

KML Placemark Sample

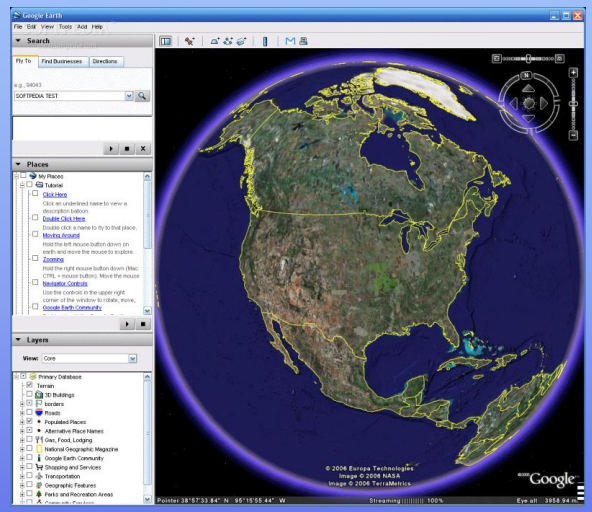
```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<Placemark>
    <description>New York City</description>
    <name>The Big Apple</name>
    <Point>
        <coordinates>-74.006393,40.714172,0</coordinates>
    </Point>
</Placemark>
</kml>
```

Creating KML Files

- You can create KML files in Google Earth
- Once you create a KML file, you can copy and paste the KML into a text editor or XML editor to edit
- ArcGIS can create a KML file from features using an ArcScript
- MapInfo can create a KML file using the Google Earth Link Utility (8.5+)

Using KML Files

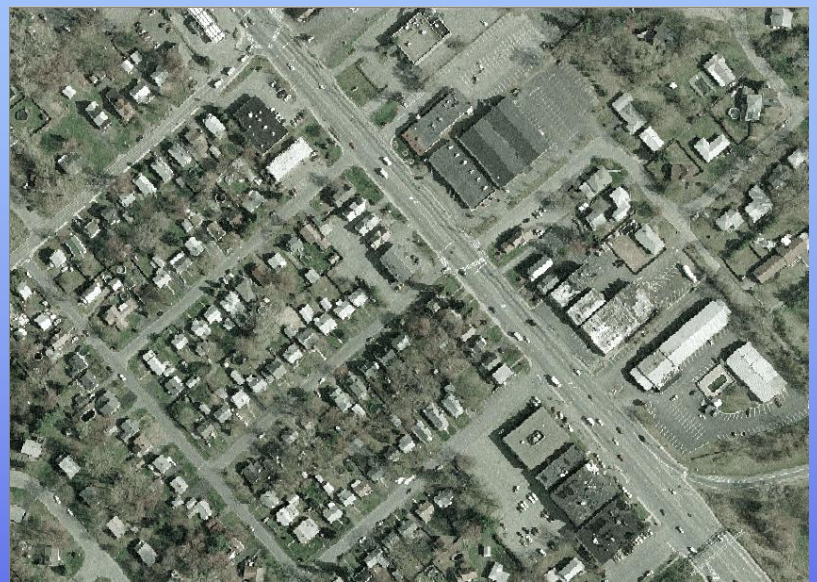
- Double-clicking on a KML file will automatically launch Google Earth and zoom to the feature(s)
- KML and KMZ files can be used in Google Maps, however your file must be hosted on a web server
- Google Maps doesn't support every type of KML feature
- ArcGIS Explorer utilizes KML files
- KML files can be shared through the KML Gallery and Google Earth Community



Geospatial Portals and Finding Data Services

NYS Orthophotography

- CSCIC has partnered with the USGS to provide the latest NYS Orthophotography as a FREE, WMS service from the USGS EROS Data Center in South Dakota
- The URL is:
  - http://gisdata.usgs.gov/wmsconnector/com.esri.wms.Esrimap/USGS_EDC_Ortho_NYSDOP?



