White Paper

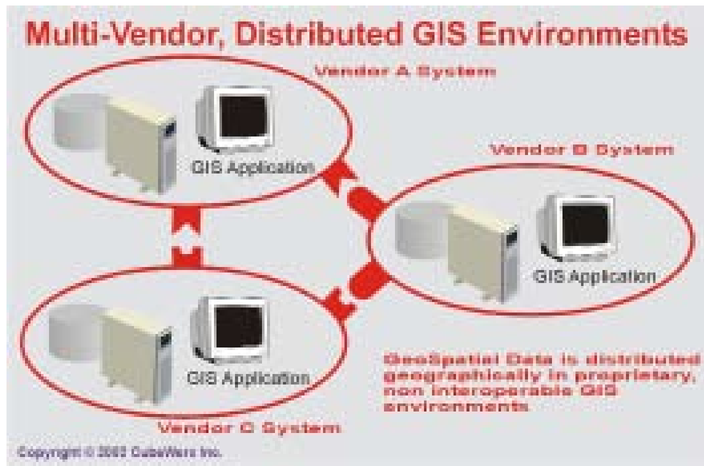
July - 2002

CubeWerx®

Optimizing consolidation, integration and sharing of geospatial data in a multi-vendor, distributed GIS environment

Organizations have long recognized the value of their geospatial data assets. But in order to more effectively manage and share these assets, they must surmount the considerable challenge of building a common framework to integrate and consolidate them. This can be quite a daunting task when these assets have been created with disparate, proprietary GIS solutions, and are dispersed geographically.

This paper will show how Open Geospatial Consortium® Web Server interfaces, when properly supported by a truly integrative and scalable data repository, allow organizations to build an effective infrastructure for the integration, consolidation, management and sharing of their geospatial data assets.



The TCP/IP-based Internet has become the de facto backbone of choice for sharing and distribution of complex information types —it provides seamless access to heterogenous networks of servers through a standard Web Browser.

The essence of the Internet is the seamless presentation of multiple layers of information using a simple Graphical User Interface (GUI), while keeping the aggregation process entirely transparent to the end-user. The Internet could therefore serve as the ideal platform for organizations who wish to build a framework for the more effective sharing and distribution of their geospatial data.

However, organizations are faced with the immediate realization that the convergence of geospatial data towards a Browser-based client is rapidly complicated by the proprietary nature of GIS applications. The Browser-based client will need to 'speak' the language of the local GIS application, in order to access the data stored in that particular location.

The processing capabilities required to interact with several different GIS applications would therefore eliminate any possibility of deploying a lightweight, Browser-based client. This scenario would automatically require the deployment of 'thick' clients, with tremendous processing capabilities and high bandwidth requirements. But thick clients prove completely impractical in instances where the end-user simply wants to view or 'browse' the data.

‘Thinning’ the Web client is not simply an issue of building a Web server with the required processing capabilities to speak and handle the various ‘languages’; it requires a common framework, or interoperability between GIS applications, in order to be truly effective.

The Different Paths to Interoperability

There have been a number of initiatives launched in recent years attempting to facilitate the creation of a common framework for consolidation, integration and sharing of geospatial data.

Metadata Standards. The primary goal of metadata standards is to facilitate the discovery of data, regardless of where or how it is stored. The aim is to define how metadata should be used when describing the content of a spatial data file.

Universal Spatial Data Standards. The primary goal of the universal spatial data standard is to define a file format that would transcend all vendor- and industry-specific formats.

Data Access. The primary goal of data access standards work is to define a common conversion format that allows one GIS application to access and retrieve files, which have been created by another proprietary GIS application.

While participants to these initiatives all share a common objective, i.e. improving the free-flow of data between applications and industries, issues such as the costs arising from conversion of legacy data have raised some doubts as to the ‘traction’ that the standards would ultimately have in the marketplace.

The following table outlines some of the hurdles facing interoperability initiatives.

Metadata Standards	Data Access Standards	Data Standards
• Adoption and acceptance of a universal metadata standard • GeoData remains in its native, proprietary file format • Cost of converting legacy data, once an universal standard is adopted • No automatic update mechanism present in Z39.50	• Little commercial ‘traction’ for standards such as OGDI • High processing and bandwidth requirements for bringing data to a thick client • Deployment of thin Web client is made difficult by the amount of processing required.	• Adoption and acceptance of a universal GeoData standard • Cost of converting legacy data, once a universal standard is adopted • Deployment of thin Web client is made difficult by the nature of GIS application that required ‘thick’ clients

The OpenGIS® Consortium Approach: Open Web Service Interfaces

The OpenGIS® Consortium (OGC) puts the emphasis on standardizing 'interfaces' between processes rather than content and format.

