

WADC

GCI Domain Model

-- Version 0.1 --

Fundamental Suggestions

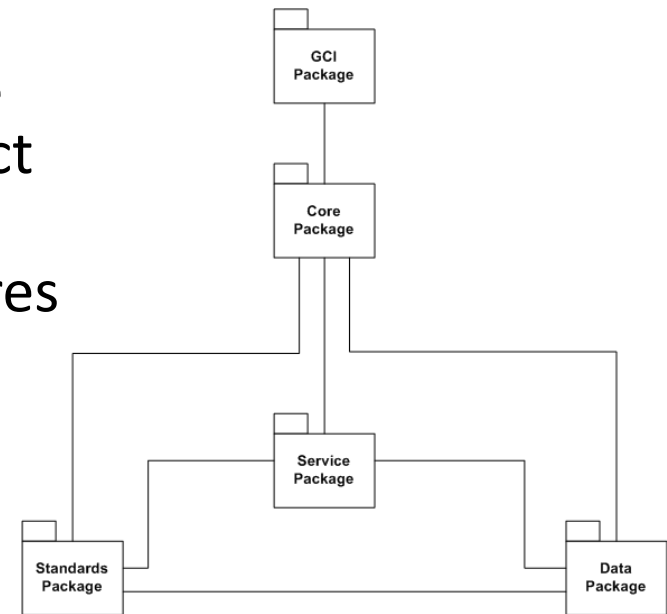
- Relationship between “system” and “component”
- Clarification between “type” and “instance”

	Data	Services	Standards
Type	DataSet/Collection	Interface	Specification
Item	Granule	Binding	Adherence

- Remote sensing data inventories should not be harvested
- Authenticated Registration
- Openness of portals
 - Some may be “endorsed”
- Assertions
 - Events are Services
 - Models are Services (or components offering services)
 - Categorization is flexible
 - Type (technically)
 - Number of schemes
 - Lifecycle of a scheme

Organization

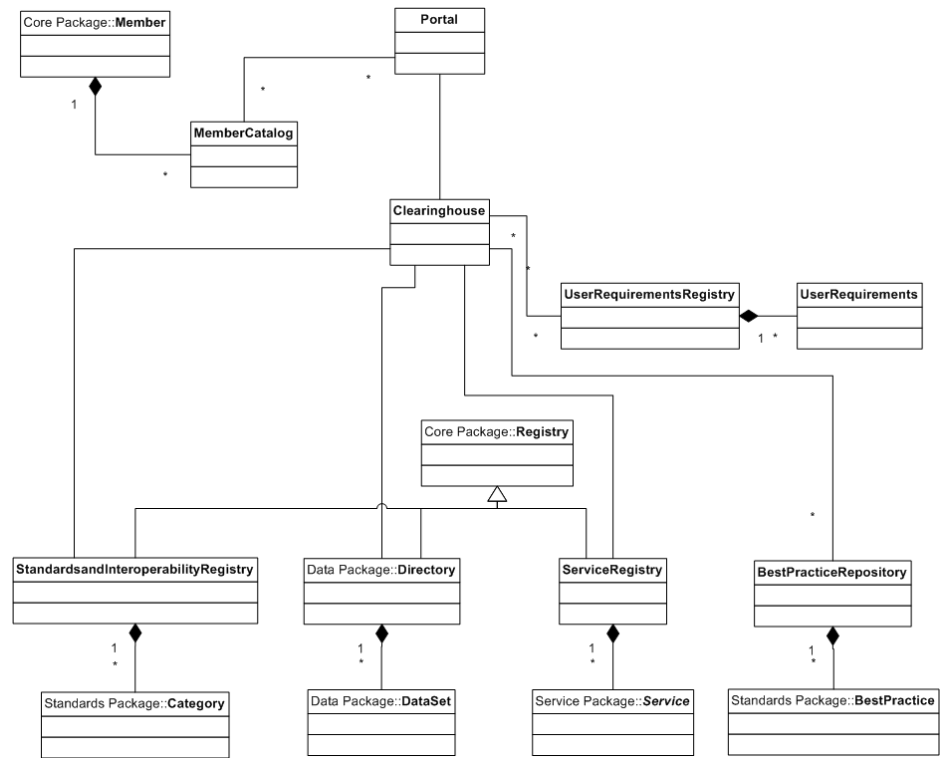
- **5 Basic Packages** (*Packages are logical constructs, meant to organize related items*)
 - **GCI** – Identified pieces of the GEOSS Common Infrastructure. These are the entities that GCI Users relate to/interact with directly.
 - **Core** – Underlying entities and structures required to support GCI.
 - **Data** – Entities related to Data contributions.
 - **Service** – Entities related to service/functional contributions.
 - **Standards** – Agreements and standards within the GEOSS community.



