GFD-R-P.110
Category: Recommendation
INFOD-WG

Stephen Davey, National e-Science Centre
Vijay Dialani, IBM Corporation
Ronny Fehling, Oracle Corporation
Steve Fisher, Rutherford Appleton Laboratory
Dieter Gawlick, Oracle Corporation
Christopher Kantarjiev, Oracle Corporation
Cecile Madsen, IBM Corporation
Susan Malaika, IBM Corporation
Shailendra Mishra, Oracle Corporation
Mallikarjun Shankar, Oak Ridge National Laboratory
Created: 2 Oct 2003
Revised Date: 5 May 2007
Published 3 July, 2007

Information Dissemination in the Grid Environment – Base Specifications

Status of This Memo

This memo provides a recommendation to the Grid communities. The intention is to define a standard. Distribution is unlimited.

Copyright Notice

Abstract

INFOD (Information Dissemination) provides a general means to determine which messages are to be sent from which publishers to which consumers based upon information kept in a registry. To support this, INFOD specifies interfaces that allow the characterization (in the registry) of publishers, consumers and various other components using vocabularies that are meaningful to members of the communities they belong to. INFOD makes use of a notify operation similar to that defined by the WS-Notification specification to send information between publishers and consumers.

INFOD also extends the publish/subscribe paradigm by allowing consumers to be determined dynamically based on the message content. Additionally, INFOD allows subscribers to determine what defines an event and which messages should be created in response to these events.

Table of Contents

1 Introduction

Having the most up-to-date information available is becoming increasingly important. Rick Hayes-Roth uses the term VIRT (Valuable Information at the Right Time) to capture this requirement[1, 2]. The core idea of VIRT is that consumers of information should receive the information that is relevant to them as soon as it is available or whenever it is needed. COI (Conditions Of Interest) determine which information is needed when and by whom. Information Dissemination[3] (INFOD) provides core technology to support the VIRT objectives for a wide range of applications.

Technology to support basic aspects of VIRT has been well established; JMS, the Java Messaging System, is a good example. JMS supports publishers as information providers and consumers as information recipients. The selection of the information is driven through subscriptions, which represent the COI.

This basic model has been extended by INFOD with:

- **Subscribers**: Subscriptions are typically specified by consumers. By assigning this task to subscribers they can determine a fitting subset of potential consumers based on some properties associated with them; e.g., notify the two security agents closest to an incident. Furthermore, consumers can get information that they did not subscribe to; a chemical spill ahead of me (the consumer) is an example.
- **Data Sources**: Publishers may be able to publish a wide variety of information. This information is organized as data sources. Examples of data sources are queues, (RSS) streams, files, (temporal) databases and applications. The structure, and to some extent the meaning, of the information of each data source is defined by one or more data vocabularies.
- **Data Vocabularies**: Data vocabularies are used to define the structure of information independent of the publishers and the data sources. Data vocabularies can be specified using SQL, XML, RDF or any other method as long as this method supports at least one query/filter language.
- **Property Vocabularies**: Property vocabularies are used to specify XML schemas that can be used to describe a class of publishers, consumers, subscribers and data sources in a way that is meaningful to a community that intends to share information. For example: all the publishers of the car dealer and the consumers of car buyer communities share property vocabularies.
- **Property Vocabulary Instances**: Property vocabulary instances are used to describe specific publishers, consumers, subscribers and data sources; e.g., a publisher who is characterized as a car dealer and further described by its location, its business rating and any other information that may be of interest.

---

[1] Model Based Communication Network and VIRT: Orders of magnitude better for Information Superiority (http://www.w2cog.org/revamp/files/MICOM2006-RHR-VIRT-final-1751.pdf?PHPSESSID=7f4fae6171ea8ef9204bcb000a6d2b67)

[2] Event Processing in the Global Information Grid (GIG) (http://complexevents.com/wp-content/uploads/2007/01/Event_Processing_GIG_RHR.ppt)

[3] The INFOD Base Use Case Scenarios (see http://forge.gridforum.org/sf/go/doc13626?nav=1) provide helpful background information. It describes INFOD patterns and their implementation as well as INFOD Use Cases

Using this extended model, the effort to establish and maintain the desired information flow, i.e., the effort to define and maintain subscriptions, can be significantly reduced. Without the extended model a subscriber needs to determine explicitly which publishers and data sources are of interest. With the extended model subscribers specify the type of information of interest along with the required properties of the publishers and data sources; e.g., the subscriber wants to make sure that the information from all the sensors providing a certain type of data and owned by a well accepted organization are sent to all agencies of a certain type located within 30 miles of each sensor. If sensors (or agencies) are added, relocated or removed the information flow must be adjusted.

INFOD captures information about publishers, consumers, subscriptions, subscribers, data sources, data vocabularies and property vocabularies in a registry, called the INFOD registry. The information in this registry is organized as resources. The main objective of the INFOD registry is to match publishers and consumers and to notify publishers which information has to be delivered to which consumer.

Some of the resources in the INFOD registry capture information about objects that exist outside of INFOD; e.g., a publisher and consumer are typically web services. Resources that capture information about an external object are called entries. Figure 1 shows the INFOD resources.

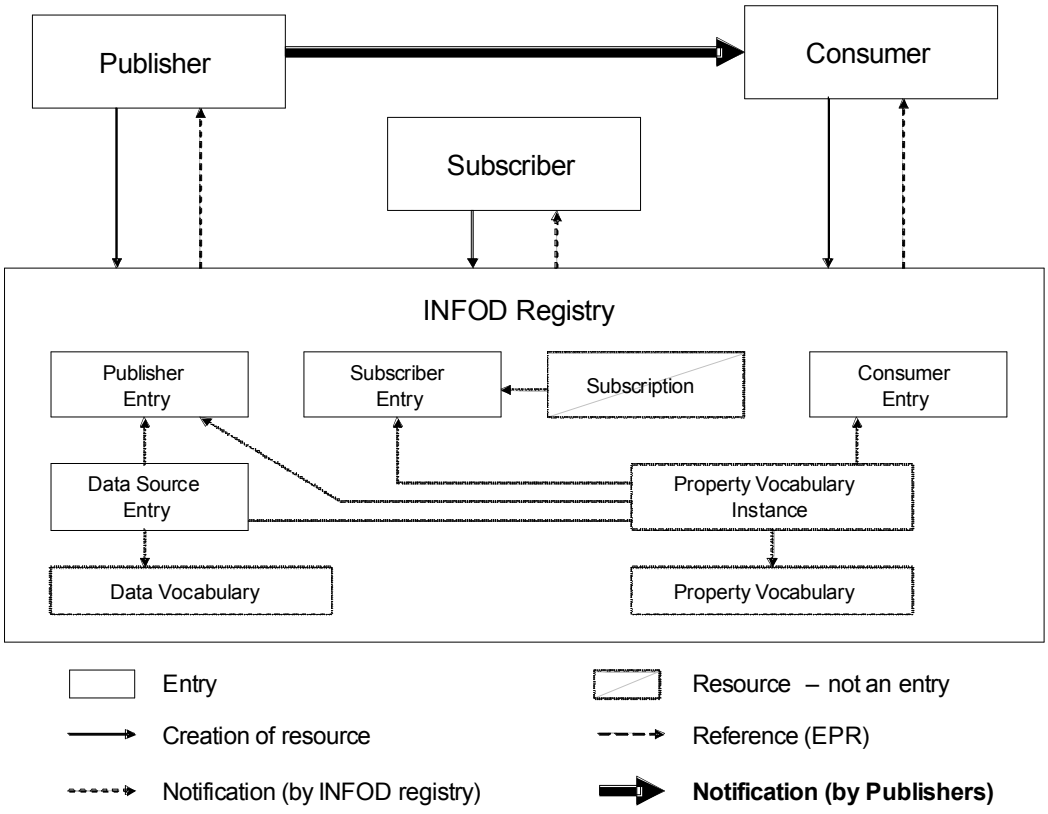

Figure 1: INFOD Resources

The registry is used to manage the information that is required to determine which information (messages) has to flow from which publishers to which consumers. The messages flow directly from the publishers to the consumers making use of a notification system similar to WS-Notification.

Here is a list of the contributions of the INFOD model:

- **Property Constraints and Mutual Filtering**: Each entry can specify a set of property constraints referencing information related to other entries; a property constraint is an XQuery or XPath expression referencing entries and property vocabulary instances. For example, a car buyer can specify an interest in information from car dealers within 30 miles and having an exceptional business rating. A car dealer could specify that the dealership is only interacting with customers with a high credit rating. This mutual filtering ensures that the buyer will not get information from a dealer too far away and that the car dealer does not contact buyers without the proper financial status.

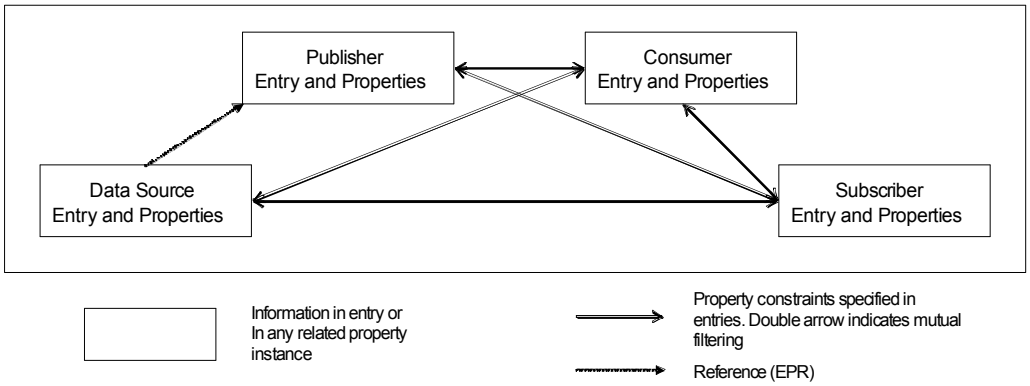

Figure 2: Property Constraints

Property constraints are used to specify which other entries are eligible to interact with a given entry. Examples of interactions are sending or receiving a message or reacting to a subscription. Property constraints can reference properties of other entries as well as properties captured in property vocabulary instances. Figure 2 shows all property constraints that can be specified between entries. The absence of constraints shows that the interaction is unrestricted.

- **Property constraints (in Subscriptions)**: Property constraints can be used in subscriptions to define publishers and consumers instead of identifying publisher and consumers explicitly. The INFOD registry will determine which publishers and consumers conform to the constraints. Any limitation imposed through mutual filtering will be taken into account. This support simplifies the task of the subscribers of matching publishers with subscribers. The INFOD registry also adapts the information flow to changes of resources; e.g., the INFOD registry will react to new, modified or deleted publisher entries as soon as they become available. A subscriber would not be able to achieve this.

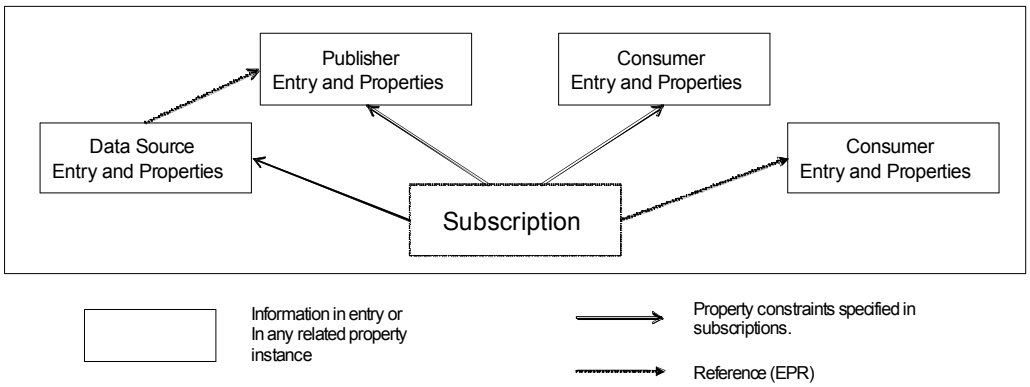

Figure 3: Property Constraints in Subscriptions

Figure 3 shows all property constraints that can be specified by subscriptions. EPRs can be used to identify entries explicitly. The absence of property constraints shows that there is no limitation in the selection of publishers, data sources and consumers.

- **Data Constraints (in Subscription)**: A data constraint is a query supported by a data vocabulary. Data constraints are used to specify which information is of interest. To make the information *as valuable as possible* subscribers can specify what an event is by defining conditions or patterns on (temporal) data sources. Additionally, subscribers can specify which information or message should be disseminated in response to an event. Data constraints can only be specified for subscriptions.

  An example may illustrate this. A banking customer may be interested in a visualization of the development of his/her portfolio over the last 12 months when the moving 4-week average of one of the stocks changes by twice as fast as the Dow Jones Industrial Index.

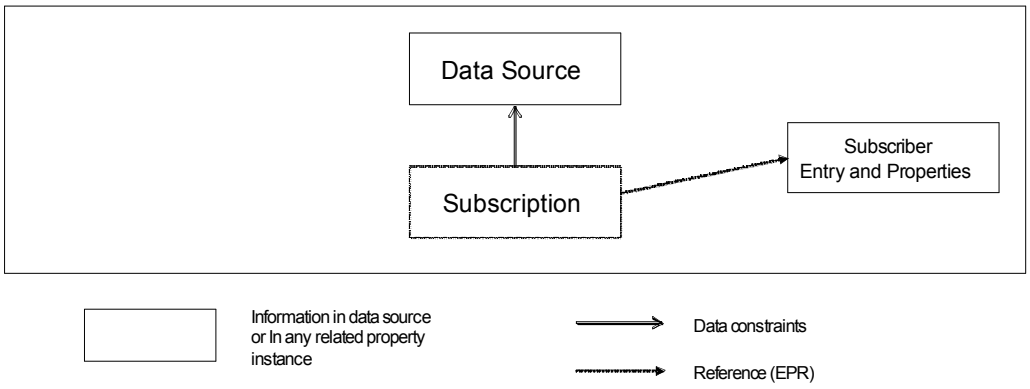

Figure 4: Data Constraints in Subscriptions

Figure 4 shows those data constraints which data sources eligible. The absence of a data constraint indicates interest in all data.

Property and data constraints in subscriptions represent the COI in VIRT. Data and property constraints specified in subscription are complemented by property constraints specified in entries.

1.1 The Registry

1.1.1 Resources

The registry manages various resources as listed below. A resource is used here as meaning something which is held in the registry. Each resource type has calls to create and drop it from the registry. Some resources have a call to replace them.

1.1.1.1 Publisher Entry, Consumer Entry and Subscriber Entry

As already explained, an entry is the information stored in the registry about an external object. Each is identified in the registry by a unique EPR (endpoint reference). Operations are provided to create, replace and drop these entries. Note that these verbs are with respect to the entries in the registry and not the external object, so we talk about creating a publisher entry rather than registering a publisher. The act of creation involves storing information and returning the EPR of the entry. The creation operation will often store the EPR of the external object. This is the only place the external EPR, identifying the external object, is stored. All other references to EPRs are to EPRs of resources.

