Open Geospatial Consortium

Technology Office

4899 North Old State Road 37
Bloomington, IN 47408

Telephone: +1-812-334-0601
Facsimile: +1-812-961-2053

Request For Quotation

And

Call For Participation

In the

OGC® Kentucky Watershed Modeling Information Portal Pilot Project

Annex B: KWMIP Architecture

(Updated)

RFQ Issuance Date: February 14, 2006

Proposal Due Date: March 7, 2006

TABLE OF CONTENTS

Annex B: KWMIP Architecture

1 Overview

The Kentucky Watershed Modeling Information Portal (KWMIP) project plays an important role in the overall realization of the Commonwealth of Kentucky's plan to improve and increase the use of watershed modeling in the management of the State's resources. KWMIP provides the architectural framework, and an initial Commonwealth-wide system to support the needs of many Kentucky citizens' needs that can best be satisfied by watershed modeling.

One of the main objectives of the KWMIP is to implement a working Portal providing access to geospatial data needed to support the use of watershed modeling. Surveys indicate that as much as 70% of the time expended to produce a watershed model is used to assemble the data and prepare it for use. KWMIP aims to reduce that time significantly for a selected set of models by providing a single point of access to data maintained on Commonwealth servers and by third parties.

The architecture presented in this document describes the planned implementation of KWMIP. The design of the architecture takes into account the reference model of OGC and its broad spectrum of specifications and discussion papers.

The as built system in Kentucky includes two different platforms, one provided by the Kentucky Landscape Census project, and one provided by the Kentucky Geonet Portal. Respondents may propose to add KWMIP functionality to either of these platforms, or to mix and match as appropriate. For instance, a proposal could be based entirely on one system or the other or it could include use of the KLC Catalog and the KYGeonet graphic user interface or vice versa.

In addition to identifying the general requirement of an effective geospatial portal, this annex highlights how OpenGIS® Standards explicitly support the capabilities of the reference architecture. The KWMIP will be deployed using several OpenGIS® Web Services interface specifications, based on the requirements and corresponding Use Cases that were derived from the KWMIP User Needs Assessment.

1.1 KWMIP Portal Reference Architecture

The KWMIP Portal Reference Architecture specifies four service classes that are needed to procure a comprehensive geospatial portal implementation and it identifies the OpenGIS Interoperability Standards that are applicable to the services. The five service classes are:

- **Portal Services** –Provide the single point access to the geospatial information on the portal. In addition, these services provide the management and administration of the portal.
- **Catalog Services** –Used to locate geospatial services and information wherever it is located and provide information on the services and information if finds to the user.
- **Portrayal Services** –Used to process the geospatial information and prepare it for presentation to the user.
- **Data Services** – Used to provide geospatial content and data processing.
- **KWMIP Services** – Though a type of Data Service, they are shown separately in this document because they are the additional work to be added to the Kentucky system.

The KWMIP Portal Reference Architecture is shown in **Figure 1**.

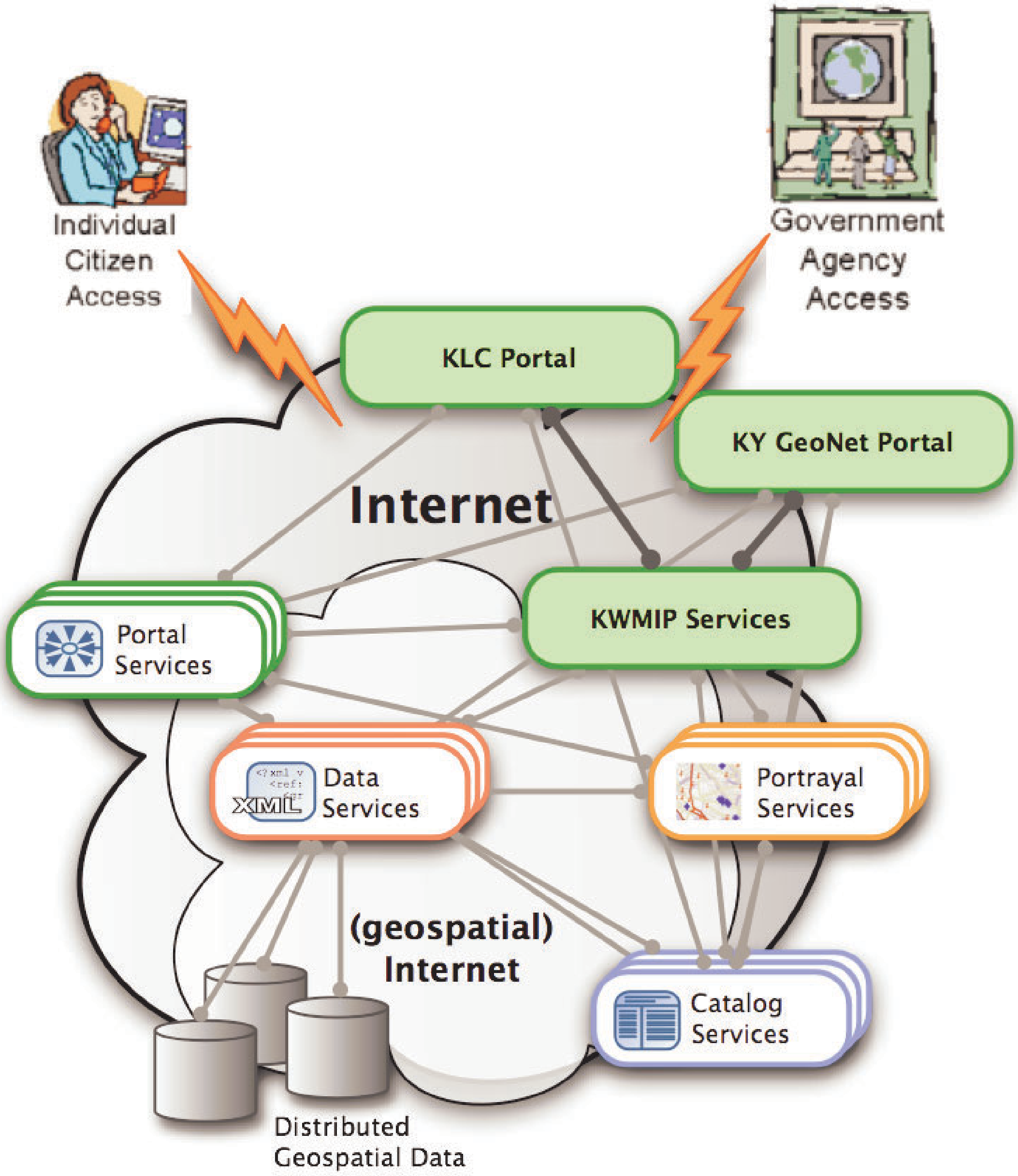

Figure 1: KWMIP Portal Reference Architecture

It is important to understand that the Portal Services and any required Infrastructure Services are the only ones that need be resident on the platform on which the portal is operating. All of the other services can be distributed across the Internet and can be dynamically registered and executed. Also notice that the Portal does not store the geospatial data processed by the distributed services. This loosely coupled service orientation is known as a service oriented architecture, which is described in more detail subsequence section.

The KWMIP Portal Reference Architecture follows the tenets set forth in the OpenGIS Reference Model (ORM), which is explained in the next section.

1.2 Service Oriented Architecture (SOA)

Service orientation is a way of viewing software assets on the network—fundamentally, the perspective of IT functionality being available as discoverable Services on the network. Essentially, Service orientation provides business users with understandable, high-level business Services they can call upon and incorporate into business processes as needed. The Service orientation vision is therefore one of agility and flexibility for users of technology, coupled with an abstraction layer that hides the complexity of today's heterogeneous IT environments from those users.

Service-Oriented Architecture (SOA) is an architecture that represents software functionality as discoverable Services on the network. SOAs have been around for many years, but the difference with the SOAs we talk about today is that they are based on standards, in particular, *Web Services*. Web Services provide standards-based interfaces to software functionality. Producers of these Services may publish information about them in a Service registry, where Service consumers can then look up the Services they need and retrieve the information about those Services they need to bind to them.

Applications designed using SOA can provide the same functionality as that found in a monolithic architecture coupled with the following additional benefits:

- Easier extension of legacy logic to work with new business functionality
- Greater flexibility to change without the need to constantly re-architect for growth
- Cost savings by providing straight-forward integration

Within the SOA context, therefore, the KWMIP Portal is the ideal mechanism for realizing the Web Services vision of "enter once, use often" by providing a bridge between "separate islands of data and processing services." The portal does not store any content. Therefore, the geospatial content and services remain with the responsible authority- those agencies or private sector organizations responsible for the collecting and maintaining geospatial content continue to retain control of those activities. The authoritative source for geospatial content and services becomes the Portal as the conduit to disparate and distributed content and services. The OGC reference architecture is a service-oriented architecture based on the now recognized Web Services paradigm of the publish/find/bind pattern and supports the dynamic binding between service/content providers and requestors since sites and applications are frequently changing in a distributed environment. Content and service providers publish their availability to the portal's catalog making their information "known" to the portal.

1.2.1 Publish-Find-Bind

The core method of communications within the portal is based on service-oriented architecture that follows a service trading paradigm. Service trading is a fundamental concept that addresses the discovery of available service instances. Publishing a capability or offering a service is called "export" (publish). Finding a service request against published offers or discovering services is called "import" (Find). Binding a client to a discovered service is called "service interaction" (Bind). This can also be depicted in an equivalent manner as the "Publish – Find – Bind" (PFB) pattern of service interaction. These fundamental roles and interactions are depicted in **Figure 2**.

This service trading function is elaborated in a separate document (ISO/IEC 13235-1) and refined somewhat in the Object Management Group (OMG) Trading specification, which is technically aligned with the computational view of the ODP trading function. Most importantly, a broker supports dynamic (i.e. run-time) binding between service providers and requesters, since sites and applications are frequently changing in large distributed systems. A broker registers service offers from provider objects and returns service offers upon request to requestor objects according to some criteria.

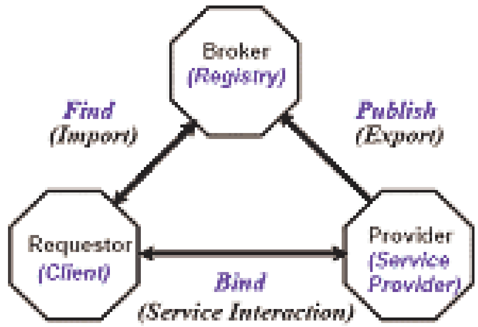

Figure 2: Service Trading Communication Structure

In the KWMIP Portal reference architecture, there are three fundamental roles that are defined to actuate the service trading. They are:

- **Broker -** a role that registers service offers from service providers and returns service offers upon request to requestor according to some criteria.
- **Provider -** a role that registers service offers with a broker and provides services to clients.
- **Requestor** - a role that obtains service offers, satisfying some criteria, from the broker and binds to discovered services provided by the provider.

To export (i.e. publish a service offer), an object gives the broker a description of a service, including a description of the interface at which that service instance is available. To import (i.e. find suitable service offers), an object asks the broker for a service having certain characteristics. The broker checks against the descriptions of services and responds to the requestor with the information required to bind with a service instance. Preferences may be applied to the set of offers matched according to service type, some constraint expression, and various policies. Application of the preferences can determine the order used to return matched offers to the requestor.

1.3 KWMIP Portal Functionality and System Use Cases

Following up to the *KWMIP System Architecture and Database Review document* and the *User Needs Analysis (UNA) and Process Flow document*, this Appendix A to this annex presents the system use cases and associates them to the KWMIP UNA requirements -- user process flows and functional requirements groupings -- and OpenGIS high-level Web Services (where applicable).

1.4 KWMIP Portal Reference Model

The structure of this Architecture is loosely based on the Reference Model for Open Distributed Processing (RM-ODP), which was the design baseline for the OpenGIS Reference Model (ORM). The model provides an architecture framework for the ongoing work of the OGC. Further, the ORM provides a framework for the OGC Technical Baseline. The OGC Technical Baseline consists of the currently approved OpenGIS Specifications as well as for a number of candidate specifications that are currently in progress.

The ORM has the following purposes:

- Provides a foundation for coordination and understanding (both internal and external to OGC) of ongoing OGC activities and the Technical Baseline;
- Update/Replacement of parts of the 1998 OpenGIS Guide;
- Describes the OGC requirements baseline for geospatial interoperability;
- Describes the OGC architecture framework through a series of non-overlapping viewpoints: including existing and future elements;
- Regularize the development of domain-specific interoperability architectures by providing examples.

The ORM is a living document that will be revised on a regular basis to continually and accurately reflect the ongoing work of the Consortium. (see http://www.opengeospatial.org/specs/?page=orm. It is encouraged that respondents to this RFQ understand the concepts that are presented in the ORM.

The five views shown in **Figure 3** into the KWMIP Portal Architecture are described in further detail in Sections 2 through 6 of this annex.