Products and services that conform to OGC's open interface specifications allow users to freely exchange and apply spatial information, application and services across networks, different platforms and products.

The following are some of the Web Server Interfaces, that have been or are in the process of being defined and adopted by OGC:

- Web Map Server (WMS) Interface
- Web Feature Server (WFS) Interface
- Web Coverage Server (WCS) Interface
- Stateless Catalogue Interface

The OpenGIS® Consortium

OGC is an international industry consortium of more than 230 companies, government agencies and universities participating in a consensus process to develop publicly available geoprocessing specifications. OpenGIS® Specifications support interoperable solutions that "geo-enable" the Web, wireless and location-based services, and mainstream IT, and empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications.

Web Map Server (WMS) Interface

A standard Web Browser client can submit a request for a map in the form of Uniform Resource Locator (URL). The content of the result will necessarily depend on which of the three requests were included in the query.

The query consists of URL parameters which :

- Define the spatial extent, i.e. which portion of the Earth is to be mapped, the coordinate system to be used, the type(s) of information to be shown, the desired output format (JPEG, GIF, PNG, etc.), the output size (pixels), the rendering style and other parameters
- In order for the content of the map to be queried, indicate the map and the location on the map that are of interest
- By including a 'capabilitlities' request can ask a Map Server to disclose its 'holdings'

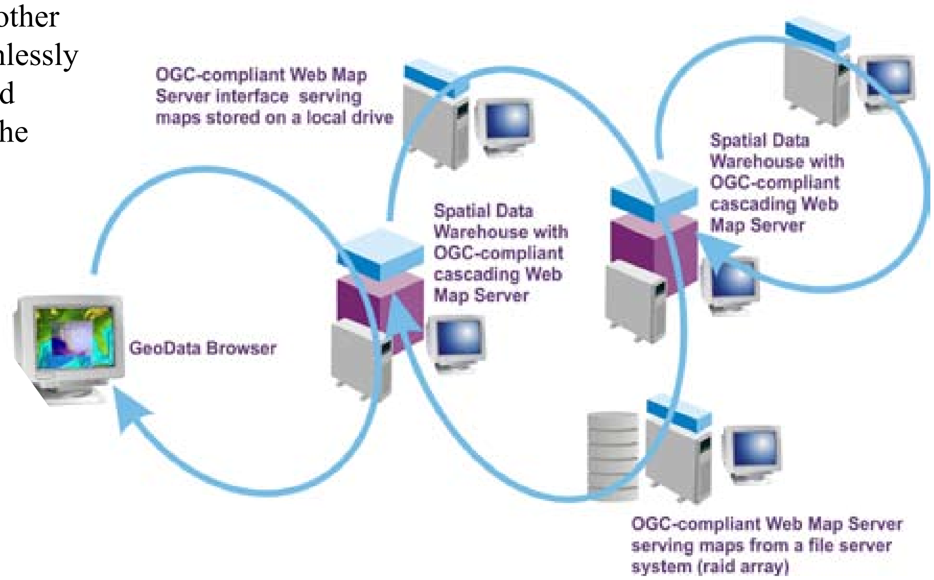
A Web Map Server Interface that complies to OGC specifications can basically handle three types of requests:

GetMap. Produce a map as a picture (JPEG, GIF, PNG, etc.) or as a series of graphical elements
GetFeatureInfo. Answer basic queries about the content of the map
GetCapabilities. Relay to other applications what types of maps can be produced and whether these maps can be queried further

Cascading Web Map Server Interfaces

This architecture is even more powerful when the Browser Client connects to a Web Map Server Interface that can cascade other WMS-compliant interfaces.

In essence, a cascading WMS Interface can relay a URL request it has received to other WMS Interfaces and seamlessly manage the integration and rendering of the results. The entire process remains transparent to the user.



The user no longer accesses a single data repository when connecting to a WMS Interface, but rather to an entire chain of distributed repositories.

Style Layer Descriptor (SLD)

Another important aspect of the WMS Interface is the user's ability to expand the URL request to include information on how the features are to be represented, i.e. what symbology is to be employed when rendering the features.

A Style Layer Descriptor (SLD) allows the user to define the what's, when's and how's relating to the required symbology. A SLD consists of a XML document, residing locally or externally, which will be referred to within the URL request sent to the WMS Interface.

Web Feature Server (WFS) Interface

A Basic Web Feature Server Interface can handle three types of requests:

DescribeFeatureType. Provides a description of the structure of a given feature type
GetFeature. Provides access to features based on a Filter that constrains the returned set of features spatially and non-spatially
GetCapabilities. Provides a list of the feature types available in the WFS instance and what operations each supports.