Finding Geospatial Data and Services

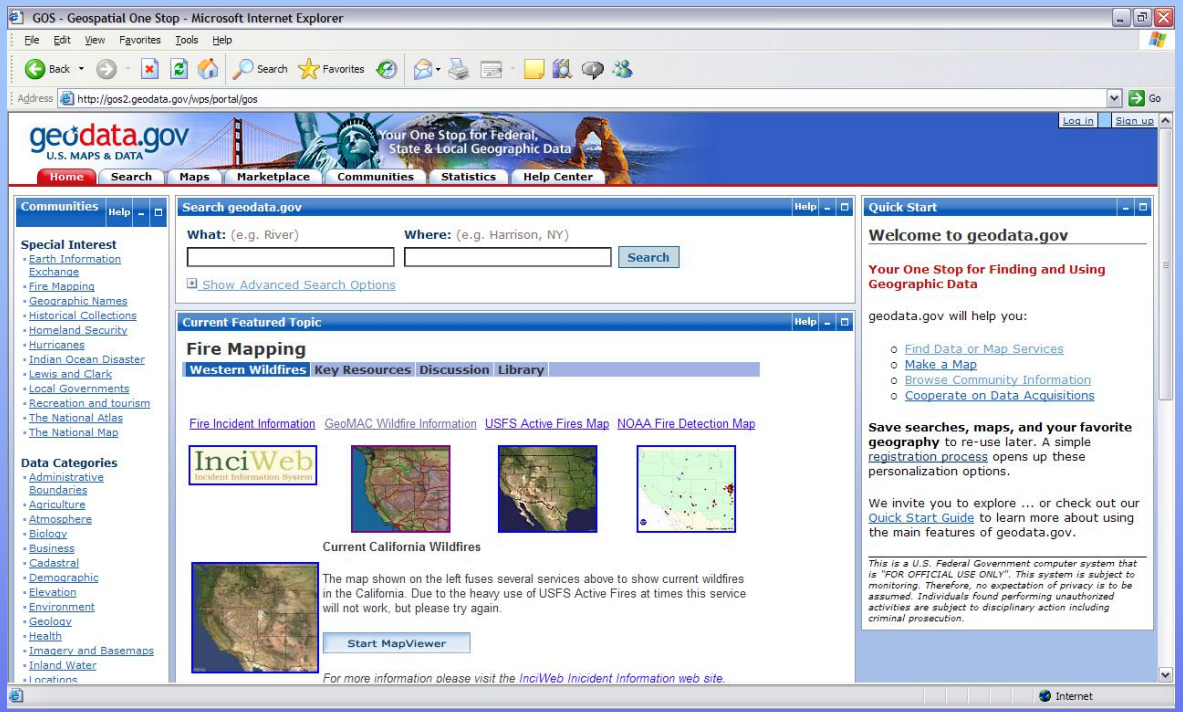
- Portals
  - A portal is a web site that serves as a point of entry to access data and related resources on the web
  - GeoSpatial One Stop (GOS) (http://www.geodata.gov)
  - Geography Network (www.geographynetwork.com)

- Web Sites
  - ArcWeb Services (http://www.arcwebservices.com)
  - Envinsa On-Line Services (http://www.mapinfo.com/envinsa)
  - TerraServer (http://terraserver.microsoft.com)
  - State/County Web Sites
    - NYS GIS Clearinghouse (http://www.nysgis.state.ny.us)
    - MassGIS (http://lyceum.massgis.state.ma.us)
    - VCGI (http://www.vcgi.org)

- Web Searching
  - Search on WFS or WMS

Geospatial One Stop (GOS)

- Federally hosted portal to serve as a central point of access to federal, state and local geospatial data resources
- Managed by Department of Interior, in partnership with Federal Geographic Data Committee (FGDC)
- Can search for map services and downloadable content
- Includes an on-line map viewer



Other Geospatial Services and Final Thoughts

Task Oriented Services

Geocoding Services

- Client passes in one or more addresses to a web service which geocodes the data and returns one or more x,y coordinates

Routing Services

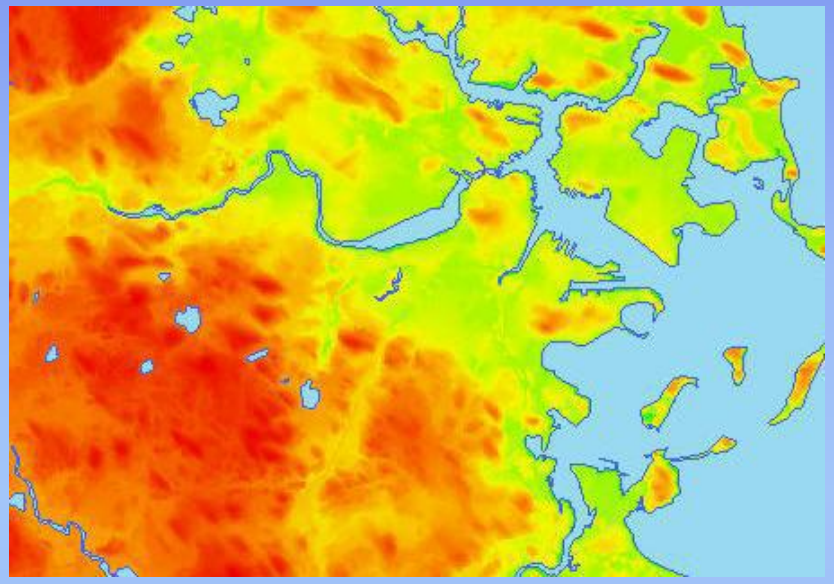
- Client passes in one or more locations to a web service which computes the optimal route and returns the route and optional text based directions

Geoprocessing Services

- Any type of geoprocessing/spatial analysis can be done via a web service (buffer, clip, multi-step model, etc.)
- Input and output varies based on the geoprocessing performed
- Client passes input parameters to a web service which performs the geoprocessing and returns the results

Web Coverage Services (WCS)

- WCS is the OGC specification for implementing a coverage service
- Similar to a WFS, however for "coverage" data of continuous surfaces such as a DEM
- Results returned to client as grid data in one of the following formats:
  - GeoTIFF
  - NITF
  - DTED
- Similar functions such as GetCapabilities and GetCoverage
- Still in it's infancy compared to WMS and WFS