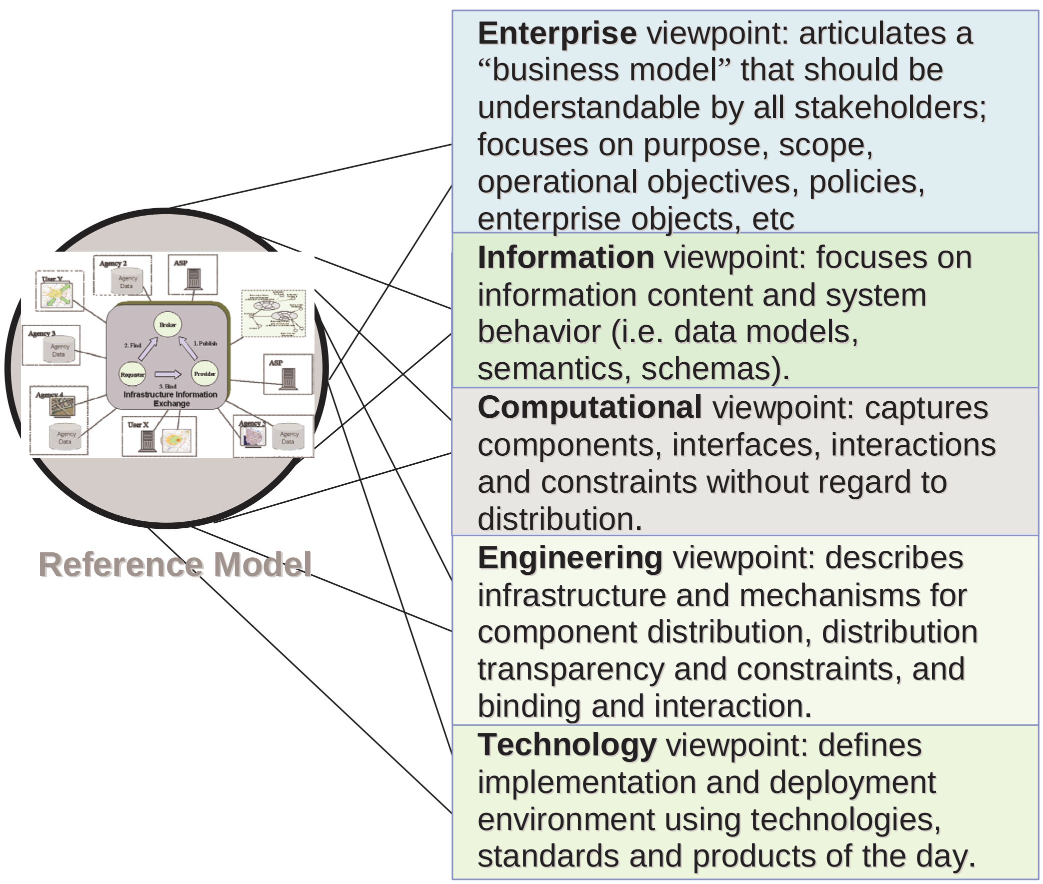

Figure 3: KWMIP Portal Reference Model

- Section 2 (Enterprise view) describes the Enterprise Architecture for the KWMIP Portal. This architecture describes the high-level system concepts.

- Section 3 (Information View) describes the Information Architecture for the KWMIP Portal. This architecture describes the basic information building blocks of KWMIP Portal.

- Section 4 (Computational View) describes the Computational Architecture for the KWMIP Portal. This architecture describes the basic service building blocks of KWMIP Portal.

- Section 5 (Engineering View) describes the Engineering architecture for the KWMIP Portal. This architecture describes the core components that are to be deployed and the infrastructure to integrate them into a single environment.

- Section 6 (Technology View) describes the target deployment environment for KWMIP Portal components in terms of technologies, standards and products.

2 Enterprise Viewpoint

The KWMIP Portal Enterprise Architecture captures the capabilities that must be present in support of allowing users to view and obtain desired data for a particular area, without needing to know the details of how the data are stored and maintained by independent organizations. The capabilities identified in the enterprise view provide the requirements to be met by the KWMIP services and information architecture. Success of the KWMIP is measured by the accuracy of the enterprise view and how well the information and systems architectures support that view. The system concept illustrates the operational setting, major system components, and major interfaces.

Under the current system architecture vision, the Commonwealth Office for Technology (COT will host the KWMIP services including the web server hardware and software that supports the system.

It will allow custom coded functionality to be stored and administered in a central location. The state will also host a data node for the KWMIP services as well to provide user access to some suite of data holdings within the Kentucky portal. Permissions settings to access data and other administrative functions for each data node will be left to the node administrator(s). In this manner, some level of security for the portal will be provided at the state level via infrastructure already in place.

A registry of available data will be hosted at the state level within the main hub for the portal. This registry will maintain a list of available datasets over all nodes on the KWMIP system and provide pointers to this data at each local node. Users need never be aware that they are utilizing data from multiple system nodes within their KWMIP system map as the portal architecture will allow for seamless integration of data from all nodes on the portal. The critical point of the proposed architecture is that the KWMIP system will not store or maintain all of the data and its associated services; rather, these are distributed in many computers across the state and perhaps nationwide and maintained by the agency or organization that is responsible for its data and services. The actual mapping, or translation, occurs on the fly upon request of a data set by a user. All that will be required for a user to access data and services provided by the KWMIP system is the use of a standard web browser.

The KWMIP system is based on the Internet environment where the applications/clients can have access to one or more data holdings via the Internet. OGC interfaces provide access to the many disparate, geographically distributed geographic data holdings. This includes the use of OGC interfaces on both the client and the server sides. For example, the KWMIP system will provide a standard Web Feature Service

(WFS) interface and Web Mapping Service (WMS) interface, as well as ftp access to allow download of entire files.

2.1 KWMIP Concepts

The KWMIP Technical Advisory Group (TAG) has participated in a formal User Needs Assessment document, available on request, and a formal Use Case document published as Appendix A to this document. Together these define the needs expressed by the potential users and a Use Case to be used in satisfying those needs. The ‘data provider’ use case has not been defined at this time, but will be done in conjunction with the selected proposer. .

2.2 KWMIP Goals

The KWMIP acts as a primary data source for watershed modeling in Kentucky. The goals are listed below:

- To provide fast, low cost, reliable access to Geospatial Data needed for watershed modeling. Note that the KWMIP Portal is only for data access. The KWMIP Portal is not conceived as a re-write of desktop GIS functionalities.
- To facilitate the alignment of roles, responsibilities and resources.

2.3 KWMIP Requirements

1) The KWMIP system is based on the same open standards and specifications that are used in the KLC, the National Map, the NSDI, and the federal government’s Geospatial One Stop.

2.3.1 KWMIP Portal Service Requirements

The KWMIP Reference Architecture describes a framework within which an organized collection of open standard specifications can be implemented to allow discover of an access to data required to conduct watershed modeling..

It is explicitly open, vendor-neutral, and not dependent on any particular GIS, programming language, database, middleware or operating system. Components of portals based on the KWMIP Reference Architecture – servers, clients, servlets, applets, middleware, databases, applications, etc. – and the external resources that link to them can be commercial-off-the-shelf (COTS), government-off-the-shelf (GOTS), custom, shareware, open source, and/or legacy. These components communicate through the standard interfaces, protocols and schemas defined in the OGC Reference Model (ORM) and referenced throughout this annex.

In **Figure 4**, the KWMIP Portal Reference Architecture is presented again this time identifying the particular services that fall within each class of service.

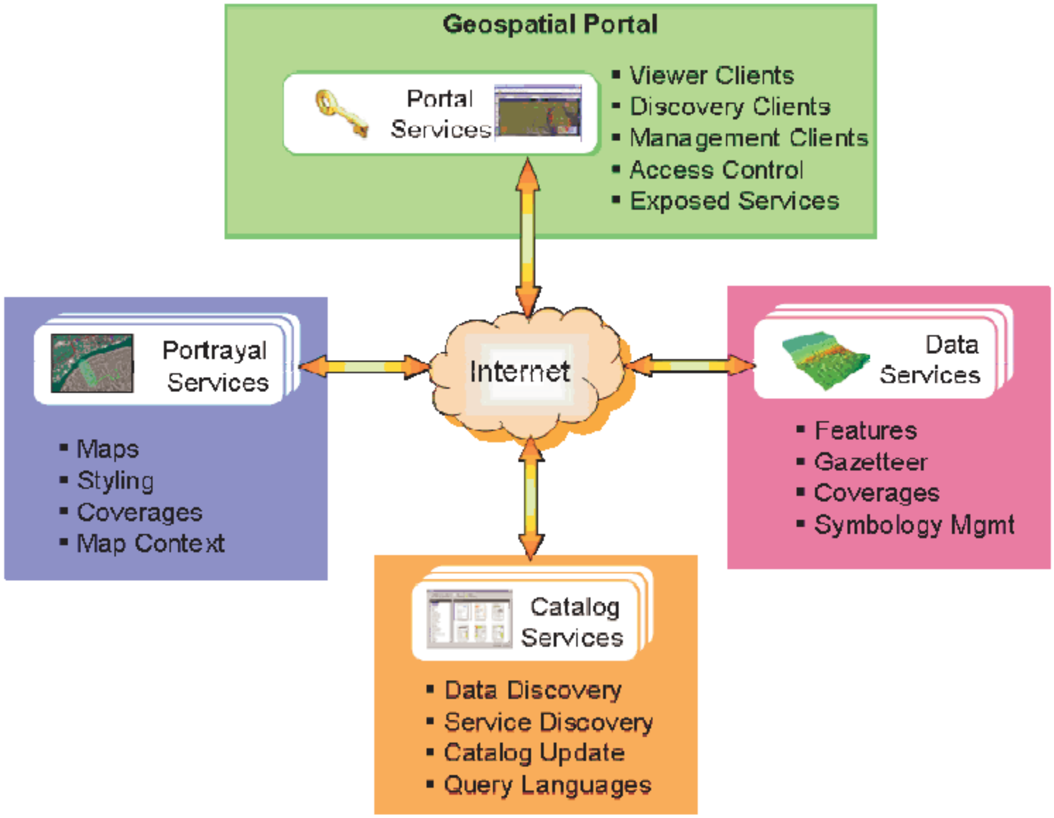

Figure 4: KWMIP Services Distribution

In the subsequent section, each of the services is discussed along with identifying which OGC Implementation Specification is applicable for that service.

2.3.1.1 Portal Services

The Portal Services are accessible from the Portal Platform (e.g. desktop, laptop, etc.) or servers that have network connectivity. Users may leverage Portal Services to access the distributed Portrayal, Catalog and Data services, depending upon the requirements and designed implementation of the application. The access to these services is provided by client software that is resident on the portal platform. More specifically, when accessed as a World Wide Web application a Client runs on an HTTP server and generates HTML pages to be displayed in the User's web browser (the thin client). The Portal Services are provided by adding KWMIP to either the KLC Portal or the KYGeonet portal.

2.3.1.1.1 Viewer Client

The Viewer Client provides a visualization user interface to display and navigate content retrieved from the Portrayal and Data services. It may be proposed as an extension of either the KLC Viewer Client or the KYGeonet Viewer Client and must be capable of connecting to data services using OpenGIS interfaces, selection and version to be negotiated.

2.3.1.1.2 Discovery Client

The Discovery Client provides means for users to locate needed content and services according to user-defined criteria. More specifically, the Discovery Client enables the portal catalog containing the information about the content and services that have been registered to the portal to be searched and the request displayed to the user. The Discovery Client will also allow the user to select a desired content or service and have the service invoked for presentation in the viewer client. It may be proposed as an extension of either the KLC Discover Client or the KYGeonet Discovery Client and must be capable of connecting to catalog services using OpenGIS interfaces, selection and version to be negiated.

2.3.1.1.3 Publisher Client

The Publisher Client provides means for portal maintainers and authorized users (e.g., admin users) to publish services or content discovered using the portal catalog. The Publisher Client allows authorized users to register primary information sources, pre-defined symbolization rules, and possibly other information. This published information is then made available to the Discovery Client where the published information can be found and services invoked. It may be proposed as an extension of either the KLC Publisher Client or the KYGeonet Publisher Client. As it is envisioned as an entirely COT internal client, it is preferred that it use OpenGIS interfaces, but not required to do so.

2.3.1.1.4 Gazetteer Client

The Gazetteer Client provides users the ability to navigate through spatially organized features with well-known feature names. Gazetteer Clients allow users to formulate queries to retrieve named features. The Gazetteer Client should utilize the USGS Geographic Names Information System as a primary database of feature names. It may be proposed as an extension of either the KLC Gazetteer Client or the KYGeonet Gazetteer Client

2.3.1.1.5 Data Extraction Client

Data Extraction Clients provide users the ability to extract specific content from the Data Services class of services. It may be proposed as an extension of either the KLC Data Extraction Client or the KYGeonet Data Extraction Client

2.3.1.1.6 Authentication and Access Control

The KWMIP system will enable Authentication and Access Control that restricts access to an organization's content and service offerings based on criteria that are controlled locally and documented in the Portal site as a set of rules. At a minimum, the Portal should not prohibit Providers from defining access restrictions. In other words, the system must not cause all Users to appear as a single anonymous user when invoking services.