Each entry has a name and description, both of which are optional, not necessarily unique and have string values. They are also both expected to be meaningful to humans.

The replace operation (for example ReplacePublisher in Section 2.1.2) takes the EPR that was returned by the create operation as an additional parameter and keeps only the identity of the entry: all the data associated with it by the create operation is replaced by new data however all relations established after the original entry was created are preserved as the identity of the entry remains unchanged. The drop operation (for example DropPublisher in Section 2.1.3) takes the EPR of the entry and makes the stored entry unavailable and so makes subsequent use of the EPR invalid. The drop operation is not allowed to make the system inconsistent (see Section 1.1.2) so, by default, an error will be reported if an attempt is made to drop an entry which is still referenced. There is an optional flag which can be set to “DISABLE NEW REFERENCES” which results in the entry being dropped when the last reference to the entry has been removed and “CASCADE”, which also drops (recursively) all entries referencing that entry.

1.1.1.2 Data Vocabularies and Data Source Entries

Data are only useful if there is a shared understanding of these data by publishers, consumers and subscribers. For this purpose INFOD uses vocabularies, which are maintained within the registry. Data vocabularies describe the structure of the data that is available from publishers. It is the responsibility of a community of users with a common interest to define a data vocabulary and register it as the first step in using INFOD. For flexibility, data vocabularies can be specified using SQL, XML, RDF or any other data model. The INFOD registry will *not* manage instances of user data. A data vocabulary is used by the registry to carry out vocabulary specific operations. Vocabularies are managed, with operations such as CreateDataVocabulary (Section 2.5.5) to store information about the data vocabulary in the registry.

A data source entry is created by an operation called CreateDataSourceEntry. This represents an association between data vocabularies and entries – specifically publisher entries thereby identifying the publisher as a source of some specific type of information.

Data Source Entries, like other entries have their own EPR and an optional name and description. In addition they have the EPR of the two things they are relating.

1.1.1.3 Property Vocabularies and Property Vocabulary Instances

A user community may also define property vocabularies to allow property constraints to be defined. For example a business community may decide that consumers should have a postal address. This mechanism allows this postal address to be precisely defined. These vocabularies, which are optional, are expressed by an XML schema.

The CreatePropertyVocabularyInstance call (Section 2.5.2) is then used to create a The Property Vocabulary Instance which holds actual values for a particular Publisher, Consumer or Subscriber entry. The Property Vocabulary Instance references a Property Vocabulary.

Constraints identifying which other resources are of interest or unacceptable may be expressed using these properties. For example a publisher may choose to only send messages to consumers whose address matches some pattern.

Property vocabularies can be used as an extension mechanism to define notions such as quality of service. In a future version of the document this extension mechanism may be used to formalize properties such as operational characteristics.

1.1.1.4 Subscriptions and Constraints

No information starts flowing in an INFOD system until a subscription is created. A subscription normally defines various constraints. In the absence of all constraints a subscription will cause all messages to be sent from all publishers to all consumers. In practice producers have constraints to indicate who they will send messages to, consumers have constraints to say who they will get messages from and a subscription will normally have at least a data constraint indicating what kind of messages are wanted. The registry acts on subscriptions by finding matching publishers and consumers using the property constraints of publisher entries, consumer entries, subscriber entries, subscriptions and data source entries along with data constraints of subscriptions expressed in terms of a data vocabulary.

In addition a subscription may include dynamic consumer constraints. These are constraints which are evaluated by the consumer rather than the registry by looking at the contents of a potential message.

As already mentioned the subscription is not an entry as it has no counterpart outside the registry. The create operation returns an EPR.

1.1.2 Dependencies

A basic dependency rule governs the creation, modification and removal of resources within the INFOD registry: only resources that are registered in the INFOD registry can be referenced.

Figure 5 shows the relations between the various INFOD resources. The arrows show the direction of reference. In addition the P or D in the box shows which resources may hold property or data constraints respectively.

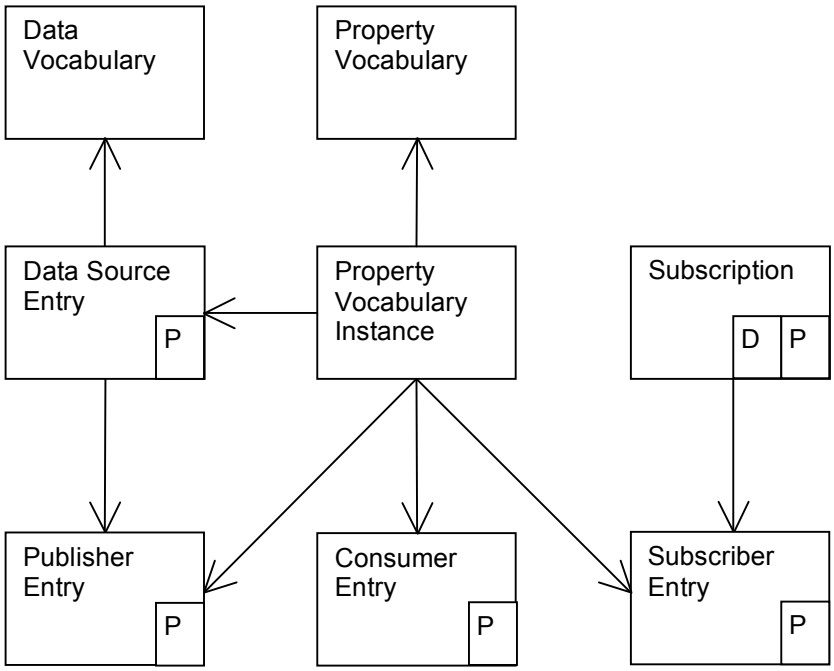

Figure 5: Relation between INFOD resources

1.1.3 Matching Publishers to Subscriptions

Discovery of publishers to match a specific subscription requires the registry to examine the vocabularies and all the constraints so that it can generate correct notifications. Instead of using notifications the GetMetaData operation (see section 2.7) may be used to query the information in the INFOD registry and, most importantly, to look up matching subscriptions and publishers.

INFOD objects, especially publishers need to react immediately to changes in the INFO registry. They may register to be notified of any changes that are significant for them.

1.2 Security

INFOD uses existing security mechanisms to ensure that the dissemination of information happens according to security policies. The specification of communities can be used to complement and enhance security policies.

1.3 Lifetime Management

The INFOD specification does not contain any specific resource lifetime management other than the facilities to remove INFOD resources, for example *DropSubscription* etc. However, to ensure that in cases where a client becomes disconnected from the INFOD Registry and is unable or unwilling to destroy obsolete INFOD resources, some form of lifetime management should be employed such as WS-ResourceLifetime (see http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-03.pdf). This should provide a mechanism by which resources may be destroyed after a period of time unless the scheduled termination time is extended.

1.4 Summary of key aspects of INFOD

The INFOD base specification may be summarized:

- Publishers should be able to describe their available messages, events and states in terms of a data vocabulary
- Subscribers must be able to constrain messages based on message content and publisher and consumer information.
- Publishers must be able to choose what messages to publish based on consumer and subscription information.
- Consumers must be able to constrain messages based on message content, publisher information and subscription information.
- Any service can request that it be notified by the registry of changes that it considers relevant.
- The INFOD registry can apply constraints simultaneously.

1.5 Glossary

Constraint	Contraints are used to specify which conditions must be safisfied to be eligible for an interaction. Constraints must be formulated in the constraint language(s) that are associated to the vocabularies, which are used to structure the referenced data. Most constraints are evaluated by the registry but dynamic consumer constraints are dealt with by the consumer. The absence of constraints shows that the interaction is unrestricted.
Constraint Language	The grammar of the constraints specification associated to a type system.
Entry	An entry is the information about an external object that is stored in the registry. There of four types of entry: publisher, consumer, subscriber and data source.
Data Vocabulary	A data vocabulary defines the structure of the data associated to a data source. Data vocabularies can be specified using any type system.
Consumer	A consumer is able to receive messages delivered by publishers. Property vocabularies can be used to extend the description of consumers; consumers can limit the flow of messages by defining constraints.
Consumer Entry	Information about a consumer stored in a registry

EPR	An EPR (Endpoint Reference) is an XML structure encapsulating information useful for addressing a message to a Web service. This includes the destination address of the message, any additional parameters (called reference parameters) necessary to route the message to the destination, and optional metadata about the service.
Event	An event is a view at a state transition specified by a publisher or a subscriber. Publishers may allow subscribers to reference events (those defined by publishers) to create messages or to define events by referencing state transitions. In many cases, publishers do not provide access to events but allow only access to (and selection of) messages. In this case the state and event definitions are hidden to subscribers.
Message	A message is used to deliver data from publishers to consumers. A message normally contains information about an event that is observed by a publisher.
Property Vocabulary	A property vocabulary specifies the structure of properties associated to entries.
Property Vocabulary Instance	A property vocabulary instance represents the (values of) properties that are associated to specific entries. A property vocabulary instance has to be structured according to a property vocabulary.
Publisher	A publisher is able to create and deliver data in the form of messages to consumers. Property vocabularies can be used to extend the description of publishers; publishers can limit subscriptions requests by defining constraints. Publishers may create and deliver messages unconditionally or make the delivery of messages dependent on subscriptions. Publisher may allow subscribers (using subscriptions) to specify which messages should be created in response to which events; events maybe pre-defined or based on (subscriptions) specifications referencing state changes.
Publisher Entry	Information about a publisher that is stored in a registry as a resource
Registry	A repository of INFOD resources able to deliver notifications
Resource	A resource is an elementary object in the registry that may be created, replaced or dropped.
Subscriber	A subscriber specifies subscriptions. Subscriptions are the primary means of specifying the message flow from publishers to consumers.
Subscriber Entry	Information about a subscriber that is stored in a registry as a resource.
Subscription	A subscription defines which information has to be delivered by which publishers to which consumers. The information is selected by constraint specifications; publishers and consumers are identified through explicit references (EPR's) or constraints on property vocabularies.

Type System	A type system is an enumeration that defines the list of acceptable value domains, their value ranges and binary representation in a digital system.
Vocabulary	A vocabulary defines the structure of data in the context of a type system; e.g., a schema in the context of XML. Vocabularies are used to facilitate a common understanding of data between publishers, consumers and subscribers.
WSN	Web Service Notification is a pattern-based approach to allow Web services to disseminate information to one another
Data Source entry	A data source entry specifies that data structured with the referenced vocabulary (and constraint language) is offered by the referenced publisher.

1.6 Terminology

Except in this introductory chapter, the keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [IETF RFC 2119].

When describing abstract data models, this specification uses the notational convention used by the “XML Information Set” (see http://www.w3.org/TR/xml-infoset/). Specifically, abstract property names always appear in square brackets (e.g., [some property]).

This specification uses a notational convention, referred to as “Pseudo-schemas”. A Pseudo-schema uses a BNF-style convention to describe attributes and elements:

- `?′ denotes optionality (i.e. zero or one occurrences),
- `*′ denotes zero or more occurrences,
- `+′ one or more occurrences,
- `[′ and `]′ are used to form groups,
- `|′ represents choice.

- Attributes are conventionally assigned a value which corresponds to their type, as defined in the normative schema.
- Elements with simple content are conventionally assigned a value which corresponds to the type of their content, as defined in the normative schema.
- The use of {any} indicates the presence of an element wildcard (<xs:any/>).
- The use of @{any} indicates the presence of an attribute wildcard (<xs:anyAttribute/>).
- In the interest of brevity, some extensibility points have been omitted from the Pseudo-schemas.

```
<!-sample pseudo-schema -->
<element
      required_attribute_of_type_QName="xs:QName"
      optional_attribute_of_type_string="xs:string"? >
  <required_element />
  <optional_element /> ?
  <one_or_more_of_these_elements /> +
  [ <choice_1 /> | <choice_2 /> ] *
</element>
```

Where there is disagreement between the separate XML schema and WSDL files describing the messages defined by this specification and the normative descriptive text (excluding any pseudo-schema) in this document, the normative descriptive text will take precedence over the separate files. The separate files take precedence over any pseudo-schema and over any schema and WSDL included in the appendices.

1.7 Namespaces

The following namespaces are used in this document:

Prefix	Namespace		Meaning
S	http://schemas.xmlsoap.org/soap/envelope/ http://www.w3.org/2003/05/soap-envelope	OR	SOAP Envelope
xsd	http://www.w3.org/2001/XMLSchema		XML Schema
wsa	http://www.w3.org/2005/08/addressing		WS-Addressing
wsrf-bf	http://docs.oasis-open.org/wsrf/bf-2		WS Base Faults
wsnt	http://docs.oasis-open.org/wsn/b-2		WS-Base Notification
wsntw	http://docs.oasis-open.org/wsn/bw-2		WS-Base Notification
wstop	http://docs.oasis-open.org/wsn/t-1		WS-Topics
infod	http://www.ogf.org/infod		INFOD

The INFOD name space is divided into two subcomponents (INFODRegistry and INFODNotify)

1.8 Fault Definitions

All faults generated by a NotificationProducer or SubscriptionManager should be compliant with the WS-BaseFaults (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf) specification.

All faults defined by this specification MUST use the following URI for the WS-Addressing [action]:

http://www.ogf.org/infod/fault.

2 The Base INFOD Registry Interface

The tables below list the operations of the base INFOD registry interface and the section that describes them in detail.

The Base INFOD Registry Interface:

Operation		Description	Section
Managing Publisher Entries	CreatePublisherEntry	This operation defines how to create a new Publisher entry in an INFOD registry.	2.1.1
	ReplacePublisherEntry	This operation defines how to replace a particular Publisher entry in an INFOD registry.	2.1.2
	DropPublisherEntry	This operation defines how to drop an existing Publisher entry from an INFOD registry.	2.1.3
Managing Subscriber Entries	CreateSubscriberEntry	This operation defines how to create a new Subscriber entry in an INFOD registry.	2.2.1
	ReplaceSubscriberEntry	This operation defines how to replace a particular Subscriber entry in an INFOD registry.	2.2.2
	DropSubscriberEntry	This operation defines how to drop an existing Subscriber entry from an INFOD registry.	2.2.3
Managing Consumer Entries	CreateConsumerEntry	This operation defines how to create a new Consumer entry in an INFOD registry.	2.3.1
	ReplaceConsumerEntry	This operation defines how to replace a particular Consumer entry in an INFOD registry.	2.3.2
	DropConsumerEntry	This operation defines how to drop an existing Consumer entry from an INFOD registry.	2.3.3
Managing Subscriptions	CreateSubscription	This operation defines how to create a new Subscription in an INFOD registry.	2.4.1
	ReplaceSubscription	This operation defines how to replace a particular Subscription in an INFOD registry.	2.4.2
	DropSubscription	This operation defines how to drop an existing Subscription from an INFOD registry.	2.4.3
Man agin	CreatePropertyVocabulary	This operation defines how to create a property vocabulary to an INFOD registry	2.5.1

	DropPropertyVocabulary	This operation defines how to drop a property vocabulary from an INFOD registry.	2.5.2
	CreatePropertyVocabularyInstance	This operation creates a new instance of a property vocabulary that is already registered in an INFOD registry.	2.5.3
	DropPropertyVocabularyInstance	This operation drops an existing instance of a particular property vocabulary registered in an INFOD registry.	2.5.4
	CreateDataVocabulary	This operation defines how to create a data vocabulary to an INFOD registry	2.5.5
	DropDataVocabulary	This operation defines how to drop a data vocabulary from an INFOD registry.	2.5.6
Managing Data Sources	CreateDataSource	This operation defines how to create a data source in an INFOD registry.	2.6.1
	DropDataSource	This operation defines how to drop a data source from a particular Publisher entry in an INFOD registry.	2.6.2
	GetMetaData	This operation queries the metadata of resources defined in a particular INFOD registry.	2.7

2.1 Managing Publisher Entries

These operations are used to manage publishers

- CreatePublisherEntry (section 2.1.1)
- ReplacePublisherEntry (section 2.1.2)
- DropPublisherEntry (section 2.1.3)

2.1.1 CreatePublisherEntry

As part of the processing of a CreatePublisherEntry request message, the INFOD registry MUST create an INFOD entry and an EPR representing the publisher entry.