NOTE: For a quick guide on how to interpret the symbols in the diagrams on this and the following slides, please see the last slide.

GCI Package

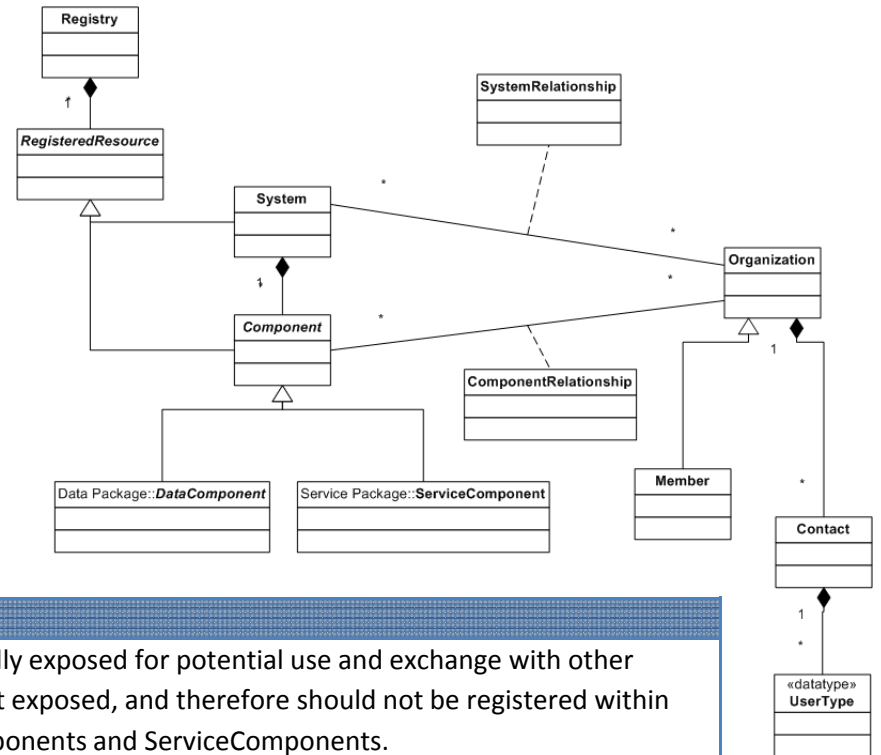
These are the publicly identified pieces of the GEOSS Common Infrastructure. These are the entities that GCI Users relate to/interact with directly.



Element	Description
BestPracticesRegistry	A compendium of BestPractices. Has a searchable interface.
Clearinghouse	The entity which provides a common discovery interface for all GCI queries.
Directory	A searchable container for types of data (datasets).
MemberCatalog	A searchable container for GEOSS Member organizations.
Portal	Human interface for interaction with GCI. Highly interactive with Clearinghouse for discovery of GEOSS-related resources.
ServiceRegistry	A searchable container for services.
StandardsandInteroperabilityRegistry	A compendium of Standards and other related agreements for interoperability. Has a searchable interface.
UserRequirement	A textual request for something to be done within the GEOSS community. Normally a request for a resource that is needed, but not available. Examples might include specific observations, model predictions or algorithm improvements.
UserRequirementsRegistry	A compendium of UserRequirments. Has a searchable interface.