Providers may enforce access restrictions at the network TCP/IP level, at the HTTP server level, at the web service component level, or at any other point of the service request's passage through the Provider's network. Providers should not be required to register their access restrictions at the Portal; however, service metadata and dataset metadata should include information about restrictions to minimize failed access requests. Portal participants may select and document appropriate metadata fields for this purpose. It may be proposed as an extension of either the KLC or the KYGeonet portals.

2.3.1.2 Portrayal Services

Portrayal Services provide specialized capabilities supporting visualization of geospatial information. Portrayal Services are components that, given one or more inputs, produce rendered outputs (e.g., cartographically portrayed maps). Portrayal Services can be tightly or loosely coupled with other services such as Data and Processing Services and transform, combine, or create portrayed outputs. Portrayal Services may use styling rules specified during configuration or dynamically at runtime by Application Services.

Portrayal Services provide specialized capabilities supporting visualization of geospatial information. Portrayal Services are components that, given one or more inputs, produce rendered outputs (e.g., cartographically portrayed maps) or leverage the parameters of rendered outputs to coordinate multi-source display (e.g. create scale and view-dependent displays). Portrayal Services are loosely coupled with other services such as Map and Data Services and transform, combine, or create portrayed outputs. Five possible components are identified with the OGC Reference Architecture are described below. It may be proposed as an extension of either the KLC Portrayal or the KYGeonet Portrayal service.

2.3.1.2.1 Map Portrayal

The OpenGIS WMS Specification is a set of protocols that provide access by Web clients to maps rendered by map servers on the Internet. The WMS interface allows the client to query the "capabilities" of a given map server. Based on the capabilities, the WMS interface allows a server to return a Portable Network Graphics (PNG), Graphics Interchange Format (GIF), Joint Photographic Expert Group format (JPEG), or Tagged Image File Format (TIFF) image for a given area of interest and a specified coordinate reference system.

These returned images (pictures) can be viewed in transparency mode, thus allowing for example, the display of roads on top of a satellite image. The WMS interface support user queries of displayed spatial content that indicates the Spatial Reference System (SRS) and Bounding Box of the portion of the Earth to be mapped, and the output width, height and format of the picture.

The WMS can have an addition capability to define styles that control the presentational rules that are to be used when displaying geographic features. This ability to control the display styles in defined in the Styled Layer Descriptors Specification (SLD).

2.3.1.3 Catalog Services

Catalog Services provide a common mechanism to classify, register, describe, search, maintain and access information about *resources* available on a network. Resources are network addressable instances of typed data or services. Types of registries are differentiated by their role such as registries for cataloging data types (e.g., types of geographic features, coverages, sensors, and symbols), online data instances (e.g., datasets, repositories, and symbol libraries), service types and online service instances. Catalog services allow:

1) <u>Providers</u> of resources to publish descriptive information about resource types and instances;

2) <u>Requestors</u> of resources to discover information about resource types and instances; and

3) <u>Requestors</u> of resources to access (bind to) resource providers.

It may be proposed as an extension of either the KLC Catalog or the KYGeonet Catalog, but will be provided using the OpenGIS Catalog Specification v 2.0, CS-W Profile.

2.3.1.4 Data Services

Data Services provide access to collections of content in repositories and databases. Resources accessible by Data Services can generally be referenced by a name (identity, address, etc). Given a name, Data Services can then find the resource. Data Services usually maintain indexes to help speed up the process of finding items by name or by other attributes of the item. The OpenGIS Framework defines common encodings and interfaces in which multiple, distributed Data Services are accessed and their contents "exposed" in a consistent manner to other major components. The sections below describe the set of Data Services of the OpenGIS Framework to be used in KWMIP.

2.3.1.4.1 Feature Services

The OpenGIS WFS Specification supports the query and discovery of geographic features and attributes. In a typical Web-base scenario, WFS delivers GML representations of simple geospatial features in response to queries from HTTP clients. Clients (service requestors) access geographic feature data through a WFS by submitting a request for just those features that are needed for an application. The client generates a request posts it to a WFS instance (a WFS server on the Web). The WFS instance executes the request, returning the results to the client as GML. A GML-enabled client can manipulate or operate on the returned features.

2.3.1.4.2 FTP Service

KWMIP FTP service, not shown in the Data Services box on the diagram, will also be provided to enable download of full data sets using that protocol.

2.3.1.4.3 Gazetteer

A Gazetteer is a directory of features containing some information regarding position. A Gazetteer Service is a network-accessible service that retrieves one or more features (after the ISO feature model), given a query (filter) request. This filter request must support selection by well-known feature attribute values, and especially by published or context-unique identifiers. The queryable feature attributes are any properties that describe the features, including but not limited to feature type, feature name, authority, or identification code. Each instance of a Gazetteer Service has an associated vocabulary of identifiers. Thus, a Gazetteer Service may apply to a given region, such as a country, or some other specialized grouping of features. The returned features will include one or more geometries expressed in an OGC Spatial Reference System. The Gazetteer interface extends the WFS specification by defining additional behavior and formalizing the response schema elements.

It will be used in conjunction with the Discovery Client and can be proposed as an extension of either the KLC or the KYGeonet functionality.

2.3.1.4.4 Coverage Services

The OpenGIS Web Coverage Service Specification (WCS) supports the networked interchange of geospatial content as "coverages" containing values or properties of geographic locations. Unlike the Web Map Service, which filters and portrays spatial content to return static maps (server-rendered as pictures), the Web Coverage Service provides access to intact (unrendered) geospatial information, as needed for client-side rendering, multi-valued coverages, and input into scientific models and other clients beyond simple viewers.

3 Information Viewpoint

The information viewpoint describes the basic information building blocks of the Portal and is concerned with the semantics of information and information processing. It defines classes of information that are defined by the Portal. It then enumerates various encodings, formats, and languages of relevance.

3.1 Information Classes

- As mentioned before, the KWMIP is intended to become a primary data source for watershed modeling in Kentucky. A detailed enumeration of the data types is found in Annex B, Appendix A.

3.2 Geospatial Information Formats

3.2.1 Vector Data

Vector data is the class of data that represents point, line and polygon geospatial features. This data contains detailed geometry and attribute information about individual geospatial features. In the case of KWMIP, the principal vector data will be the content defined in the FGDC Framework standard. Note that the data needs to be formatted to be displayed in an application. If provided via WFS this data is encoded based on a GML (version to be determined) application schema. If provided via FTP it will be sent in its native format.

3.2.2 Picture Maps

Maps are defined as visual portrayals of feature or coverage data--a map is a picture that can be readily viewed by a user. For purposes of The KWMIP, maps are considered a separate data class from imagery or coverage data and are typically the output from a WMS server that might be connected to a WFS server. Relevant picture map formats for the KWMIP Portal are:

- PNG - Portable Network Graphics
- GIF - Graphics Interchange Format
- JPEG - Joint Photographic Expert Group format

3.2.3 Coverage/Grid Data

A coverage is an association of points within a spatial/temporal domain to a value (of a defined data type, possibly a complex type). A coverage in the OpenGIS® Specification is a function which can return its value at a geometric point. Scalar fields (such as temperature distribution), terrain models, population distributions, satellite images and digital aerial photographs, bathymetry data, gravitometric surveys, and soil maps can all be regarded as coverages. Coverages usually represent phenomena.

Most Coverages depend on two stored functions. The functions map respectively “to” and “from” a mathematical coordinate space called the Coverage Extent. The first function, the Coverage Generation Function, maps from Earth coordinates to Coverage Extent Coordinates, and provides geolocation. The second, the Schema Mapping, maps from the Coverage Extent to some range of values. In general, both functions must be stored. This data may be provided via an OpenGIS Coverage service or an FTP interface.

3.3 Data Encoding for Geographic Information

The Geography Markup Language (GML) is an XML encoding for the transport and storage of GI, including both the geometry and properties of geographic features[1]. GML is a set of rules and schemas for

---

[1] As with the OpenGIS® Simple Feature Specification, GML utilizes the OpenGIS® Abstract Specification geometry model. However, unlike the Simple Features Specification, the GML Specification includes the ability to handle complex properties.

encoding geographic information in XML. Because ordinary Web browsers “understand” XML, users with no special GI software can be given access to geoprocessing and semantic processing capabilities delivered through their Web browsers.

3.4 Management and Control of Data

3.4.1 Data Metadata

Data Metadata is metadata that describes data holdings. The relevant specification for KWMIP Data Metadata is:

- CSDGM - FGDC Content Standard for Digital Geospatial Metadata

Note: the specification "ISO 19115 Geographic Information - Metadata" is expected to be profiled by FGDC as a new version of its metadata standard. When that has occurred and been deployed in the NSDI Clearinghouse network, The KWMIP system will adopt that specification as well.

3.4.2 Service Metadata

Service Metadata is metadata that describes services. Relevant specifications for The KWMIP Service Metadata include (but are not limited to):

- WMS - OGC Web Map Service - Capabilities XML
- WFS - OGC Web Feature Service - Capabilities XML
- WCS – OGC Web Coverage Service – Capabilities XML

3.4.3 Query Languages

Most of the services used in KWMIP have a queryable interface. To use those interfaces, it is necessary that the client and server have a common understanding of what constitutes a valid query. Standardized query languages provide the common vocabulary and grammar needed to enable this sliver of interoperability. Relevant specifications for The Query Languages are:

- Catalog Service for the Web (CS-W) part of the OpenGIS Catalog Implementation Specification
- OGC Web Map Service - GetFeatureInfo operation
- OGC Filter Specification (associated with the WFS specification)

3.5 Application Schemas

Application schema is a set of conceptual schema for data required by one or more applications (i.e., describes the structure of GML for a specific application). It contains the descriptions of both geographic data and other related data. Any description is always an abstraction, always partial, and always just one of many possible views. Designers of application schemas may extend or restrict the types defined in the base schemas (i.e., Standard Application Schema) to define appropriate types for an application domain.

In brief, the purpose of an application schema is twofold:

- to provide a computer-readable data description defining the data structure, which makes it possible to apply automated mechanisms for data management;

- to achieve a common and correct understanding of the data, by documenting the data content of the particular application field, thereby making it possible to unambiguously retrieve information from the data.

3.5.1 GML Application Schemas

Feature types for a GML application language are defined in an XML Schema defining the language, known as a GML application schema. The conceptual role of a GML application schema, therefore, is to formally define the members of a catalog of feature types for a particular information community. The operational role is to validate XML instance documents describing information of interest to members of the information community.

Given an existing datasource with, for example, a pre-defined table structure (designed to support the existing primary business processes) and an application domain GML application schema that the WFS should use, the configuration of any general purpose WFS service requires the following:

1. specification of the mapping for each of the element and attribute content in the XML document that is generated in response to a WFS GetFeature request -> the datasource operation that supplies the required information;

2. specification of the mapping for any supported WFS filter request -> an operation on the datasource.

4 Computational Viewpoint

The computational viewpoint is concerned with the functional decomposition of the system into a set of services that interact at interfaces. This viewpoint captures the details of these components and interfaces without regard to distribution.

The KWMIP Computational Viewpoint is represented graphically in the Concepts diagram in **Figure 3**. That figure represents data providers as cylinders near the bottom of the diagram. Each provider supports several online services that are accessed by the Portal.

> *Note: Although the diagram shows the data and services as tightly coupled, in practice there may be services that have no associated data, or data providers that rely on third-party services; conceptually, however, all of the data of interest are accessed through a service.*

Also, providers will need to register their metadata about their offerings by publishing themselves in the Portal's Registry. The registry provides dataset-level metadata for their holdings.

