National Biological Information Infrastructure

# NBII GEOSPATIAL INTEROPERABILITY FRAMEWORK

## FUNCTIONAL REQUIREMENTS DOCUMENT

December 8, 2004

12201 SUNRISE VALLEY DRIVE, MS 302
RESTON, VA 20192

National Biological Information Infrastructure

# TABLE OF CONTENTS

National Biological Information Infrastructure

# NBII GEOSPATIAL INTEROPERABILITY FRAMEWORK

FUNCTIONAL REQUIREMENTS DOCUMENT

## INTRODUCTION

This report describes the functional requirements for a geospatial information framework (GIF), to be part of the National Biological Information Infrastructure (NBII). The goals of GIF are aligned with those of NBII as a whole. The nature of geospatial information–its generation, management, communication, and visualization–lead to requirements which are in some measure a superset of those for non-geospatial information. Addressing those requirements within the context of the federated organizational nodes comprising the NBII leads to a component architecture for the GIF which is described in this report.

Goals, use cases, component requirements, and other considerations are also described in the present report. Specific designs and technologies for the GIF are then presented in a companion document: NBII GIF General Design. Another important aspect of the GIF is begun here and continued in the general design: adoption and implementation of standards for representation and exchange of geospatial information is critical for establishing the level of interoperability between technologies and organizations, without which the GIF and NBII itself cannot achieve its goals.

## GIF FUNCTIONAL GOALS

The functional requirements developed in this document are rooted in a set of fundamental goals for a geospatial interoperability framework for the NBII. These goals have been laid out in concordance with goals expressed in the enterprise architecture for NBII as a whole, but put a particular emphasis on ways in which geography and location can further the achievement of those goals.

### GEOSPATIAL DISCOVERY OF FEDERATED CONTENT

A primary goal of the NBII GIF will be to enable geospatial discovery of content federated under the NBII. Geospatial discovery may or may not involve a map; it involves searching for and accessing federated documents and datasets using any criteria of geography or location. This may involve searches based on a map location or extent expressed in spatial coordinates. It may also involve criteria with geographic significance, such as

- ✓ Place names
- ✓ Political or demographic divisions
- ✓ Land use or landscape areas

- ✓ Species type localities or ranges

Geospatial search criteria are not necessarily Boolean (either/or) but may involve scoring search results based on geography, as in nearness to a geographic feature or the proportion of overlap with it.

This primary goal leads in two directions to other GIF goals:

- ✓ tools and applications for performing geospatial discovery and related tasks

- ✓ preparation and management of content to be discovered successfully with those tools.

The following goals expand on these directions.

## INTEGRATED ACCESS TO FEDERATED CONTENT

User tools for performing geospatial discovery fall into a number of functional areas. It is clear, however, that along with integration of content, a goal of the GIF is the integration of functionality. This means applications (and application toolkits) which can seamlessly support end-to-end workflows and use cases, from formulation of questions, through discovery, consumption, analysis, visualization, interpretation, and communication.

This should not be taken to mean that one application will support every user role and activity within the NBII, but that the business of users and not individual operations should be the guiding principle for applications which are developed or supported within the GIF.

## ACCESS PERSONALIZATION AND COLLABORATION

A goal related to that of application integration, this goal involves assisting the productivity of a user or group of users by allowing application configurations and intermediate results to be saved from session to session. Collaboration among users may then be enhanced by allowing these state records, such as map contexts and analysis plans, to be published themselves and shared between users.

An important corollary of this goal is that applications and workflows should be designed to allow configuration by the user whenever appropriate, in order to balance of the concerns of personal productivity against the benefits of a standardized user interface and practicality of application maintenance.

## SECONDARY CONTENT CREATION AND COMMUNICATION

To expand on the corollary goal in the last section, an inevitable and indeed desirable consequence of facilitating access to diverse content is the generation of new derivative content. This content may be as simple as a list of bookmarks, a map context (definition of a map composition), or a comment. It may be as complex as the result of a sophisticated model simulation. This secondary content may be just as significant as any primary content in the NBII, but cannot be fully understood or appreciated without an accessible record of its relationship primary content. A goal of the GIF is both to support the generation / use of secondary content, and to maintain / facilitate its connection to the primary sources from which it was derived. By "facilitate" is meant not only to document the links, but facilitate their traversal by whatever application tools have been developed.

## COMMON INFRASTRUCTURE SUPPORT FOR APPLICATION DEVELOPMENT

Further along the application direction of the GIF, the goal of providing useful applications is extended to include support for the ongoing and widespread development of useful applications. There are a number of levels on which this support may be developed, for a variety of NBII user and partner audiences. An essential principle of the support will be to incorporate common functionality wherever possible into a common infrastructure to minimize the effort of developing and deploying innovative and/or specialized applications. Another part of this goal will be empowering NBII nodes and partners to make full use of the NBII regardless of their level of IT resources and sophistication.

## GEO-ENABLEMENT OF CONTENT AND METADATA

On the content side of GIF goals, geospatial discovery is always complicated by the need to put discoverable content on a common geospatial footing. This is sometimes termed "Geo-enablement". One one level, the goal is to use (and convert to) common representations of geospatial properties, such as geometric models and coordinate reference systems. On another level, the goal is to add such representations to documents and data which have only indirect links to geography, such as place name references or references to other data with quantitative geospatial properties. This goal can be a difficult and ongoing challenge, yet is essential if the GIF is to be of utility across the broadest sectors of the NBII.

## COMMON INFRASTRUCTURE SUPPORT FOR NODE MANAGEMENT

Stepping back from the immediate goals of GIF functionality leads to the goal of sustaining GIF capabilities over time across a spectrum of node capabilities and resources. As in the case of application development, this involves placing the means of GIF management and administration as much as possible within a common infrastructure, which is accessible to all nodes. The more that NBII nodes can manage their own contributions and use of the GIF, the more likely that it will remain up to date and capable of meeting everyone's needs.

## GEOSPATIAL INTEROPERABILITY

Finally, the GIF will only succeed by pursuing whenever possible the goal of geospatial interoperability. Interoperability must be pursued in many forms and on many levels; it often appears opposed to goals of efficiency and practicality. Yet, in the long run, it will be the essential quality of the GIF which enables it to serve the diversity of technologies, disciplines, and knowledge communities which must characterize the NBII.

An immediate goal is to provide interoperability both between content sources and between application functions. Content source interoperability means that content may be discovered and utilized in a similar fashion no matter where in the NBII it may happen to be situated; specific standards and technologies within the GIF will be selected with this goal in mind.

Application function interoperability means that each application supported by the GIF will strive to present similar user interfaces and paradigms for similar functionality. It also means that operational information (particularly metadata) used for one application function (e.g. discovery) can be carried over for use by another application function (e.g. visualization). Again, use of specific standards and technologies within the GIF will focus on this interoperability goal.

## USE CASES

### SEARCH FOR FEDERATED CONTENT

| Use Case Description | |
|---|---|
| Name | Search for federated content |
| Priority | High |
| Description | Scientist searches for content metadata within federated registries of the NBII. |
| Precondition | User has access to a search application<br>Search application has access to data registry<br>Data are listed in data registry somewhere<br>User or client knows query able attributes of the registry (e.g. bounding box, keyword, Dublin Core, Darwin Core)<br>Data registry contains a list of data collections. A data collection is a set of data instances of one or more data types (see collections in catalog or ISO 19115)<br>Data registry has access information to other federated registries |
| Flow of Events – Basic Path | |
| 1. | Scientist is accessing a client application that allows him to access data registry to find data. |
| 2. | Scientist provides location as bounding box and nuisance species as keyword |
| 3. | Data registry returns XML document listing available data collections and record hits |
| 4. | Client renders XML document |
| 5. | Scientist selects one or more data collections of interest and may refine query |
| 6. | Client queries data collection and gets back a list of data instances |
| 7. | Scientist selects one data instance from list and displays detailed metadata about selected data instance |
| 8. | |
| 9. | |
| Alternate Flows | |
| Step 2a. | Scientist provides location as a place name or address |

| Use Case Description | |
|---|---|
| Step 2b. | Client converts place name or address to a bounding box, possibly using a remote service to do so. |
| Step 2c. | Return to Step 3. |
| Step 5a. | User elects to consider federated registries which are not cached by the immediate registry |
| Step 5b. | Client re-submits query to registry with cascade option |
| Step 5c. | Registry cascades query to federated registries, then aggregates their replies into a single response and returns as XML document |
| Step 5d. | Return to Step 5. |
| Step 5i. | Scientist selects classification or taxonomy record collection |
| Step 5ii. | Client fetches record collection or initial record from data registry |
| Step 5iii. | Scientist browses classifications or taxonomies to find desired data instance |
| Step 5iv. | Go to Step 7 |
| Step 8. | Scientist requests metadata about additional data instance which are linked or connected to the returned data instance. |
| Postcondition | User is able to access desired data using the returned metadata |

## SEARCH FOR CONTENT AND THEN DISCOVER SERVICES

| Use Case Description | |
|---|---|
| Name | Search for content and then discover services |
| Priority | High |
| Description | Scientist uses an integrated client application to find content of interest and then seamlessly find appropriate services which offer that content. |
| Precondition | User has access to a search application<br>Search application has access to data registry and a service registry<br>Data are listed in data registry somewhere<br>User or client knows query able attributes of the registry (e.g. bounding box, keyword)<br>Data registry contains a list of data collections. A data collection is a set of data instances of one or more data types (see collections in catalog or ISO 19115)<br>Registered data instances have unique identifiers<br>Services are listed in service registry somewhere<br>Service registry entries reference the unique identifiers of data instances being offered |
| Flow of Events – Basic Path | |
| 1. | Scientist is accessing a client application that allows him to access both data registry and service registry. |
| 2. | Scientist provides location as bounding box and pollution as keyword<br>Client application may restrict query to desired service type |

| Use Case Description | |
|---|---|
| 3. | Data registry returns XML document listing available data collections |
| 4. | Client renders XML document |
| 5. | Scientist selects one or more data collections of interest |
| 6. | Client queries data collection and gets back a list of data instances |
| 7. | Scientist selects one data instance from list and displays detailed metadata about selected data instance |
| 8. | Scientist selects service type(s) of interest |
| 9. | Client uses metadata record to query service registry for corresponding content offers |
| Alternate Flows | |
| Step 1. | See use case 4.1 for alternate data discovery flows |
| Step 9a. | Client automatically queries service registry for content offers of types it can consume. |
| | |
| Postcondition | Scientist has enough access information and appropriate client applications to consume content offer(s) of interest |

## PUBLISH FEDERATED CONTENT

| Use Case Description | |
|---|---|
| Name | Publish federated content |
| Priority | High |
| Description | Scientist or collaborator creates, publishes, and maintains content metadata record(s) in a registry federated to the NBII |
| Precondition | User has access to a registry management application<br>Application has transactional access to at least one data registry and optionally at least one service registry<br>User or client knows the appropriate metadata record schema for this content |
| Flow of Events – Basic Path | |
| 1. | Scientist uses a geoparsing, geolinking, or validating application or service to geo-enable her data / document |
| 2. | Scientist creates or updates a metadata record for his dataset |
| 3. | Scientist uses the management client to publish her record to the appropriate registry |

| Use Case Description | |
|---|---|
| 4. | Client creates an appropriate record identifier or obtains one from the registry. |
| 5. | Client creates a transaction to insert the record into the registry |
| 6. | Scientist optionally uses the management client to update relevant service metadata record(s) in the service registry for service(s) offering the dataset, using the data registry record identifier. |
| 7. | Scientist creates lifecycle information for his metadata record |
| 8. | Scientist optionally creates classifications and/or associations between her record and others |
| 9. | Data registry processes the metadata records for this dataset as needed to make them discoverable by registry queries |
| Alternate Flows | |
| Step 6a. | Scientist or collaborator updates the capabilities information for services offering the dataset, using means appropriate to that service implementation; the service capabilities are then harvested by the service registry |
| | |
| | |
| Postcondition | Content is now discoverable across the registry federation |

## PUBLISH FEDERATED SERVICES

| Use Case Description | |
|---|---|
| Name | Publish federated services |
| Priority | Moderate |
| Description | Scientist publishes descriptions of the capabilities and offerings of geospatial web services offered by the scientist or her organization / node |
| Precondition | Scientist or collaborator user has access to a service and/or service registry management application<br>Application has access to at least one service and/or service registry<br>User is responsible for configuration / maintenance of one or more service instances. |
| Flow of Events – Basic Path | |
| 1. | User deploys a local service instance. |
| 2. | User customizes service information of local instance (e.g. WSDL ports) |
| 3. | Management application registers service instance with a federated service registry |

| Use Case Description | |
|---|---|
| 4. | User configures service instance to serve data which have been registered with one or more data registries and has been assigned a unique dataset identifier |
| 5. | Service registry harvests service capabilities including reference to dataset identifier. |
| 6. | Management application optionally updates dataset registration with data registry to include links to the service metadata record and/or endpoint of the local service offering the dataset. |
| Alternate Flows | |
| Step 5a. | Management application updates service metadata record in the service registry directly as needed |
| Step 5b. | Go to Step 6. |
| | |
| Postcondition | Local service instance is discoverable through one or more federated service registries. Local content offers are discoverable in a one- or two-step process through one or more federated data registries |