Core Package

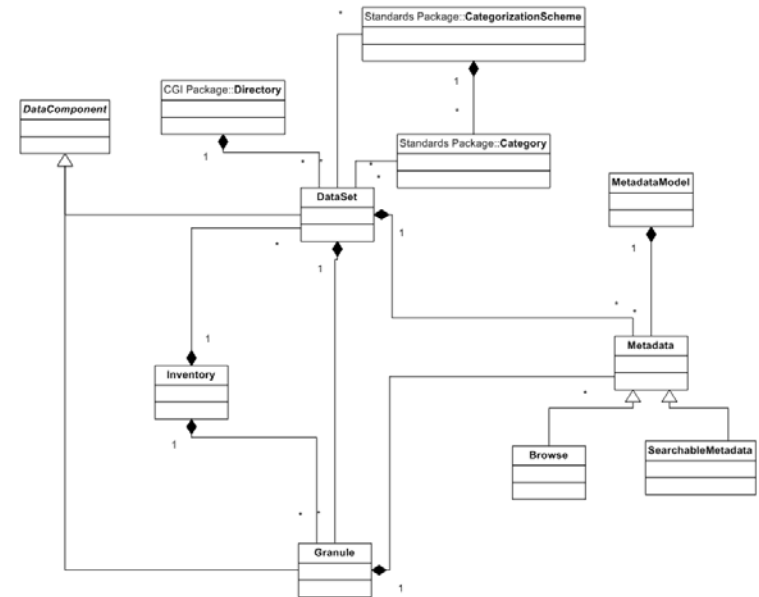
Underlying entities and structures required to support GCI.



Element	Description
Component	Components are parts of systems that are intentionally exposed for potential use and exchange with other systems. Systems may have components that are not exposed, and therefore should not be registered within GCI. There are two types of Components: DataComponents and ServiceComponents.
Contact	An individual registered within the GCI. Contacts are parts of Organizations and may have multiple roles (UserTypes).
Member	A special kind of Organization that has met the criteria to be a GEOSS participant.
Organization	A collective entity that has its own collective goals and purpose.
RegisteredResource	This is an abstract base class. RegisteredResources are published in Registries. They provide a set of metadata that supports discovery and understanding to those registries upon publication.
Registry	This is an abstract base class. A searchable container of RegisteredResources. Registries share a common metadata model, to which all of the items registered must adhere. Registries represent their contents (RegisteredResources) in an organized fashion, allowing for navigation by other clients and portals.
System	Systems are sets of elements, comprised and organized to work together to meet a specific need. These elements have responsibilities to interconnect with other elements within the system in a predefined way, in order to contribute to the system's goal. In GCI, those elements are represented as Components.
UserType	A role that a particular Contact has. A Contact may have multiple UserTypes, as they relate to different Organizations or Systems.

Data Package

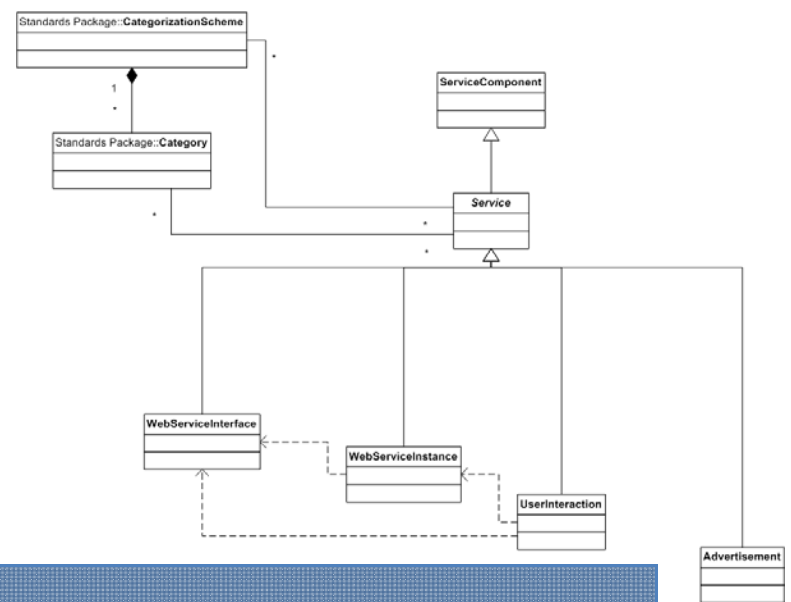
Entities related to Data contributions.