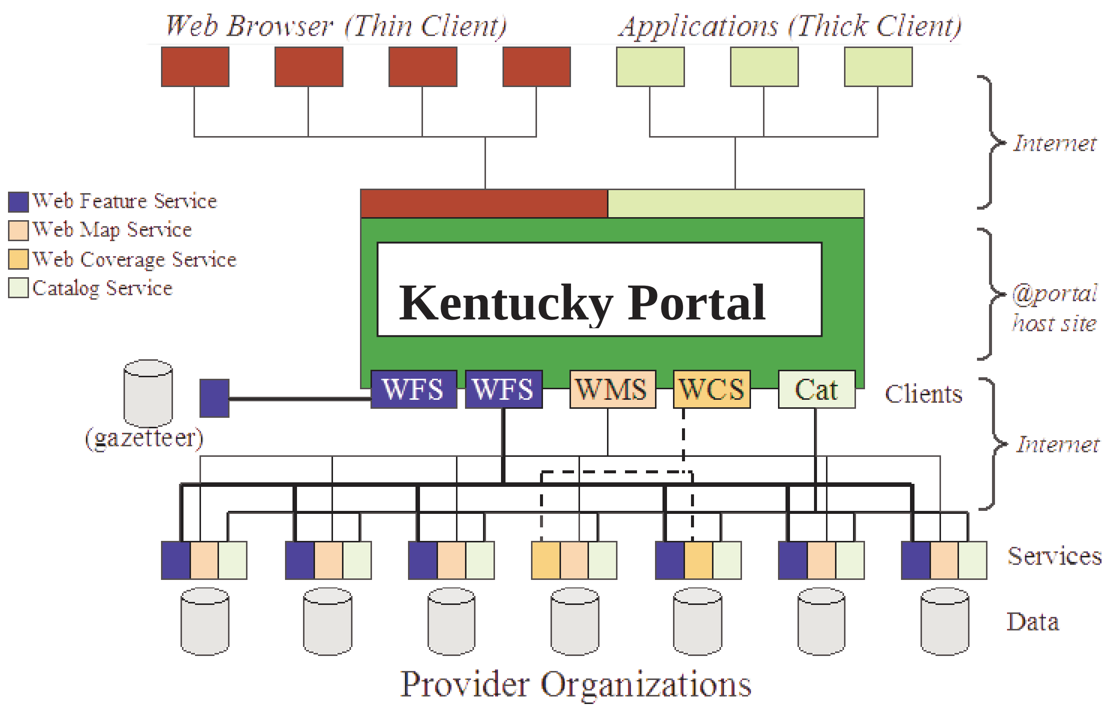

Figure 5: KWMIP Computational Viewpoint

In the center of the diagram is the Portal's application environment, which can be thought of as an assemblage of clients of these various service types. The Portal can make requests to the providers' OGC Web Services distributed throughout the Internet, receive responses, and present those responses to Portal users in an appropriate manner.

The top row of the diagram represents Portal users. The default method of access will be via a standard web browser (a "thin client") operated by a human. The system will also support any desktop client capable of using OGC interfaces. Users satisfied with the functions provided by the Portal may not require any other type of access client.

The KWMIP Computational Viewpoint is based on a subset of the output of OGC activities to date and includes concepts described in the draft OpenGIS® Reference Model.

- The OGC Reference Model establishes the baseline of common interfaces, exchange protocols, and services that have been developed or adopted by the OGC community and describes a framework that can be profiled for use in application domains like KWMIP. It can be found at: http://www.opengeospatial.org/specs/?page=orm
- OpenGIS Implementation Specifications provide guidance to application developers on how to build their products to comply with this framework.
- OpenGIS services are implementations of services that conform to OpenGIS® Implementation Specifications.
- OpenGIS applications are compliant applications that can "plug into" the network of OpenGIS services. The KLC and the KYGeonet node on The National Map are examples OpenGIS applications.

4.1 KWMIP Portal Interfaces and Data Flow Use Cases

4.1.1 Catalog Interface

The OpenGIS® Catalog Services Specification version 2[2] specifies the interfaces between clients and catalogue services, through the presentation of abstract and implementation-specific models. Catalogue services support the ability to publish and search collections of descriptive information (metadata) for data, services, and related information objects. Metadata in catalogues represent resource characteristics that can be queried and presented for evaluation and further processing by both humans and software. Catalogue services are required to support the discovery and binding to registered information resources within an information community.

Catalogue services support the use of one of several identified query languages to find and return results using well-known content models (metadata schemas) and encodings. It specifies open APIs that provide discovery services, access services and interfaces for catalog managers, including a complete Catalog Query Language. Detailed implementation guidance is provided for establishing and ending a stateful catalog session to: query the catalog server properties, check the status of a request, cancel a request, issue a query, present the query results, and get the schema of a discovered collection.

Moreover, Catalog Services support access to catalogs and registries, which are comprised of collections of metadata and types. (Catalogs and registries are essentially repositories for metadata.) Catalogs contain information about *instances* of datasets and services. Catalog Services provide a search operation that can return metadata or the names of *instances* of datasets and services. Registries contain information about *types*. *Types* are defined by well-known vocabularies. Registry Services implement a search operation that can return metadata or the names of *types*.

Catalogs based on the OpenGIS® Catalog Services Specification have the following features:

1) Catalogs enable automated discovery of and automated access to and management of machine-readable metadata describing data that are held in online spatial data repositories (and perhaps off-line data repositories) and also metadata describing online OGC Web Services[3].
2) The metadata (for both data and services) registered in catalogs must adhere to certain metadata schema standards (i.e., FGDC, ISO). Other metadata schema standards and data content standards not mandatory but are also important.
3) Whatever schemas are employed to structure the metadata, all metadata involved with Web Services are encoded using the XML.
4) The OpenGIS® Catalog Services Specification defines an SQL-like Common Query Language for search and retrieval of metadata, along with profiles of it for the OLEDB, CORBA, and Web computing environments. The Web profile uses the ANSI/NISO Z39.50 (a.k.a. ISO 23950) protocol, either on its own Internet port, or via HTTP using XML-encoded requests.
5) The data metadata provides information about how to access (view, retrieve, manipulate) the data. The data can be in any raster or vector data format (or even text or video), and they can be held in any data server. However, the data server will not be able to respond automatically to

[2] The Catalog Services Revision Work Group of the OGC Technical Committee has developed a new revision of the Catalog Services Implementation Specification 2.0 that incorporates, as a profile, a “stateless” Web interface called CS-W that is derived from earlier work on WRS including implementations developed and demonstrated in previous Interoperability Program initiatives. So in addition to the Catalog Service 1.1.1 Implementation Specification, the WRS Interoperability Program Report (Project Document 03-024) comprise the technical baseline for Catalog/Registry services for the KWMIP Portal.

[3] Note that a catalog need not provide all of these capabilities. It may provide only automated discovery of metadata about data.

access requests unless the system is online and fitted with interfaces enabling client/server communication. Typically, these will be interoperability interfaces that conform to OpenGIS® Specifications.

6) Spatial catalogs are designed to be distributed. Owner-imposed access control and security will, quite appropriately, limit people's access to some catalogs and cataloged resources, but access will not be arbitrarily limited by closed, proprietary software interfaces.

7) The OGC Catalog Interface is *stateful:* servers "remember" their clients and fill later requests based on earlier ones. However, the Web (linked by the HTTP protocol) is *stateless*: servers treat each request independently. The Web profile of the OGC Catalog Interface simulates a stateful session using an HTTP "cookie."

4.1.2 WMS Interface

The OpenGIS® Web Map Service (WMS) Implementation Specification 1.1.1 generates “pictures” of georeferenced data and allows a client to overlay map images (raster-based) for display served from multiple WMS services on the Internet. This interface, when implemented in web based GI servers and clients, enables the user to access GI from any number of sources on the web, regardless of vendor brand computing platform.

Independent of whether the underlying data are simple features (such as points, lines and polygons) or coverages (such as grided fields), the WMS produces an image of the data that can be directly viewed in a graphical Web browser or other picture-viewing software. Users can pan and zoom to find an area of interest, and then get a raster image “view” of the data. A major benefit of this approach is to provide useful information while controlling access to detailed and perhaps sensitive GI data and attributes. The displayed maps are only raster images of the source spatial data, which is a benefit if a data provider prefers not to provide the actual data. This approach is also highly beneficial as it can be quickly implemented, does not require high speed access, and provides huge benefit to the end user with little incremental cost.

The WMS specification standardizes three operations (two required and one optional) by which maps are requested by clients, and it standardizes the way that servers describe their data holdings. In addition, the WMS Specification defines a set of query parameters and associated behaviors.

The three operations (requests) are listed below and shown in Figure 6:

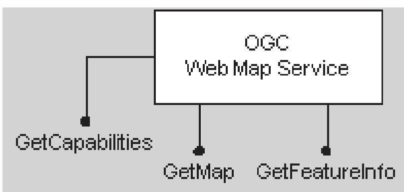

Figure 6: WMS Interface Operations

1. **GetCapabilities** (required) - returns the WMS server's service-level metadata, which is a machine-readable (and human-readable), description of the WMS service's information content and acceptable request parameters.

2. **GetMap** (required) - returns a map image whose geospatial and dimensional parameters are well defined.

3. **GetFeatureInfo** (optional) - returns information about particular features shown on a map. If a WMS server supports this operation, its maps are said to be "queryable," and a WMS client can request information about features on a map by adding to the map URL additional parameters specifying a location (as an X, Y offset from the upper left corner) and the number of nearby features about which to return information.

A WMS labels its data as one or more “Layers,” each of which is available in one or more “Styles.” Upon request a WMS makes an image of the requested Layer(s), in either the specified or default rendering Style(s). The image request (GetMap) indicates the Spatial Reference System (SRS) and Bounding Box of the portion of the Earth to be mapped, and the output width, height and format of the picture.

In essence, a WMS server can do three things:

1) Produce a map (as a picture, as a series of graphical elements, or as a packaged set of geographic feature data),

2) Answer basic queries about the content of the map, and

3) Tell other programs what maps it can produce and which of those can be queried further.

A WMS client can issue GetMap requests for different maps to several independent Map servers. If each map has the same geographic area and physical dimensions, and if their backgrounds are transparent, then they can be overlaid in a single window to produce a combined map. For example, server A might produce a topography image, server B a map of rivers and lakes, and server C a map of watershed boundaries. Each server maintains the type of data in which it specializes, but the end user can obtain a combined presentation of the three Layers.

4.1.2.1 WMS SLD Enabled Operations

The WMS Specification applies to a WMS service that publishes its ability to produce maps rather than its ability to access specific data holdings. A basic WMS classifies its georeferenced information holdings into "Layers" and offers a finite number of predefined "Styles" in which to display those layers. This basic behavior of a WMS service can be extended to allow user-defined symbolization of feature data instead of named Layers and Styles.

While a WMS currently can provide the user with a choice of style options, the WMS can only tell the user the name of each style. It cannot tell the user what each portrayal will look like on the map. More importantly, the user has no way of defining unique styling rules. The capability for a human or machine client to define these rules requires an extension - a styling language that the WMS client and WMS server can both understand.

The OpenGIS® Styled Layer Descriptor (SLD) Specification describes this extension; it provides the schema for implementing common and consistent communication of map representation, independent of the underlying GIS or database technology. The SLD specification adds the following additional operations that are not available on a basic WMS:

1. **DescribeLayer** – asks for an XML description of a map layer. The result is the URL of the WFS server containing the data and the feature type names included in the layer.

2. **GetLegendGraphic** – provides a general mechanism for acquiring legend symbols, beyond the LegendURL reference of WMS Capabilities.

3. **GetStyles** – used to retrieve user-defined styles from a WMS.

4. **PutStyles** – used to store user-defined styles into a WMS

In brief, an SLD-enabled WMS retrieves features from a WFS service and applies explicit styling information provided by the user in order to render a map. A WMS client retrieves capabilities from a WMS server. If the WMS server supports SLD, the WMS client allows the user to create custom styles on traditional WMS layers (in SLD terminology, UserStyles for NamedLayers), which then makes an SLD-enabled GetMap request to retrieve a map.

SLD is robust enough to fulfill a wide range of cartographic needs and is terse enough to be useful even using only HTTP GET as a transport method. However, some of the current SLD limitations are: (1) there is no elegant way to specify a thematic or chloropleth map. For example, the user can not encode data in four colors starting with gray and ending with black without specifying the exact data ranges for each color and the exact color value for each range: (2) the ability to create styles lacks a style library service.

In short, the WMS interface provides protocols (mainly, the GetCapabilities, GetMap, GetFeatureInfo and DescribeLayer, the last being defined as part of the SLD specification) in support of the creation and display of registered and superimposed map-like views of information that come simultaneously from multiple sources that are both remote and heterogeneous.

See the OGC WMS Cookbook[4] for implementation guidelines and examples.

4.1.2.2 WMS Use Cases

The use cases in this section describe general scenarios that involve WMS services in support of KWMIP services. These should be viewed as context for the requirements listed in the Enterprise, Information, and Engineering viewpoints presented in other sections of this Annex as well as the use cases presented in Appendix A associated with the specific functionality of the KWMIP services.

4.1.2.2.1 WMS GetCapabilities Use Cases

WMS GetCapabilities Use Case	
Name	A WMS Client Requests Capabilities XML Document
Priority	High
Description	This use case allows a WMS Client (or client proxy) to request a server to expose its service metadata [1], its mapping content [2], and capability metadata [3].
Precondition	There is a WMS available that provides its capabilities in the form of an XML document that the WMS Client understands.
Flow of Events – Basic Path	
1.	WMS Client submits a GetCapabilities request.

[4] OGC WMS Cookbook, http://www.opengeospatial.org/resources/?page=cookbooks

WMS GetCapabilities Use Case	
2.	WMS validates the request.
3.	WMS returns the capabilities. These include a list of layers that the WMS Client wants to display.
4.	WMS Client accepts the WMS capabilities, parses them and uses the information for subsequent queries (GetFeatureInfo, GetMap, or DesribeLayer, last being defined as part of the SLD specification).
Flow of Events – Alternative Paths	
2	If the request is invalid, it is rejected and a service exception report is returned.
Postcondition	WMS Clients would generally run the GetCapabilities operation and cache its result for use throughout a session or reuse it for multiple sessions.