The format of the request message for the CreatePublisherEntry operation is based on the schema provided in Appendix I – XML Schema definition for an INFOD entry. Details are as follows:

```
<infod:CreatePublisherEntry>
  <infod:WSReference>
    wsa:EndPointReferenceType
  </infod:WSReference> ?
  <infod:PublisherName> xsd:string </infod:PublisherName> ?
  <infod:PublisherDescription>
    xsd:string
  </infod:PublisherDescription> ?
  <infod:PropertyConstraint>
    xsd:any
  </infod:PropertyConstraint> *
  <infod:Notification>
```

```
    xsd:boolean default ”FALSE”
  </infod:Notification> ?
</infod:CreatePublisherEntry>
```

The elements of the CreatePublisherEntry message are further described as follows:

/infod:WSReference

An endpoint reference element, as defined by WS-Addressing, used to identify the WS endpoint for the entry. Note that this MAY be the WS EPR of the requesting service, but does not have to be.

/infod:PublisherName

A string representing the name of the publisher. This name MAY NOT be unique.

/infod:PublisherDescription

A string representing a description of the publisher.

/infod:PropertyConstraint

Property contraints are used to specify which conditions must be satisfied by other entries (consumers, data sources and subscribers) to be eligible for interaction with this publisher. A property constraint MUST be formulated as an XQuery. The INFOD Base Use Case Scenarios (see http://forge.gridforum.org/sf/go/doc13626?nav=1) provide examples of XQueries.

For example, a publisher identifies the set of consumers that are eligible to receive data by formulating property constraints.

Note that the XQuery statement MUST be encoded correctly, i.e. characters such as “>” would be represented as “>”

/infod:Notification

When used, the registry MUST notify the publisher about changes relevant in the registry. A fault MUST be returned if infod:WSReference is not specified.

For further details see section 3.2.1

A WS-Addressing Action header with the value
http://www.ogf.org/infod/INFODRegistry/CreatePublisherEntry MUST accompany the message.

INFOD Registry Response

If the INFOD registry accepts the CreatePublisherEntry message, it MUST respond to the WS endpoint specified in the request message with a CreatePublisherEntryResponse message. The CreatePublisherEntryResponse message is a message of the following form:

```
<infod:CreatePublisherEntryResponse>
    <infod:PublisherEntryReference>
       wsa:EndPointReferenceType
    </infod:PublisherEntryReference>
</infod:CreatePublisherEntryResponse>
```

The elements of the CreatePublisherEntryResponse message are further described as follows:

/infod:PublisherEntryReference

An endpoint reference element, as defined by WS-Addressing, used to identify the newly created publisher entry in the INFOD registry.

One of the following faults MUST be sent if the operation fails:

- CreateResourceAuthorizationFault: User not authorized to create the INFOD resource at this INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- UnsupportedXQueryFault: The XQuery specified could not be parsed correctly

The message MUST be structured according to the WS-Base Faults specification. For examples using SOAP, see the SOAP v1.2 Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

Example SOAP Encoding of the Create Publisher Message

The following is a non-normative example of a CreatePublisherEntry request message using SOAP:

```
<s:Envelope ... >
  <s:Header>
    <wsa:Action>
      http://www.ogf.org/infod/INFODRegistry/CreatePublisherEntry
    </wsa:Action>
    ...
  </s:Header>
  <s:Body>
    <infod:CreatePublisherEntry>
      <infod:WSReference>
        <wsa:Address>
          http://www.example.org/SomePublisher
        </wsa:Address>
      </infod:WSReference>
      <infod:PublisherName>
        SomePublisher
      </infod:PublisherName>
      <infod:PublisherDescription>
        This publisher can publish some information
      </infod:PublisherDescription>
      <infod:PropertyConstraints>
       fn:doc("INFODRegistry.xml")/Consumers/infodConsumer
         [fn:contains(ConsumerName,"Ronny"}]
      </infod:PropertyConstraints>
      <infod:Notification>
        TRUE
      </infod:Notification>
    </infod:CreatePublisherEntry>
  </s:Body>
</s:Envelope>
```

2.1.2 ReplacePublisherEntry

The ReplacePublisherEntry operation replaces an INFOD publisher entry's metadata information at a given INFOD registry. As part of the processing of a ReplacePublisherEntry message, the INFOD registry MUST replace the entire INFOD metadata for the entry representing the publisher. All previously defined values MUST be deleted. The ReplacePublisherEntry differs from the CreatePublisherEntry interface in that it replaces an existing publisher entry and assigns the original EPR to the replaced publisher.

The format of the request message for a ReplacePublisherEntry operation is also based on the schema definition provided in Appendix I – XML Schema for an INFOD entry. Details are as follows:

```
<infod:ReplacePublisherEntry>
```

```
  <infod:WSReference>
    wsa:EndPointReferenceType
  </infod:WSReference> ?
  <infod:PublisherEntryReference>
    wsa:EndPointReferenceType
  </infod:PublisherEntryReference>
  <infod:PublisherName> xsd:string </infod:PublisherName> ?
  <infod:PublisherDescription>
    xsd:string
  </infod:PublisherDescription> ?
  <infod:PropertyConstraint>
    xsd:any
  </infod:PropertyConstraint> *
  <infod:Notification>
    xsd:boolean "FALSE"
  </infod:Notification> ?
</infod:ReplacePublisherEntry>
```

The elements of the ReplacePublisherEntry message are further described as follows:

/infod:WSReference

An endpoint reference element, as defined by WS-Addressing, used to identify the WS endpoint for the entry. Note that this MAY be the WS EPR of the requesting service, but does not have to be. The request MAY be made 'on behalf' of the actual service.

/infod:PublisherEntryReference

An endpoint reference element, as defined by WS-Addressing, used to identify the publisher entry in the INFOD registry that will be replaced.

/infod:PublisherName

A string representing the name of the publisher. This name MAY NOT be unique.

/infod:PublisherDescription

A string representing a description of the publisher.

/infod:PropertyConstraint

Property contraints are used to specify which conditions must be safisfied by other entries (consumers, data sources and subscribers) to be eligible for interaction with this publisher. A property constraint MUST be formulated as an XQuery. The INFOD Base Use Case Scenarios (see http://forge.gridforum.org/sf/go/doc13626?nav=1) provide examples of XQueries.

For example, a publisher identifies the set of consumers that are eligible to receive data by formulating property constraints.

Note that the XQuery statement MUST be encoded correctly, i.e. characters such as ">" would be represented as ">"

/infod:Notification

When used, the registry MUST notify the publisher about changes relevant in the registry. A fault MUST be returned if infod:WSReference is not specified.

For further details see section 3.2.1

A WS-Addressing Action header with the value
http://www.ogf.org/infod/INFODRegistry/ReplacePublisherEntry MUST accompany the message.

INFOD Registry Response

If the INFOD registry accepts the ReplacePublisherEntry message, it MUST respond to the WS endpoint specified in the request message with a ReplacePublisherEntryResponse message. The ReplacePublisherEntryResponse message is a message of the following form:

```
<infod:ReplacePublisherEntryResponse>
  <infod:Status>
 xsd:string default ”COMPLETED”
  </infod:Status>
</infod:ReplacePublisherEntryResponse>
```

The elements of the ReplacePublisherEntryResponse message are further described as follows:

/infod:Status

An indication that the request has been successfully executed.

One of the following faults MUST be sent if the operation fails:

- ReplaceResourceAuthorizationFault: User not authorized to replace the INFOD resource at this INFOD registry
- UnknownResourceReferenceFault: An resource has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- UnsupportedXQueryFault: The XQuery specified could not be parsed correctly

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.1.3 DropPublisherEntry

The DropPublisherEntry operation removes an INFOD publisher entry from an INFOD registry.

The format of the request message for a DropPublisherEntry operation is:

```
<infod:DropPublisherEntry>
  <infod:PublisherEntryReference>
    wsa:EndPointReferenceType
  </infod:PublisherEntryReference>
  <infod:ExecutionMode> xsd:string </infod:ExecutionMode> ?
</infod:DropPublisherEntry>
```

The elements of the DropPublisherEntry message are further described as follows:

/infod:PublisherEntryReference

An endpoint reference element, as defined by WS-Addressing, used to identify the INFOD resource in the registry to drop.

/infod:ExecutionMode

A parameter indicating the mode of execution of the drop request. Possible values are:

“IF UNUSED” The drop request will execute only if the resource is unreferenced

"DISABLE NEW" No new references are possible for the resource. The resource will be dropped when the last reference to this resource is gone

"CASCADE" The drop request will execute immediately and all references to the resource will be removed recursively

If this parameter is not specified, the default value "IF UNUSED" MUST be used.

A WS-Addressing Action header with the value http://www.ogf.org/infod/INFODRegistry/DropPublisherEntry MUST accompany the message.

INFOD Registry Response

If the INFOD registry accepts the DropPublisherEntry message, it MUST respond to the WS endpoint specified in the request message with a DropPublisherEntryResponse message. The DropPublisherEntryResponse message is a message of the following form:

```
<infod:DropPublisherEntryResponse>
  <infod:Status>
 xsd:string default "COMPLETED"
  </infod:Status>
</infod:DropPublisherEntryResponse>
```

The elements of the DropPublisherEntryResponse message are further described as follows:

/infod:Status

An indication that the request has been successfully executed.

One of the following faults MUST be sent if the operation fails:

- DropResourceAuthorizationFailure: User not authorized to drop the INFOD resource at this INFOD registry
- UnknownResourceReferenceFault: An resource has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- ExecutionModeFault: Cannot use ExecutionMode provided

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.2 Managing Subscriber Entries

The following operations are used to manage subscribers:

- CreateSubscriberEntry (section 2.2.1)
- ReplaceSubscriberEntry (section 2.2.2)
- DropSubscriberEntry (section 2.2.3)

2.2.1 CreateSubscriberEntry

As part of the processing of a CreateSubscriberEntry request message, the INFOD registry MUST create an INFOD entry representing the subscriber.

The format of the request message for CreateSubscriberEntry operation is based on the schema provided in Appendix I – XML Schema for an INFOD entry. Details are as follows:

```
<infod:CreateSubscriberEntry>
  <infod:WSReference>
    wsa:EndPointReferenceType
  </infod:WSReference> ?
  <infod:SubscriberName> xsd:string </infod:SubscriberName> ?
  <infod:SubscriberDescription>
    xsd:string
  </infod:SubscriberDescription> ?
  <infod:PropertyConstraint>
    xsd:any
</infod:PropertyConstraint> *
<infod:Notification>
  xsd:Boolean default "FALSE"
</infod:Notification> ?
</infod:CreateSubscriber>
```

The elements of the CreateSubscriberEntry message are further described as follows:

/infod:WSReference

An endpoint reference element, as defined by WS-Addressing, used to identify the WS endpoint for the entry. Note that this MAY be the WS EPR of the requesting service, but does not have to be. The request MAY be made 'on behalf' of the actual service.

/infod:SubscriberName

A string representing the name of the subscriber name, this name MAY NOT be unique.

/infod:SubscriberDescription

A string representing a description of the subscriber.

/infod:PropertyConstraint

Property contraints are used to specify which conditions must be safisfied by other entries (publishers, data sources, and consumers) to be eligible for interaction with this publisher. A property constraint MUST be formulated as an XQuery. The INFOD Base Use Case Scenarios (see http://forge.gridforum.org/sf/go/doc13626?nav=1) provide examples of XQueries.

For example, a subscriber identifies the set of publishers that are eligible to react to subscriptions specified by this subscriber.

Note that the XQuery statement MUST be encoded correctly, i.e. characters such as ">" would be represented as ">"

infod:Notification

When used, the registry MUST notify the subscriber about relevant changes in the INFOD registry. A fault MUST be returned if infod:WSReference is not specified.

For further details see section 3.2.2.

A WS-Addressing Action header with the value
http://www.ogf.org/infod/INFODRegistry/CreateSubscriberEntry MUST accompany the message

INFOD Registry Response

If the INFOD registry accepts the CreateSubscriberEntry message, it MUST respond to the WS endpoint specified in the request message with a CreateSubscriberEntryResponse message. The CreateSubscriberEntry response message is a message of the following form:

```
<infod:CreateSubscriberEntryResponse>
    <infod:SubscriberEntityReference>
       wsa:EndPointReferenceType
    </infod:SubscriberEntityReference>
</infod:CreateSubscriberEntryResponse>
```

The elements of the CreateSubscriberEntryResponse message are further described as follows:

/infod:SubscriberEntityReference

An endpoint reference element, as defined by WS-Addressing, used to identify the newly created subscriber entry in the INFOD registry.

One of the following faults MUST be sent if the operation fails:

- CreateResourceAuthorizationFault: User not authorized to create the INFOD resource at this INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- UnsupportedXQueryFault: The XQuery specified could not be parsed correctly

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.2.2 ReplaceSubscriberEntry

The ReplaceSubscriberEntry operation replaces an INFOD subscriber entry's metadata information at a given INFOD registry. As part of the processing of a ReplaceSubscriberEntry request message, the INFOD Registry MUST replace the entire INFOD metadata for the entry representing the subscriber. All previously defined values MUST be deleted. The ReplaceSubscriberEntry differs from the CreateSubscriberEntry interface in that it replaces an existing subscriber entry and assigns the original EPR to the replaced subscriber.