On top of the basic requests, a transactional WFS supports the following requests:

LockFeature. Provides the ability to allow long term feature locking
Transaction. Provides the ability to Insert, Update and Delete features.

Web Coverage Server (WCS) Interface

The Web Coverage Service extends the Web Map Server interface to allow access to geo-spatial 'coverages' containing values or properties of geographic locations, rather than static maps. Access to native (unrendered) geospatial data is required for thick client rendering, multi-valued coverages and input into scientific models and other clients beyond simple viewers.

The Web Coverage Server can handle two types of requests

GetCapabilities. Provides a description of coverages (i.e. values or properties of spatio-temporal locations culled from various collections. Each collection of available data represents a 'layer'.
GetCoverage. Provides access to data in a given coverage layer. Coverage queries can be constrained in several dimensions (geographic, elevation, time and sample dimension.) Data can be returned either in native resolution or in subsampled form.

Stateless Catalog Interface

The Stateless Catalog interface will provide access to metadata via a sessionless (Web) protocol. This specification will allow for the attachment of metadata to two types of objects

Services, hence the Service Registry Server (SRS) Interface, following OGC-capabilities specifications
Data, hence the Data Registry Server (DRS) Interface, following FGDC standards.

The Role of the Geospatial Data Repository

Implementation of OpenGIS® Web Server Interfaces alone does not guarantee that the proper platform will exist, to support the expanded functionality of applications.

There are two additional characteristics which a data repository must possess in order to fully realize its promises :

Integrative. Repository should ingest and store data, regardless of types, file size or data volumes. The results of any search (within a given spatial extent) must offer a truly seamless visualization of all available data layers

Scalable. Repository should scale to large volumes and access time should not be affected by the size of the data store

Current Repository Limitations

Most data repositories currently used in basic OpenGIS® Web Server Interfaces implementations have some or all of the following limitations, which prevent them from being truly effective:

Lack of support for some spatial data types. The current repository cannot ingest or store types other than the ones used by the map publishing application it was designed for. Some spatial data types must either be converted to a format supported by the application (time consuming and costly) or maintained in a separate data store (ineffective).

Inability to effectively handle large data volumes. The current repository cannot effectively handle queries placed either against large volumes of spatial data or large data sets (e.g. satellite imagery). Spatial data may have to be partitioned into several smaller repositories, resulting often in data duplication and poor data currency.

Inability to provide a seamless, integrated view of imagery data spanning more than one scene. The current repository cannot handle any query requesting a mosaic view of all available imagery data. This creates a reliance on either a thumbnail (low resolution) representation of the imagery data or a cataloguing (metadata) scheme, both of which can render the search ineffective if they are not properly maintained.

Inability to provide a seamless,overlaying view of disparate spatial data types. The current repository cannot handle any query for a layered view of all disparate spatial data types (vector, raster and satellite imagery.

CubeWerx® Geospatial Data Repository Technologies

CubeWerx provides commercial-off-the-shelf (COTS) software for building an optimal data repository. CubeWerx Suite consists of these software components:

- CubeSTOR™ Spatial Data Warehouse: an extremely robust and highly performant Oracle RDBMS tool set
- CubeSERV™ Web Service Interfaces: fully compliant to OpenGIS® specifications for interoperable access to the Spatial Data Warehouse.

What is a Spatial Data Warehouse?
The concept of a spatial data warehouse evolves from the same concepts and definition that are applied to business data warehousing:

A subject-oriented, integrated, time-variant, non-volatile collection of spatial data.

In essence, GeoData that:
- Is composed of multiple (disparate) data sources and data products
- Is optimized to support data production and data distribution requirements
- Is updated by operational systems

A spatial data warehouse is a multi-dimensional solution:

- It stores all types of data (vector, matrix, raster, associated metadata, etc.) directly into the repository at different levels of resolution
- It allows end-users to query vast amounts (terabytes) of disparate data with a very fast response time.
- It allows for visual discovery (or navigation) of all data relating to a subject, or spatial extent:
  - End-users can "slice and dice" through different (and dispate) layers of data
  - End-users can can drill-down or roll-up through various dimensions (resolutions) of data
  - End-users obtain a generalized or aggregated view of the data

CubeWerx®
15 Rue Gamelin, Suite 506
Gatineau, Quebec J8Y 6N5
Canada

North American sales
sales@cubewerx.com
Telephone: (819) 771-8303
Facsimile: (819) 771-8388

Australia and New Zealand sales
CubeWerx Australia Pty Ltd
Brad Spencer, General Manager
ABN: 37 115 163 285
Mob: +61 (0)404 841 131
Tel/Fax: +61 (0)2 9481 7024
brad.spencer@cubewerx.com.au
www.cubewerx.com.au