[1] Service metadata includes the following: Names, Titles, Online Resource URL. Optionally, Abstract, Keyword List, Contact Information, Fees, and Access Constraints may be provided.

[2] The most critical part of the WMS Capabilities XML is the Layers and Styles it defines.

A single parent Layer encloses any number of additional layers, which may be hierarchically nested as desired. Some properties defined in a parent layer are inherited by the children it encloses. These inherited properties may be either redefined or added to by the child.

A WMS server shall include at least one <Layer> element for each map layer offered. If desired, layers may be repeated in different categories when relevant.

A Layer is said to have been "cascaded" if it was obtained from an originating server and then included in the Capabilities XML of a different server. The second server may simply offer an additional access point for the Layer, or may add value by offering additional output formats or spatial reference systems.

If the optional opaque attribute is missing or has a value of "0," then maps made from that Layer will generally have significant "no-data" areas that a client may display as transparent. Vector features such as points and lines are considered not to be opaque in this context (even though at some scales and symbol sizes a collection of features might fill the map area). A value of "1" indicates that the Layer represents an area-filling coverage of space. For example, a map that represents topography and bathymetry as regions of differing colors will have no transparent areas. The "opaque" declaration should be taken as a hint to the Client to place such a Layer at the bottom of a stack of maps.

[3] Capability metadata includes names the actual operations that are supported by the service instance, the output formats offered for those operations, and the URL prefix for each operation.

4.1.2.2.2 GetFeatureInfo Use Cases

WMS GetFeatureInfo Use Case	
Name	A WMS Client Request

WMS GetFeatureInfo Use Case	
Priority	Medium (optional request)
Description	This use case enables a user to click on a pixel and inquire about the schema and metadata values of the feature(s) represented there [1].
Precondition	
Flow of Events – Basic Path	
1.	WMS Client submits a GetFeatureInfo request, specifying the layers to be queried [2], the format info [3], the feature count [4], and the x and y [5].
2.	WMS validates the request.
3.	WMS returns a description for the specific features selected [6].
4.	WMS Client accepts the requests and process it for further requests (i.e., GetMap).
Flow of Events – Alternative Paths	
2	If the request is invalid, it is rejected and a service exception report is returned [7].
Postcondition	

[1] The GetFeatureInfo operation is designed to provide clients of a WMS with more information about features in the pictures of maps that were returned by previous Map requests. The canonical use case for GetFeatureInfo is that a user sees the response of a Map request and chooses a point on that map for which to obtain more information. The basic operation provides the ability for a client to specify which pixel is being asked about, which layer(s) should be investigated, and what format the information should be returned in.

Because the WMS protocol is stateless, the GetFeatureInfo request indicates to the WMS what map the user is viewing by including most of the original GetMap request parameters (all but VERSION and REQUEST). From the spatial context information (BBOX, SRS, WIDTH, HEIGHT) in that GetMap request, along with the X,Y position the user chose, the WMS can (possibly) return additional information about that position.

The actual semantics of how a WMS decides what to return more information about, or what exactly to return is left up to the WMS provider.

[2] GetFeatureInfo is only supported for those Layers for which the attribute queryable="1" (true) has been defined or inherited

[3] The optional INFO_FORMAT indicates what format to use when returning the feature information (i.e., GML).

[4] The optional FEATURE_COUNT parameter states the maximum number of features for which feature information should be returned.

[5] The required X and Y parameters indicate a point of interest on the map. X and Y identify a single point within the borders of the WIDTH and HEIGHT parameters of the embedded GetMap request.

[6] The GetFeatureInfo response is according to the requested INFO_FORMAT if the request is valid, or issue an exception otherwise. The nature of the response is at the discretion of the WMS provider, but it shall pertain to the feature(s) nearest to (X,Y).

[7] For example, a client shall not issue a GetFeatureInfo request for layers for which the attribute queryable="0" (false) has been defined. A WMS should respond with a properly formatted Service Exception response if it encounters that request but does not support it.

4.1.2.2.3 GetMap Use Cases

WMS GetMap Use Case	
Name	A WMS Client Requests a Map
Priority	High
Description	This use case allows a WMS Client (or client proxy) to request multiple servers to craft “map overlays”, possessing the identical spatial reference system, size, scale, and pixel geometry. These overlays can be ordered and placed by the client into a display, and optionally, using transparent pixel technology, the information from several sources can be rendered for immediate viewing (display).
Precondition	
Flow of Events – Basic Path	
1.	WMS Client submits a GetMap request, specifying the Layers, Styles [1], SRS [2], BBox [3], Format [4], Width and Height [5], Transparent [6], BGColor [7].
2.	WMS validates the request.
3.	WMS returns the maps generated from the request parameters [8]
4.	WMS Client accepts the responses and process it for display.
Flow of Events – Alternative Paths	
2	If the request is invalid, it is rejected and a service exception report is returned [9].
Postcondition	

[1] Each style Name shall be one that was defined in the <Name> element of a <Style> element that is either directly contained within, or inherited by, the associated <Layer> element in Capabilities XML. (In other words, the Client may not request a Layer in a Style that was only defined for a different Layer.)

[2] If the WMS server has declared SRS=NONE for a Layer then the Layer does not have a well-defined spatial reference system and should not be shown in conjunction with other layers. The Client shall specify SRS=NONE in the GetMap request and the Server may issue a Service Exception otherwise.

[3] If the WMS server has declared that a Layer is not subsettable/resizable, then the Client shall specify exactly the declared Bounding Box values in the GetMap request and the Server may issue a Service Exception otherwise.

[4] Typical output formats include PNG, GIF, JPEG, TIFF, etc. When the data do not cover the entire field of view (such as a network of roads that includes the space between the roads), the background can be made transparent in some output formats.

[5] In the case where the aspect ratio of the BBOX and the ratio width/height are different, the WMS shall stretch the returned map so that the resulting pixels could themselves be rendered in the aspect ratio of the BBOX. In other words, it should be possible using this definition to request a map for a device whose output pixels are themselves non-square, or to stretch a map into an image area of a different aspect ratio.

NOTE: Map distortions will be introduced if the aspect ratio WIDTH/HEIGHT is not commensurate with X, Y and the pixel aspect. Client developers are cautioned to minimize the possibility that users will inadvertently request or unknowingly receive distorted maps.

If the WMS server has declared that a Layer has fixed width and height (it's not subsettable and resizable, then the Client shall specify exactly those WIDTH and HEIGHT values in the GetMap request and the Server may issue a Service Exception otherwise.

[6] Transparent specifies whether the map background is to be made transparent or not. The ability to return pictures drawn with transparent pixels allows results of different Map requests to be overlaid, producing a composite map. It is strongly recommended that every WMS offer a format that provides transparency for layers which could sensibly be overlaid above others.

NOTE: At the time of this writing, the image/gif format provides transparency and is properly displayed by common web clients. The image/png format provides a range of transparency options.

When TRANSPARENT is set to TRUE and the FORMAT parameter contains a Picture format (e.g., image/gif), then a WMS shall return (when permitted by the requested format) a result where all of the pixels not representing features or data values in that Layer are set to a transparent value. For example, a "roads" layer would be transparent wherever no road is shown. When TRANSPARENT is set to FALSE, those pixels shall be set to the value of BGCOLOR

When the Layer has been declared as "opaque", then significant portions, or the entirety, of the map may not be able to made transparent.

When the FORMAT parameter contains a Graphic Element format, the TRANSPARENT parameter may be included in the request but its value shall be ignored by the WMS.

[7] When FORMAT is a Picture format, a WMS shall render its output on a background whose pixels were initially uniformly of the color encoded in BGCOLOR. When FORMAT is a Graphic Element format (which does not have an explicit background), a WMS should avoid use of the BGCOLOR value for foreground elements because they would not be visible against a background picture of the same color.

When the Layer has been declared as "opaque," then significant portions, or the entirety, of the map may not show any background at all.

[8] The response to a valid GetMap request shall be a map of the georeferenced information layer requested, in the desired style, and having the specified spatial reference system, bounding box, size, format and transparency.

[9] For example, a server shall throw an exception (code=StyleNotDefined) if an unadvertised Style is requested.

4.1.3 WFS Interface

The OpenGIS® WFS Implementation Specification describes feature data access such that servers and clients can “communicate” at the feature level. The WFS interface also defines simple transaction operations (i.e., Create a Feature, Delete a feature, and Update a feature) on OpenGIS® Simple Features (feature instances). Note, however, that since the Kentucky Portal will act as a discovery and access gateway for data query, mapping, and reporting, it is not conceived to utilize functionality associated with data editing.

Whereas WMS delivers a picture, WFS implemented in a client supports the dynamic exploitation and access of feature data and associated attributes on the Web from any server product that implements WFS. This capability opens the door to enhanced spatial analysis, modeling and other operations based on the intelligence of the attributed data.

A Web-accessible commercial GIS server with an interface based on the WFS specification provides a similar capability to the WMS except that the actual vector GIS data can be transmitted to the client for further processing. The WFS interface provides the ability to have a common and consistent access mechanism to vector (coordinate) data stores, such as ESRI Shape files, Oracle Spatial tables, and so forth. It allows applications to be built and deployed independent of the underlying GIS technology.

Therefore, a WFS request – like those supported in many GIS and RDBMS packages – consists of a description of the query and data transformation operations that are to be applied to WFS enabled spatial data warehouses on the Web. The request is generated on the client and is posted to a WFS server. The WFS Server “reads” and executes the request returned in a feature set as GML. A GML enabled client then can use the feature set.

The WFS specification standardizes the following operations:

1) **GetCapabilities** (required for Basic WFS): requests a basic description of the WFS service instance. The service responds with a Capabilities XML document that contains a description of all the operations that the WFS supports and a list of all feature types that it can service.

2) **DescribeFeatureType** (optional for Basic WFS): requests a detailed description of specific feature types. The response is a GML Application Schema documents specifying the GML encoding for the feature type. Allows the client to infer the format of a GML encoded representation of the feature, which it may use (for example) to form the details of a GetFeature request. For example, DescribeFeatureType (Road) request, gives road and its properties and their types or content models.

3) **GetFeature** (required for Basic WFS): requests the digital representation of specific feature instances. The request specifies: (a) the feature type(s) of interest, (b) conditions to select the set of instances, and (c) the subset of properties that should be included in the response (using filter expressions (queries). The service responds with a Feature or FeatureCollection(s) (complex object(s)) containing the requested features.

The other two operations (Transaction and LockFeautre) are associated with data editing and are not conceived to be part of the KWMIP application.

The WFS specification is accompanied by the OpenGIS® Filter Encoding Implementation Specification, which defines a grammar for query filters for the GetFeature request. If a WFS request contains a filter, the WFS server only selects those features that pass through it. A filter is like a WHERE clause in SQL, but written in XML and capable of expressing conditions (i.e., spatial, comparison, logical, arithmetic).

A client is not restricted to requesting information. It can also post new or updated information to a WFS server. This operation is called a "transaction" and it enables data manipulation operations on geographic features (over HTTP) that include the ability to create, delete, update, and/or get (query) features based on spatial and non-spatial constraints.

In brief, the WFS and the Filter Encoding specifications together support the following operations:

- Retrieving the capabilities of a WFS
- Retrieving feature description in the form of GML application schemas
- Retrieving feature instances based on criteria that are expressed as OGC Filters (using only the spatial, comparison, logical, or arithmetic operators defined in the Filter Encoding specification)

The following example outlines, in general terms, the typical interaction sequence of a WFS request:

1. The Web client sends a GetCapabilities request to get a Capabilities XML document from a WFS server, which it then processes.
2. The Web client (optionally) makes a DescribeFeatureType request to a WFS server to get Feature schema – the definition of one or more of the feature types to get the names and types of the properties in order to be able to form meaningful query or transaction requests. Also, it is this stage where Level 0 Profile of GML for WFS is connected with.
3. Based on the definition of the feature type(s) in the received GML Application Schema, the client constructs scripts, data structures, etc. for data processing.
4. The Web client generates a GetFeature request as specified in the Capabilities XML document,
5. The request is posted to a Web (http) server.
6. The WFS server is invoked to read and service the request.
7. The WFS server sends back a Feature or a FeatureCollection. (In the event that an error has occurred, the WFS server's status report will indicate that fact.)
8. The Web client process the received FeatureCollection and graphically displays the results.

4.1.3.1 WFS Use Cases

The use cases in this section describe general scenarios that involve WFS services in support of KWMIP. These should be viewed as context for the requirements listed in the Enterprise, Information, and Engineering viewpoints presented in other sections of this Annex as well as the use cases presented in Appendix A associated with the specific functionality of the KWMIP.