Element	Description
Browse	A type of metadata that provides a visual representation of a particular data object. This is intended to facilitate the quick understanding of whether or not the data may be of value to the user. There can be many types of browse, and a data object can have multiple browse items.
DataComponent	An abstract base class. DataComponents represent the informational entities available from systems. There are two kinds of DataComponents: DataSets and Granules.
DataSet	DataSets are types of data resources. These are defined by a common set of metadata which describe the common attributes of either all the items (Granules) that comprise the DataSet, or the attributes of the set in its entirety (e.g. – The first observation date and the last observation data of a mission that is no longer active.).
Granule	A specific data object. Granules belong to DataSets. This is the atomic level of data, as it is represented in a data system or Inventory. Granules may be further de-composed, but those further decompositions are not represented in an Inventory.
Inventory	This is an abstract base class. RegisteredResources are published in Registries. They provide a set of metadata that supports discovery and understanding to those registries upon publication.
Metadata	Information that, while data itself, is really describing another entity. Metadata is most valuable for discovery and understanding of objects.
MetadataModel	A pre-defined way to organize related sets of metadata. MetadataModels define things such as: Fields, types, ranges, optional/mandatory, multiplicity.
SearchableMetadata	SearchableMetadata is a Metadata that has been identified as being an acceptable field for searching through a Metadata Inventory. Not all Metadata is searchable.

Service Package

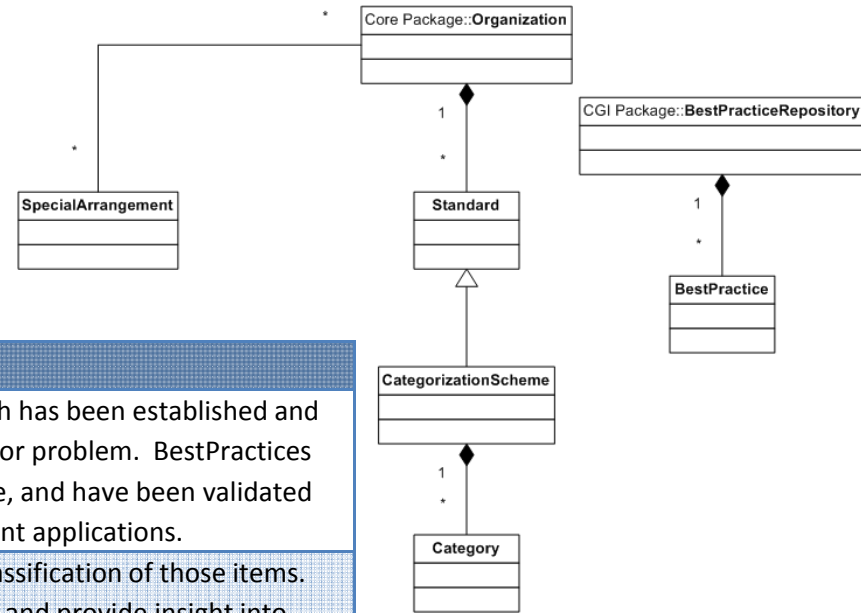
Entities related to service/functional contributions.