The format of the request message for a ReplaceSubscriberEntry operation is also based on the schema definition provided in Appendix I – XML Schema for an INFOD entry. Details are as follows:

```
<infod:ReplaceSubscriberEntryEntry>
  <infod:WSReference>
    wsa:EndPointReferenceType
  </infod:WSReference> ?
  <infod:SubscriberEntryReference>
    wsa:EndPointReferenceType
  </infod:SubscriberEntryReference>
  <infod:SubscriberName> xsd:string </infod:SubscriberName> ?
  <infod:SubscriberDescription>
    xsd:string
  </infod:SubscriberDescription> ?
  <infod:PropertyConstraint>
    xsd:any
</infod:PropertyConstraint> *
 <infod:Notification>
  xsd:Boolean default "FALSE"
</infod:Notification> ?
</infod:ReplaceSubscriberEntry>
```

The elements of the ReplaceSubscriberEntry message are further described as follows:

/infod:WSReference

An endpoint reference element, as defined by WS-Addressing, used to identify the WS endpoint for the entry. Note that this MAY be the WS EPR of the requesting service, but does not have to be. The request MAY be made 'on behalf' of the actual service.

/infod:SubscriberEntryReference

An endpoint reference element, as defined by WS-Addressing, used to identify the subscriber entry in the INFOD registry that will be replaced.

/infod:SubscriberName

A string representing the name of the subscriber. This name MAY NOT be unique.

/infod:SubscriberDescription

A string representing a description of the subscriber.

/infod:PropertyConstraint

Property contraints are used to specify which conditions must be safisfied by other entries (publishers, data sources, and consumers) to be eligible for interaction with this subscriber. A property constraint MUST be formulated as an XQuery. The INFOD Base Use Case Scenarios (see http://forge.gridforum.org/sf/go/doc13626?nav=1) provide examples of XQueries.

For example, a subscriber identifies the set of publishers that are eligible to react to subscriptions specified by this subscriber.

Note that the XQuery statement MUST be encoded correctly, i.e. characters such as ">" would be represented as ">"

infod:Notification

When used, the registry MUST notify the subscriber about relevant changes in the INFOD registry. A fault MUST be returned if infod:WSReference is not specified.

For further details see section 3.2.2.

A WS-Addressing Action header with the value
http://www.ogf.org/infod/INFODRegistry/ReplaceSubscriberEntry MUST accompany the message

INFOD Registry Response

If the INFOD registry accepts the ReplaceSubscriberEntry message, it MUST respond to the WS endpoint specified in the request message with a ReplaceSubscriberEntryResponse message. The ReplaceEntrySubscriber response message is a message of the following form:

```
<infod:ReplaceSubscriberEntryResponse>
  <infod:Status>
 xsd:string default "COMPLETED"
  </infod:Status>
</infod:ReplaceSubscriberEntryResponse>
```

The elements of the ReplaceSubscriberEntryResponse message are further described as follows:

/infod:Status

An indication that the request has been successfully executed. .

One of the following faults MUST be sent if the operation fails:

- ReplaceResourceAuthorizationFault: User not authorized to replace the INFOD resource at this INFOD registry
- UnknownResourceReferenceFault: An resource has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- UnsupportedXQueryFault: The XQuery specified could not be parsed correctly

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.2.3 DropSubscriberEntry

The DropSubscriberEntry operation removes an INFOD subscriber entry from an INFOD registry.

The format of the request message for a DropSubscriberEntry operation is:

```
<infod:DropSubscriberEntry>
  <infod:SubscriberEntryReference>
    wsa:EndPointReferenceType
  </infod:SubscriberEntryReference>
  <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
</infod:DropSubscriberEntry>
```

The elements of the DropSubscriberEntry message are further described as follows:

/infod:ResourceReference

An endpoint reference element, as defined by WS-Addressing, used to identify the INFOD resource in the registry to drop.

/infod:ExecutionMode

A parameter indicating the mode of execution of the drop request. Possible values are:

- "IF UNUSED" The drop request will execute only if the resource is unreferenced
- "DISABLE NEW" No new references are possible for the resource. The resource will be dropped when the last reference to this resource is gone
- "CASCADE" The drop request will execute immediately and all references to the resource will be removed recursively

If this parameter is not specified, the default value "IF UNUSED" MUST be used.

A WS-Addressing Action header with the value http://www.ogf.org/infod/INFODRegistry/DropSubscriberEntry MUST accompany the message

INFOD Registry Response

If the INFOD registry accepts the DropSubscriberEntry message, it MUST respond to the WS endpoint specified in the request message with a DropSubscriberEntryResponse message. The DropSubscriberEntry response message is a message of the following form:

```
<infod:DropSubscriberEntryResponse>
  <infod:Status>
 xsd:string default "COMPLETED"
  </infod:Status>
</infod:DropSubscriberEntryResponse>
```

The elements of the DropSubscriberEntryResponse message are further described as follows:

/infod:Status

An indication that the request has been successfully executed.

One of the following faults MUST be sent if the operation fails:

- DropResourceAuthorizationFailure: User not authorized to drop the INFOD resource at this INFOD registry
- UnknownResourceReferenceFault: An resource has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- ExecutionModeFault: Cannot use ExecutionMode provided

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.3 Managing Consumer Entries

The following operations are used to manage consumers:

- CreateConsumerEntry (section 2.3.1)
- ReplaceConsumerEntry (section 2.3.2)
- DropConsumerEntry (section 2.3.3)

2.3.1 CreateConsumerEntry

As part of the processing of a CreateConsumerEntry request message, the INFOD registry MUST create an INFOD entry representing the consumer.

The format of the request message for CreateConsumerEntry operation is based on the schema provided in Appendix I – XML Schema for an INFOD entry. Details are as follows:

```
<infod:CreateConsumerEntry>
  <infod:WSReference>
    wsa:EndPointReferenceType
  </infod:WSReference>
  <infod:ConsumerName> xsd:string </infod:ConsumerName> ?
  <infod:ConsumerDescription>
    xsd:string
  </infod:ConsumerDescription> ?
  <infod:PropertyConstraint>
    xsd:any
</infod:PropertyConstraint> *
  <infod:Notification>
  xsd:Boolean default "FALSE"
</infod:Notification> ?
</infod:CreateConsumerEntry>
```

The elements of the CreateConsumerEntry message are further described as follows:

/infod:WSReference

An endpoint reference element, as defined by WS-Addressing, used to identify the WS endpoint for the entry. Note that this MAY be the WS EPR of the requesting service, but does not have to be. The request MAY be made 'on behalf' of the actual service.

/infod:ConsumerName

A string representing the name of the consumer. This name MAY NOT be unique.

/infod:ConsumerDescription

A string representing a description of the consumer

/infod:PropertyConstraint

Property contraints are used to specify which conditions must be safisfied by other entries (publishers, data sources, and subscribers) to be eligible for interaction with this consumer. A property constraint MUST be formulated as an XQuery. The INFOD Base Use Case Scenarios (see http://forge.gridforum.org/sf/go/doc13626?nav=1) provide examples of XQueries.

For example, a consumer identifies the set of publishers that are eligible to react to subscriptions.

Note that the XQuery statement MUST be encoded correctly, i.e. characters such as ">" would be represented as ">"

infod:Notification

When used, the registry MUST notify the consumer about relevant changes in the INFOD registry. A fault MUST be returned if infod:WSReference is not specified.

For further details see section 3.2.3.

A WS-Addressing Action header with the value http://www.ogf.org/infod/INFODRegistry/CreateConsumerEntry MUST accompany the message

INFOD Registry Response

If the INFOD registry accepts the CreateConsumerEntry message, it MUST respond to the WS endpoint specified in the request message with a CreateConsumerEntryResponse message. The CreateConsumerEntry response message is a message of the following form:

```
<infod:CreateConsumerEntryResponse>
  <infod:ConsumerEntryReference>
    wsa:EndPointReferenceType
  </infod:ConsumerEntryReference>
</infod:CreateConsumerEntryResponse>
```

The elements of the CreateConsumerEntryResponse message are further described as follows:

/infod:ConsumerEntryReference

An endpoint reference element, as defined by WS-Addressing, used to identify the newly created consumer entry in the INFOD registry.

One of the following faults MUST be sent if the operation fails:

- CreateResourceAuthorizationFault: User not authorized to create the INFOD resource at this INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified

827 • UnsupportedXQueryFault: The XQuery specified could not be parsed correctly

828 The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP
829 v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

830 2.3.2 ReplaceConsumerEntry

831 As part of the processing of a ReplaceConsumerEntry request message, the INFOD registry MUST
832 replace the entire INFOD metadata for the entry representing the consumer. All previously defined
833 values MUST be deleted. The ReplaceConsumerEntry differs from the CreateConsumerEntry
834 interface in that it replaces an existing consumer entry and assigns the original EPR to the replaced
835 consumer.

836 The format of the request message for a ReplaceConsumer operation is also based on the schema
837 definition provided in Appendix I – XML Schema for an INFOD entry. Details are as follows:

```
838 <infod:ReplaceConsumerEntry>
839   <infod:WSReference>
840     wsa:EndPointReferenceType
841   </infod:WSReference>
842   <infod:ConsumerEntryReference>
843     wsa:EndPointReferenceType
844   </infod:ConsumerEntryReference>
845   <infod:ConsumerName> xsd:string </infod:ConsumerName> ?
846   <infod:ConsumerDescription>
847     xsd:string
848   </infod:ConsumerDescription> ?
849   <infod:PropertyConstraint>
850     xsd:any
851 </infod:PropertyConstraint> *
852   <infod:Notification>
853   xsd:Boolean default "FALSE"
854 </infod:Notification> ?
855 </infod:ReplaceConsumerEntry>
```

856 The elements of the ReplaceConsumerEntry message are further described as follows:

857 /infod:WSReference

858 A REQUIRED endpoint reference element, as defined by WS-Addressing, used to identify the
859 WS endpoint for the entry. Note that this MAY be the WS EPR of the requesting service, but
860 does not have to be. The request MAY be made 'on behalf' of the actual service.

861 /infod:ConsumerEntryReference

862 A REQUIRED endpoint reference element, as defined by WS-Addressing, used to identify the
863 resource in the INFOD registry that will be replaced.

864 /infod:ConsumerName

865 A string representing the name of the consumer. This name MAY NOT be unique.

866 /infod:ConsumerDescription

867 A string representing a description of the consumer

868 /infod:PropertyConstraint

869 Property contraints are used to specify which conditions must be safisfied by other entries
870 (publishers, data sources, and subscribers) to be eligible for interaction with this consumer. A
871 property constraint MUST be formulated as an XQuery. The INFOD Base Use Case

Scenarios (see http://forge.gridforum.org/sf/go/doc13626?nav=1) provide examples of XQueries.

For example, a consumer identifies the set of publishers that are eligible to react to subscriptions.

Note that the XQuery statement MUST be encoded correctly, i.e. characters such as ">" would be represented as ">"

infod:Notification

When used, the registry MUST notify the consumer about relevant changes in the INFOD registry. A fault MUST be returned if infod:WSReference is not specified.

For further details see section 3.2.3.

A WS-Addressing Action header with the value http://www.ogf.org/infod/INFODRegistry/ReplaceConsumerEntry MUST accompany the message

INFOD Registry Response

If the INFOD registry accepts the ReplaceConsumerEntry message, it MUST respond to the WS endpoint specified in the request message with a ReplaceConsumerEntryResponse message. The ReplaceConsumerEntry response message is a message of the following form:

```
<infod:ReplaceConsumerEntryResponse>
  <infod:Status>
 xsd:string default "COMPLETED"
  </infod:Status>
</infod:ReplaceConsumerEntryResponse>
```

The elements of the ReplaceConsumerEntryResponse message are further described as follows:

/infod:Status

An indication that the request has been successfully executed.

One of the following faults MUST be sent if the operation fails:

- ReplaceResourceAuthorizationFault: User not authorized to replace the INFOD resource at this INFOD registry
- UnknownResourceReferenceFault: An resource has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- UnsupportedXQueryFault: The XQuery specified could not be parsed correctly

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.3.3 DropConsumerEntry

The DropConsumerEntry operation removes an INFOD consumer entry from an INFOD registry.

The format of the request message for a DropConsumerEntry operation is:

```
<infod:DropConsumerEntry>
  <infod:ConsumerEntryReference>
    wsa:EndPointReferenceType
  </infod:ConsumerEntryReference>
```

```
  <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
</infod:DropConsumerEntry>
```

The elements of the DropConsumerEntry message are further described as follows:

/infod:ConsumerEntryReference

An endpoint reference element, as defined by WS-Addressing, used to identify the INFOD resource in the registry to drop.

/infod:ExecutionMode

A parameter indicating the mode of execution of the drop request. Possible values are:

"IF UNUSED"	The drop request will execute only if the resource is unreferenced
"DISABLE NEW"	No new references are possible for the resource. The resource will be dropped when the last reference to this resource is gone
"CASCADE"	The drop request will execute immediately and all references to the resource will be removed recursively

If this parameter is not specified, the default value "IF UNUSED" MUST be used.

A WS-Addressing Action header with the value
http://www.ogf.org/infod/INFODRegistry/DropConsumerEntry MUST accompany the message

INFOD Registry Response

If the INFOD registry accepts the DropConsumerEntry message, it MUST respond to the WS endpoint specified in the request message with a DropConsumerResponseEntry message. The DropConsumerEntry response message is a message of the following form:

```
<infod:DropConsumerEntryResponse>
  <infod:Status>
 xsd:string default "COMPLETED"
  </infod:Status>
</infod:DropConsumerEntryResponse>
```

The elements of the DropConsumerResponseEntry message are further described as follows:

/infod:Status

An indication that the request has been successfully executed.

One of the following faults MUST be sent if the operation fails:

- DropResourceAuthorizationFailure: User not authorized to drop the INFOD resource at this INFOD registry
- UnknownResourceReferenceFault: An resource has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- ExecutionModeFault: Cannot use ExecutionMode provided

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.4 Managing Subscriptions

The following operations are used to manage subscriptions:

- CreateSubscription (section 2.4.1)
- ReplaceSubscription (section 2.4.2)
- DropSubscription (section 2.4.3)

2.4.1 CreateSubscription

The CreateSubscription operation is used by a subscriber, to create an INFOD subscription in an INFOD registry.

This subscription resource is responsible to describe the conditions of interest of potential consumers for potential publishers.

As part of the processing of a CreateSubscription request message, the INFOD registry MUST create an INFOD resource representing the subscription.

The format of the request message for CreateSubscription operation is based on the schema provided in Appendix I – XML Schema for an INFOD resource. Details are as follows:

```
<infod:CreateSubscription>
  <infod:SubscriptionName> xsd:string </infod:SubscriptionName> ?
  <infod:SubscriptionDescription>
    xsd:string
  </infod:SubscriptionDescription> ?
  <infod:SubscriberEntryReference>
    wsa:EndPointReferenceType
  </infod:SubscriberEntryReference>
  <infod:DataConstraint >
    xsd:anyType
  </infod:DataConstraint> *
  <infod:PropertyConstraint>
    xsd:any
  </infod:PropertyConstraint> *
  <infod:DynamicConsumerConstraint>
    xsd:anyType
  </infod:DynamicConsumerConstraint> *
</infod:CreateSubscription>
```

The elements of the CreateSubscription message are further described as follows:

/infod:SubscriptionName

A string representing the name for the subscription. This name MAY NOT be unique.