### VISUALIZE GEOSPATIAL CONTENT

| Use Case Description | |
|---|---|
| Name | Visualize geospatial content |
| Priority | High |
| Description | Scientist |
| Precondition | Scientist uses a client application able to visualize geospatial content. Scientist has already discovered content she would like to visualize and discovered service offers which can be used by the client application |
| Flow of Events – Basic Path | |
| 1. | User enters or selects information to connect to a desired service offer such as a Web Map Service layer. |
| 2. | Client application adds the layer to a map legend |
| 3. | User selects an initial coordinate system, map extent, and style. |
| 4. | Client requests a map image in the desired extent, style, format, etc. and displays it within map visualization window |
| 5. | User repeats steps 1-4 until all layers of interest have been added to the map visualization |
| 6. | User navigates to areas of interest within map visualization with pan-zoom tools, or by selecting place names from a gazetteer menu. |

| Use Case Description | |
|---|---|
| 7. | Client requests each subsequent map image from the service as the desired extent changes. |
| 8. | User drills down into attributes of one or more visualized datasets using a GetInfo tool. |
| 9. | User defines a one-dimensional profile or track within the map extent, and an attribute or function or query (spatial or non-spatial) of multiple attributes, to produce an X-Y chart. |
| 10. | User hides or exposes map layers to improve visualization |
| 11. | User prints or saves their map composition |
| 12. | User selects a dataset or subset for download using spatial and non-spatial selection tools. User also selects options (format, projection, etc.) for the download "package" |
| 13. | Client requests map data from the same service or from a different service offering the equivalent data in the desired download format. |
| Alternate Flows | |
| Step 1a. | User opens a previously saved and/or registered map context document into the client application, which then performs Steps 2 and 4. |
| Step 2i. | User selects a non-geospatial dataset and links it's attributes to those of one of the previously selected geospatial datasets by defining a key relationship |
| Step 2ii. | Client application adds the joined dataset to a map legend as a single layer, but performs a compound service request and response-processing step whenever visualization of the layer is updated. |
| Step 3a. | User selects saved and/or registered map styles which the client then applies to the map layer, either by rendering features locally, or by supplying the style reference in its map image request in Step 4. |
| Step 3i. | User creates thematic styles for one or more layers by selecting feature attribute(s), rules, and symbolizations for a theme. |
| Step 6a. | User enters an address into a geocoding window. |
| Step 6b. | Client requests an address geocode from a remote geocoding service, calculates an initial map extent around the geocode coordinates, then requests new map image(s) to visualize |
| Step 6i. | User performs non-spatial query of data from one or more visualized layers and selects "Zoom To" option |
| Step 6ii. | Client application formulates query as one or more data or feature info request(s) and calculates a new map extent from the response(s) |
| Step 9a. | User selects a data attribute or function or query (spatial or non-spatial) of multiple attributes to serve as a z-coordinate. User also selects options of orientation, rendering, and illumination |
| Step 9b. | Client provides a 3-D view of a map visualization, either by locally translating the 2-D map image to the new x-y-z surface, or by requesting a terrain view image from a service capable of providing it. |

| Use Case Description | |
|---|---|
| Postcondition | Visualization composition possibly including multiple map, legend, data table, and/or chart elements provides useful insight into the geospatial data for the user or others. |

### ANALYZE GEOSPATIAL CONTENT

| Use Case Description | |
|---|---|
| Name | Analyze or model geospatial content |
| Priority | Moderate |
| Description | Scientist |
| Precondition | User has previously discovered geospatial data of interest and formulated a hypothesis of significance to be tested using the geospatial data.<br>User has access to a geospatial analysis application.<br>User has discovered services that offer the geospatial data of interest. |
| Flow of Events – Basic Path | |
| 1. | User opens the analysis application, begins a new analysis scenario, and enters or selects the sources and identifiers of the desired geospatial dataset(s). User also selects initial parameters such as area of interest or other sub-selection of the data. |
| 2. | Application requests either the datasets themselves or (if sufficient) metadata such as dataset structure and domain (e.g. data dictionary or schema). |
| 3. | User selects an analytical or modeling procedure and selects a dataset attribute, function, or query to link with required and optional parameters of the procedure |
| 4. | Application requests the datasets or subsets as necessary and carries out the requested procedure. |
| 5. | Application presents the results of the analytical or model procedure in report or table format |
| 6. | Application adds results as a new layer to the present map visualization |
| 7. | User adjusts input parameters as needed and re-runs procedure until desired results have been obtained |
| 8. | User prints results or saves them as<br>✓ Extended map context with procedure input values<br>✓ Secondary dataset, such as an interpolated grid, set of contour lines, or predicted future feature attributes. |
| 9. | |
| Alternate Flows | |

| Use Case Description | |
|---|---|
| Step 3a. | User formulates a custom procedure by graphically linking together data sources, processing parameters and processing modules |
| Step 4a. | Application formulates a request with input parameters and dataset references, then posts request to a geoprocessing service. Application then requests status updates from the service until it returns the completed procedure results to the application |
| | |
| Postcondition | Secondary content and presentation materials that facilitate interpretation and communication of existing datasets. |

## CREATE AND PUBLISH SECONDARY (DERIVATIVE) CONTENT

| Use Case Description | |
|---|---|
| Name | Create and publish secondary (derivative) content |
| Priority | Moderate |
| Description | Scientist or collaborator creates and publishes secondary content so that it's links to relevant primary content may be discovered and traversed. |
| Precondition | User has discovered primary content and wishes to create new content by interpreting or processing that content.<br>User has access to an application for creating, documenting, and publishing derivative content.<br>Application has access to primary content offered by one or more federated services<br>Application has access to a data registry which can accept association metadata |
| Flow of Events – Basic Path | |
| 1. | User selects one or more primary datasets or subsets |
| 2. | Client application requests selected datasets or map images from appropriate federated services and displays them |
| 3. | User initiates a map algebra procedure with the selected datasets or draws new features in new feature layer on top of the existing layers. |
| 4. | Application creates a new feature layer in response to use input |
| 5. | When finished creating the new feature layer, the user finishes and approves a metadata record for the new layer that has been created by the application. |
| 6. | The application uploads the new metadata record to one or more federated data registries |
| 7. | The application creates association metadata records which link the metadata for the new layer to the metadata records for the primary layers |
| 8. | The user selects an online repository for storage of the new feature layer. |
| 9. | The application uploads the new layer to a transactional or upload-capable data repository. |

| Use Case Description | |
|---|---|
| 10. | User saves a context document incorporating primary and derivative data layers, either locally, or to a context-capable registry |

**DEVELOP A GEOSPATIAL SERVICE OFFERING**

| Use Case Description | |
|---|---|
| Name | Develop a geospatial service offering |
| Priority | High |
| Description | Scientist or collaborator deploys a geospatial service offering by standing up server hardware and software, configuring the software, and making geospatial content available through the service. |
| Precondition | User is a member or contractor of an NBII node<br>User has access to tools for server node development from NBII or collaborator |
| Flow of Events – Basic Path | |
| 1. | User selects and downloads an appropriate server software distribution kit from the NBII Node Tools website |
| 2. | User deploys hardware and system software supported by the selected distribution kit |
| 3. | User runs an installer routine to install the server distribution kit |
| 4. | Using a web-based administration tool, the user configures the installed software for operation and uploads local content to be offered through the deployed server. |
| 5. | User configures map styles and other content offer parameters |
| 6. | User publishes the service and content metadata for their server as described in User Cases 4.3 and 4.4 |
| 7. | |
| 8. | |
| 9. | |
| Alternate Flows | |
| Step 1a. | User logs into a virtual server account offered by a geospatial services provider within NBII |
| Step 2a. | User avoids any hardware or software installation or maintenance and proceeds directly to Step 4: Upload and configure local content |

| Use Case Description | |
|---|---|
| Step 1i. | User selects and installs 3rd-party software which is not distributed by NBII but conforms to service interface standards established for the GIF. |
| Postcondition | Geospatial service offering and its contents are discoverable and consumable across the NBII |

**DEVELOP A GEOSPATIAL APPLICATION**

| Use Case Description | |
|---|---|
| Name | Develop a geospatial application |
| Priority | Moderate |
| Description | Scientist or collaborator uses a RAD (Rapid Application Development) toolkit to build a focused client application which consumes geospatial Web Services and NBII content |
| Precondition | Developer is a member or contractor of an NBII node<br>Developer has access to GIF-enabled RAD tools for development of geospatial client applications which are interoperable with NBII federated services. |
| Flow of Events – Basic Path | |
| 1. | Developer obtains a GIF application toolkit by download or otherwise in the programming language and technology platform of interest. |
| 2. | Developer discovers content and services on the GIF which she would like to incorporate into an application. |
| 3. | Developer builds an application use case and designs an initial application based on the discovered services and toolkit capabilities. |
| 4. | Developer selects and configures components from the toolkit for the service requests and user interface elements in the application design |
| 5. | Developer fills in the interactions between application components by modifying existing sample application code to reflect the program logic of his own application. |
| 6. | Developer deploys functioning application as either a browser-accessible or downloadable Web application |
| 7. | Developer publishes a description of the application as a "service client" to a federated service registry |
| 8. | Developer repeats Steps 4-7 periodically in order to gradually fill out and refine the application functionality as the application use case and design evolve with user needs. |
| 9. | Developer takes the present application as a template and returns to Step 2. to build a new application. |
| Alternate Flows | |

| Use Case Description | |
|---|---|
| Step 4a. | Developer configures a browser-based hosted application entirely online by using a Web-based configuration/administration application |
| Step 4b. | Validated application is immediately online so that no custom programming, deployment, or registration is needed. |
| Step 9a. | Developer uses an existing design and use case with a different toolkit to generate an application for a different technology platform. |
| Postcondition | New application is discoverable and available for use by one or more NBII nodes, within a short period of time. |

## APPLICATION REQUIREMENTS

This section details the application functional requirements derived substantially from consideration of the use cases in Section 4. These requirements fall under a number of categories, including requirements for the manner in which GIF applications should be developed

### DEVELOPMENT

The functionality of the NBII GIF begins with development of applications that make effective use of its service-oriented architecture, and also address the evolving needs of its users. A GIF development methodology needs particularly to address the case that a diversity of developers with a range of experience and resources will be building applications to work with a common information and services framework. The methodology needs to be effective, for example, for scientists with minor development experience whose task focus for each application is a moving target.

#### DEVELOPMENT PRINCIPLES

- ✓ Address a variety of client technologies
- ✓ Encapsulate the complexity of service request-response interactions
- ✓ Support information interoperability with compatible objects / components in each client technology
- ✓ Support a client “thickness” which is appropriate for each audience and task
- ✓ Emphasize rapid development of smaller client applications or application components that focus on a particular task

#### TARGET AUDIENCES

As a federation, the NBII has a mission of working with a diverse audience of nodes and partners. It is clear that one-size-fits-all will not be a successful development methodology. It will be helpful, however, to address a small number of particular target audience profiles, in order to focus efforts and produce effective tools, rather than taking on a service task which is too large to be accomplished.

The various targets seem to fall into three general categories, which are suitable for a first pass at user/developer/node profiles:

- ✓ "Central Node" – The large and experienced resources available for central NBII development. Enterprise-scale development and Commercial-Off-The-Shelf (COTS) software are possibilities here, as well as the resources to develop and maintain toolkits for wider use.

- ✓ "Large Node"– The substantial but variable resources which are likely to be available at larger NBII nodes. While they may be capable of enterprise-scale development and make use of Commercial-Off-The-Shelf (COTS) software, there may be many other tasks in play besides participation in NBII, and generally more of a focus on scientific tasks than on software or software tool development. The large node may both appreciate the support that GIF gives their development efforts and trend in a direction more cooperative with NBII with that support.

- ✓ "Small Node" – Many nodes and partners of NBII will be smaller organizations with limited resources and somewhat of an indirect interest in NBII participation. These organizations will frequently face the choice between using resources and free software provided by NBII, and not participating in the NBII to any substantial degree. This target audience will benefit particularly from largely pre-built, no-cost "node kits" which are either physical distributions of free software, or virtual access to publishing capabilities on other organizations' systems.

These three target profiles span a range in both technological ambition and in the degree of support and handholding that may be required.

The profiles also correspond to three user profiles that span a range, respectively, from large to small administrative and management capabilities:

- ✓ Central-node users may be comfortable installing complicated client and server software, as well as utilizing administration clients for same. Such users may be less familiar with particular tasks on which a GIF application might be focused.

- ✓ Large-node users may still be comfortable (or have assistance with) installation and administration. These users may be more focused, however, on individual scientific or analytical tasks rather than general geospatial technology

- ✓ Small-node users are most likely to prefer thin-client browser applications and other out-of-the-box solutions that require a minimum of administration and training in order accomplish their work. They may be less willing to learn GIS techniques, but more interested in having geospatial technology melded into their own dynamic workflow.

In addition to development requirements, all of the various application requirements below will need to address in some fashion the diverse audience for GIF applications that are roughly represented by these target profiles.

## DEPLOYMENT

Once applications have been developed, comes the challenge of deploying them. Deployment requirements are addressed in three areas, which relate in large part to peaceful and useful coexistence across the GIF

### DEPLOYMENT MODES

There are three broad modes of deployment at present, each of which have their pluses and minuses in different contexts:

- ✓ Server-based browser applications – the main bulk of these applications is deployed onto a server that needs only to be connected with users' computers by the Internet. A minimum of user installation and configuration is generally required, but there are limits to the user interactions that can be supported, and risky assumptions as to compatibility from one user computer to another.

- ✓ Server-provisioned stand-alone applications – the main bulk of these applications is loaded onto the user's computer, partly with standard software installs or plug-in configurations, and partly with custom application components downloaded (or updated) from the application Web site. A substantial amount of the application processing may still be accomplished on server tiers through Web services requests. There is both significantly greater installation effort and greater potential user interface capability in this mode. Significant compatibility issues may be encountered during the initial install, but are generally minimized within the application itself.