Element	Description
Advertisement	A representation of a service that does not directly interface with software systems through interfaces. Examples of Advertisements represent an organizations expert ability to utilize MODIS Ocean data, or the ability to deliver sensors in the Mozambique Channel.
Service	A functional element of a component. Services define mechanisms by which consumers can access their capabilities, primarily through interfaces. There are four types of Services: WebServiceInterface, WebServiceInstance, UserInteraction and Advertisement.
ServiceComponent	An abstract base class. Service Components represent all the different types of Services, as they are integrated within, and discovered through ServiceRegistries.
UserInteraction	This is a type of Service that is consumed by humans. UserInteraction services provide a computer based interface (normally a graphical user interface), that allow humans to interact with other services. Userinteraction services may be dedicated to specific WebServiceInstances, or may allow the flexibility to interact with any WebServiceInstance that meets a WebServiceInterface.
WebServiceInstance	A specific location that supports an identified WebServiceInterface. In its essence, this is an address (i.e – IP address and port) of a machine that listens, and responds to, messages with an IT protocol, according to the format as defined in a WebServiceInterface.
WebServiceInterface	A WebServiceInterface defines of the “contract” by which a service provider commits to deliver the functionality of a service. The WebServiceInterface defines a service, in terms of its operations, parameters, types, exceptions, as well as some non-functional attributes. A WebServiceInterface is “owned” by one organization, but many WebServiceInstances that meet a WebServiceInterface may be deployed by many organizations.






Standards Package

Agreements and standards within the GEOSS community.



Element	Description
BestPractice	A documented approach, process, or methodology which has been established and proven to be an effective way of addressing a challenge or problem. BestPractices are authored by experts in the particular field of practice, and have been validated as providing value or added efficiency in multiple different applications.
CategorizationScheme	A way of organizing items in a way that facilitates the classification of those items. CategorizationSchemes enable more efficient discovery, and provide insight into understanding the relevance of an item. In general, CategorizationSchemes are orthogonal to one another. Items can be classified in multiple CategorizationSchemes. CategorizationSchemes do evolve over time allowing new Categories to be added. In addition new CategorizationSchemes can be identified providing new ways of organizing items. Examples of CategorizationSchemes might be Color, and Manufacturer. Each automobile might be classified in both CategorizationSchemes.
Category	A particular element of classification within a CategorizationScheme. Items may “fit into” a category that is defined by a Categorization Scheme. An example might be an automobile is “Yellow” within the Color CategorizationScheme, and classified as “SEAT” in the manufacture CategorizationScheme. Items may be classified in multiple Categories within a Categorization Scheme. For example, a two-tone car might be both “Yellow” and “Black”.
SpecialArrangement	SpecialArrangements are agreements between organizations to collaborate in a defined manner. These are not as “formal” as standards, but are still shared between Members and are worthy of capture, documentation and publication.
Standard	An established and agreed upon method, approach or technical specification that has been adopted by a formal standards organization.

Symbol Definition

Element	Symbol	Description
Package		A logical construct for grouping related entities. This is a way of decomposing a domain to make it more digestible and understandable by people.
Class		An entity that has a defined role within the domain. Classes belong to Packages and have relationships with other Classes. Classes support encapsulation by having attributes (state) and operations (behavior). Class symbols are named (sometimes preceded by its Package name), and have two sections below the name. The first is for the list of attributes. The second is for the list of operations.
DataType		Similar to a Class, but with no encapsulated behavior.
Multiplicity	*	Indicates how many times a relationship may exist. Examples are ordinal (i.e. – a specific number), 0..1 (zero or one), * (any number). Displayed at the ends of association lines between classes or datatypes.
Generalization/ Specialization		A mechanism to identify types and sub-types. The entity at the top of the triangle is a generalized type, and those items “below” the triangle are subtypes (or specializations) of that type.
Contains		A mechanism reflecting composition. One entity is composed of, or contains, some number of other entities. For example, a country has one capital. An automobile has four tires.