/infod:SubscriptionDescription

A string representing a description of the subscription.

/infod:SubscriberEntryReference

An endpoint reference element to the INFOD EPR, as defined by WS-Addressing, used to identify the subscriber entry responsible for the subscription.

/infod:DataConstraint

DataConstraint specifies which information is of interest to consumers. The constraint(s) language(s) is/are implicitly defined through the reference of the vocabulary EPR. Data Constraints are not applied by the INFOD registry but by the publishers.

See 2.5 for more details on how to define a vocabulary referenced by such constraints.

/infod:PropertyConstraint

Property contraints are used to specify which conditions must be safisfied by entries (publishers, data sources, and consumers) to be eligible for this subsription. A property constraint MUST be formulated as an XQuery. The INFOD Base Use Case Scenarios (see http://forge.gridforum.org/sf/go/doc13626?nav=1) provide examples of XQueries.

For example, a subscription identifies the set of publishers that are eligible to react to this subscription.

Note that the XQuery statement MUST be encoded correctly, i.e. characters such as ">" would be represented as ">"

/infod:DynamicConsumerConstraint

An element specifying which consumers receive a specific message. The constraint(s) language(s) is/are implicitly defined through the reference of the vocabulary EPR.

These Constraints are designed to determine the consumers of each message based on its content; i.e., a Dynamic Consumer Constraint cannot be applied by the INFOD registry and is processed by the publishers.

infod:PropertyConstraint should be used to specify consumer constraints if all messages created in response to the subscription are published to the same set of consumers.

For example, a message representing a bill should be *published* to the payee.

A WS-Addressing Action header with the value
http://www.ogf.org/infod/INFODRegistry/CreateSubscription MUST accompany the message

INFOD Registry Response

If the INFOD registry accepts the CreateSubscription message, it MUST respond to the WS endpoint specified in the request message with a CreateSubscriptionResponse message. The CreateSubscription response message is a message of the following form:

```
<infod:CreateSubscriptionResponse>
  <infod:SubscriptionReference>
    wsa:EndPointReferenceType
  </infod:SubscriptionReference>
</infod:CreateSubscriptionResponse>
```

The elements of the CreateSubscriptionResponse message are further described as follows:

/infod:SubscriptionReference

An endpoint reference element, as defined by WS-Addressing, used to identify the newly created subscription in the INFOD registry.

One of the following faults MUST be sent if the operation fails:

- CreateResourceAuthorizationFault: User not authorized to create the INFOD resource at this INFOD registry

- UnknownResourceReferenceFault: An resource has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- UnsupportedXQueryFault: The XQuery specified could not be parsed correctly

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.4.2 ReplaceSubscription

As part of the processing of a ReplaceSubscription request message, the INFOD registry MUST replace the entire INFOD metadata for the resource representing the subscription. All previously defined values MUST be deleted. The ReplaceSubscription differs from the CreateSubscription interface in that it replaces an existing subscription resource and assigns the original EPR to the replaced subscription.

The format of the request message for a ReplaceSubscription operation is also based on the schema definition provided in Appendix I – XML Schema for an INFOD resource. Details are as follows:

```
<infod:ReplaceSubscription>
  <infod:SubscriptionReference>
    wsa:EndPointReferenceType
  </infod:SubscriptionReference>
  <infod:SubscriptionName> xsd:string </infod:SubscriptionName> ?
  <infod:SubscriptionDescription>
    xsd:string
  </infod:SubscriptionDescription> ?
  <infod:SubscriberReference>
    wsa:EndPointReferenceType
  </infod:SubscriberReference>
  <infod:DataConstraint>
    xsd:anyType
  </infod:DataConstraint> *
  <infod:PropertyConstraint>
    xsd:any
  </infod:PropertyConstraint> *
  <infod:DynamicConsumerConstraint>
    xsd:anyType
  </infod:DynamicConsumerConstraint> *
</infod:ReplaceSubscription>
```

The elements of the ReplaceSubscription message are further described as follows:

/infod:SubscriptionReference

An endpoint reference element, as defined by WS-Addressing, used to identify the subscription resource in the INFOD registry that will be replaced.

/infod:SubscriptionName

A string representing the name of the subscription. This name MAY NOT be unique.

/infod:SubscriptionDescription

A string representing a description of the subscription.

/infod:SubscriberEntryReference

An endpoint reference element to the INFOD EPR, as defined by WS-Addressing, used to identify the subscriber entry responsible for the subscription.

/infod:DataConstraint

DataConstraint specifies which information is of interest to consumers. The constraint(s) language(s) is/are implicitly defined through the reference of the vocabulary EPR. Data Constraints are not applied by the INFOD registry but by the publishers.

See 2.5 for more details on how to define a vocabulary referenced by such constraints.

Note: If no data constraint is specified all messages published by publishers are of interest.

/infod:PropertyConstraint

Property contraints are used to specify which conditions must be safisfied by entries (publishers, data sources, and consumers) to be eligible for this subsription. A property constraint MUST be formulated as an XQuery. The INFOD Base Use Case Scenarios (see http://forge.gridforum.org/sf/go/doc13626?nav=1) provide examples of XQueries.

For example, a subscription identifies the set of publishers that are eligible to react to this subscription.

Note that the XQuery statement MUST be encoded correctly, i.e. characters such as “>” would be represented as “>”

/infod:DynamicConsumerConstraint

An element specifying which consumers receive a specific message. The constraint(s) language(s) is/are implicitly defined through the reference of the vocabulary EPR.

These Constraints are designed to determine the consumers of each message based on its content; i.e., a Dynamic Consumer Constraint cannot be applied by the INFOD registry and is processed by the publishers.

infod:PropertyConstraint should be used to specify consumer constraints if all messages created in response to the subscription are disseminated to the same set of consumers.

For example, a message representing a bill should be disseminated to the payee.

A WS-Addressing Action header with the value
http://www.ogf.org/infod/INFODRegistry/ReplaceSubscription MUST accompany the message

INFOD Registry Response

If the INFOD registry accepts the ReplaceSubscriptionRequest, it MUST respond to the WS endpoint specified in the request message with a ReplaceSubscription message. The ReplaceSubscription response message is a message of the following form:

```
<infod:ReplaceSubscriptionResponse>
  <infod:Status>
 xsd:string default ”COMPLETED”
  </infod:Status>
</infod:ReplaceSubscriptionResponse>
```

The elements of the ReplaceSubscriptionResponse message are further described as follows:

/infod:SubscriptionReference

An endpoint reference element, as defined by WS-Addressing, used to identify the subscription resource in the INFOD registry to replace.

One of the following faults MUST be sent if the operation fails:

- ReplaceResourceAuthorizationFault: User not authorized to replace the INFOD resource at this INFOD registry
- UnknownResourceReferenceFault: An resource has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- UnsupportedXQueryFault: The XQuery specified could not be parsed correctly

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.4.3 DropSubscription

The DropSubscription operation MUST be used to remove an INFOD subscription resource from an INFOD registry.

The format of the request message for a DropSubscription operation is:

```
<infod:DropSubscription>
  <infod:SubscriptionReference>
     wsa:EndPointReferenceType
  </infod:SubscriptionReference>
  <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
</infod:DropSubscription>
```

The elements of the DropSubscription message are further described as follows:

/infod:SubscriptionReference

An endpoint reference element, as defined by WS-Addressing, used to identify the INFOD subscription resource in the registry to drop.

/infod:ExecutionMode

An optional parameter indicating the mode of execution of the drop request. Possible values are:

“IF UNUSED”	The drop request will execute only if the resource is unreferenced
“DISABLE NEW”	No new references are possible for the resource. The resource will be dropped when the last reference to this resource is gone
“CASCADE”	The drop request will execute immediately and all references to the resource will be removed recursively

If this parameter is not specified, the default value “IF UNUSED” MUST be used.

A WS-Addressing Action header with the value http://www.ogf.org/infod/INFODRegistry/DropSubscription MUST accompany the message

INFOD Registry Response

If the INFOD registry accepts the DropSubscription request, it MUST respond to the WS endpoint specified in the request message with a DropSubscriptionResponse message. The DropSubscriptionResponse message is a message of the following form:

```
<infod:DropSubscriptionResponse>
  <infod:Status>
 xsd:string default ”COMPLETED”
  </infod:Status>
```

```
</infod:DropSubscriptionResponse>
```

The elements of the ReplaceSubscriptionResponse message are further described as follows:

/infod:Status

An indication that the request has been successfully executed.

One of the following faults MUST be sent if the operation fails:

- DropResourceAuthorizationFailure: User not authorized to drop the INFOD resource at this INFOD registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- ExecutionModeFault: Cannot use ExecutionMode provided

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.5 Managing Vocabularies

INFOD has a set of predefined vocabularies. These are REQUIRED vocabularies for the INFOD registry:

- INFOD PublisherEntry Vocabulary
- INFOD SubscriberEntry Vocabulary
- INFOD ConsumerEntry Vocabulary
- INFOD Subscription Vocabulary
- INFOD DataSourceEntry Vocabulary

These vocabularies are used by the INFOD registry to match publishers with consumers through subscriptions and ensure that property constraints and data constraints are validated. All of these vocabularies are described in xml and detailed in section 5

Users MAY also define two additional types of vocabularies:

Property Vocabularies: Entries may specify properties that define their characteristics. They do that using a property vocabulary that may be queried. If two or more entries share the same property vocabulary, they can specify constraints on each other. The INFOD registry MAY manage constraints on these property vocabularies in addition to constraints formulated in the INFOD vocabularies. Property Vocabularies MUST be defined in xml.

***Data Vocabularies*:** In order to tell publishers which messages a subscription is interested in, they MUST agree on the data vocabulary. The data vocabulary is referenced in the *DataConstraints* component of a subscription resource, which allows INFOD subscribers to describe the structure of the published data/data of interest to them.

Data constraints' definitions MUST point to an existing data vocabulary and thus are simply equivalent to defining operations on top of an existing vocabulary (i.e. selection criteria, etc. on top of published data). Data Vocabularies are not limited to xml.

This section describes how these two types of vocabulary are created and dropped from an INFOD registry. It also includes operations for creating and dropping instances of a registered property vocabulary.

2.5.1 CreatePropertyVocabulary

The CreatePropertyVocabulary creates a property vocabulary in an INFOD registry. The Property Vocabulary is an XML schema. As part of the processing of a CreatePropertyVocabulary request message, the INFOD registry MUST create a new resource for that vocabulary.

The format of the request message for CreatePropertyVocabulary operation is as follows:

```
<infod:CreatePropertyVocabulary>
  <infod:PropertyVocabularyName>
    xsd:string
  </infod:PropertyVocabularyName> ?
  <infod:PropertyVocabularyDescription>
    xsd:string
  </infod:PropertyVocabularyDescription> ?
  <infod:PropertyVocabularyBody>
    xsd:schema
  </infod:PropertyVocabularyBody>
</infod:CreatePropertyVocabulary>
```

The elements of the CreatePropertyVocabulary message are further described as follows:

/infod:PropertyVocabularyName

A string representing a name that is local to the INFOD registry where the CreatePropertyVocabulary operation takes place. This name MAY NOT be unique.

Names MUST NOT start with $$infod.

/infod:PropertyVocabularyDescription

A string representing a description of the vocabulary.

/infod:PropertyVocabularyBody

An element defining an XML Schema. This is an extensibility mechanism to allow XML elements to be specified for the defined property vocabulary.

A WS-Addressing Action header with the value http://www.ogf.org/infod/INFODRegistry/CreatePropertyVocabulary MUST accompany the message.

INFOD Registry Response

If the INFOD registry accepts the CreatePropertyVocabulary request, it MUST respond to the WS endpoint specified in the request message with a CreateVocabularyResponse message.

In case of a successful registration, the CreateVocabularyResponse message is a message of the following form:

```
<infod:CreatePropertyVocabularyResponse>
    <infod:PropertyVocabularyReference>
       wsa:EndPointReferenceType
    </infod:PropertyVocabularyReference>
</infod:CreateVocabularyResponse>
```

The elements of the CreateVocabularyResponse message are further described as follows:

/infod:PropertyVocabularyReference

An endpoint reference element, as defined by WS-Addressing, used to identify the newly created property vocabulary.

One of the following faults MUST be sent if the operation fails::

- CreateResourceAuthorizationFault: User not authorized to create a resource at this INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- UnSupportedVocabularyFault: Vocabulary Language not supported

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.5.2 DropPropertyVocabulary

The DropPropertyVocabulary operation drops a particular property vocabulary from an INFOD registry.

The format of the request message for an DropPropertyVocabulary operation is:

```
<infod:DropPropertyVocabulary>
  <infod:PropertyVocabularyReference>
    wsa:EndPointReferenceType
  </infod:PropertyVocabularyReference
  <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
</infod:DropPropertyVocabulary>
```

The elements of the DropPropertyVocabulary message are further described as follows:

/infod: PropertyVocabularyReference

An endpoint reference element, as defined by WS-Addressing, used to identify the vocabulary to drop from the Registry.

/infod:ExecutionMode

A parameter indicating the mode of execution of the drop request. Possible values are:

“IF UNUSED”	The drop request will execute only if the resource is unreferenced
“DISABLE NEW”	No new references are possible for the resource. The resource will be dropped when the last reference to this resource is gone
“CASCADE”	The drop request will execute immediately and all references to the resource will be removed recursively

If this parameter is not specified, the default value “IF UNUSED” MUST be used.

A WS-Addressing Action header with the value http://www.ogf.org/infod/INFODRegistry/DropPropertyVocabulary MUST accompany the message

INFOD Registry Response

If the INFOD registry accepts the DropPropertyVocabulary request, it MUST respond to the WS endpoint specified in the request message with an DropPropertyVocabularyResponse message. The DropPropertyVocabulary response message is a message of the following form:

```
<infod:DropPropertyVocabularyResponse>
  <infod:Status>
 xsd:string default ”COMPLETED”
```

```
  </infod:Status>
</infod:DropPropertyVocabularyResponse>
```

The elements of the DropPropertyVocabularyResponse message are further described as follows:

/infod:Status

An indication that the request has been successfully executed.

One of the following faults MUST be sent if the operation fails:

- DropResourceAuthorizationFailure: User not authorized to drop the resource at this INFOD registry
- UnknownResourceReferenceFault: An element has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- ExecutionModeFault: Cannot use ExecutionMode provided

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.5.3 CreatePropertyVocabularyInstance

The CreatePropertyVocabularyInstance operation creates a new instance of a particular property vocabulary previously created in the INFOD registry. An instance of a property vocabulary fills in values into the vocabulary structure defined by the Property Vocabulary (section 2.5.1) and relates a particular INFOD entry to the instance. The referenced entry is now identified to use the property vocabulary.

As part of the processing of a CreatePropertyVocabularyInstance request message, the INFOD registry MUST create a new instance for that vocabulary.