- ✓ Desktop applications – these are traditional desktop applications which are installed in one go and may perform all of their processing on the user's computer, making use of remote services only as required. Traditional GIS applications fall into this last category

An application's choice of mode should emphasize the simplest user experience that is capable of accomplishing the task, but the user's prior experience, as well as the maintainability of the application should also be taken into account. An application that is not utilized by its target users is clearly not being effective.

### VALIDATION AND COMPLIANCE

This requirement is a broad one, but relates to whether an application is dealing correctly with the service interfaces and information schemas (as well as semantics) provided by the GIF. Problems may be as simple as typographical errors in service requests and as complicated as subtle disconnects in the information semantics. This is especially difficult in an environment of diverse voluntary federated services and clients, but some form of application testing regime will be required if only to give developers and users a broad sense of confidence in the framework itself.

PUBLISHING STANDARDS

The issue of exchange of valid information becomes more acute when users are actually updating or creating online information (whether data or metadata) through their applications. In this case it may be much more severe for one user to be affected by the errors and incompatibilities introduced (however inadvertently) by another user. Going up the administration and development chain, the consequences can become that much more serious.

Along with testing, this suggests the need for application "sandboxes", a common facet of much open-source development of distributed components. This allows the consequences of using a new application to be monitored without their causing ripple effects (e.g. in registry holdings) throughout the GIF.

APPLICATION DISCOVERY

It is a current curiosity that much more effort has gone into discovery of Web services than into Web applications; the latter is much more likely to be found by word of mouth or through search engines. While the NBII Portal provides a general mechanism for advertising and promoting applications, a better approach is to explicit publish them as "client services" to service registry, where they can be easily matched in queries with the services and content they are designed to make use of. The fundamental requirement, then, is the creation and publishing of metadata records for client applications as well as services within the GIF.

## DISCOVERY

A fundamental application requirement within a federated, distributed, and service-oriented environment, is the capability to search for and discover the far-flung (or nearby) resources of interest to the user, which the application is then able to utilize. There are a number of basic principles to facilitating this:

- ✓ As simple as possible to publish and to search (but not simpler)
- ✓ Information interoperability that allows the registry federation to be extensible into other communities
- ✓ Free-text searches are necessary but not sufficient for effective discovery
- ✓ Discovery should lead with minimum effort to utilization

SERVICE ACCESS

Access to discovery services is critical to effective discovery. This includes both ability to publish resources through an application and ability to search for resources through an application. The first level of discovery is then the discovery services themselves. This is a bootstrap problem of providing the initial information in order to look for additional information. Presumably this will mainly be a problem for developers as long as the federation of discovery resources is effective, i.e. that one built-in resource per application will be sufficient for finding and utilizing any others.

## SEARCHES

The basic application requirement for searches is that searches may become extremely complex and resource-intensive–it is up to the application to make this as transparent and easy for the user as possible, and up to the GIF to make this as straightforward as possible for the application. After all, discovery is almost always a means to another end, not the end in itself.

The user and the application need to know, easily, what types of resources can be searched for, what resource properties can be queried, and what property values are appropriate. There are four general modes of searching which should be addressed by most applications:

- ✓ Search on common properties – this is the role of metadata standards such as Dublin Core and Darwin Core, to provide structured sets of resource properties which nonetheless can be attached to virtually any type of distributed resource.

- ✓ Search on specific properties – this requires the application to access and present to the user the query-able record collections, the query-able properties for each record type, and the domains for each property. Formulating the right query may still be difficult unless metadata for the above parameters is sufficiently extensive.

- ✓ Free-text search – although users perhaps too easily revert to this type of search, it is often the only way of gleaning information from poorly organized and documented data; one-reason search engines of this type are so popular on the Web.

- ✓ Structured browsing – various forms of metadata organization such as classification, taxonomies, link frequencies, etc., provide added value to the basic collection of records being searched. This has the benefit of guiding the user to find the most appropriate search term for his purposes, and allowing a sort of search collaboration where intentional or inadvertent advice is given from one user to another as to the most relevant search results.

As discussed further below, saving and sharing searches that have proved useful are some of the most common ways of personalizing application use and supporting collaboration

## TRANSITION TO UTILIZATION

Discovery is only important to a geospatial application if something useful can be done with the search results. That use may only be printing out metadata records or sending an email to the author of a dataset, but more commonly the user would like to immediately obtain, map, or otherwise work with what has been found. This usually means proceeding to discover an online service that offers the desired content in a form useable by the application.

In principle, metadata records may contain online resource links that lead to the offered content. In practice, datasets and other content are less dynamic than their provision through online services. This means that it is much harder to keep service references current in a data registry than it is to keep data references current in a service registry. By whatever means, most useful applications will provide the user with an easy and direct path from discovered content of interest to visualization or other use of that content.

## VISUALIZATION

While there are many geospatial tasks that can be accomplished without maps, it almost always makes sense in some part of a geospatial application to visualize the data. This usually takes the form of a 2D map, but could also be a profile, a chart, or a 3D scene. Even within the realm of a 2D map, there are many ways of visualizing information. Almost any user can make use of some map, but the visual literacy of most users drops steeply with anything beyond a road map. Effective visualization applications for GIF will require a balance between innovative visualization and familiar map paradigms.

A particular requirement will be consistency in both the map styles and the tools offered by each application. The rapid development of applications will not be matched by rapid user acceptance if each new application requires training in a new visual paradigm and toolset.

### PARADIGMS

#### *Road map*

Features in this view are shown styled by feature type

#### *Thematic Map*

Features in this view vary in style according to the values of one or more attributes of the features. Common style variations include color, pattern, size, and symbol type. This type of map requires a particularly involved chart legend in order for the user to clearly interpret the thematic symbolization.

#### *Chart Map*

In this view, small charts overlaid at locations on the map portray multiple attributes at each location

*Image – overlay*

In this paradigm, a layer of features with which the user is actively interacting overlies a static background map image. Interaction tools may allow the user to "promote" features from the background into the active foreground for use.

*Profile*

A cross-section map or chart shows elevation or the variation in some other feature attribute along a path delineated across the 2D map extent

*3D Scene*

Features in this view are shown in 3D perspective, usually draped over an elevation model of the map extent to provide the 3rd dimension. This paradigm may include any of the other 2D paradigms as part of the scene but care is required to preserve clarity.

*3D Thematic Map*

In this variant of the 3D scene, the 3rd dimension is provided by some continuously or discretely varying feature attribute other than elevation. The appearance is similar to that of the 3D map, but extra effort is required to understand the significance of the scene "topography" since it may represent a conceptual elevation rather than a physical one.

TOOLS

*Zoom*

Change scale of view

*Pan*

Translate extent of view

*Previous view*

Return to previous view extent(s). This may be augmented as a "map cache" which saves a complete map view configuration with thumbnail image, allowing the viewer to bring up the same map again, as well as save that map configuration into a portable context document.

*Full extent*

Return to full extent of one or more map layers.

*Go to place*

Choose a geographic place name and zoom to it (or its vicinity) by picking, browsing within the current extent, or searching a gazetteer.

*Go to query*

Select a subset of features within one or more map layers and zoom to the bounding box of that subset. The inverse of this tool is "select within extent" which varies the selection of features in another view according to the present map extent.

*Go to point(s)*

It is sometimes important to have the capability of entering a new point of interest (POI) or bounding box (BBOX) directly into the viewer. This is particularly useful when the user can enter those coordinates in one coordinate reference system (CRS) and have them transformed into the present map CRS.

*Add to (thematic) class*

In association with thematic analysis, this tool allows the user to build ad hoc thematic feature classes within a layer by selecting them interactively (point or lasso). When this capability is linked to thematic representation of those features in other views (tables, charts), the operation is referred to in statistical analysis as "brushing"

*Get info – drill down*

This tool is the primary link between visualization and analysis, but allowing the user to select a map feature or features and examine or otherwise work with its data attributes. This is termed drill-down or drill-through when the tool allows the user to link through to other datasets or follow other types of links (e.g. href's) within those attributes.

**ANALYSIS**

Analysis of geospatial content is a very broad topic, but we identify here a limited number of analysis categories. Three modes of analysis should be considered. The first is linking to content which has already been derived by analysis; for example contoured sample locations. The second is performing a "canned" analysis whose parameters have already been configured; for example a habitat metric calculation. The third is configuring and running a full model simulation such as a population model; this mode is likely to be the most compute-intensive, user-interface-intensive, and interactive.

THEMATIC ANALYSIS

This type of analysis involves the calculation and modeling which goes into determining thematic styling, from simple binning to geostatistics. The result is largely the thematic visualization and accompanying legend.

MAP ALGEBRA

This covers spatial analyses such as spatial selects, intersects, merges, grid algebra, and so on; the result is usually a derivative dataset and map layer.

STATISTICAL ANALYSIS

There are of course a great variety of statistical analyses that may be performed on geospatial and associated datasets. For application purposes, these are divided into those generating new map visualizations, those that generate derivative data, and those which produce non-spatial summary reports.

SIMULATION

In the context of a geospatial application, simulations generate derivative geospatial data that are critically dependent on choice of model and input data. Often the most appropriate input settings are not known in advance and have to be arrived at iteratively. Model results are very powerful in advancing understanding and very dangerous in being confused with actual observations (the boundary itself is a gray one). It is particular incumbent upon modeling applications to promote appropriate model use (beyond blanket disclaimers of liability).

DOCUMENTATION

Much of geospatial analysis generates derivative data whose meaning can only be understood in the context of that derivation. Just as metadata are required for original data explaining where it came from, the same thing and more is required for derivative data, including if possible enough information for someone else to reproduce the analytical process which created it. Just as a map context allows the recreation of a map, an analysis context document can be created, updated, stored, discovered, communicated, and re-used. An analytical application should provide these capabilities.

ADMINISTRATION

Particularly in the area of simulations, but with other analytical tasks, most users will require assistance and guidance in their application. An important consideration should be the administration of such metadata as model setups and presets, valid model result domains, and other aids whose state of the art and applicability to particular datasets may be expected to keep evolving.

## PERSONALIZATION

While fine-grained access control is not a primary design goal of GIF, there are two main uses for reliably determining the identity (or at least the uniqueness) of users. The first is in order to personalize their application experience with preferences and saved configurations. The second is in order to provide attribution of content that they themselves create for use by others.

### AUTHENTICATION

There are many means of authenticating web application users. A simple method such as username – password will probably suffice here. Both logging in and maintenance of username – password lists are tedious and error-prone; hence there is a preference for these tasks to be done as globally as possible within NBII. This is one area where integration with the NBII portal will be particularly important.

### SAVING STATE

This refers to leaving an application in mid-task and being able to return and restore it to that condition in the future. Documents such as map contexts, which record the necessary information for this, can also be treated as any other derivative content–metadata to be registered, discovered, re-used.

### CONFIGURATION

Application configurations may be part of an application state, but also consist of those setting required for the application to function at all. As such, they are likely to be read in at start-up and administrated either in an "advanced" UI component or in a separate application altogether.

Another important requirement for application configuration is the ability to configure the application for individual groups or communities. For example, particular map symbologies, place name gazetteers, or content menus may be configured upon login / startup for users belonging to specific groups.

### CONTENT CREATION

Personalized content to be created by one or more applications may include:

- ✓ Context documents
- ✓ Style documents (e.g. thematic styles)
- ✓ Annotations or comments

- ✓ Classifications of content (e.g. "My Top 10 datasets")
- ✓ Derivative data and supporting metadata
- ✓ Queries

## COLLABORATION

Too often the paradigm of client-server applications leads to undue focus on the interaction between user and machines, with too little focus on that between users and other users. There are several means by which collaboration between GIF users may be enhanced with a geospatial component

### SHARING DATA

This paradigm emphasizes the ability of a user to publish primary or secondary data so that they can be discovered and "subscribed to" from appropriate services by other users in their applications. Communications such as annotations or location-specific commentary also constitute share-able data.

### SHARING METADATA

This paradigm emphasizes sharing various forms of metadata such as context documents, catalog queries, through publishing or by more informal means such as email.

### SHARING APPLICATIONS

When the sharing of such things as application context documents and annotations reaches an interactive pace, it becomes in essence a form of application sharing, where one or more users' changes to the application state and accessed data can be shared (like instant messages) with one or more other users in virtually real time. While generalized application-sharing and whiteboard applications are another paradigm altogether, the capability of linking application instances between several users during, for example, a teleconference could be a useful and relatively small increase in application capability.

# SERVICE REQUIREMENTS

With goals of providing common data and operations capabilities across many nodes, applications, and users, it is clear that the GIF requires a set of shared, distributed, possibly federated, and robust application services. These services need to be accessible across application platforms and technologies, between NBII nodes that are in general connected only by the public Internet.

## GENERAL REQUIREMENTS

The use of shared (Web) services for distributed application development only really provides the benefits of increased functionality and lowered cost when a set of general requirements for those services can be met.

It is important to point out in terms of implementation requirements that a service may be "local" to an application, even being deployed on the same host, so that any of a variety of binary, platform-specific access and transport protocols might provide the highest performance. The most general case, however, has to be considered one in which server and client are maintained by very different organizations, separated by multiple routers, firewalls, proxies, and other restrictive network components; in this case little other than text messages exchanged by http protocol can be considered reliable. Hence, a Web-accessible interface must be considered a minimum general requirement for any of the services described below.

### SHARED

An important criterion for a shared service is that it really can be shared. A service needs to represent a common set of content and functionality that is applicable to many tasks. An application-specific service may be a way of distributing a specific application, but offers few other benefits for all the effort. An important corollary is that a relatively small number of service types are likely to be of more benefit than a large number.