4.1.3.1.1 WFS GetCapabilities Use Case (Basic WFS)

WFS GetCapabilities Use Case	
Name	A WFS Client Requests Capabilities XML Document
Priority	High (required for a basic WFS)
Description	This use case allows a WFS Client to learn about the WFS service itself [1], the capabilities of a WFS service [2], a list of all feature types that a WCS can service [3], and the WCS's filter capabilities [4].
Precondition	There is a WFS available that provides its capabilities in the form of an XML document that the WFS Client understands.
Flow of Events – Basic Path	
1.	WFS Client submits an GetCapabilities request.
2.	WFS validates the request.
3.	WFS returns the capabilities. These include a list of feature types that the WFS Client wants to query.
4.	WFS Client accepts the WFS capabilities, parses them and uses the information for subsequent queries (i.e., DescribeFeatureType, GetFeature).
Flow of Events – Alternative Paths	
2	If the request is invalid, it is rejected and a service exception report is returned.
Postcondition	WFS Clients would generally run the GetCapabilities operation and cache its result for use throughout a session or reuse it for multiple sessions.

[1] The following content can be specified for the service metadata: Name, title, abstract, keyword (to aid catalog searching), HTTP URL (home page), fees, and access constraints.

[2] The capabilities section of the Capabilities XML document specifies the list of requests that the WFS can handle. Two classes of WFS services, based on the capabilities they support, are WFS Basic and WFS Transaction.

[3] The feature info section of the Capabilities XML document defines the list of feature types (and operations on each feature type) that are available from a WFS service. The following elements can be used to describe each feature type contained in a feature type list: Name, Title, Abstract, Keyword (to aid catalog searching), SRS (i.e., 'EPSG:<POSC Code>', URL format), Operation, , MetadataURL ('TC211= ISO TC211 19115; 'FGDC' = FGDC CSDGM), and LatLongBoundingBox (facilitates geographic searches by indicating where instances of the particular feature type exist. Since multiple LatLongBoundingBoxes can be specified, a WFS can indicate where various clusters of data may exist. This knowledge aids client applications by letting them know where they should query in order to have a high probability of finding data.)

[4] This is an optional section. If it exists, then the WFS should support the operations advertised/defined in the Filter Encoding Implementation Specification. If the Filter Capabilities Section as part of the

GetCapabilities request is not defined, then the client should assume that the server only supports the minimum default set of filter operators as defined in the Filter Encoding Implementation Specification.

4.1.3.1.2 WFS DescribeFeatureType Use Case (Basic WFS)

WFS DescribeFeatureType Use Case	
Name	A WFS Client Requests a Description of Specific Feature Types
Priority	High (required for a basic WFS)
Description	This use case allows a WFS Client to infer the format of a GML encoded representation of a road feature. It uses this information (i.e., feature properties, feature types, feature content models) to form a specific GetFeature request.
Precondition	WFS Client has previously obtained the capabilities of the WFS. From inspecting the capabilities document, the client was able to determine what the WFS server's feature types are.
Flow of Events – Basic Path	
1.	WFS Client submits a DescribeFeatureType request [1].
2.	WFS is invoked to read and service the request.
3.	WFS validates the request and executes it by returning the schema for the Feature or FeatureCollection [2].
4.	WFS Client uses the definitions of one or more of the feature types defined in the schema to get the names and types of the properties for forming a specific GetFeature request (or a Transaction request) [3].
Flow of Events – Alternative Paths	
2	If the request is invalid, it is rejected and a service exception report is returned.
3	If the execution of the request fails for some reason, WFS returns a service exception report.
Postcondition	N/A

[1] Indicates what schema description languages can be used to describe the schema of a feature type when a client requests such a description; it will define how feature instances are to be encoded on input and hot they will be generated on output. XMLSCHEMA is the only mandatory language that must be available. The SCHEMALANGUAGES entity can be redefined to include vendor specific formats/languages.

[2] The DescribeFeature request contains zero or more TypeName elements that encode the names of feature types that are to be described. If the content is empty, then that shall be interpreted as requesting a description of all feature types that a WFS can service.

[3] In response to a DescribeFeatureType request, where the value of the output format attribute has been set to XMLSCHEMA, a WFS implementation must be able to present an XML Schema document that is a valid GML application schema and defines the schema of the feature types listed in the request. The document(s) presented by the DescribeFeatureType request may be used to validate feature instances generated by the WFS in the form of feature collections on output or feature instances specified as input for transaction operations.

As specified by GML, the feature schema definition is entirely at the discretion of the particular WFS implementation that is describing its feature types. The only caveats are:

1. Feature geometry must be expressed using the GML geometry description. (geometry.xsd).

2. Spatial Reference Systems must be expressed as defined in the GML Implementation Specification, version 2.1.1.

3. The feature schema must be consistent with the OGC feature model. This means that the feature schema defines properties of the feature. The GML interpretation of this statement is that the elements nested below the root element of a feature type define properties of that feature.

4.1.3.1.3 WFS GetFeature Use Case (Basic WFS)

WFS GetFeature Use Case	
Name	A WFS Client Request Feature(s)
Priority	High (required for a basic WFS)
Description	This use case allows a WFS Client to retrieve one or more feature instances of a Road. These instances conform to a GML 3.0 application schema that is understood by the WFS Client [1].
Precondition	WFS Client has previously obtained the capabilities of the WFS. From inspecting the Capabilities XML document, the client was able to determine that the WFS can export Road feature types that the client can understand. In addition, a client application can determine the properties of a feature by making a DescribeFeatureType request before composing a GetFeature request [2].
Flow of Events – Basic Path	
1.	WFS Client submits a GetFeature request [3]. The request specifies: (a) that the feature is a Road, (b) to select the Road (or a segment of it) by using the BBOX filter or the other ten spatial operators defined in the OGC Filter Encoding specification, i.e., Crosses, Overlaps, etc. (a filter could also be non-geographic, i.e., select all Roads that are “two lanes”), and (c) to use the geometry property of the Road feature (i.e., LineString) that should be included in the response.
2.	WFS is invoked to read and service the request.
3.	WFS validates the request and executes it by returning the feature instances matching the request parameters. An XML document, containing the result set, is returned to the client [4].

WFS GetFeature Use Case	
4.	WFS Client accepts the response message and processes the received feature instances. It then provides the information to the end-user by graphically displaying the results (or downloading it to a local repository).
Flow of Events – Alternative Paths	
2	If the request is invalid, it is rejected and a service exception report is returned.
3	In the event that an error has occurred, the WFS's status report will indicate that fact.
Postcondition	N/A

[1] GetFeature enumerates the formats available for expressing the results of a query. The RESULTFORMATS entity defines the mandatory output format of GML but can be redefined to include additional vendor specific formats.

[2] The DescribeFeatureType operation will generate a GML application schema defining the schema of the feature type. The client can then select the properties to be fetched. In addition, the client can determine which feature properties are mandatory and must be fetched in order for the WFS to be able to generate an instance of the feature type that will validate againt the generated GML application schema. In the event that a WFS encounters a query that does not select all mandatory properties of a feature, the WFS will internally augment the property name list to include all necessary property names. A WFS client must thus be prepared to deal with a situation where it receives more property values than it requests.

[3] The optional maxFeatures attribute can be used to limit the number of features that a GetFeature request retrieves. Once the maxFeatures limit is reached, the result set is truncated at that point.

Also, a GetFeature request can issue a GetFeatureWithLock to indicate to a WFS service to attempt to lock the features that are selected; presumably to update the features.

[4] The format of the response to a GetFeature request is controlled by the outputFormat attribute. The default value for the outputFormat attribute shall be GML2. This will indicate that a WFS must generate a GML document of the result set, and more specifically, the output must validate against the GML application schema generated by the DescribeFeatureType operation (see above).

Any GML document generated by a WFS implementation, in response to a query where the outputFormat is GML2, must reference an appropriate GML application schema document so that the output can be validated. This can be accomplished using the schemaLocation attribute. This attribute provides hints as to the physical location (in the form of a URI) of one or more schema documents which may be used for local validation and schema-validity assessment.

For the GetFeatureWithLock request, a WFS must generate a result that includes the lock identifier.

4.1.4 WCS Interface

The WCS Implementation Specification v1.0.0 specification defines three interface operations that explain how WCS serves to describe, request, and deliver multi-dimensional coverage data over the Web that represent values or properties of geographic locations. Version 1.0.0 of the specification emphasizes "simple" coverages (defined on some regular, rectangular grid or tesselation of space); and anticipates other coverage types as defined in the OpenGIS® Abstract Specification (Topic 6, "The Coverage Type," OGC document #99-106).

The three WCS interface operations are:

1) **GetCapabilities** (required): this operation is used by a client to request WCS server's capabilities, which are defined in an XML document conveying general information about the service itself, and specific information about the available data collections from which coverages may be requested. Current limitation (version 1.0.0) of this operation is that there is no ability to retrieve desired parts of the full Capabilities XML document.

2) **DescribeCoverage** (optional): Client may be able to formulate simple GetCoverage requests based only on the information received from the GetCapabilities XML document (elements defined in the CoverageOfferingBrief provide a summary-level description of coverage data available from a given service). However, in order to make more finely tuned GetCoverage requests, WCS clients will need to obtain further details about a particular coverage, using the DescribeCoverage operation. Current limitation (version 1.0.0) of this operation is that coverage range sets are defined only as single homogenous "range component." Also, only points at an external description are available even to get basic info like coverage range, observable or value space.

3) **GetCoverage** (required): this operation allows retrieval of coverages from a coverage layer. A WCS server processes this request from a client and returns a result set to the client. Current limitation (version 1.0.0) of this operation is that there is no ability to retrieve elevation subsets of a coverage beyond current regularly spaced (grid) elevation. Also, there is no ability to retrieve spatial subsets of a coverage, beyond current regularly spaced (grid) elevation.

Both the WMS and the WCS servers provide for the generation and delivery of raster-based information. The WMS server returns an "image map," that is, an array of pixel values ready for portrayal. In contrast to the WMS server, where only visualization is accomplished, the WCS server preserve data values from grid coverages (i.e., Elevation, Temperature, etc.) and enables more than picture display (i.e., provides numerical input to models, supports multi-valued coverages, allows client-side rendering). This capability enables analysis involving the evaluation/manipulation/combination of multiple coverages to answer specific questions. These values must be further processed if they are to be portrayed. This access to intact, unprocessed GI is needed for client side processing, multi-valued coverages, and input into scientific models and other clients beyond simple viewers. (The WFS service, by contrast to both the WMS and WCS services, returns a collection of vectors (features) that inform the client of values of interest, such as temperature, ownership, average rainfall, and so on.)

WCS was designed to remain compatible with WMS (i.e., retain similar query and metadata syntax), but it does provide for richer query and metadata such as multi-dimensional data and queries (e.g., time series, multi-band imagery). Other key aspects of WCS are one Layer at a time. Any overlays are on the client-side. Also, there are no styles or legends, nor pixel width or height with WCS.

Some of WCS implementation challenges include the following. Thin clients are inadequate and, as a result, writing a special-purpose coverage client is essential. This might include adding a Web connection to Raster GIS and configuring the browser with a "helper app." Also, there is no consensus on coverage encodings, which currently may include the following:

- Proprietary formats (ECW, ESRI BIL, etc.)
- Specialized formats (HDF-EOS, DTED, etc.)
- Common web formats (PNG, GeoTIFF)
- XML Coverage encodings:

- Earth Science Markup Language
- eXtensible Data Format (XDF)
- GML Coverage encoding

4.1.4.1 WCS Use Cases

The use cases in this section describe general scenarios that involve WCS services in support of KWMIP. These should be viewed as context for the requirements listed in the Enterprise, Information, and Engineering viewpoints presented in other sections of this Annex as well as the use cases presented in Appendix A associated with the specific functionality of the KWMIP.

4.1.4.1.1 GetCapabilities Use Case

WCS GetCapabilities Use Case	
Name	A WCS Client Requests Capabilities XML Document
Priority	High
Description	This use case allows a WCS client to get an XML document containing WCS service information and data collections information [1].
Precondition	There is a WCS available that provides its capabilities in the form of an XML document that the WCS Client understands.
Flow of Events – Basic Path	
1.	WCS Client submits a GetCapabilities request.
2.	WCS validates the request.
3.	WCS returns the capabilities. These include a list of basic coverage properties that the WCS Client wants to retrieve.
4.	WCS Client accepts the WCS capabilities, parses them and uses the information for subsequent requests (DescribeCoverage or GetCoverage).
Flow of Events – Alternative Paths	
2	If the request is invalid, it is rejected and a service exception XML message is returned to describe to the client application or its user the reason(s) that the request is invalid.
3	If the GetCapabilities operation cannot return descriptions of the WCS server's available data, this metadata information must be available from a separate source, i.e., image catalog service [2].
Postcondition	WCS Client knows which coverage types it can query. WCS Clients would generally run the GetCapabilities operation and cache its result for use throughout a session or reuse it for multiple sessions.

[1] Coverage properties include service metadata (i.e., access constraints, fees, etc.), service capability (i.e., the requests that the WCS supports, the formats in which exceptions are returned, and any other vendor-specific service capabilities); and content metadata (i.e., the bounding box for the spatial extent).