The format of the request message for CreatePropertyVocabularyInstance operation is as follows:

```
<infod:CreatePropertyVocabularyInstance>
  <infod:EntryReference>
    wsa:EndPointReferenceType
  </infod:EntryReference>
  <infod:PropertyVocabularyReference>
    wsa:EndPointReferenceType
  </infod:PropertyVocabularyReference>
  <infod:PropertyVocabularyInstanceBody>
    {xsd:anyType} ?
  </infod:PropertyVocabularyInstanceBody>
</infod:CreatePropertyVocabularyInstance>
```

The elements of the CreatePropertyVocabularyInstance message are further described as follows:

/infod:EntryReference

EPR of the INFOD entry that the instance of the property vocabulary will be identified with.

/infod:PropertyVocabularyReference

EPR of a vocabulary that will be referenced to the INFOD resource.

/infod:PropertyVocabularyInstanceBody

An element that contains specific instance information that needs to match the structure of the vocabulary defined in VocabularyReference.

A WS-Addressing Action header with the value http://www.ogf.org/infod/INFODRegistry/CreatePropertyVocabularyInstance MUST accompany the message.

INFOD Registry Response

If the INFOD registry accepts the CreatePropertyVocabularyInstance request, it MUST respond to the WS endpoint specified in the request message with a CreatePropertyVocabularyInstance response message.

The CreatePropertyVocabularyInstanceResponse message is a message of the following form:

```
<infod:CreatePropertyVocabularyInstanceResponse>
  <infod:PropertyVocabularyInstanceReference>
    wsa:EndPointReferenceType
  </infod:PropertyVocabularyInstanceReference>
</infod:CreatePropertyVocabularyInstanceResponse>
```

The elements of the CreatePropertyVocabularyInstanceResponse message are further described as follows:

/infod:PropertyVocabularyInstanceReference

An endpoint reference element, as defined by WS-Addressing, used to identify the newly created vocabulary instance.

One of the following faults MUST be sent if the operation fails:

- CreateResourceAuthorizationFault: User not authorized to create the INFOD resource at this INFOD registry
- UnknownResourceReferenceFault: An resource has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- UnSupportedVocabularyFault: Vocabulary Language not supported

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf) .

2.5.4 DropPropertyVocabularyInstance

The DropPropertyVocabularyInstance operation drops an existing instance of a particular property vocabulary previously created in the INFOD registry.

The format of the request message for a DropPropertyVocabularyInstance operation is:

```
<infod:DropPropertyVocabularyInstance>
  <infod:PropertyVocabularyInstanceReference>
     wsa:EndPointReferenceType
  </infod:PropertyVocabularyInstanceReference>
  <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
</infod:DropPropertyVocabularyInstance>
```

The elements of the DropPropertyVocabularyInstance message are further described as follows:

/infod:PropertyVocabularyInstanceReference

An endpoint reference element, as defined by WS-Addressing, used to identify the property vocabulary instance to drop from the Registry.

/infod:ExecutionMode

A parameter indicating the mode of execution of the drop request. Possible values are:

"IF UNUSED"	The drop request will execute only if the resource is unreferenced
"DISABLE NEW"	No new references are possible for the resource. The resource will be dropped when the last reference to this resource is gone
"CASCADE"	The drop request will execute immediately and all references to the resource will be removed recursively

If this parameter is not specified, the default value "IF UNUSED" MUST be used.

A WS-Addressing Action header with the value http://www.ogf.org/infod/INFODRegistry/DropVocabularyInstance MUST accompany the message.

INFOD Registry Response

If the INFOD registry accepts the DropPropertyVocabularyInstance request, it MUST respond to the WS endpoint specified in the request message with a DropPropertyVocabularyInstanceResponse message in the following form:

```
<infod:DropPropertyVocabularyInstanceResponse>
  <infod:Status>
 xsd:string default "COMPLETED"
  </infod:Status>
</infod:DropPropertyVocabularyInstanceResponse>
```

The elements of the DropPropertyVocabularyInstanceResponse message are further described as follows:

/infod:Status

An indication that the request has been successfully executed.

One of the following faults MUST be sent if the operation fails:

- DropResourceAuthorizationFailure: User not authorized to drop the resource at this INFOD registry
- UnknownResourceReferenceFault: An resource has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- ExecutionModeFault: Cannot use ExecutionMode provided

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.5.5 CreateDataVocabulary

As part of the processing of a CreateDataVocabulary request message, the INFOD registry MUST create a new resource for that vocabulary.

The format of the request message for CreateDataVocabulary operation is:

```
<infod:CreateDataVocabulary>
```

```
  <infod:DataVocabularyName> xsd:string </infod:DataVocabularyName> ?
  <infod:DataVocabularyDescription>
    xsd:string
  </infod:DataVocabularyDescription> ?
  <infod:DataVocabularyLanguage>
    {anyURI} (Namespace/URI of DataFormat)
  </infod:DataVocabularyLanguage>
  <infod:LanguageUsageDescription>
    xsd:string
  </infod:LanguageUsageDescription> ?
  <infod:DataVocabularyBody>
    xsd:anyType
  </infod:DataVocabularyBody>
</infod:CreateDataVocabulary>
```

The elements of the CreateDataVocabulary message are further described as follows:

/infod:DataVocabularyName

A string.representing a name in the INFOD registry where the CreateDataVocabulary operation takes place. This name MAY NOT be unique.

Names MUST NOT start with $$infod.

/infod:DataVocabularyDescription

A string representing a description of the vocabulary.

/infod:DataVocabularyLanguage

A URI defining the format of the data vocabulary.

/infod:DataVocabularyBody

A string representing a data vocabulary.

This embedded string represents the vocabulary and MUST be encoded correctly as defined through the DataVocabularyLanguage definition (escape characters etc.)

A WS-Addressing Action header with the value
http://www.ogf.org/infod/INFODRegistry/CreateDataVocabulary MUST accompany the message.

INFOD Registry Response

If the INFOD registry accepts the CreateDataVocabulary request, it MUST respond to the WS endpoint specified in the request message with a CreateVocabularyResponse message.

In case of a successful registration, the CreateVocabularyResponse message is a message of the following form:

```
<infod:CreateDataVocabularyResponse>
  <infod:DataVocabularyReference>
    wsa:EndPointReferenceType
  </infod:DataVocabularyReference>
</infod:CreateDataVocabularyResponse>
```

The elements of the CreateVocabularyResponse message are further described as follows:

/infod:DataVocabularyReference

An endpoint reference element, as defined by WS-Addressing, used to identify the newly created vocabulary.

One of the following faults MUST be sent if the operation fails:

- CreateResourceAuthorizationFault: User not authorized to create a resource at this INFOD registry
- UnknownResourceReferenceFault: An resource has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- UnSupportedVocabularyFault: Vocabulary Language not supported

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.5.6 DropDataVocabulary

The DropDataVocabulary operation drops a particular data vocabulary from an INFOD registry.

The format of the request message for an DropDataVocabulary operation is:

```
<infod:DropDataVocabulary>
  <infod:DataVocabularyReference>
    wsa:EndPointReferenceType
  </infod:DataVocabularyReference
  <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
</infod:DropDataVocabulary>
```

The elements of the DropDataVocabulary message are further described as follows:

/infod: DataVocabularyReference

An endpoint reference element, as defined by WS-Addressing, used to identify the vocabulary to drop from the Registry.

/infod:ExecutionMode

A parameter indicating the mode of execution of the drop request. Possible values are:

“IF UNUSED”	The drop request will execute only if the resource is unreferenced
“DISABLE NEW”	No new references are possible for the resource. The resource will be dropped when the last reference to this resource is gone
“CASCADE”	The drop request will execute immediately and all references to the resource will be removed recursively

If this parameter is not specified, the default value “IF UNUSED” MUST be used.

A WS-Addressing Action header with the value http://www.ogf.org/infod/INFODRegistry/DropDataVocabulary MUST accompany the message

INFOD Registry Response

If the INFOD registry accepts the DropDataVocabulary request, it MUST respond to the WS endpoint specified in the request message with a DropDataVocabularyResponse message. The DropDataVocabulary response message is a message of the following form:

```
<infod:DropDataVocabularyResponse>
  <infod:Status>
 xsd:string default ”COMPLETED”
  </infod:Status>
```

```
</infod:DropDataVocabularyResponse>
```

The elements of the DropDataVocabularyResponse message are further described as follows:

/infod:Status

An indication that the request has been successfully executed.

One of the following faults MUST be sent if the operation fails:

- DropResourceAuthorizationFailure: User not authorized to drop the resource at this INFOD registry
- UnknownResourceReferenceFault: An element has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- ExecutionModeFault: Cannot use ExecutionMode provided

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.6 Data Source Entries

A Data Source Entry relates a publisher entry with a data vocabulary.

The following operations are used to manage data sources:

- CreateDataSourceEntry (section 2.6.1)
- DropDataSourceEntry (section 2.6.2)

2.6.1 CreateDataSourceEntry

The CreateDataSourceEntry operation creates a relation between an INFOD publisher entry and a data vocabulary at the INFOD registry. As part of the processing of an CreateDataSourceEntry operation message, the INFOD registry MUST create an INFOD vocabulary association resource.

The format of the request message for an CreateDataSourceEntry operation is:

```
<infod:CreateDataSourceEntry>
  <infod:DataSourceEntryName> ?
    xsd:string
  </infod:DataSourceEntryName>
  <infod:DataSourceEntryDescription>
     xsd:string
  </infod:DataSourceEntryDescription> ?
  <infod:PublisherEntryReference>
     wsa:EndPointReferenceType
  </infod:PublisherEntryReference>
  <infod:DataVocabularyReference>
     wsa:EndPointReferenceType
  </infod:DataVocabularyReference> +
  <infod:PropertyConstraint>
    xsd:any
  </infod:PropertyConstraint> *
</infod:CreateDataSourceEntry>
```

The elements of the CreateDataSourceEntry message are further described as follows:

/infod:DataSourceEntryName

A string representing the name of the data source entry. This name MAY NOT be unique.

/infod:DataSourceEntryDescription

A string representing a description of the data source entry.

/infod:PublisherEntryReference

The EPR of the publisher entry for which a data source entry is created.

/infod:DataVocabularyReference

The EPR(s) of a vocabulary with which to associate the publisher entry.

/infod:PropertyConstraint

Property contraints are used to specify which conditions must be safisfied by entries (subscribers and consumers) to be eligible to receive data from this data source. A property constraint MUST be formulated as an XQuery. The INFOD Base Use Case Scenarios (see http://forge.gridforum.org/sf/go/doc13626?nav=1) provide examples of XQueries.

For example, a data sources identifies the set of consumers that are eligible to receive data from this data source.

Note that the XQuery statement MUST be encoded correctly, i.e. characters such as “>” would be represented as “>”

A WS-Addressing Action header with the value http://www.ogf.org/infod/INFODRegistry/CreateDataSourceEntry MUST accompany the message.

INFOD Registry Response

If the INFOD registry accepts the CreateDataSourceEntry request, it MUST respond to the WS endpoint specified in the request message with a CreateDataSourceEntryResponse message.

The CreateDataSourceEntryResponse message is a message of the following form:

```
<infod:CreateDataSourceEntryResponse>
  <infod:DataSourceEntryReference>
    wsa:EndPointReferenceType
  </infod:DataSourceEntryReference>
</infod:CreateDataSourceEntryResponse>
```

The elements of the response message are further described as follows:

/infod:DataSourceEntryReference

An endpoint reference element, as defined by WS-Addressing, used to identify the newly created vocabulary association.

One of the following faults MUST be sent if the operation fails:

- CreateResourceAuthorizationFault: User not authorized to create the resource at this INFOD registry
- UnknownResourceReferenceFault: An resource has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- UnsupportedXQueryFault: The XQuery specified could not be parsed correctly

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.6.2 DropDataSourceEntry

The DropDataSourceEntry operation drops a data source entry from an INFOD registry.

The format of the request message for a DropDataSourceEntry operation is:

```
<infod:DropDataSourceEntry>
  <infod:DataSourceEntryReference>
    wsa:EndPointReferenceType
  </infod:DataSourceEntryReference>
  <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
</infod:DropDataSourceEntry>
```

The elements of the DropDataSourceEntry message are further described as follows:

/infod: DataSourceEntryReference

An endpoint reference element, as defined by WS-Addressing, used to identify the association to drop from the Registry.

/infod:ExecutionMode

A parameter indicating the mode of execution of the drop request. Possible values are:

"IF UNUSED"	The drop request will execute only if the resource is unreferenced
"DISABLE NEW"	No new references are possible for the resource. The resource will be dropped when the last reference to this resource is gone
"CASCADE"	The drop request will execute immediately and all references to the resource will be removed recursively

If this parameter is not specified, the default value "IF UNUSED" MUST be used.

A WS-Addressing Action header with the value http://www.ogf.org/infod/INFODRegistry/DropDataSourceEntry MUST accompany the message.

INFOD Registry Response

If the INFOD registry accepts the DropDataSourceEntry request, it MUST respond to the WS endpoint specified in the request message with a DropDataSourceEntryResponse message. The DisCreateDataSourceEntryResponse message is a message of the following form:

```
<infod:DropDataSourceEntryResponse>
  <infod:Status>
 xsd:string default "COMPLETED"
  </infod:Status>
</infod:DropDataSourceEntryResponse>
```

The elements of the DropDataSourceEntryResponse message are further described as follows:

/infod:Status

An indication that the request has been successfully executed.

One of the following faults MUST be sent if the operation fails:

- DropResourceAuthorizationFailure: User not authorized to drop the resource at this INFOD registry

- UnknownResourceReferenceFault: An resource has been referenced that is unknown to the INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- ExecutionModeFault: Cannot use ExecutionMode provided

The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

2.7 The GetMetaData Operation

The Base Meta Data Access interface provides access to data contained in an INFOD registry. The request is formulated as an XQuery and the result is returned according to the specification in the return clause of the XQuery.

The format of the request message for a GetMetadata operation is:

```
<infod:GetMetaData>
  <infod:MetaDataQueryExpression>
 {xsd:anyType}
  </infod:MetaDataQueryExpression>
</infod:GetMetadata>
```

The elements of the GetMetadata message are further described as follows:

/infod:MetaDataQueryExpression

The element MUST be a valid XQuery or an XPath expression for the INFOD registry.

The INFOD registry is fully qualified by an INFOD registry service name appended to the string “INFODRegistry.xml”. A fully qualified name allows the registry instance to be referenced uniquely.

An example for a fully qualified INFOD registry is: http://www.w3c.org/OGF/INFOD/Instance/INFODRegistry.xml.

The INFOD registry service name need not be hard coded into the XQuery fn:doc but could be specified by setting the base-URI to be the service name e.g. declare base-URI "http://www.w3c.org/OGF/INFOD/Instance/INFODRegistry.xml". This indirection allows us to specify specific a registry amongst many in a given environment.

Default XPath expresssions:

In addition to supporting user defined Xpath/XQuery expressions, INFOD reserves the following paths and mandates their implementation.