### CURRENT

One way that a service-oriented architecture provides efficiency is by allowing the maintenance of content to be separated from the utilization of it. The application builder will only realize the benefit of this, however, when the content providers themselves can effectively maintain the service. Either the federated service itself originates from the same organization and/or locality as the offered content, or online administration of a more centralized service provides the same level of access virtually (e.g. "My_NBII_WMS").

### AVAILABLE

A service-based application can only be depended upon if the service access itself is dependable. This is in part a system administration challenge–enough redundant hardware and network connections. It is also a service administration and federation issue, however. Reliable services require good tools for administrating monitoring them, and maintaining their content. Information on replicated service offerings and current quality of service within a service federation are also important to reliable service access.

Another aspect of availability is affordability, both in terms of acquisition and operations cost, and of maintenance time. A service that is not deployed cannot be consumed.

DISCOVERABLE

The importance of making complete and up-to-date service metadata available to applications cannot be overestimated. In the end, a good service is an invisible service, which magically empowers a small application to provide immense capabilities; this only works if the application can find and continue to find the right services to make use of.

There are two basic approaches to this issue of service access when multiple services in a federation are required. Which one is most appropriate depends to some extent on the complexity of an particular application

- ✓ Direct access–in this approach, the client application manages individual connections to however many service instances are required to carry out its tasks, obtaining metadata on which services to use from a catalog or its own metadata store.

- ✓ Cascade or façade–here a single service is used for all service requests of a given type (e.g. GetMap). Those requests which cannot be satisfied "locally" are then passed on to other federated service instances which can satisfy them, and the originating service then receives all responses, aggregates them as needed, and returns them to the client. The client application has only to manage connecting to the one service and making a single request for content that may be spread across many distributed service instances.

While the former approach allows the client application much more flexibility in managing connections, for example displaying some responses while waiting for others, the latter approach provides both simplicity and stability for thinner applications. The latter approach may also provide greater reliability by transparently selecting the currently most performant-federated service to satisfy a given request.

SELF-DESCRIBABLE

There is not complete agreement on this point within the Web Services community, as some feel that any needed description of a service should be contained within a WSDL document obtained from a catalog. The experience within OGC has certainly been that catalogs may not choose to maintain complete descriptions of individual services. WSDL is not yet entirely sufficient for describing data-rich services. Service descriptions are also much more likely to be maintained (and accessible for harvesting by catalogs) if they are associated directly with the services themselves, i.e. self-describing services.

## CATALOGS

The terms "catalog" and "registry" will be used more-or-less synonymously in this document. While there are differing definitions of the two, there is little agreement on what that difference is. Either term will refer to a searchable, maintainable online store of uniquely identified metadata records, without necessarily restricting the scope of search methods, maintenance methods, or record types.

The important requirements for catalog functionality from an application standpoint are that queries can be iteratively refined and limited to control the size of returned record sets; that guidance is provided to clients in terms of query parameters, domains, and operators; and that the returned records can be usefully parsed by clients.

To the extent that the OGC Catalog Specification v. 2 is a conceptual and technology-independent description of catalog interfaces and functionality, it covers in great detail the functionality outlined above, including interfaces for maintaining the catalog contents manually or automatically through transactions.

DATA CATALOG

In this type of catalog, the metadata records describe datasets or documents such as a feature collection, a set of observations, or a report. Each record describes the data format, source, domain, precision, etc., but without necessarily indicating a specific copy or instance of that content. In a sense these are data type catalogs except where they include references to one or more data "locations".

The important specific requirement for data catalogs is that they provide common query parameters to search for metadata for a diverse range of data and documents. Either a common record format or at least a common summary of different formats also enhances accessibility. A data catalog should also support other more structured means of finding the desired content, by supporting the association and taxonomic classification of records

SERVICES

While a services catalog requires the same general functionality as any other catalog, it particularly depends on providing links between specific types of metadata. In particular, it needs to provide access to metadata about

- ✓ Service type – the general information needed for a client to interact with the service (e.g. WMS)
- ✓ Service instance – the specific information for a client to reach a particular deployed service (e.g. service endpoint, transport protocol)
- ✓ Content offering – what specific data or content are available from a specific service and how can it be requested. A unique identifier (at least within the catalog) for the content which is being served provides a critical link between data and service catalog holdings.

While a data catalog can also provide links to offered content (e.g. Online Resource tags), service (and content) offers tend to be much more dynamic than dataset descriptions. It therefore is more reliable to search an up-to-date services catalog for the identifier of a desired dataset than to count on references in a dataset description for sufficient information to connect to the right service type and instance. Given the existence of a central services catalog, it even makes sense to include an Online Resource element which is a query of the services catalog for appropriate services offerings, rather than a direct invocation of a particular service which may or may not still be useable.

OTHER CATALOGS

Application requirements define a need for managing other metadata which are operational if not application specific in nature, but are not necessarily descriptions of either data or services themselves. These include but are not limited to:

- ✓ Application contexts
- ✓ Personal profiles
- ✓ Map styles
- ✓ Symbols
- ✓ Schemas
- ✓ Annotations

Catalogs are still useful services for storing, finding, and retrieving such metadata, but finding the right records also requires "metadata about the metadata" as well as a common query approach. The searchable metadata can be included directly with the retrievable metadata. Another possibility is a separate repository for the above records, which are then referenced by searchable catalog records.

## DATA

While the ultimate use for geodata may be map visualization, there are many requirements for direct access to the data themselves. Data services provide both efficient and valid access to a variety of types of geodata. In most cases–especially for large-scale datasets–the data may be maintained in a database of some sort. Certainly a direct database connection and SQL queries are one form of data service; this is usually not completely platform and technology independent, however, nor are database port connections usually a good choice for widely distributed service federations. Data services seek to hide database complexities and dependencies behind a more standardized and robust messaging interface.

Content to be accessed from data services includes the following:

FEATURES

There are various definitions of a feature, but a generous one would take it as any data which include a geometric representation of a geographic entity, perhaps as little as a point location or a rectangular extent. A feature service will provide the ability to query either on the spatial or non-spatial attributes of a feature and return matching feature records. It will provide the ability to customize what feature attributes are returned, but may enforce a minimum set for feature validity.

Other optional feature service capabilities include descriptions of feature schemas, transformation of geometries into other coordinate reference systems, and online transactions (inserts, updates, deletes) of features.

IMAGES

In one sense, an image can be considered a grid feature, but handling of imagery is sufficiently specialized as usually to constitute a separate service type. In particular, imagery may be collected and/or available at a variety of defined resolutions (grid spacing or cell sizes). There also may be great variation in the amount and diversity of information (ranges) for any one cell (domain), such as pixel depth, spectral bands, time series, etc. Imagery services therefore usually operate on an information paradigm of a multi-dimensional cube where two dimensions are geographic and others dimensions constitute (selectable) axes of image attributes.

Another requirement for serving imagery is dealing with coordinate referencing situations. The cells in an image may not be at all geo-referenced, such as in a photograph taken at a sampling location. They may be geo-referenced but not geo-rectified (image grid aligned to coordinate reference system axes), such as in a raw LandSat scene. They may be fully geo-rectified, but not clipped into a seamless mosaic. Finally, imagery may have been processed into a completely geo-rectified, seamless image coverage. The latter is generally the most useful image product for mapping purposes, but does involve loss of information from the less-processed data. While a minimum requirements for an imagery service might be satisfied by handling seamless imagery, the other cases will at some point become important for specific applications, particularly in using imagery for remote sensing and secular change analysis.

DOCUMENTS / OBJECTS

In the case of other metadata records discussed in section 6.2.3, but also in the case of many other documents and datasets, applications will require access to stores of this information, but geolocation information may only exist elsewhere. For example, sample analyses may contain a location id that is an attribute of a separate sample location feature set. A content management system such as the NBII portal may provide the repository for such documents and datasets, but a service interface is also required in order for other applications to bring together such content with geospatial content such as feature datasets. For example, this is an important requirement for enabling drill-down in visualization applications.

As mentioned above, this functionality can sometimes be implemented with direct local connections from applications to databases or CMSs, but in the general case a Web-accessible content repository will be required which minimally allows diverse content to be stored and retrieved by unique identifier, and may also permit more advanced means of querying and sub-setting the information.

LINKS AND JOINS

There is some overlap here with section 6.2.3, but some links and joins between spatial and/or non-spatial datasets may also be considered data for some applications. The data services described above may implement storage and use of links or joins themselves. For example, an enhanced feature service capability might involve resolving links (e.g. Xlinks) contained within stored features and returning the linked content along with the requested features. Otherwise, a separate link-join service may provide association records, which a client application would use to request records, form the other data services (using unique identifiers in the association records) and perform merging or joining of those records itself.

## PORTRAYAL

A client application may choose to access geodata directly and render maps based on it, but most commonly that task is left to a portrayal service such as a map server. There are two main reasons for this. First, performance is usually better when rendering is done "close" to the data themselves with server-class hardware, with a small map image returned to the client application rather than large feature datasets. This may not be much of an issue, of course, when fast client hardware and broadband connections are in use. The second reason is that map composition is still an art and discipline that may not be central to the concerns of a particular client application. The styling of maps is often a matter that is best left to the maintainer of a map server, whose expertise can then be shared among many application users and developers.

As in maps themselves, the issues are rarely black and white. Portrayal services come in different colors that provide a range of opportunity and complexity for client applications

A minimum requirement for portrayal service output is some form of map image, in formats such as GIF, PNG, or JPEG. There are other output formats which may be required by some client applications which begin to blur the line between feature and portrayal services by including graphical representations of individual features (even including feature attributes). These include formats such as SVG and Flash. The size of a service response in one of these formats is almost always larger than that of a simple image, but a client may be able to interact with them more extensively (pan, zoom, identify) and hence make fewer service requests overall.

### FIXED STYLES

This simple and opaque form of portrayal service presents a pre-defined set of map layers in one or more pre-defined styles which are simply referenced by name in a client request. The data being used by such a service may be dynamically updated and/or available in different versions (e.g. time steps), but the choice of data is hidden behind the service interface and not of concern to the client

### DYNAMIC STYLES

A more advanced form of portrayal service will offer an interface where the client can specify the styling of a map as well. In the most advanced case, the client may even specify from where the portrayal service should obtain the data to be portrayed, causing the service to chain to an external data service before carrying out map creation. Dynamic styling is a requirement for many visualization and analysis applications, especially those that provide visualization of other linked data which is not directly geo-located and/or not available to the portrayal service itself.

STYLE MANAGEMENT

Even with dynamic styling, creating an effective map visualization is hard work, and it is usually advantageous for effective map compositions and map layer styles to be saved, re-used, and shared with other users or applications, usually in some form of style catalog as discussed in section 6.2.3

## GEOLOCATION

As defined under feature services, a feature contains some geometric representation of a geographic entity. There are other mainly textual ways of representing a geographic entity, principally as place names or addresses. Geolocation services provide the means of converting between feature and textual representations.

### GAZETTEER

A gazetteer service provides as a minimum an interface to request a feature representation of a place name, such as a town, river, mountain, etc. A gazetteer may additionally organize place names into geographic or other hierarchies–this is especially useful for use by an application's place name navigator function. It may additionally provide equivalences between place names, indicate the preferred place name out of a set of equivalent names, and also provide more than one geometric representation (e.g. center point, bounding box, outline polygon) for a given place name.

"Place name" may be defined narrowly to mean widely accepted geographic place names (for example in the USGS' GNIS), but it may also be defined broadly to mean any reasonable unique textual representation of geography (for example, a sample ID). For this reason, there is a good deal of overlap in functionality between a feature service and a gazetteer service, since a place name query may look essentially like a feature query on a place name attribute.

### GEOCODER

Another textual representation of location is the address. In principle this is not so different from any other location attribute, searchable on a feature service. In practice, the implementation of geocoders may be vastly different, because there is no one universal encoding for addresses, and because a directly addressed feature collection may not exist. A useful geocoding service generally requires effective and tunable heuristic rules for putting addresses into a "canonical form". It then generally needs to be able to calculate address location "features" dynamically by processing other feature types such as streets with address ranges, zip code areas, and so on. A geocoder service may optionally indicate the quality of a calculated location and/or return a list of possible locations for the client to choose from.

### GEOPARSER

A geoparser is a service that makes use of other geolocation services to parse arbitrary amounts of text for place names and addresses. This is an essential, but difficult step in the process of "geo-enabling" non-spatial content for use in the GIF. The first step involves searching for, identifying, validating, and optionally converting place names or addresses to features. This may be entirely sufficient for geo-enabling semantically clear documents such as sample analysis spreadsheets or other observational data, and really constitutes the most important GIF requirement for geoparsing.

The second step involves determining the significance of a geoparsed feature by analyzing the textual context in which the geoparsed term is found. The general case of this problem is nearly intractable, but specific cases involving documents and language common within NBII may still be practical to implement as a service in the GIF.

## PROCESSING

(Geo-) processing services are generally characterized as those that accept one form or format of geodata and return another. There are several forms of processing which are good candidates for shared services within GIF. There are also several challenges to doing this effectively:

- ✓ **Derivation**–the response from a processing service is a derivative dataset and is only properly understood in that context, requiring the creation, maintenance, and accessibility of metadata which reference the original data, as well as any processing directives

- ✓ **Performance**–while shared remote processing service may be useful for many application tasks, performance considerations for something like coordinate transformation may require this processing to be done locally as well, either as part of other services or by a client.

- ✓ **Time**–many processing operations such as simulations or workflow are time-intensive and unsuitable for the usual services request-response paradigm. A different service invocation paradigm may be required, where a processing operation is "inserted" into the service and the client then requests status information (or is notified e.g. by email) until the operation is done, whereupon the result becomes (at least temporarily) part of the service content which may be requested by the client in a separate operation.