[2] This is done using XLink pointer that lists the URL of a catalog that clients can search for coverage descriptions (and GML's remoteSchema for stating the schema of the remote resource) in order to make appropriate DescribeCoverage or GetCoverage requests. This is intended for servers with thousands/millions of coverage offerings, for which searching a catalog is more feasible than fetching a long XML document.

4.1.4.1.2 DescribeCoverage Use Case

WCS DescribeCoverage Use Case	
Name	A WFS Client Requests a Description of Coverage(s)
Priority	Medium (optional operation)
Description	This use case allows a WCS Client to get a full description of one or more coverages served by a particular WCS server. It uses the coverage properties (i.e., coverage type [1], spatial [2] /temporal [3] domain, range [4], internal grid structure of a coverage [5], supported formats [6], interpolation methods [7]) to form a fine-grained GetCoverage request [8].
Precondition	WCS Client has previously obtained the capabilities of the WCS. From inspecting the capabilities document, the client was able to determine what the WCS server's coverage type is.
Flow of Events – Basic Path	
1.	WCS Client issues the DescribeCoverage request [9].
2.	WCS validates the request and executes it by returning the description for a specified coverage.
3.	WCS server responds with XML document that fully describes the identified coverage.
Flow of Events – Alternative Paths	
2	If the request is invalid, it is rejected and a service exception XML message is returned to describe to the client application or its user the reason(s) that the request is invalid.
3	If the execution of the request fails for some reason, the WCS server returns a service exception report [10].
Postcondition	N/A

[1] Coverage type is limited to grid type only in the current WCS specification. Image types (a coverage that is not available in georectified form) are not supported as the WCS spec does not specify how to request, encode, or transmit the additional information (i.e., control points or sensor metadata).

Nevertheless, an image CRS can be embedded in the coverage response, available from a separate source, or otherwise known to the client.

[2] A detailed spatial domain of a coverage would entail using (repeatable) GML Polygon, representing the polygon(s) covered by the coverage spatial domain. This is particularly useful for areas that are poorly approximated by a GML Envelope (i.e., satellite image swaths, island groups, other non-convex areas).

For each coverage offering, supported coordinate reference systems (CRSs) can be listed by which the server understands (and respond to) incoming GetCoverage requests.

[3] The temporal domain describes the valid time constraints (the times for which valid data are available) for GetCoverage requests (either as sequence of time instants or continuous).

[4] Range can be defined for the coverage properties (categories, measures, or values) assigned to each location in the domain. Any such property may be scalar -- numeric or text – value (i.e., population density, terrain elevation, yesterday's max temperature) or a compound -- vector or tensor – value (i.e., income by race, climate pattern, multi-spectral radiance: brightness by wavelength).

Compound values consist of a set of identically defined measurements or observations, reported for each of several values of control variable, or aggregated into several bins (e.g., any independent variable besides those in the domain), which a GetCoverage request may use for constraints. A compound range set may have more than one control parameter or a set of bins, for quantities related to values of several parameters (e.g., counts of wildlife tabulated both by size and species).

Compound valued range set is designed for observations that are identically defined (report the same property expressed in the same reference system). If a set of observations has any semantic variation or any differences in the reference system, then the different kinds of observations belong in different coverages.

[5]This can include information such as the coverage's native resolution so that the client can formulate grid coverage requests expressed in the internal grid coordinate reference system.

[6] Any output format(s) is acceptable, provided that at least one of the following formats is supported: GeoTIFF, HDF-EOS, DTED, NITF, GML. Other formats need to be listed as part of the DescribeCoverage response.

[7] A WCS server may have the capability to interpolate coverage values over the spatial domain when a request requires resampling, reprojection, or other generalization. A coverage offering may list any of six spatial interpolation methods: (1) nearest neighbor (default); (2) bilinear; (3) bicubic; (4) lost area; (5) barycentirc; and (6) none.

[8] Note that the spatial extent can be specified using the bounding box information from within the Capabilities XML document returned after the GetCapabilities request. However, the intent is to describe the locations in more detail, e.g., in several different CRSs, or several rectangular areas instead of one overall bounding box.

[9] A request that lists no coverages shall be interpreted as requesting descriptions of all coverages that a WCS can serve.

[10] Since server support for DescribeCoverage is optional, therefore, if a server does not support it, it must return an exception rather than the requested list.

4.1.4.1.3 GetCoverage Use Case

WCS GetCoverage Use Case	
Name	A WFS Client Requests Coverage(s)
Priority	High
Description	This use case allows a WCS client to request a coverage (that is, values or properties of a set of geographic locations) bundled in a well-known coverage format.
Precondition	Normally run after GetCapabilities and DescribeCoverage replies have shown what request is allowed and what data are available.
Flow of Events – Basic Path	
1.	WCS Client submits a GetCoverage request, specifying the following parameters: CRS [1], BBox [2], Time [3], grid size and resolution [4], format, source [5], domain subset [6], range subset [7], interpolation method [8], output CRS and format [9].
2.	WCS validates the request and executes it.
3.	WCS returns the coverage extracted from the coverage request, with the specified spatial reference system, bounding box, size, and format.
4.	WCS Client accepts the response and provides the information to the end-user application.
Flow of Events – Alternative Paths	
2	If the request is invalid, it is rejected and a service exception XML message is returned to describe to the client application or its user the reason(s) that the request is invalid.
3	If the execution of the request fails for some reason, the WCS server returns a service exception report.
Postcondition	N/A

[1] Future versions of WCS spec may address ways to request multiple coverages, combining them according to mathematical or logical operators (Boolean or other ruled-based overlay).

The CRS parameter is required. Note that if the Capabilities XML document lists only “Image” type, then client must request that coverage offering in its internal (local/pixel) coordinate reference system, by specifying ‘CRS=Image’ in the GetCoverage request.

Some WCS servers may support on-the-fly georectification of coverages that are georeferenced but not already georectified. Such servers accept requests expressed in a coverage’s internal pixel/local coordinate system, but are able to express coverage replies in a ground coordinate system (this requires adding a ‘RESPONSE_CRS’ value).

[2] A GetCoverage request may include a 1-D, 2-D, or 3-D spatial constraints.

[3] If the DescribeCoverage XML reply defines a temporal domain on the selected coverage, GetCoverage requests may use a separate TIME parameter to constrain the request in time, thus supplementing a spatial BBOX.

[4] If the Capabilities XML document reports only the Interpolation method “None” for the queried coverage, then GetCoverage request must be for the full native resolution of the data; they may not use RESX, RESY, RESZ, or WIDTH, HEIGHT, DEPTH to change the coverage resolution. In this case, BBOX alone is used for subsetting.

[5] The source (any URI) for the coverage specifies a single coverage available from the WCS server. It’s value must match that of the value obtained from the WCS server’s Capabilities XML document, or from a third-party catalog.

[6] Similar to the domain specified in the DescribeCoverage return description.

Note: In response to a GetCoverage request, a WCS server will return a grid of the requested size covering the requested area. This usually requires interpolating/resampling the coverage values stored on the server. To avoid this, clients should request the coverage in a native CRS stated by the server and select a GML Envelope whose extent exactly matches that of the requested GML Grid (or GML Rectified Grid). For such a request, if the chosen CRS is “Image,” the Envelope and Grid must both describe grids of the same size. For other CRSs, the Envelope and Grid must be related by vector offset values in the coverage description (if supplied in the coverage description).

[7] In the case of a compound range set, client may request subsets by constraining the value of a range axis/parameter.

[8] Any of the six spatial interpolation methods: (1) nearest neighbor (default); (2) bilinear; (3) bicubic; (4) lost area; (5) barycentirc; or (6) none.

[9] Values for these elements must be among the listed in the DescribeCoverage XML reply.

5 Engineering Viewpoint

The Enterprise, Information, and Computation viewpoints describe a system in terms of its purposes, its content, and its functions. The Engineering viewpoint relates these to specific components linked by a communications network. This viewpoint is concerned primarily with the interaction between distinct computational objects: its chief concerns are communication, computing systems, software processes and the clustering of computational functions at physical nodes of a communications network.

Figure 7 shows the engineering view of the KWMIP. The server is located at the Governor’s Office of Technology. The portal server consists of the Portal Application Environment, the Viewer Client Generator, the WFS Client Generator and the WRS Client Generator (Note: WRS became WC-S). The WFS, WMS, and WCS servers are located TBD in the commonwealth. The Portal will open with access to two remote WMS and WFS data sources: Cities and Counties TBD. The modular nature of the system and the online registry allows additional cities and counties to join the system without requiring any physical changes in the system.

The server of the portal also hosts the Portal’s registry. This is a registry service, which is maintained and operated locally by the Portal service provider. It is accessible by the Publisher Client Generator. It maintains a list of Primary information sources and their service metadata. It may temporarily cache metadata regarding user-discovered Secondary and Tertiary information sources. It may store predefined SLD documents for WMS related services.

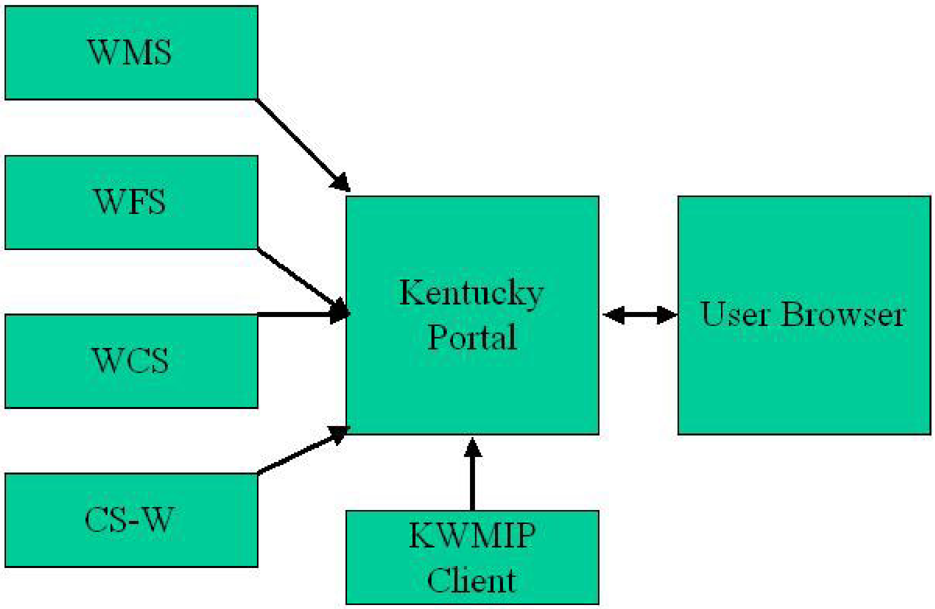

Figure 7: KWMIP Engineering Viewpoint

5.1 KWMIP UML Component Diagrams

In this section we present the component diagrams of the Portal.

Figure 8 shows the User Interface component diagram. The KWMIP Application Environment provides the integrated user interface to access the main Portal services. The Discovery Client provides interfaces to search for web resources that are registered in the Portal's registry. The Viewer Client allows users to create and display maps.

The WFS Client allows users to formulate queries that can be submitted to the accessible WFSs. These queries are based on the OGC Filter Encoding Service.The thin clients enable users to access the following services:

Search and discovery services: users can search for services in the local registry. The Portal has a discovery user interface, which allows them to search for desired services based on the stored metadata as well as capabilities elements.

Map viewing service: the Portal is envisioned to have a powerful and flexible map viewing user interface with the following elements:

- A map showing all of the displayed data types.
- Controls for navigating, zooming and panning the map.
- Menu of active data types, allowing user to toggle displayed state on/off.
- Metadata elements for each active type.
- Link to display full metadata record for each active data type.
- Link to help page(s) appropriate to current application state.

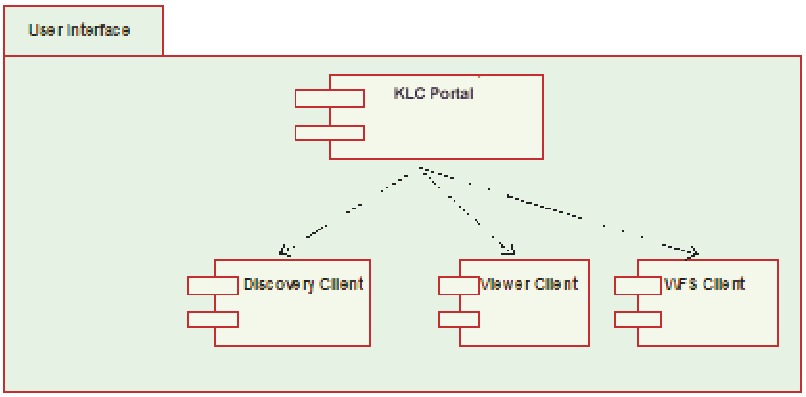

Figure 8: User Interface Package

Figure 9 shows the Discovery Client, which allows users to search and display the metadata through the getRecord and the getResourceByID respectively.