- **All publishers** - fn:doc('INFODRegistry.xml')/publishers/$$infodPublisher
- **All subscribers** - fn:doc('INFODRegistry.xml')/subscribers/$$infodSubscriber
- **All consumers** - fn:doc('INFODRegistry.xml')/consumers/$$infodConsumer
- **All subscriptions** - fn:doc('INFODRegistry.xml')/subscriptions/$$infodSubscription
- **All property vocabularies** - fn:doc('INFODRegistry.xml')/propertyvocabularies/$$infodPropertyVocabulary

- **All property vocabulary instances** - fn:doc('INFODRegistry.xml')/propertyvocabularyinstances/$$infodPropertyVocabularyInstance
- **All data vocabulary** - fn:doc('INFODRegistry.xml')/datavocabularies/$$infodDataVocabulary

A WS-Addressing Action header with the value http://www.ogf.org/infod/INFODRegistry/GetMetaData MUST accompany the message.

INFOD Registry Response

The response of the INFOD registry is:

```
<infod:GetMetaDataQueryResponse>
  <infod:MetaDataQueryResult>
    {xsd:anyType}
  <infod:GetMetaDataQueryResult>
<infod:MetaDataQueryResponse>
```

The content of infod:GetMetaDataQueryResult MUST be structured according to the return specification in the GetMetaData request.

One of the following faults MUST be sent if the operation fails:

- GetMetaDataAuthorizationFailure: User not authorized to use the operation at this INFOD registry
- MissingRequiredParameterFault: A required parameter was not specified
- UnsupportedXQueryFault: The XQuery specified could not be parsed correctly

The message MUST be structured according to the WS-Base Faults specification. For examples using SOAP, see the SOAP v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

3 Base INFOD Notification Interfaces

We divide notifications between INFOD components into two major categories: notifications from publishers to consumers which carry the actual data, and notifications from the registry to publishers, subscribers, and consumers which contain information about relevant state changes in the registry. INFOD does not use the WSN notify interface due to different header requirements.

3.1 Notifications from Publishers to Consumers

An INFOD publisher uses a Notify operation similar to that defined by WS-Notification to send messages to an INFOD consumer (see http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.3-draft-01.pdf).

The following xml describes the format of an INFOD Notify message:

```
<infod:Notify>
  <infod:NotificationMessage>
    <infod:SubscriptionReference>
      wsa:EndpointReferenceType
    </infod:SubscriptionReference> ?
    <infod:Topic Dialect="xsd:anyURI">
      {any} ?
    </infod:Topic>?
    <infod:PublisherReference>
      wsa:EndpointReferenceType
    </infod:PublisherReference> ?
    <infod:Message>
      {any}
    </infod:Message>
  </infod:NotificationMessage> +
  {any} *
</infod:Notify>
```

The components of the Notify message are further described as follows:

/infod:Notify

Contains a collection of one or more Notifications.

/infod:NotificationMessage

Contains a Notification payload.

/infod:SubscriptionReference

An endpoint reference to the Subscription that is associated with the Notify message.

/infod:Topic

An endpoint reference to the VocabularyAssociation respresenting the source of the payload.

/infod:Topic/@Dialect

An endpoint reference to the vocabulary that was used to structure the payload.

/infod:ProducerReference

An endpoint reference to the Publisher that produced the Notification.

/infod:Message

The actual Notification payload.

/infod:Notify/{any}

The Notify message also allows for open content, in order to accommodate elements that may be needed by extensions built on the WSN BaseNotification (see http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.3-draft-01.pdf), including those providing additional filtering mechanisms.

A WS-Addressing Action header with the value http://www.ogf.org/infod/INFODNotify/Notify MUST accompany the message

INFOD Registry Response

No response is expected from the INFOD consumer upon receipt of this message.

Example SOAP Encoding of the Notify Message

The following is a non-normative example of a Notify request message using SOAP:

```
<s:Envelope ... >
  <s:Header>
    <wsa:Action>
      http://www.ogf.org/infod/INFODNotify/Notify
    </wsa:Action>
    ...
  </s:Header>
  <s:Body>
    <infod:Notify>
      <infod:NotificationMessage>
        <infod:SubscriptionReference>
          <wsa:Address>
            http://www.example.org/SomeSubscripiton
          </wsa:Address>
        </infod:SubscriptionReference>
        <infod:Topic Dialect=
     "http://www.myinfodregistry.com/infod/MyDataVocabularyEPR">
          infod:DatavocabularyEPR
        </infod:Topic>
        <infod:ProducerReference>
          <wsa:Address>
            http://www.example.org/Publisher
          </wsa:Address>
        </infod:ProducerReference>
        <infod:Message>
         <MyDataVocabulary:MessageContent>MessageDataContent</MyDataVocabul
ary:MessageContent>
        </infod:Message>
      </infod:NotificationMessage>
    </infod:Notify>

  </s:Body>
</s:Envelope>
```

3.2 Notification from the Registry

The registry sends notifications to those publishers, subscribers and consumers that have registered for them. Changes of state within the registry lead to generation of events. The specifics of the payload and the condition under which a notification MUST be sent are described in the following section:

- Notification of publishers (section 3.2.1)
- Notification of subscribers (section 3.2.2)
- Notification of consumers (section 3.2.3)

3.2.1 Notification of Publishers

The INFOD registry will inform publishers that need to react to changes in the INFOD registry. The notification is conditional on the information in the publisher entry. Publishers SHOULD react immediately to these notifications.

A new publisher MUST be informed about each subscription that requires[4] this publisher to send messages; there will be one notification per subscription.

For existing publishers notifications MUST be sent about those subscriptions that mandate different messages or mandate messages to be sent to different consumers. An empty list of static and dynamic consumers indicates that a publisher MUST stop publishing for the referenced subscription

Notifications are determined by processing the property constraints and the vocabulary reference in the data constraints.

The notification contains the following message body:

```
<infod:PublisherNotification>
  <infod:SubscriptionReference>
    wsa:EndPointReferenceType
  </infod:SubscriptionReference>
  <infod:ConsumerEntryReference>
    wsa:EndPointReferenceType
  <infod:ConsumerEntryReference> *
  <infod:DynamicConsumerConstraint>
    {xsd:anyType}
  <infod:DynamicConsumerConstraint> *
  <infod:DataConstraint>
    {xsd:anyType}
  <infod:DataConstraint> *
<infod:PublisherNotification>
```

The message content is further described as follows:

/infod:SubscriptionReference

This is the EPR of the subscription for which the information is provided.

If all other parameters are omitted the publisher does not need to process this subscription any longer. This EPR is not valid after the subscription is dropped. However, the no longer valid EPR is propagated, as some of the publishers may be using the EPR for their internal references.

[4] Static and dynamic constraints are evaluated to determine if and whether the event notification should be propagated to the recipient.

/infod:ConsumerEntryReference

This is a list of 0 to n EPR references of consumer entries. The list of consumers is computed by the INFOD Registry and given to each publisher.

/infod:DynamicConsumerConstraint

This is an expression that directs the publisher to determine the consumer(s) based on the listed expressions. Each expression references data that are created by the publishers, e.g. messages to be published, and references properties of INFOD Registry resources.

The subscription should be discarded if there is no entry for StaticConsumers and for DynamicConsumerConstraint.

/infod:DataConstraint

These are the data constraints as specified in the referenced subscription.

WS-Addressing of the action MUST contain the URI
http://www.ogf.org/infod/INFODNotify/SubscriptionNotification.

3.2.2 Notification of Subscribers

The INFOD registry MUST inform subscriber that need to know the impact of changes in the INFOD registry on their subscriptions; e.g., subscription with an EPR pointing to them. The notification is conditional on the information in the subscriber entry.

In reaction to a newly created or replaced subscription the subscriber MUST be informed which publishers send and consumers receive messages based on that subscription.

In reaction to any other change in the INFOD registry the subscriber MUST be informed about those subscription for which the list of publishers or consumers has changed.

Notifications are determined by processing the property constraints and the vocabulary reference in the data constraints.

The notification contains the following message body:

```
<infod:SubscriberNotification.
<infodSubscriptionReference>
  wsa:endPointReferenceType
</infodSubscriptionReference>
<infod:PublisherEntryReference>
  wsa:endPointReferenceType
</infod:PublisherEntryReference> *
<infod:ConsumerEntryReference>
  wsa:endPointReferenceType
</infod:ConsumerEntryReference> *
<infod:SubscriberNotification
```

The message content is further described as follows:

/infod:SubscriptionReference

This is the EPR of the subscription for which the information is provided

/infod:PublisherEntryReference

This is a list of 0 to n EPR references of publisher entries. The list of publisher entries is computed by the INFOD registry.

Infod:ConsumerEntryReference

1830 This is a list of 0 to n references to static consumers. The list of consumer entries is computed
1831 by the INFOD Registry.

1832 WS-Addressing of the action MUST contain the URI
1833 http://www.ogf.org/infod/INFODNotify/SubscriptionNotification.

1834 3.2.3 Notification of Consumers

1835 The INFOD registry will inform consumers that need to know about changes in the INFOD registry that
1836 result in different messages being received or different publishers sending messages. The notification
1837 is conditional on the information in the consumer entry.

1838 A new consumer MUST be informed about those subscriptions that result in messages being send to
1839 this consumer.

1840 An existing consumer MUST be informed about any change in the INFOD registry that adds or
1841 removes subscriptions applying to this consumer. The consumer MUST also be notified if the list of
1842 publishers of a subscription, already referenced in previous notification to that consumer, has
1843 changed.

1844 Notifications are determined by processing the property constraints and the vocabulary reference in
1845 the data constraints.

1846 The notification will not be send to dynamic consumers.

1847 The notification contains the following message body:

```
1848 <infod:ConsumerNotification.
1849 <infodSubscriptionReference>
1850   wsa:endPointReferenceType
1851 </infodSubscriptionReference>
1852 <infod:PublisherEntryReference>
1853   wsa:endPointReferenceType
1854 </infod:PublisherEntryReference> *
1855 <infod:ConsumerNotification
```

1856 The message content is further described as follows:

1857 /infod:SubscriptionReference

1858 This is the EPR of the subscription for which the information is provided

1859 /infod:PublisherEntryReference

1860 This is a list of 0 to n EPR references of publisher entries. The list of publisher entries is
1861 computed by the INFOD registry.

1862 WS-Addressing of the action MUST contain the URI
1863 http://www.ogf.org/infod/INFODNotify/SubscriptionNotification.

4 Security Considerations

An INFOD operating environment consists of a set of publishers, consumers and registries. All the above service components operate in different security domains and require “long-term” secure communication of messages. Additionally, as the INFOD services operate in a web services environment, SOAP may be used as the base communication protocol. SOAP based communication between services can be secured by using the mechanisms described by the *WS security* specification (see http://www.oasis-open.org/committees/download.php/5531/oasis-200401-wsssoap-message-security-1.0.pdf). Although, the use of WS-Security provides the mechanisms to accommodate multiple security tokens and encryption technologies, it remains limited to providing a secured point-to-point communication mechanism on a message level. However, INFOD services need to build upon this security mechanism to describe the security context under which they could sustain long running exchanges of messages. A communication session between the two parties such as publisher and consumer serves as the basis for establishing the security context. Establishing a security context between system entries allows secured messaging on the session level and reduces the synchronization overheads required to obtain it on per-message basis. *WS-Secure Conversation* (see ftp://www6.software.ibm.com/software/developer/library/ws-secureconversation.pdf) provides the mechanism for maintaining such long-term contexts for message exchange.

The INFOD model **RECOMMENDS** the establishment of the following contexts:

- Publisher – Registry secured context, with Registry as the context security token creator.
- Consumer – Registry secured context, with Registry as the context security token creator.
- Subscriber – Registry secured context, with Registry as the context security token creator.
- Publisher – Consumer secured context, with Publisher as the context security token creator. It may be possible to support registry mediated delegation, where the registry mediates the establishment of trust between producer and consumer.

Authentication remains a crucial aspect of formation of a secured conversation. Hence, the specification identifies the objects that create the secured context. It is envisaged that an INFOD-Registry will provide services to multiple publishers/consumers/subscriptions and controls the access to this shared state. Hence, it is imperative to have the INFOD-Registry act as the authenticator for other services. Similarly, a publisher controls the dissemination of the messages and hence is deemed responsible for establishing the context with the consumers. In the future, it is envisaged that in later versions INFOD may introduce mechanisms for mutual authentication based on trust mechanisms. An example, is that future authentication of consumers by the publishers could be mediated by the registry.

4.1 Message Encryption and Data Privacy Requirements

INFOD advocates the use of mutual filtering techniques to provide smart dissemination of the messages. Mutual filtering requires the publishers and consumers to be able to interpret the contents of the messages being routed. As INFOD isolates a publisher from a consumer and does not require either the publishers or the consumers to authenticate each other, secured point-to-point communication becomes a non-issue for the base specification. It is assumed that publishers are able to authenticate the consumers based on their EPR references.

INFOD system provides non-repudiation of transmitted messages. It is recommended that the publisher signs its message and also provides its public key for subsequent verification by the recipients. It is suggested that the public key of each publisher is registered with the INFOD registry

1907 for retrieval by the network entities, such a public key should be registered with the
1908 PropertyVocabulary.

1909 In some cases, INFOD publishers can determine the list of consumers and can provide messages for
1910 consumption by a single consumer or a group of consumers. No present security mechanism supports
1911 such communication pattern without the establishment of a shared key between the group of
1912 consumers and the publisher.

1913 4.2 Integration with Authorization Model

1914 Access control mechanisms for management of resources rely on the authentication mechanisms to
1915 authorize the access to the resources. Only authorized principals are allowed to register the
1916 publishers publish messages, create and manage the subscription and manage the consumers. It is
1917 recommended that the authorization model should provide a fine-grained control, preferably at the
1918 level of the evaluation context/ topics. Authorization models can be divided into two categories:

1919 • Access model for INFOD resources

1920 • Access model for INFOD messages

1921 Access models for the INFOD resources enforce the policies to allow restricted access to creation,
1922 deletion, and invocation of methods on service interfaces. Access models for resources can be
1923 maintained individually by each of the INFOD services as they are directly associated with the state
1924 maintained by the service. For example, an access model of INFOD registry resources controls the
1925 process of registering a publication and remains solely responsible for enforcing the related access
1926 policies.

1927 Access model for INFOD messages allows association of the dynamic authorization policies that
1928 control the access to the contents and the routing of the messages. Candidate examples include a
1929 publisher restricting dissemination of messages to a restricted list of consumers. Dynamic
1930 authorization policies may be propagated as a part of the secured conversation context and will need
1931 to be enforced by each participant that shares the context.