There are simple processing tasks such as geocoding, where the processing result is one or more features which can be efficiently invoked with essentially a feature service interface. In other cases, the input data are more extensive and/or the output data are very different from the input data. In such cases the most appropriate common service interface may be difficult to determine; it may be that use of persistent transactions against a feature service interface may be a good choice. This does require that "feature collections" representing the input data, output data and status be effectively linked within the feature service.

### TRANSFORMATIONS

These services tend to provide utility operations, such as transformation from one coordinate reference system (CRS) to another. A coordinate transformation service is a particular requirement of the GIF because requirements for measurement and analysis involve projected CRS, but no one CRS is suitable across North America, let alone globally. It is important just for enabling applications to transform extent parameters so that users don't become lost in switching CRS'. Other transformations may involve conversion from one data format or schema to another.

Transformations such as transforming point datasets into grid/ contoured ones involve significant choices in processing directives and begin to look more like model or simulation tasks.

There are a significant number of transformations which are really GIS tasks, such as cartographic modeling, topological operations, or grid algebra, which are also good candidates for shared services, and also may serve as building blocks for constructing composite services to perform modeling and simulation as discussed below.

## SIMULATIONS

Services which perform modeling or simulation are not the first priority for the GIF, largely because the case for scale and ease of use which would translate into a good shared service, is hard to make given the great variety of models being used in the scientific community. There are three approaches that may be taken to defining specific requirements for modeling services:

Identify a modeling process whose use is so standardized and widespread that it make sense to develop a service interface specifically for the purpose.

Make use of the "persistent transaction" paradigm discussed above and work on schemas for three feature collections that are sufficiently common across models or at least communities, that they are reusable from application to application. There are some efforts known in this area (e.g. XMSF) but they are far from conclusive.

Build up model processes from simpler but more common building block services, as discussed above for some GIS operations. For example, many landscape models are built up from a common set of grid algebra operations. Either an application would then chain data from one service request to another, or a workflow service would perform that function, following instructions given by the client in a workflow language.

## WORKFLOW

Particularly where modeling, sensor collection, or other time intensive operations are involved, there is room for services that act essentially as managers of other services, or as application valets perhaps. Such services, some of which are referred to as chain managers, are able to invoke and control a chain of service requests and processing steps, either from a pre-defined workflow or following a requested workflow definition. There are a number of implementations both of workflow languages and of workflow engines which orchestrate not only multiple tasks and datasets, but also multiple users or actors in the case where approval, scheduling, or validation steps require intervention by multiple people.

A more specific definition of the requirements for a workflow service has not been done at this time, awaiting the definition of candidate workflow use cases from which such requirements can be practically derived.

Human Interaction

User applications are mentioned her under the rubric of human interaction services in order to emphasize some of the requirements which are common both to service interfaces and user interfaces.

STANDARDS

There tend to be precise definitions of service interfaces in terms of message schemas and data formats. Such definitions are more difficult for human interfaces, but no less important, if only because many developers reserve the right to use their creativity in developing mapping interfaces. The resulting mapping applications typically still look alike but function differently enough to frustrate users as they switch from one to another. NBII has made some progress in codifying standards for geospatial user interfaces. These standards should be cataloged and available to assist their adoption, in the same way as standards for any other services; a necessary step in allowing users to work with a variety of focused applications without having to learn new user interfaces each time.

REGISTRATION AND DISCOVERY

The other reason to look at applications as services is to make sure that metadata about them are catalogued and discoverable just as for other services. This avoids the situation where a service is easy to find in the catalog, but emails end up flying around as users look for an available client application with which perform a particular task with that service.

UTILITY SERVICES

There are some operations that are specific just to the functioning of a user application, but common enough that application support is worth providing in the form of a shared service. A coordinate transformation service was mentioned above as one example; another might be a service for generation of buffer features or composite extents. Although other examples will be determined as more applications are built within the GIF

## AGENTS

An agent service is one which, once an event type has been registered with it, waits for such an event and then performs a task or makes some other response which is asynchronous to the original registration request. In its most general form a client registers an event and requests an action. The event may be built into the agent's capabilities, or may be provided by the client as a combination of a service request, the response triggering action, and a frequency with which the agent should issue the service request. The action may be a pre-defined form of notification (email, IM, pager) or another submitted service request (e.g. contour this new dataset).

Agent services have at least two important uses in the GIF, in performing notification, and in automating workflow. Other use cases for agents are likely to arise in the future as workflow needs are developed within NBII.

NOTIFICATION AGENT

A notification service is an important aspect of some other required services in the GIF, whether it is implemented as an integral component or as a standalone "helper". It is important, for example, that a catalog be able to harvest new dataset descriptions or service capabilities that have been registered with it, then notify users who have requested news of changes to certain holdings or types of holdings, or areas of holdings.

WORKFLOW AGENT

Workflow agents are particularly important in the GIF because there are many datasets in the scientific community that are not kept up to date because the processing tasks to update them are so tedious and time-consuming. An agent is a way of scripting such tasks– for example, by initiating a new contouring request when a point dataset has been updated. In this fashion, update of the contours doesn't have to depend on someone checking personally that new data have come in, and then finding the time to perform the necessary processing, or worse, having to track down someone else to do so.

## CONTENT REQUIREMENTS

The use cases described in this report generate requirements not only for client applications and shared services, but also for the substance and format of content that is to be exchanged and transformed by those components. Specific content types (i.e. grouped information and semantics) are required in order to carry out specific tasks in the use cases, such searching for datasets (content metadata) or mapping observation locations (point geodatasets). Specific formats are important for several levels of interoperability. On a technology level, heterogeneous distributed applications require predictable if not standard data formats for exchanging information. On a content level, format or structure is (and will be for the foreseeable future) an important mechanism for communicating meaning or semantics.

Discussed below are general requirements for geodata, specific requirements such as gazetteer data, and other data such as dictionaries, leading into the discussion of metadata requirements in Sec. 8.

It is worth re-stating at this point that there is no a priori distinction between data and metadata. What are data in one context may be metadata in another. The distinction is more properly dependent on the particular user and application requirements being considered. A distinction is drawn here only to the extent that anything identified as data for GIF applications purposes will generally have corresponding descriptions (i.e. metadata) which allow it to be published, discovered, communicated, and utilized. The descriptions themselves also require descriptions, ad absurdum; the similarity to ontologies is not accidental. For example, the FGDC metadata standard or ISO 19115 may be thought of as "meta-metadata"

In the interest of practicality, the guiding principle for brevity here will be application relevance. In other words, ISO19115 (or ISO 19139) may be an important reference document for maintaining metadata and building discovery applications, but it is not in itself operational content for GIF. In the end, distinctions between content, meta-content, and reference materials will evolve along with the GIF applications and the needs of NBII user communities.

## FEATURE DATA

Feature data, the bread and butter of the GIF, combines geometric representations of *geographic entities* with attributes of those entities and/or other related information. There are definitions of features which more or less precisely circumscribe what is meant by *geographic entity*; for example, determining whether a sampling location or a polygon drawn around all sightings of a particular species is truly a *geographic entity*. For the purposes of the GIF a broad definition will be required, as there is much information to be visualized on a map which will not necessarily relate to something like visible geography.

### COMPLEXITY, SCALE, SYSTEMS OF REPRESENTATION

There are many possible geometric representations for geography, which can become quite complex in the pursuit of precision and realism. The overwhelming requirement of the GIF, however, is the efficient exchange of information. In this respect, experience shows that simple representations are appropriate to the scale and precision of most underlying location data, as well as to most application needs. Most tasks can be accomplished with (multi-)point, (multi-)line, and (multi-)polygon geometries, as defined, for example, in the OGC Simple Features Specification. In the limited cases where this is not sufficient, additional information (e.g. topology) can usually be layered on top of these representations to elaborate them, leaving the basic information still available as needed.

As geometric representations, feature data may be not tied to a particular screen resolution, but still have particular scales of relevance to reality. For example, the number of straight segments chosen to represent a curving river may be a good approximation at 1:50000, but a poor one at 1:5000, and an excessive amount of data at 1:1000000. Applications in the GIF which provide overview and drilldown of NBII data will require not only documentation of the appropriate scales for viewed datasets, but also links between original datasets and those which have been derived by generalization for broader overviews (either on-the-fly or pre-prepared). In some cases, generalization may important for other reasons, such as protecting the nests of endangered species, but it needs to be equally clear what are the original and what are the derived data.

Coordinate reference systems (CRS) for feature data may be basic geospatial knowledge, but problems with them are responsible for more breakdowns in distributed geospatial applications than any other. There are two common issues:

- ✓ Geographic coordinates (latitude – longitude in some order, datum, etc) are more widely applicable but only projected coordinates provide credible (flat) maps and ability to do geographic analysis.

- ✓ Data can be transformed to a common CRS for overlay maps, but this only works if the original coordinate system is *accurately* documented.

Another less significant issue is that, like data compression, data transformations may degrade the original data if the transformed straight segments representing a polygon, for example, no longer accurately represent curved entities.

Data to be managed in the GIF cover a greater geographic area than can be represented accurately in any one projected coordinate system. This suggests that publishers should be encouraged to make their data available in geographic coordinates (WGS 84 or ITRS-derived CRS when developed), but provide wherever possible the capability in services or applications to project those data to a local CRS.

## GEOLOCATION

Another form of derivation will occur when geography is represented in datasets not by geometries which can be directly mapped, but by indirect geolocation information such as place names, area ID's, or addresses. In other cases, direct geographic coordinates–as from GPS–are available, but intended to match a standard location for visualization (e.g. stations for repeat sampling).

There are at least three ways in which these types of datasets may be "geo-enabled":

- ✓ Geometry may be constructed and added to the dataset permanently using a gazetteer, geocoder, or key-join operation
- ✓ A separate, derivative dataset may be created containing constructed geometries and key fields that may be joined on the fly to the original dataset for publishing purposes.
- ✓ A link dataset may be created which contains two key field for each record, one pointing at the original dataset and another pointing at a feature dataset. All three datasets are dynamically joined for publication.

## FORMATS AND STANDARDS

There are any number of incompatibilities between feature datasets that make it difficult to exchange them across a federation of services and client applications such as the GIF. These barriers to interoperability fall into distinct levels.

**Formats** are generally the first level of difficulty. Many imposed and de facto standard file formats exist, but have fallen short of universality because they were either too complex (e.g. SDTS), or not sufficiently expressive (e.g. shape). The best present candidate is Geographic Markup Language (GML), which is exceedingly expressive and leverages the use of XML. Its adoption is as yet hindered by both potential complexity and XML bulkiness.

**Schemas** or information models are generally the second level of difficulty. For example, one road feature may be modeled in many different ways even just using GML; neither machines nor humans are able a priori to recognize the different forms as being the same feature. This incompatibility can be addressed by agreement on standard feature models (e.g. application profiles and schemas in GML); the models and corresponding schemas then need to be well documented and available to data creators and maintainers.

**Semantics** or expression of meaning within a given schema pose a third level of difficulty. For example, if two data schemas both have a field <owner>, do they refer to the same concept, or does one field mean "owner of the feature" and the other mean "maintainer of the feature data"? This difficulty is acute even when a standard schema is well "explained" but the translation from other schemas is not clear. A schema language such as XML Schema is not very helpful in this regard because it mainly constrains the syntax of data, not its semantics. Metadata such as FGDC cover some of this ground, but focus more on the dataset instance rather than the schema or type. Until more semantically focused schema languages (and their accompanying ontologies) have been established as common practice, the most practical approach to semantic interoperability at present seems to be to agree on a limited number of standard schemas and fields, then to reach agreement operationally on their semantic content through guidelines and transformation procedures. Clearly this is a huge challenge for an organization such as NBII, but an important one to be undertaken albeit in manageable portions.

## IMAGERY AND RASTER DATA

In one sense, grid or raster-based geodata are just more feature data in which the features are regularly arranged cells filling space (2D, 2.5D, or 3D). Data of this class, however, are sufficiently different in so many ways from other feature data, as discussed in Sec. 6.3.2, that they merit a separate requirements focus.

### ORTHOPHOTOGRAPHY

Aerial orthophotography serving as background for other map layers, perhaps along the elevation shading, is probably the most basic imagery requirement for GIF operations. Challenges – scale-appropriate resolution, CRS transformation, color match, cloud cover, diachronous image sources, etc. – are the same ones faced by the National Map program in serving an ortho framework layer; the GIF would most likely follow its lead as well as making use of the layer itself if/when possible.

### MULTISPECTRAL IMAGERY

An additional layer of complexity is introduced with multi-spectral imagery such as LandSat, because both visualization and analysis of the image values involves processing and interpretation. Initial requirements in this area will likely involve metadata for map products that have been pre-processed from such imagery and are served as map layers. Subsequent development of GIF applications involving remote-sensing interpretation will introduce requirements for serving and exchanging imagery formats which support multiple bands. Here the picture gets cloudy, so to speak, with the same format trade-offs between complexity (e.g. HDF-EOS) and limitation (e.g. GeoTIFF) which have plagued feature data formats. One hopes that support (e.g. HDF/EOS documentation) for a common imagery format will be further developed in the near future.

### INTERPOLATED COVERAGES

In this category are digital elevation models and other raster data which involve a (model-dependent) generalization of limited observations or measurements to a continuous field of parameter values across a geographic extent. In addition to portrayal and processing requirements, these datasets have additional status as derivative content. Many such coverages may be generated from the same observational data depending on interpolation method, model assumptions, and application needs.

### SIMULATED COVERAGES

Many models and modeling methods that may at some point form the basis for a GIF application, provide grids or other coverages as outputs (as well as inputs). Examples include finite difference and finite element models, as well as landscape analysis and other grid-algebraic procedures. In these cases there will be close correspondences between a stack of grid datasets, an application state, and documentation of a model run.