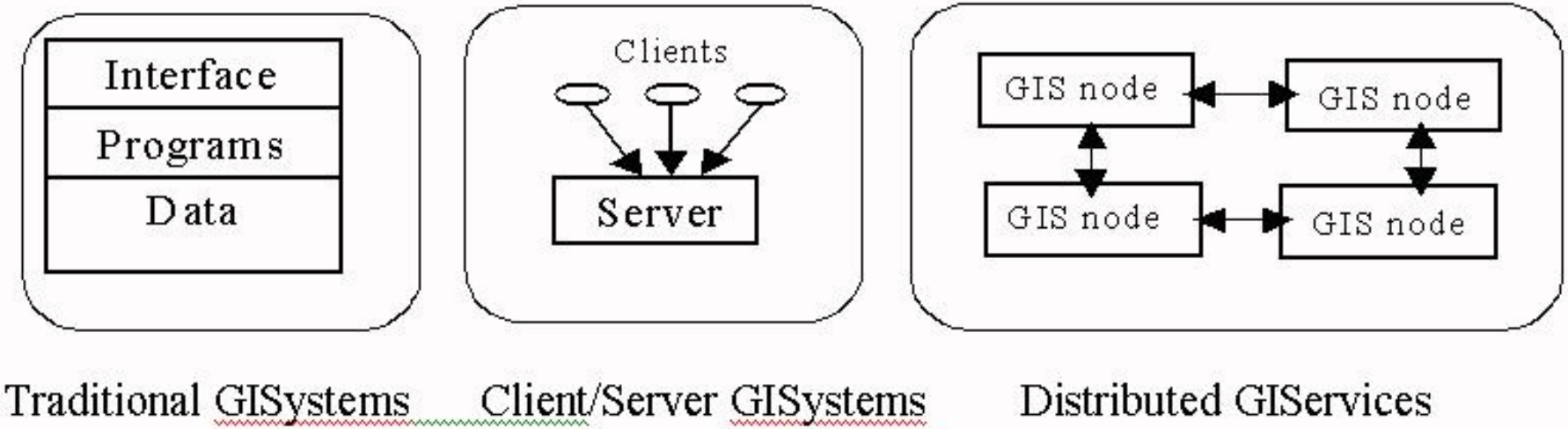
Benefits of Serving Data with OpenGIS®

- Control the data that you are providing
- Ensure users are consuming the correct version/vintage
- Share your data with the widest possible audience
- Reduce the amount of time handling "requests"
- Much more simple to publish regular changes

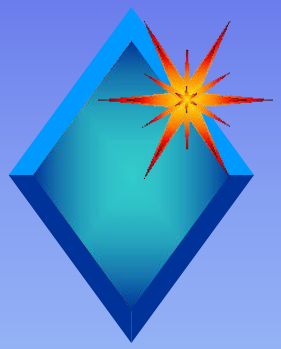
Future of Geospatial Web Services

- Web 2.0 (web as a platform) is here
- Distributed GIS will be one of the primary focuses over the next few years due to our dependency on external data
- Increased focus on use of web services in GIS Desktop products (not implemented as an afterthought)
- Increased focus for mobile devices with wireless comm.
- Increased bandwidth will expand the usage of WFS
- More focus in 3D (beyond Google Earth)
- Increased use of geoprocessing services
- Commercial data companies moving towards a more prominent delivery option via services
- GeoRSS feeds

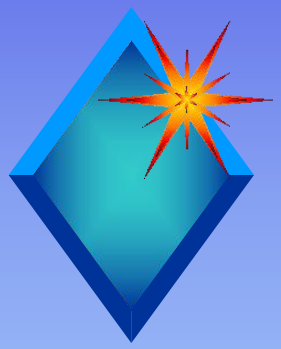
The Future of Geographic Information Services



- Three alternatives for GIS architecture.



- **Traditional GISystems** are closed, centralized systems, incorporating interfaces, programs and data.
- **Client/Server GISystems** are based on generic client/server architecture in network design. The client-side components are separated from server-side components (databases and programs).
- **Distributed GIServices (Peer-to-Peer GIS nodes)** are built upon a more advanced architecture. The most significant difference is the adoption of distributed component technology, which can access and interact with multiple and heterogeneous systems and platforms, without the constraints of traditional client/server relationships.



Related Links:

- What is P2P ... and What isn't ? by Clay Shirky 11/24/2000 http://www.openp2p.com/pub/a/p2p/2000/11/24/shirky1-whatisp2p.html
- What Can P2P Do for B2B? By Mark Merkow, CCP, CISSP http://ecommerce.internet.com/outlook/article/0%2C1467%2C7761_486031%2C00.html
- http://compnetworking.miningco.com/compute/compnetworking/cs/peertopeer/
- CNN - TIME: Napster the Revolution http://www.cnn.com/ALLPOLITICS/time/2000/10/02/revolution.html
-