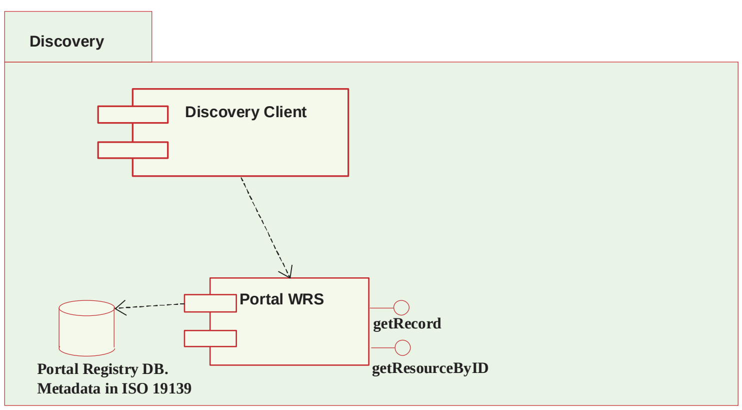

Figure 9: Discovery Package

Once the resources of interest are discovered the user can either invoke the WFS Query Client or the WMS Viewer Client. It is expected that the registry records include the URL to the capabilities document of the discovered WFS or the WMS resources.

As shown in **Figure 10**, when the Viewer Client is launched, the client parses the capabilities document to display information about the layers stored in each WMS resource.

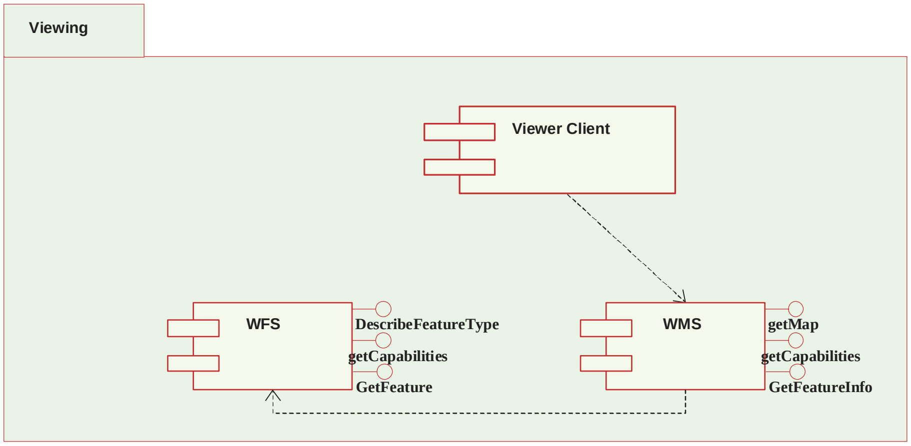

Figure 10: Viewing Package

6 Technology Viewpoint

The technology viewpoint is concerned with the underlying infrastructure in a distributed system. It describes the hardware and software components used in a distributed system. The infrastructure, which may be provided by a Distributed Computing Platform (DCP), allows objects to interoperate across computer networks, hardware platforms, operating systems and programming languages.

This section provides an overview of multiple DCPs implementation approaches, followed by a list of the current encodings used, including XML, imagery and well-known binary and text encodings. The Technology viewpoint also summarizes the technologies chosen for the Web Services platform implementations.

Figure 11 shows the technology viewpoint of the KWMIP Architecture. The Portal's thin client can run on any standard web browser. The Portal is based on standard HTTP protocol operations, GET and POST, and does not require any additional software to be installed on the end-user machine. In The KWMIP Portal, Application Services are realized as, or requested from, web pages transferred across a network from the Portal application server and supported by Portal-side capabilities to generate various clients and process requests. Thus, each item in the list below is called a "Client Generator" of some type.

Client Generators process requests from Thin Clients, maintain or transfer state between requests, and return responses to the requesting Thin Client. More specifically, when accessed as a World Wide Web application a Client Generator runs on an HTTP server and generates HTML pages to be displayed in the User's web browser (the thin client). In KWMIP, Client Generators are termed "Portal-side" components because they run on the computer hosting the Portal, not on the end-user's machine or on the data service provider's machine.

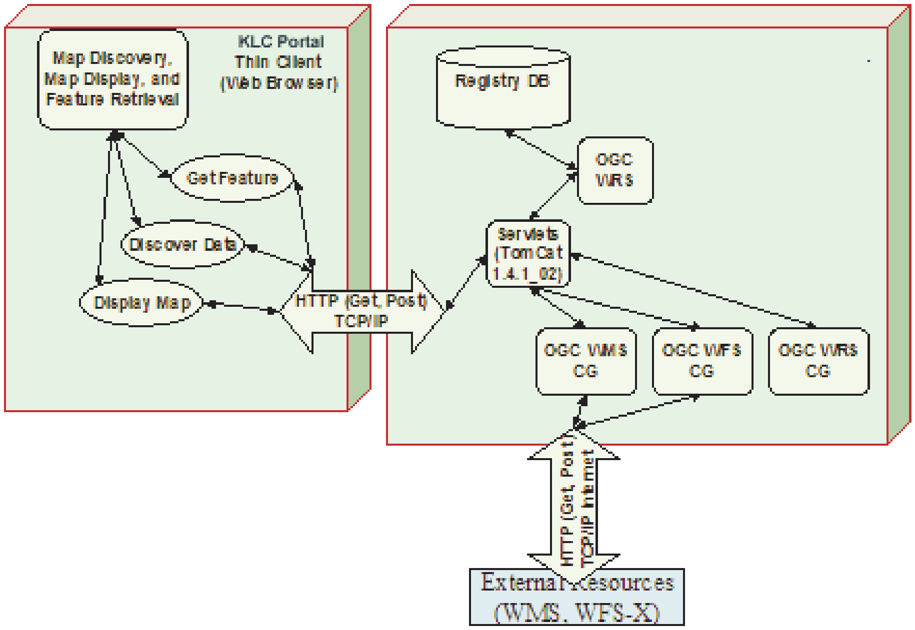

Figure 11: KWMIP Portal Technology Viewpoint

Appendix A: Functionality for KWMIP Portal

See attached Appendix A.

Appendix B: KWMIP Portal Architecture References

Refer to the OGC website (http://www.opengeospatial.org/specs/?page=baseline) for the authoritative listing of adopted documents.

Note: Please contact the OGC Tech Desk if you need assistance in gaining access to these documents (techdesk@opengeospatial.org).

OGC Specifications and Supporting Documents Relevant to KWMIP Portal:

1) OpenGIS® Geography Markup Language (GML) Implementation Specification (version 3.0), available at: http://www.opengeospatial.org/specs/?page=specs
2) OpenGIS® Filter Encoding Implementation Specification, version 1.0, available at: http://www.opengeospatial.org/specs/?page=specs
3) OpenGIS® Style Layered Description (SLD) Implementation Specification, version 1.0, available at: ***http://www.opengeospatial.org/specs/?page=specs***
4) OpenGIS® Web Map Service (WMS) Implementation Specification, version 1.1.1, available at: ***http://www.opengeospatial.org/specs/?page=specs***
5) OpenGIS® Web Coverage Service Implementation Specification, version 1.0, available at: ***http://www.opengeospatial.org/specs/?page=specs***

6) OpenGIS® Map Context Documents Implementation Specification, version 1.0, available at: ***http://www.opengeospatial.org/specs/?page=specs***

7) OpenGIS® Project Document 02-076r3: Gazetteer Service Profile of the Web Feature Service Implementation Specification, Version 0.9, Rob Atkinson and Jens Fitzke (eds.) , September 2002, <http://www.opengeospatial.org/techno/discussions/02-076r3.pdf>

8) OpenGIS® Web Feature Server (WFS) Implementation Specification, version 1.0, available at: ***http://www.opengeospatial.org/specs/?page=specs***

9) Gazetteer Service Profile of a WFS, available at: https://portal.opengeospatial.org/files/?artifact_id=7175

10) OpenGIS® Catalog Service Implementation Specification, version 2.0, available at: ***http://www.opengeospatial.org/specs/?page=specs***

11) OpenGIS® Project Document 03-024: OWS1 Registry Service, Richard Martell (ed.), January 2003, <not available electronically, please contact creed@opengeospatial.org >

Other OGC Specifications and Supporting Documents

12) OpenGIS® Abstract Specification Topic 11: OpenGIS® Metadata (ISO/TC 211 DIS 19115) May 2001, <http://www.opengeospatial.org/techno/abstract/01-111.pdf>

13) OpenGIS® Abstract Specification Topic 12: OpenGIS® Service Architecture (Version 4.3), Percival, G. (ed.), January 2002, < http://www.opengeospatial.org/techno/abstract/02-112.pdf>

14) OGC Cookbooks website: http://www.opengeospatial.org/resources/?page=cookbooks

ISO Specifications

15) ISO 19101:2002 (Reference Model): http://webstore.ansi.org/ansidocstore/product.asp?sku=ISO+19101:2002

16) ISO 19107 (Spatial Schema) : http://www.isotc211.org/protdoc/DIS/ISO_DIS_19107_(E).pdf

17) ISO 19108 (Temporal Schema) : http://www.isotc211.org/protdoc/DIS/DIS19108.pdf

18) ISO 19109 (Rules for Application Schema) : http://www.isotc211.org/protdoc/DIS/ISO_DIS_19109_(E).pdf

19) ISO 19110 (Methodology for Feature Cataloguing) : http://www.isotc211.org/protdoc/DIS/ISO_DIS_19110_(E).pdf

20) ISO 19111 (Spatial Referencing by Coordinates) : http://www.isotc211.org/protdoc/DIS/DIS19111.pdf

21) ISO 19112 (Spatial Referencing by Geographic Identifiers) : http://www.isotc211.org/protdoc/DIS/ISO_DIS_19112_(E).pdf

22) ISO 19115 (Metadata) : http://www.isotc211.org/protdoc/DIS/ISO_DIS_19115_(E).pdf

23) ISO 19117 (Portrayal) : http://www.isotc211.org/protdoc/DIS/ISO_DIS_19117_(E).pdf

24) ISO 19119 (Services) : http://www.isotc211.org/protdoc/DIS/ISO_DIS_19119_(E).pdf

25) ISO 19123 (Schema for Coverage Geometry and Functions): http://www.isotc211.org/protdoc/211n1227/readme.htm

26) ISO 19125-1 (Simple Features Access - Part 1: Common Architecture): http://www.isotc211.org/protdoc/DIS/DIS19125-1.pdf

27) ISO 19125-2 (Simple Features Access - Part 2: SQL option): http://www.isotc211.org/protdoc/DIS/DIS19125-2.pdf

Other Related Specifications:

28) EPSG, European Petroleum Survey Group Geodesy Parameters, Lott, R., Ravanas, B., Cain, J., Girbig, J.-P., and Nicolai, R., eds., http://www.epsg.org/

29) FGDC-STD-001-1988, Content Standard for Digital Geospatial Metadata (version 2), US Federal Geographic Data Committee, http://www.fgdc.org/metadata/contstan.html

30) ANSI/NISO Z39.50 Application Service Definition and Protocol Specification [ISO 23950 http://lcweb.loc.gov/z3950/agency/document.html]

31) IETF RFC 2109: HTTP State Management Mechanism http://www.w3.org/Protocols/rfc2109/rfc2109

32) IETF RFC 1729: Using the Z39.50 Information Retrieval Protocol in the Internet Environment [ftp://ftp.ietf.org/rfc/rfc1729.txt]

33) Uniform Resource Identifiers (URI): Generic Syntax (RFC 2396) T. Berners-Lee, R. Fielding, L. Masinter, available at: http://www.ietf.org/rfc/rfc2396.txt

34) Extensible Markup Language (XML) 1.0, Second Edition, Tim Bray et al., eds., W3C, 6 October 2000. See http://www.w3.org/TR/2000/REC-xml-20001006

35) XML Schema Part 1: Structures. World Wide Web Consortium (W3C). W3C Recommendation (2 May 2001). Available [online]: http://www.w3.org/TR/xmlschema-1/

36) XML Linking Language (XLink) Version 1.0, DeRose, S., Maler, E., Orchard, D., available at http://www.w3.org/TR/xlink/

37) Web Services Description Language (WSDL) Version 1.2. W3C Working Draft (9 July 2002). World Wide Web Consortium (W3C). Available [online]: http://www.w3.org/TR/wsdl12/

38) Simple Object Access Protocol (SOAP) 1.1, Box, D., et. al., available at http://www.w3.org/TR/SOAP/

39) UDDI – Universal Description, Discovery, and Integration, see http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uddi-spec

40) Registry Information Model v2.1, OASIS/ebXML Registry Technical Committee (Approved Committee Specification, June 2002). See http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=regrep

41) Registry Services Specification v2.1. OASIS/ebXML Registry Technical Committee (Approved Committee Specification, June 2002). See http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=regrep

Related Supporting Documents:

42) Reference Model of Open Distributed Processing [ISO/IEC 10746]