## GAZETTEER DATA

Gazetteer datasets appear at first glance to consist of place names and locations, either of which may be searched to find the other. There are a few more aspects of gazetteer data, however, which are important for specific applications.

### GEOMETRIES

Navigation by place name requires an application to obtain a map extent corresponding to a chosen place name. Few gazetteer datasets actually have box geometries for their place names, however. Those focusing on place names are more likely to contain only point locations, while arbitrary feature datasets which are pressed into service as gazetteers may contain quite complex multi-line or multi-polygon geometries which are overkill and excessive overhead for simply determining a new map extent. In some cases, the dataset may be processed to replace its original geometries with plausible extent geometries, or a gazetteer service may perform this task as place name geometries are requested. In the case of point geometries, the best that can usually be done is to guess a likely extent from some indication of place type (e.g. town, county, state). This is either done in pre-processing, by the gazetteer service, or else in a (relatively smart) mapping client.

Geoparsing does not necessarily have the same requirements for place name extents, depending on the type of feature which is being derived, and the number of place names which are parsed from a given document.

### RELATIONSHIPS

Another unusual aspect of gazetteer feature collections is the likelihood of both spatial and lexical relationships between the place name features. Place names are commonly arranged in a geographic hierarchy, so that any one place name feature also refers to a broader or enclosing place name, as well as one or more narrower or enclosed place names; this taxonomic structure makes hierarchical place name navigation menus fairly simple to construct.

It is common, too, for different place names to refer to the same place. Place names may be duplicated, sequential in time, from different languages or political groups, etc. A common approach is include in each gazetteer feature an attribute which references the preferred equivalent place name out of any which are included in that gazetteer. Searching the gazetteer on this key should return the set of equivalent place names for that geographic entity.

## TAXONOMIES

Not just place names, but really any terms may be usefully arranged in a taxonomy for purposes of organization, structured searching, and/or to imply material relationships. Species taxonomies are of course central to biological studies, whatever their basis or motivation. Taxonomies will tend to straddle the line between data and metadata in the GIF, since they are one form of NBII content to manage, but also provide support application tasks such as searching, sorting, navigation, visualization, and analysis.

### CLASSIFICATIONS

The essence of applying taxonomy is the classification of other content according to its structure. For example, taxonomy of benthic invertebrates becomes a useful tool in the GIF when (geo-enabled) seafloor samples have been classified so that browsing the taxonomy enables one to highlight corresponding samples on a map. The other advantage of maintaining classifications separately from the classified data is that the classification can be performed (and re-performed) by more than one publisher to more than one taxonomy without altering the original observational data.

### CROSSWALKS

Once data or metadata have been classified according to more than one taxonomic scheme, however, it begs the question of comparison. In general, drawing connections between taxonomic elements of difference schemes will be another one of those elemental challenges / opportunities of NBII which go beyond the GIF. It must be hoped that GIF applications, though, will be able to assist this process with tools for geospatial analysis and visualization of taxonomic correspondences.

## DICTIONARIES AND THESAURI

Dictionaries and thesauri, collections of terms with less structural semantics than taxonomies, nonetheless fill important roles in the GIF, for understanding scientific and technical terms of course. Their particular importance here is to define data and parameter domains, critical for content discovery and validation; they are also important for other purposes such as definition of theme rules for map portrayal.

Many of the same requirements exist for these collections as for taxonomies, such as making them available online and current for use as application resources and in data validation routines, as well as supporting crosswalks between terms in different thesauri.

While the overall purpose of the Geospatial Information Framework may be to connect users with information, it is safe to say that the GIF engine will run on metadata–that information which allows the discovery and effective utilization of loosely federated GIF components by users in a wide assortment of roles.

This section presents GIF requirements for the set of metadata types needed to fulfill the use cases in this report. The first requirement here is for adoption across the GIF of standards and management practices for these types of metadata, to provide a common GIF "language". Ideally, the standardization will extend to other affiliated organizations as well. The second requirement, however, is that these standard metadata fulfill the operational requirements of the GIF and this requirement may bring NBII to take the lead in establishing or extending other metadata standards. While many NBII requirements are indeed unique to this organization, most are applicable to other information infrastructures and such activities, as metadata standards advancement should result in value to other organizations.

## CONTENT METADATA

Given FGDC and ISO accomplishments, standards for this type of metadata are more fully developed than for any other. There is also much work on biological metadata extensions. There are at least two areas in which further work needs to be done, however, to support GIF needs:

### DATA TYPES OR TEMPLATES

FGDC or ISO metadata are good at characterizing a dataset instance. They are less adept at characterizing a dataset type or template for creating additional, similar datasets. It is crucial for GIF functionality across the NBII federation for datasets to be created and maintained as a limited number of well-known data types. It isn't necessary to change the existing metadata standards to accomplish this, than to work on creating profiles of these standards sufficient to define consistent "types". This not only accomplishes the goal of "typing" NBII geodatasets, but actually makes it easier to produce and manage full metadata for them.

It is always difficult, of course, to persuade data creators not to create data schemas unique to their special needs. It can be hoped that persuasion will be effective in the form of easier metadata creation and access to a wider range of applications that work with their data. Creators may also be persuaded to generate derivatives of their "proprietary" datasets for use in the GIF.

### SCALE OF DESCRIPTION

Most content metadata standards focus on describing datasets, or homogeneous collections of data objects. This is one useful scale of description, but by no means the only one. It is quite common for data to have important-to-document characteristics at a number of scales of aggregation from individual features (e.g. stream reaches) or cells to entire data libraries (e.g. VMAP0). For GIF purposes, metadata for multiple scales of data aggregation may be simultaneously needed. For example, administrative metadata may exist for a single species range, while quality metadata have been developed for a set of species ranges derived from a specific set of observations, while the whole package of datasets used for a GAP analysis, in which the range collection is included, may have contact metadata for the analysts who performed the analysis. It is important, therefore, to document (meta-metadata again) what scale of data aggregation particular metadata elements pertain to, and how metadata for one scale is related to metadata for another scale.

#### SERVICE CONNECTIONS

As discussed in Sec's 5.3.3 and 6.2.2, it is possible to include within content metadata, elements with connection information for services offering that content. The disadvantage of this approach is that service metadata are by their nature more dynamic than content metadata. One approach is for content metadata to be dynamically updated at each access. It still runs the risk of going stale, however, on a user's computer. A better approach is to provide connection information to a catalog service with search parameters for discovering all current offers for that content.

The requirement for this to work is the assignment of a unique identifier for each "data product" published in the data catalog (and other federated catalogs), which can then be referred to by any relevant content "offers" in the services catalog(s). This identifier is crucial not only for making the content – service connection, but also for making connections between different service types, such as portrayal and data services.

### SERVICE METADATA

The purpose of service metadata is to allow users and applications to discover and consume services with a minimum of private understanding and out-of-band communications. It is probably not possible for computer application to connect to and consume an unknown service purely from some sort of first principles, despite claims made for some forms of service metadata. The best practical hope seems to be a strong typing of Web Services, so that each service type can be thoroughly characterized for application developers as well as users. As long as each instance of such a service type is sufficiently conformant, it should not generally be necessary for each new user to call the service provider in person.

There are still competing standards for services metadata which all have their strengths and weaknesses. None is perfectly suited to GIF requirements, yet several may be important for sharing service information between NBII and other organizations. Looking at combined GIF operational and enterprise requirements, it seems wise to develop a service metadata type for GIS that is in some sense a superset of existing prominent standards. It then becomes practical to supply service metadata records in those standard formats as needed, for example in publishing NBII content to Geospatial One-stop.

Briefly, the GIF requires that certain sections be present for each service metadata record:

#### SERVICE TYPE

This describes what well-known service type (e.g. Web Map Service v1.3) this service is an instance of.

SERVICE DESCRIPTION

Known as service metadata, sensu strictu, it describes how this service should be characterized, who is responsible, what are the service conditions, etc.

CONNECTION INFORMATION

Also known as a service offer, this is the basic protocol information for connecting to the service, what operations are supported, global options, service levels, etc.

CONTENT OFFERS

This is the meat of a content-oriented service, identifying each content item that the service offers and specifying how to request it (e.g. GetFeature), as well as what request options are available (e.g. CRS). It is this information which, when correctly configured, allows a client application to magically offer all of the service's capabilities to a user. For GIF purposes, content offers are also queried from a service catalog to match up a dataset with a service on the basis of a unique dataset identifier.

Just as content metadata may apply to different scales of aggregation, so is it the case with content offers. A content offer from a feature dataset does not commonly apply to a single feature– that would in many (but not all) cases result in excessive and repetitive metadata. In the case of imagery scenes, each one is itself already a collection of cells with corresponding radiance values, yet with image services accumulating millions of scenes, one content offer per scene also begins to sound excessive. In the end, there may need to exist content offers defining access to a variety of content aggregations, setting one as a default, but leaving it up the application to choose the most appropriate scale for its purposes.

## CONTENT CONNECTIONS AND DERIVATIONS

As discussed earlier, an important goal of the GIF (and indeed of NBII) is to facilitate users' making connections between datasets and between other works created by an entire continent of biological and related projects. Free text search in a catalog is only a first step. Some of this link information can be encoded in datasets themselves or their metadata records. One disadvantage is that, again, content metadata standards are not fully set up to perform this function. The other is that connections and derivations are often a value which is added by others, not just the primary data creators, and as such should have a separate existence from the datasets they link.

The requirement here is then for metadata records which express various forms of association between other metadata records, in which searchable association types, as well as roles for the linked content, can be recorded. These records then become actionable metadata for catalog clients and enable a range of drill-down functions in applications. For example, drill down to point observation datasets from a derived abundance contour dataset, then find other gridded datasets drawn from the same observations but with different methods.

## STATES AND CONTEXTS

Support for application saving as well as application collaboration requires a metadata record which can encode that information in a portable form which can be catalogued, communicated, and re-used. A basic form of this exists as the OGC Web Map Context, focusing on a map composition metaphor; the standard is currently being extended to accommodate other service and data types as an OWS Context document. These standards also make allowance for extended elements specific to knowledge domains or applications.

The map composition metaphor will be a useful one for recording the state of most GIF applications, but will probably be strained in advanced modeling applications. In these cases, any of a number of developing model portability languages will provide a useful addition or alternative.

## STYLES AND SYMBOLS

Cartography is a visual language and at some point needs a grammar for comprehension. As maps become greater in number and wider in availability across NBII, the need for common cartographic paradigms in them will increase, so as not to strain the ability of most users to interpret them.

A first step is document the styles which are made available through GIF map services, by encouraging software support for returning at least legend images and if possible full map composition metadata for available map images. The OGC Styled Layer Descriptor is a good first pass at accomplishing this.

Cartographic comprehension will eventually be advanced by providing finer-grained (e.g. per feature type) portrayal styles and symbols for re-use by map creators or for dynamic use by application users. This requires the creation and management of style and symbol metadata documents, so that styles and symbols appropriate to particular data, knowledge communities, and purposes, can be discovered, shared, and utilized.

This is another “meta-metadata” issue, as the style information itself is a form of metadata and the searchable description of a style may be separate from the style itself. It is probably sufficient as a first pass to place descriptive material inline with the style information itself and catalogue only one form of style metadata. The situation is slightly different for symbols, since symbol formats such as Truetype or Postscript may not be searchable. In these cases, the actual symbol definition may need to be held in a repository or even a symbol service separate from searchable descriptions of those symbols.

## OTHER METADATA

Other types of metadata will become important to advancement of the GIF, but the types described above should be sufficient for initial implementation. There is, for example, an entire class of metadata types related to personalization, authentication, and authorization. In the case of NBII and the GIF, elaboration of these types and their roles should be tied closely to the capabilities provided by the NBII portal.

## ENTERPRISE REQUIREMENTS

The foregoing requirements have focused on those directly tied to use cases and applications envisioned for the GIF. Requirements in this section are related more to some of the organizational contexts in which the GIF will be implemented.

### NBII

#### ENTERPRISE ARCHITECTURAL CONSIDERATIONS

The GIF, as described in this document must meet the requirements as stated in the NBII Enterprise Architecture Plan, as well as meet the USGS and DOI Enterprise Architecture Requirements. Namely, the GIF must support the NBII goal of distributed data ownership and stewardship, while allowing these data to be accessed centrally by a typical NBII user.

#### NODE CONSIDERATIONS

There are several considerations that should be noted in this requirements document related to the NBII Nodes:

1) There should be no or very little cost to add or register mapping resources and datasets into the GIF as well as take advantage of the Rapid Application Development Toolkit built for use by the nodes,

2) The complexity of the GIF should not become a barrier to adoption, give that the varying level of resources within the nodes. The tools will need to be easy to use and easy to understand, with clear, demonstrable benefits to the participating nodes,

3) The NBII Program office will continue to provide technical support and training to help the Nodes reach their goals of fully participating in the GIF

### NATIONAL MAP

A major new vision of the USGS is a set of nationwide framework geodata layers which will be made available online as a basic public service. GIF requirements should address both the utilization of National Map layers in GIF applications, and possible return contributions of NBII geodata. Compatibility requirements fall into several areas, subject to ongoing evolution of National Map thinking in those areas. Compatibility with GIF will have a grounding in adherence to OGC standards; a challenge will be the changes and extensions made by each organization to address their specific needs where the present OGC standards are not yet sufficiently evolved.

#### CATALOG COMPATIBILITY

The National Map cataloguing effort appears to begin with NSDI-type Z39.50 catalog interfaces and aim at OGC Catalog v2.0 compatibility, although there appears to be an intermediate Web interface in which a custom query language is employed. In order for GIF users to search National

Map content, one or more gateways to those interfaces may be necessary; The OGC catalog specification does not yet specifically address cascading or federating catalogs, but is likely to do so in a next update since requirements to do so are rapidly appearing.

#### SERVICE COMPATIBILITY

National Map services are likely to be OGC-compliant map and data services, possibly also including gazetteers and geocoders.

#### INFORMATION COMPATIBILITY

It is likely that most National Map data will be available at least in GML, although default imagery formats may not yet be established. Metadata are another issue; while content metadata will be based on FGDC standards, service and other operational metadata are likely to be different in some details from those established in the GIF, and so require some translation for effective operation of GIF applications.

#### CONTRIBUTIONS FROM NBII TO THE NATIONAL MAP

There may be some map layers or data collections (e.g. habitat information) offered by GIF services which are appropriate to become part of the National Map offering. It is clear in the National Map vision that the framework layers will draw from local offerings at larger scales, but the mechanisms for this are not yet worked out or available. It is also not yet clear how such layers would be selected, whether they would be served physically from the GIF or from National Map infrastructure, and how the National Map would update its information on availability of remote services. In all likelihood, the basic requirement would be access to an OGC-compliant service, which should be able to be met by GIF services.

### GEOSPATIAL ONE-STOP (GOS)

This project has many facets, but is essentially an effort to offer access to as much geospatial content (within and outside of government) as possible through a single Web portal. There are goals for this to become the central node to a new generation of national spatial data infrastructure with access to multiple federations of both catalogs and other services. At present, it consists of a portal application utilizing a central catalog and Web mapping service.

#### DISCOVERY OF GIF OFFERINGS

The present GOS portal provides content publishers with some ability to add metadata for their content in customized, FGDC-based text formats. Automatic harvesting of metadata only appears to be available for certain proprietary catalog interfaces. Making GIF services and content accessible through GOS will therefore require generating and submitting metadata in this format from GIF holdings deemed suitable for GOS access.

COMPATIBILITY ISSUES

Since standards support in GOS is a moving target at present, other compatibility issues with GOS may or may not arise in the future.

## REQUIREMENT SUMMARIES BY PHASE

The preceding sections of this report describe a wide range of requirements for implementing an effective Geospatial Information Framework. This section is intended to draw together those requirements into distinct and actionable phases of activity. As formulated, this summary will then provide the direct goals of the technical design described in the accompanying document Geospatial Information Framework: Technical Design and Specification.

The requirements expressed in this report are clearly more than can be accomplished in one implementation cycle or limited period of time. Many requirements laid out here are in fact not yet ready for implementation and need to be elaborated through experience. The purpose of dividing requirements into phases is not only to facilitate implementation, but also to allow experience to refine requirements in succeeding phases and lay the groundwork for ongoing cycles of implementation which continue to keep the GIF up to date.

### REQUIREMENTS OVERVIEW

Requirements have been divided into three phases of work:

- ✓ Phase 1 – work now
- ✓ Phase 2 – work next
- ✓ Phase 3 – work after that

As might be expected, Phase 1 requirements are the most succinct, while Phase 3 groups all the requirements which remain to be addressed at some point in the future and are not really ready to be implemented in their present forms.

Each phase has a concept of operations, a functional goal to be accomplished with its implementation, which is shown diagrammatically for that phase.

Within each phase, requirements are grouped into areas, such as technology requirements, application requirements, service requirements, and content / meta-content requirements.

### PHASE 1 REQUIREMENTS

Requirements in this phase are considered appropriate to address immediately for an initial stand-up of the GIF. Their focus is facilitating GIF applications to make full use of the publish – find – bind paradigm of distributed computing and jumpstarting this process with an integrated viewer application useful enough to encourage publication of content for it.

The concept of operations in this phase allows for the fundamental components of the GIF– services, catalogs, and applications–to begin interacting with one another within the context of end user applications. Most content creation, publishing and administration tasks are left behind the scenes until a later phase. Components in this phase include:

- ✓ Services such as map servers and gazetteers (other as well as other URL-accessible content)
- ✓ Catalogs for content, service, and collaboration metadata (as well as other Clearinghouse-federated catalogs)
- ✓ Applications at least one of which integrates discovery, visualization, and collaboration

Activities in this phase include:

- ✓ Harvest / management of service metadata
- ✓ Discovery of content *then* services
- ✓ Consumption of map services (maps) and gazetteer services (for navigation)
- ✓ Saving of application contexts to a collaboration catalog
- ✓ Discovery and consumption of application contexts by other users.

TECHNOLOGY REQUIREMENTS

- ✓ Basic requirements are not restricted to proprietary technology, but may be met with either commercial or open-source software implementations
- ✓ Commercial DBMS products are limited at present to SQL Server.
- ✓ GIF distributed components are not restricted to one software / hardware platform or application technology
- ✓ GIF applications are not restricted to a particular "thickness" of client implementation
- ✓ Distributed application messaging makes use of Web protocols
- ✓ Distributed content and meta-content is able to be represented in XML wherever possible
- ✓ Standard component interfaces are used for distributed / federated operations wherever possible
- ✓ Component development makes use of SOTS (standard off-the-shelf) components or toolkits wherever practical
- ✓ Components include interfaces for distributed administration as well as operation

VIEWER APPLICATION REQUIREMENTS

- ✓ Queries a content catalog using common core metadata parameters

- ✓ Queries a service catalog using a queried content identifier such as from 12.2.2.1
- ✓ Browses / searches a catalog or data taxonomy as an alternative to text queries
- ✓ Requests and overlays map images from multiple map servers using queried service metadata from a catalog or service
- ✓ Navigates directly (pan, zoom, zoom back) or by place name from a gazetteer
- ✓ Modifies requested styles for map layers where available
- ✓ Shows layer information and changes visibility in a map legend
- ✓ Changes requested map CRS and preserves map extent
- ✓ Selects / highlights one or more features on the map and displays / highlights attributes in a table where feature data are available
- ✓ Drills down to other tables or browse-able content using links in the mapped feature attributes
- ✓ Adds user annotations to the map (text) visualization
- ✓ Prints or saves a map layout image or attribute table
- ✓ Saves, publishes, and retrieves application contexts (catalog queries and/or map compositions with annotations and styles)

CATALOG SERVICE REQUIREMENTS

- ✓ Stores several (standard) catalog record types, including:
  - Data
  - Services (including their content offerings)
  - Application contexts
  - Classifications of data and services
  - Associations between datasets
  - User queries
- ✓ Assigns or manages unique identifiers for catalogued records

- ✓ Responds to standard Web record queries on both common-core and record-specific parameters:
  - Dublin Core
  - Darwin Core
  - OGC Core
- ✓ Supports online submission, harvesting, and administration of records
- ✓ Supports transactions (inserts-updates-deletes) of queries and application contexts as metadata records

GAZETTEER SERVICE REQUIREMENTS

Offers multiple collections of place name features, including at least the following attributes:

- ✓ Place name
  - Geometry (point, and/or bounding box)
  - Broader term – place names referring to an enclosing geography
  - Narrower term(s) – place names referring to an enclosed geography
  - Preferred term (connects equivalent place names)
  - Place type (especially important for inferring extents from point geometries for map navigation)
- ✓ Service metadata are able to be harvested by catalog service
- ✓ Responds to standard Web queries on place names, geometries, or other gazetteer attributes
- ✓ Returns requested subsets of feature attributes
- ✓ Is able to offer a variety of feature types as gazetteer entries

PORTRAYAL SERVICE REQUIREMENTS

- ✓ Responds to standard Web requests with images of selected map layers
- ✓ Service metadata are able to be harvested by catalog service
- ✓ Provides map images in multiple CRS'

- ✓ Provides map images with dynamic styling (including thematic styling) per request
- ✓ Responds to standard Web requests with information about features selected in map image
- ✓ Responds to standard Web requests with legend information / images
- ✓ Is able to be implemented at any node level

CONTENT REQUIREMENTS

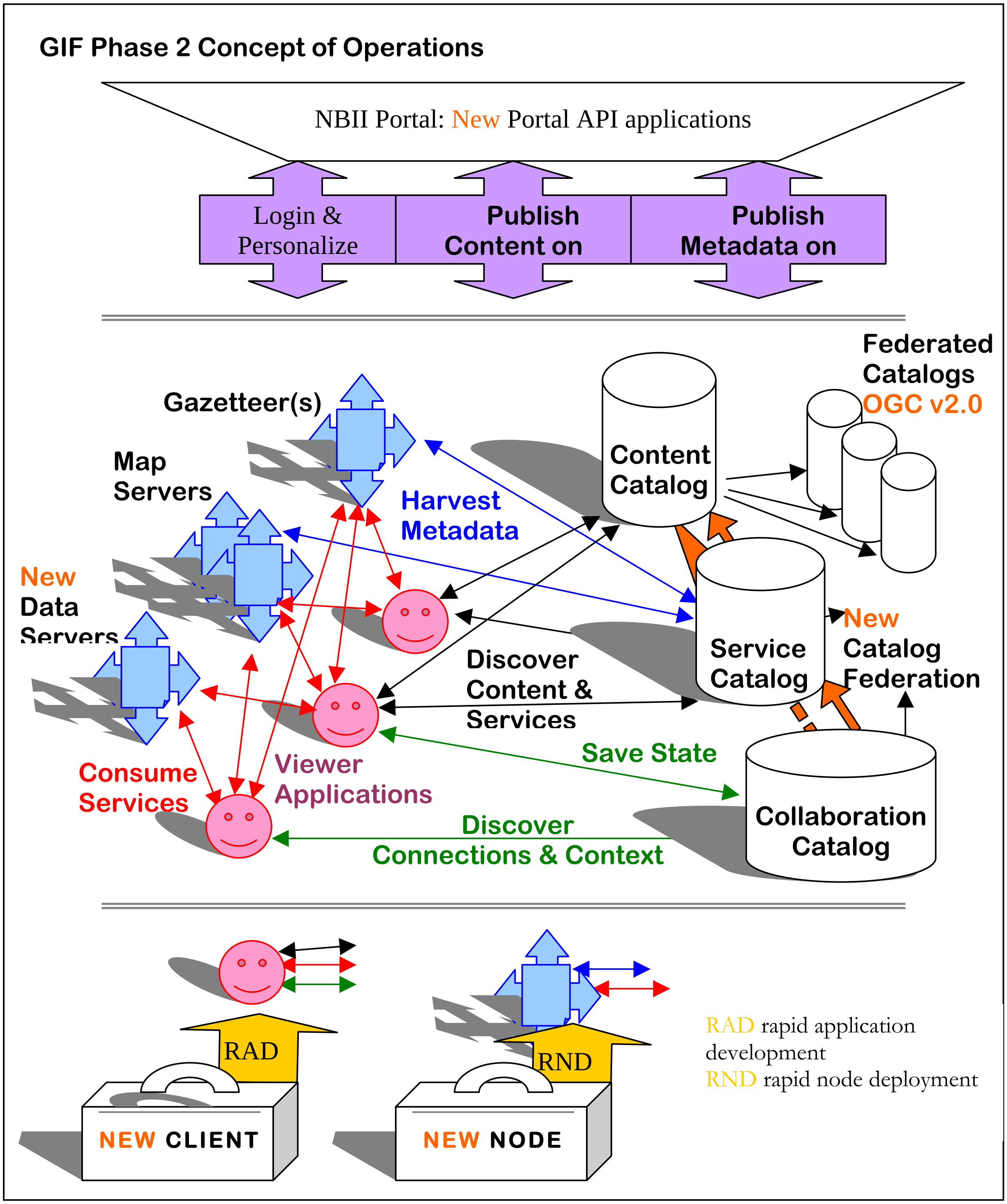
- ✓ Feature data – available in (or convertible to) GML with accompanying schema
- ✓ Feature data – offered in a standard application schema (NBII or NBII-catalogued) wherever possible
- ✓ Image, raster, or grid data – available in either GML, geotiff, png. or jpeg2000
- ✓ Other referenced NBII content – available as Web content (HTML, XHTML, XML, etc) wherever possible through a unique and stable URL.
- ✓ Geospatial data published in common geographic CRS
- ✓ Place name feature collections converted or bridged from major gazetteers (GNIS, Alexandria), as well as from NBII-specific feature datasets

METADATA REQUIREMENTS

This requirement constitutes a list of metadata record formats whose use is to be standardized within the GIF; all records are assumed to have an XML representation and accompanying schema.

- ✓ Content metadata record – including a unique identifier and service catalog search element
- ✓ Content type metadata record – the profile or archetype for standardized NBII geodata content
- ✓ Service metadata record – the metadata needed to discover and make use of a service
- ✓ Classification scheme / taxonomy record (set) –organization of content records for structured searching
- ✓ Association record – documents some actionable connection between two other content records, e.g. derivation.
- ✓ Application context record – a shareable record of viewer application state

## PHASE 2 REQUIREMENTS



Phase 2 requirements listed below focus on facilitating the publication and development processes within the GIF, by bringing the NBII portal into play through its portal API, and developing toolkits for both client applications and service nodes. These requirements include implicitly the requirements for Phase 1.

The concept of operations for phase 2 of the GIF adds functionality in several areas. Principal among them is integration with the NBII portal through its application API, in order to add authentication and content / metadata management to the GIF. The concept is to facilitate publishing on the GIF through this portal mechanism.

In addition to this facility, phase 2 envisions supporting expansion of the GIF across NBII nodes and partners with frameworks and toolkits for both application development and service node deployment. Finally, phase 2 envisions the addition of data services and additional capabilities for catalog federation.

## TECHNOLOGY REQUIREMENTS

### VIEWER APPLICATION REQUIREMENTS

- ✓ Viewer application built as a sample project for an application toolkit available to all nodes

### DATA SERVICE REQUIREMENTS

- ✓ Feature data
- ✓ Imagery data
- ✓ Database access

### PORTAL API REQUIREMENTS

- ✓ Synchronization of content types and identifiers between NBII portal and GIF catalog

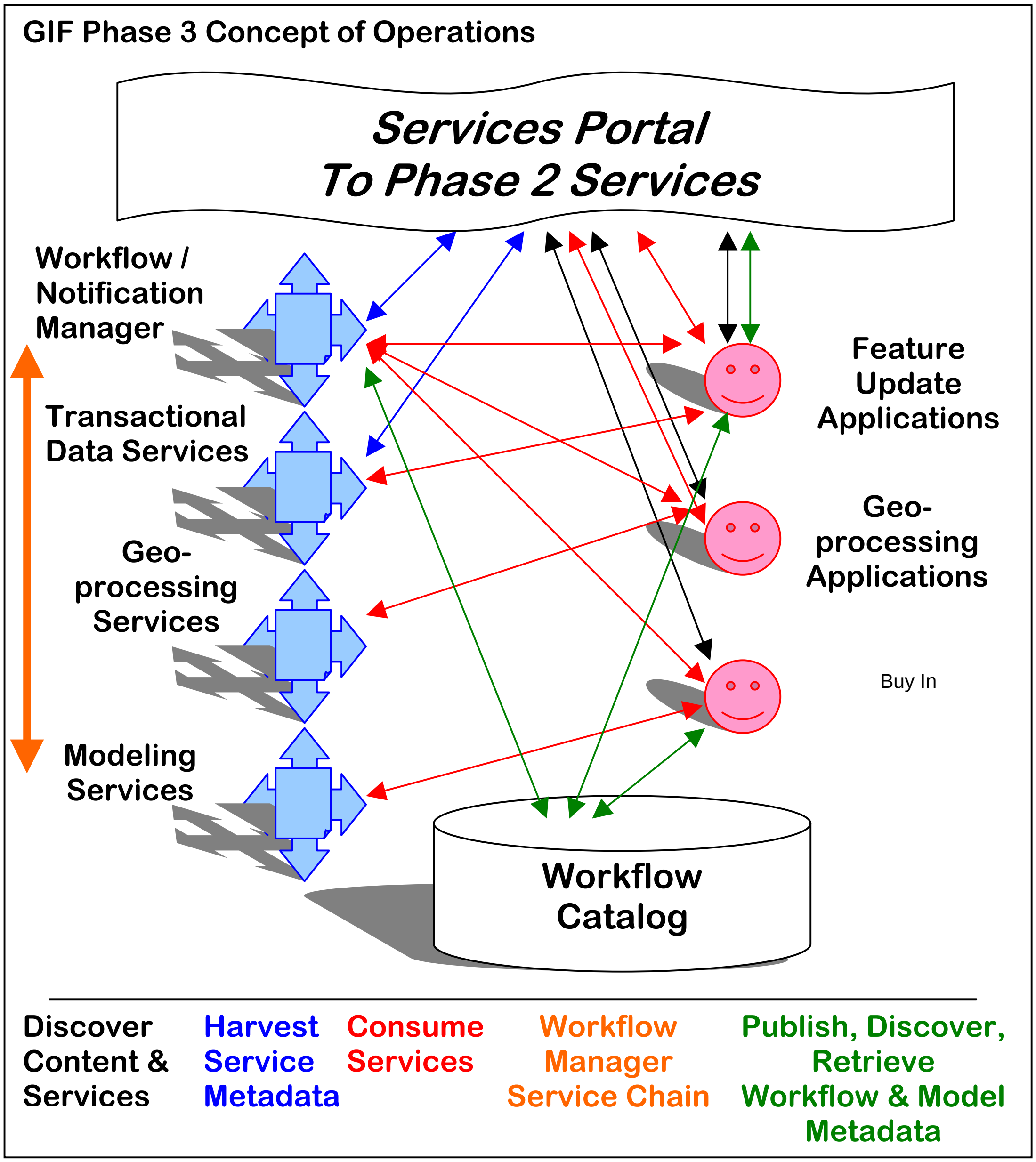
### AUTHENTICATION SERVICE REQUIREMENTS

- ✓ Catalog service(s) accept authentication decisions and identifiers from NBII portal, set ownership of metadata records, and allow owners to set public / private access rights on metadata
- ✓ Content management service requirements
- ✓ Development framework requirements
- ✓ Node kit requirements

    - Freely available to each NBII node

- ✓ Content requirements
- ✓ Metadata requirements

**PHASE 3 REQUIREMENTS**




Phase 3 introduces new client applications and services to support advanced online geospatial content creation across the GIF. The four areas of content creation comprise

- ✓ Geoparsing and geo-enablement of non-spatial content
- ✓ Transactional creation and update of feature data

- ✓ Geoprocessing operations on existing feature and grid data
- ✓ Generation of modeled or simulated data

The concept of operations for phase 3 builds on the publishing functionality of phase 2 with this additional creation functionality. An essential characteristic of this functionality is the creation and management of searchable and shareable metadata documenting the creation context and derivation of all new content.

TECHNOLOGY REQUIREMENTS

VIEWER APPLICATION REQUIREMENTS

- ✓ Data update
- ✓ Analysis – modeling
- ✓ Geoprocessing

GEOLOCATION SERVICE REQUIREMENTS

- ✓ Geocoder
- ✓ Geoparser

SERVICES PORTAL REQUIREMENTS

Undetermined at this time

TRANSACTIONAL DATA SERVICE REQUIREMENTS

Undetermined at this time

WORKFLOW REQUIREMENTS

Undetermined at this time

GEOPROCESSING REQUIREMENTS

Undetermined at this time

MODELING / SIMULATION REQUIREMENTS

Undetermined at this time

CONTENT REQUIREMENTS

Undetermined at this time

METADATA REQUIREMENTS

Undetermined at this time

| Term | Definition |
|---|---|
| Geo-enablement | The addition or linking of quantitative geospatial properties (e.g. features) to a document or dataset based on the analysis of existing location information within or related to that document or dataset. |

## BIBLIOGRAPHY


DCMI, *DCMI Metadata Terms*, DCMI Recommendation (2003-11-19). Available [online]: <http://dublincore.org/documents/dcmi-terms/>.

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

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

IETF RFC 2396 (August 1998), Uniform Resource Identifiers (URI): Generic Syntax, Berners-Lee, T., Fielding, N., and Masinter, L., eds., <http://www.ietf.org/rfc/rfc2396.txt>

IETF RFC 2616 (June 1999), Hypertext Transfer Protocol – HTTP/1.1, Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T., eds., <http://www.ietf.org/rfc/rfc2616.txt>

IETF RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1, Draft Standard (June 1999), available at <http://www.ietf.org/rfc/rfc2026.txt>.

ISO/IEC 11179-3:2003, Information technology – Metdata registries (MDR) – Part 3: Registry metamodel and basic attributes.

ISO 19101:2002, Geographic information -- Reference model

ISO 19103 (DTS), Geographic information - Conceptual schema language, (Draft Technical Specification)

ISO 19106:2002 (DIS), Geographic information - Profiles

ISO 19108:2002, Geographic information - Temporal schema

ISO 19109:2002 (DIS), Geographic information - Rules for application schema

ISO 19110:2001 (DIS), Geographic information - Methodology for feature cataloguing

ISO 19113:2002, Geographic information - Quality principles

ISO 19114:2001, (DIS) Geographic information - Quality evaluation procedures

ISO 19118:2002, (DIS) Geographic information - Encoding

ISO 23950:1998, Information and documentation -- Information retrieval (Z39.50) -- Application service definition and protocol specification

ISO/IEC 19106:2003, Geographic Information – Profiles.

ISO/IEC 19119:2003, Geographic Information – Services.

ISO/IEC 10746 (all parts), Information Technology – Open Distributed Processing –Reference Model.

ISO/IEC 11179-3:2003, Information technology – Metdata registries (MDR) – Part 3: Registry metamodel and basic attributes.

ISO/IEC 8601:2000, Data elements and interchange formats – Information interchange – Representation of dates and times.

ISO/IEC 8825:1990, Information technology -- Open Systems Interconnection -- Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)

ISO/IEC TR 10000-1:1998, Information technology – Framework and taxonomy of International Standardised Profiles – Part 1: General principles and documentation framework.

ISO/IEC TR 10000-2:1998, Information technology – Framework and taxonomy of International Standardised Profiles – Part 2: Principles and Taxonomy for OSI Profiles

ISO/WD 19135, Geographic information – Procedures for registration of geographical information items.

OASIS ebRIM, ebXML Registry Information Model v2.5, Committee Approved Specification (June 2003), available at <http://www.oasis-open.org/committees/regrep/documents/2.5/specs/ebrim-2.5.pdf>

OASIS SAML, Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1, OASIS Standard (2 September 2003). Available [online]: <http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf>.

OASIS XACML, eXtensible Access Control Markup Language (XACML) Version 1.1, OASIS Committee Specification (7 August 2003). Available [online]: <http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf>.

OGC 01-009 Coordinate Transformation Service Version 1.0.0 Implementation Specification

OGC 01-026r1, Geocoder Version 0.7.6 Discussion Paper

OGC 01-035, Geoparser Version 0.7.1 Discussion Paper

OGC 01-037, Location Organizer Folder Version 1.0 Discussion Paper

OGC 01-068r3, Web Map Service Implementation Version 1.1.1 Specification

OGC 02-006, OGC Abstract Specification Topic 12: OpenGIS Service Architecture

OGC 02-017r1, WMS Part 2: XML for Requests using HTTP POST Version 0.0.3 Discussion Paper

OGC 02-023r4, OpenGIS Geography Markup Language (GML) Implementation Specification, Version 3.0 (29 January 2003). Available [online]: <http://www.opengis.org/docs/02-023r4.pdf>.

OGC 02-058, Web Feature Service Version 1.0.0 Implementation Specification

OGC 02-070, Styled Layer Descriptor Version 1.0.0 Implementation Specification

OGC 02-076r3, Gazetteer Service Profile of Web Feature Service Version 0.0.9 Discussion Paper

OGC 02-094, Filter Encoding Implementation Specification, version 1.0 (2001-09-19), available at <http://www.opengis.org/docs/02-059.pdf>

OGC 03-003r10, Level 0 Profile of GML3 for WFS Version 0.0.10 Discussion Paper

OGC 03-022r3, Observations and Measurements Version 0.9.2 Recommendation Paper

OGC 03-025, Web Services Architecture Version 0.3 Discussion Paper

OGC 03-026, Service Information Model Version 0.3 Discussion Paper

OGC 03-036r2, Web Context Documents Implementation Specification RFC

OGC 03-040, OGC Reference Model Version 0.1.2

OGC 03-065r6, Web Coverage Service Version 1.0 Implementation Specification

OGC 03-081r2, Web Terrain Service Version 0.5 Request for Comment

OGC 04-010r1 Geolinked Data Access Service Version 0.9.1 Discussion Paper

OGC 04-011r1 Geolinking Service Version 0.9.1 Discussion Paper

OGC 04-024, Web Map Service Version 1.3 Implementation Specification

OGC 99-049, OpenGIS Simple Features Specification for SQL, Revision 1.1 (5 May 1999). Available [online]: <http://www.opengis.org/docs/99-049.pdf>.

OGC 99-113, OGC Abstract Specification Topic 13: Catalogue Services

OMG 03-03-01, Unified Modeling Language Specification, Version 1.5. Available [online]: <http://www.omg.org/docs/formal/03-03-01.pdf>.

OWL-S, OWL-S: Semantic Markup for Web Services, version 1.0 (2003-11). Available [online]: <http://www.daml.org/services/owl-s/1.0/owl-s.html>.

RFC 2246, The TLS protocol Version 1.0, IETF Proposed Standard (January 1999). Available [online]: <http://www.ietf.org/rfc/rfc2246.txt>.

RFC 2387, The MIME Multipart/Related Content-type, IETF Proposed Standard (August 1998). Available [online]: <http://www.ietf.org/rfc/rfc2387.txt>.

RFC 2388, Returning Values from Forms: multipart/form-data, IETF Proposed Standard (August 1998). Available [online]: <http://www.ietf.org/rfc/rfc2388.txt>.

RFC 2392, Content-ID and Message-ID Uniform Resource Locators, IETF Proposed Standard (August 1998). Available [online]: <http://www.ietf.org/rfc/rfc2392.txt>.

RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax, IETF Draft Standard (August 1998). Available [online]: <http://www.ietf.org/rfc/rfc2396.txt>.

RFC 2617, HTTP Authentication: Basic and Digest Access Authentication, IETF Draft Standard (June 1999). Available [online]: <http://www.ietf.org/rfc/rfc2617.txt>.

W3C SOAP-1, SOAP Version 1.2 Part 1: Messaging Framework, W3C Recommendation (24 June 2003). Available [online]: <http://www.w3.org/TR/SOAP/>.

W3C SOAP-2, SOAP Version 1.2 Part 2: Adjuncts, W3C Recommendation (24 June 2003). Available [online]: <http://www.w3.org/TR/soap12-part2/>.

W3C XPath1, XML Path Language (XPath) Version 1.0, W3C Recommendation (16 November 1999). Available [online]: <http://www.w3.org/TR/xpath>.

W3C XPointer, XPointer Framework, W3C Recommendation (25 March 2003). Available [online]: < http://www.w3.org/TR/xptr-framework/>.

XML 1.0 (October 2000), Extensible Markup Language (XML) 1.0 (2nd edition), World Wide Web Consortium Recommendation, Bray, T., Paoli, J., Sperberg-McQueen, C.M., and Maler, E., eds., <http://www.w3.org/TR/2000/REC-xml>