5 Appendix I – XML Schema

This section includes the following xml schema:

- Publisher Entry (section 5.1)
- Subscriber Entry (section 5.2)
- Consumer Entry (section 5.3)
- Subscription (section 5.4)
- Property Vocabulary (section 5.5)
- Property Vocabulary Instance (section 5.6)
- Data Vocabulary (section 5.7)
- Data Source Entry (section 5.8)
- INFOD Error Messages (section 5.9)
- INFOD Notification (section 5.10)
- Publishers Notification (section 5.11)
- Subscriber Notification (section 5.12)
- Consumer Notification (section 5.13)

The following graphic depicts the XML schema relations for the www.ogf.org/infod/INFODRegistry Namespace. The circles in the first row represent the operations of the INFOD registry. The circles in the second row show the vocabularies that are managed by the vocabulary operations. The boxes in the third row represent the resources, data entries and property vocabulary instances. Within the xml schema of those boxes, there are reference pointers to other entries or vocabularies, represented by EPRs. The honeycombs represent the external web services EPRs that are associated to the resources. Note that the same Web Service EPR can be associated to multiple INFOD resources.

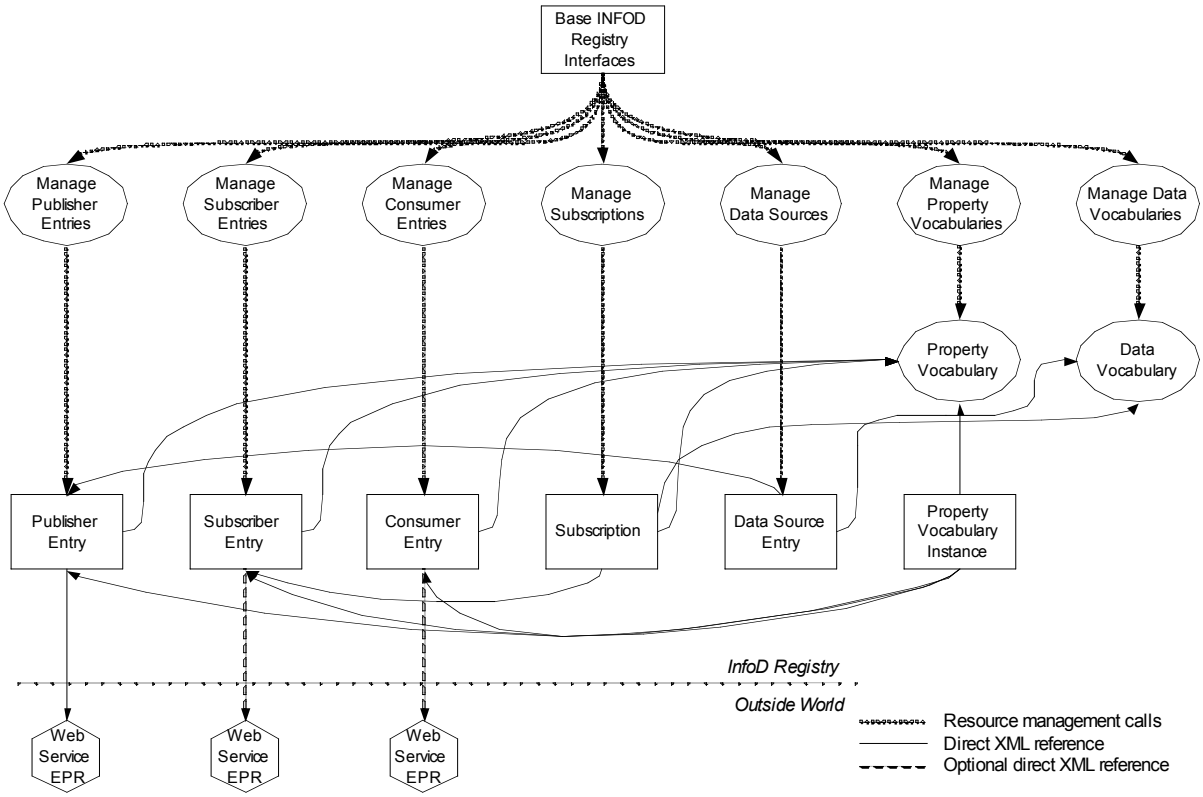

Figure 6: XML schema relations of INFODRegistry namespace

Figure 6, provides a schematic representation of internals of INFOD registry. Web Service EPRs from publisher, subscriber and consumers are associated with publisher, subscriber and consumer entries respectively. Associations between entries with data and property vocabulary instances are also highlighted.

5.1 Publisher Entry

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:ident="http://www.ogf.org/infod"
            targetNamespace="http://www.ogf.org/infod/INFODRegistry">

   <xsd:element name="infodPublisherEntry">
     <xsd:annotation>
         <xsd:documentation>
             Description of Publisher Entries
         </xsd:documentation>
     </xsd:annotation>
     <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="WSReference"
                                    type="wsa:EndpointReferenceType"
                                    minOccurs="0" maxOccurs="1"/>
          <xsd:element name="PublisherName" type="xsd:string"
                                    minOccurs="0" maxOccurs="1"/>
          <xsd:element name="PublisherDescription" type="xsd:string"
                                    minOccurs="0" maxOccurs="1"/>
          <xsd:element name="PropertyConstraint" type="xsd:any"
```

```
                                    minOccurs="0" maxOccurs="unbounded"/>
          <xsd:element name="Notification" type="xsd:boolean"
                                    nillable="true"
                                    minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        </xsd:complexType>
     </xsd:element>
</xsd:schema>
```

5.2 Subscriber Entry

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:ident="http://www.ogf.org/infod"
            targetNamespace="http://www.ogf.org/infod/INFODRegistry">

  <xsd:element name="infodSubscriberEntry">
    <xsd:annotation>
      <xsd:documentation>
        Description of Subscriber Entries
      </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="WSReference"
          type="wsa:EndpointReferenceType"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="SubscriberName" type="xsd:string"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="SubscriberDescription" type="xsd:string"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="PropertyConstraint" type="xsd:any"
          minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="Notification" type="xsd:boolean"
          nillable="true" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
   </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

5.3 Consumer Entry

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:ident="http://www.ogf.org/infod"
            targetNamespace="http://www.ogf.org/infod/INFODRegistry">

  <xsd:element name="infodConsumerEntry">
    <xsd:annotation>
      <xsd:documentation>
        Description of Consumer Entries
      </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="WSReference"
          type="wsa:EndpointReferenceType"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="ConsumerrName" type="xsd:string"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="ConsumerDescription" type="xsd:string"
```

```
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="PropertyConstraint" type="xsd:any"
          minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="Notification" type="xsd:boolean"
          nillable="true" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

5.4 Subscription

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                xmlns:ident="http://www.ogf.org/infod"
                targetNamespace="http://www.ogf.org/infod/INFODRegistry">

  <xsd:element name="infodSubscription">
    <xsd:annotation>
      <xsd:documentation>
        Description of Subscriptions
      </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="SubscriptionName" type="xsd:string"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="SubscriptionDescription" type="xsd:string"
          minOccurs="1" maxOccurs="1"/>
        <xsd:element name="SubscriberEntryReference"
          type="wsa:EndpointReferenceType"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="DataConstraint" type="xsd:any"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="PropertyConstraint" type="xsd:any"
          minOccurs="0" maxOccurs="unbounded"/>
     <xsd:element name="DynamicConsumerConstraint" type="xsd:any"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

5.5 Property Vocabulary

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                xmlns:ident="http://www.ogf.org/infod"
                targetNamespace="http://www.ogf.org/infod/INFODRegistry">

  <xsd:element name="infodPropertyVocabulary">
    <xsd:annotation>
      <xsd:documentation>
        Description of a Property Vocabulary
      </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="PropertyVocabularyName" type="xsd:string"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="PropertyVocabularyDescription" type="xsd:string"
```

```
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="PropertyVocabularyBody" type="xsd:any"
          minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

5.6 Property Vocabulary Instance

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                xmlns:ident="http://www.ogf.org/infod"
                targetNamespace="http://www.ogf.org/infod/INFODRegistry">

  <xsd:element name="infodPropertyVocabularyInstance">
    <xsd:annotation>
      <xsd:documentation>
        Description of Property Vocabulary Instance
      </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="EntryReference"
          type="wsa:EndpointReferenceType"
          minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PropertyVocabularyReference"
          type="wsa:EndpointReferenceType"
          minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PropertyVocabularyInstanceBody"
       type="xsd:schema"
          minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

5.7 Data Vocabulary

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                xmlns:ident="http://www.ogf.org/infod"
                targetNamespace="http://www.ogf.org/infod/INFODRegistry">

  <xsd:element name="infodDataVocabulary">
    <xsd:annotation>
      <xsd:documentation>
        Description of Data Vocabulary
      </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="DataVocabularyName" type="xsd:string"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="DataVocabularyDescription" type="xsd:string"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="DataVcabularyLanguage" type="xsd:string"
          minOccurs="1" maxOccurs="1"/>
        <xsd:element name="DataVocabularyBody" type="xsd:any"
          minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
```

```
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

5.8 Data Source Entry

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
               xmlns:ident="http://www.ogf.org/infod"
               targetNamespace="http://www.ogf.org/infod/INFODRegistry">

   <xsd:element name="infodDataSourceEntry">
    <xsd:annotation>
      <xsd:documentation>
        Description of Data Source Entries
      </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="CreateDataSourceEntryName" type="xsd:string"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="DataSourceEntryDescription"
          type="xsd:string"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="PublisherEntryReference"
          type="wsa:EndpointReferenceType"
          minOccurs="1" maxOccurs="1"/>
        <xsd:element name="DataVocabularyReference"
          type="wsa:EndpointReferenceType"
          minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PropertyConstraint" type="xsd:any"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

5.9 Error Messages

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:wsrf-bf="http://www.ogf.org/infod/fault"
            xmlns:ident="http://www.ogf.org/infod"
            targetNamespace="http://www.ogf.org/infod/INFODRegistry">

  <xsd:complexType name="CreateResourceAuthorizationFaultType">
    <xsd:complexContent>
      <xsd:extension base="wsrf-bf:BaseFaultType"/>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:element name="CreateResourceAuthorizationFault"
              type="infod:CreateResourceAuthorizationFaultType"/>

  <xsd:complexType name="ReplaceResourceAuthorizationFaultType">
    <xsd:complexContent>
      <xsd:extension base="wsrf-bf:BaseFaultType"/>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:element name="ReplaceResourceAuthorizationFault"
              type="infod:ReplaceResourceAuthorizationFaultType"/>

```

```
<xsd:complexType name="DropResourceAuthorizationFailureType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="DropResourceAuthorizationFailure"
             type="infod:DropResourceAuthorizationFailureType"/

<xsd:complexType name="ExecutionModeFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="ExecutionModeFault"
             type="infod:ExecutionModeFaultType"/>

<xsd:complexType name="UnSupportedVocabularyFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="UnSupportedVocabularyFault"
             type="infod:UnSupportedVocabularyFaultType"/>

<xsd:complexType name="UnsupportedXQueryFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="UnsupportedXQueryFault"
             type="infod:UnsupportedXQueryFaultType"/>

<xsd:complexType name="GetMetaDataAuthorizationFailureType">
  <xsd:complexContent>
   <xsd:extension base="wsrf-bf:BaseFaultType"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="GetMetaDataAuthorizationFailure"
             type="infod:GetMetaDataAuthorizationFailureType"/>

<xsd:complexType name="UnknownResourceReferenceFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="UnknownResourceReferenceFault"
             type="infod:UnknownElementReferenceFaultType"/>

<xsd:complexType name="MissingRequiredParameterFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="MissingRequiredParameterFault"
             type="infod:MissingRequiredParameterFaultType"/>

<xsd:complexType name="UnknownFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType"/>
  </xsd:complexContent>
</xsd:complexType>
```

```
  <xsd:element name="UnknownFault"
               type="infod:UnknownFaultType"/>

</xsd:schema>
```

5.10 INFOD Notification

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:ident="http://www.ogf.org/infod"
            targetNamespace="http://www.ogf.org/infod/INFODNotify">

<!-- ================ Notification Metadata  ==================== -->

  <xsd:element name="SubscriptionReference"
               type="wsa:EndpointReferenceType"/>
  <xsd:element name="Topic"
               type="infod:TopicExpressionType"/>
  <xsd:element name="PublisherReference"
               type="wsa:EndpointReferenceType"/>

<!-- ================= Message Helper Types  ==================== -->

  <xsd:complexType name="TopicExpressionType" mixed="true">
    <xsd:sequence>
      <xsd:any minOccurs="0" maxOccurs="1" processContents="lax"/>
    </xsd:sequence>
    <xsd:attribute name="Dialect" type="xsd:anyURI" use="required"/>
    <xsd:anyAttribute/>
  </xsd:complexType>

  <xsd:complexType name="NotificationMessageHolderType">
    <xsd:sequence>
      <xsd:element ref="infod:SubscriptionReference"
                   minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="infod:Topic"
                   minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="infod:PublisherReference"
                   minOccurs="0" maxOccurs="1"/>
      <xsd:element name="Message">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:any namespace="##any" processContents="lax"
                     minOccurs="1" maxOccurs="1"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="NotificationMessage"
               type="infod:NotificationMessageHolderType"/>

<!-- ==== Message Types for Consumer Notification by Publishers ===== -->

  <xsd:element name="Notify">
    <xsd:annotation>
       <xsd:documentation> Notification of Consumers by Publishers
       </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
```

```
        <xsd:element ref="infod:NotificationMessage"
                     minOccurs="1" maxOccurs="unbounded"/>
        <xsd:any namespace="##other" processContents="lax"
                     minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

</xsd:schema>
```

5.11 INFOD Publisher Notification

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:ident="http://www.ogf.org/infod"
            targetNamespace="http://www.ogf.org/infod/INFODNotify">

  <xsd:element name="PublisherNotification">
    <xsd:annotation>
       <xsd:documentation>
           Notification of Publishers
       </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="SubscriptionReference"
                     type="wsa:EndpointReferenceType"
                     minOccurs="1" maxOccurs="1"/>
          <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:sequence>
                <xsd:element name="ConsumerEntryReference"
                       type="xsd:any"
                       minOccurs="1" maxOccurs="unbounded"/>
                   <xsd:element name="DynamicConsumerConstraint"
                      type="xsd:any"
                       minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:sequence>
                <xsd:element name="ConsumerEntryReference"
                       type="xsd:any"
                       minOccurs="0" maxOccurs="unbounded"/>
                   <xsd:element name="DynamicConsumerConstraint"
                      type="xsd:any"
                       minOccurs="1" maxOccurs="unbounded"/>
            </xsd:sequence>
         </xsd:choice>
       <xsd:element name="DataConstraint"
                     type="xsd:any"
                     minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

5.12 INFOD Subscriber Notification

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:ident="http://www.ogf.org/infod"
            targetNamespace="http://www.ogf.org/infod/INFODNotify">

```

```
  <xsd:element name="SubscriberNotification">
    <xsd:annotation>
       <xsd:documentation>
           Notification of Subscribers
       </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="SubscriptionReference"
                     type="wsa:EndpointReferenceType"
                     minOccurs="1" maxOccurs="1"/>
           <xsd:element name="ConsumerEntryReference"
                       type="wsa:EndpointReferenceType"
                       minOccurs="0" maxOccurs="unbounded"/>
           <xsd:element name="PublisherEntryReference"
                       type="wsa:EndpointReferenceType"
                       minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

5.13 INFOD Consumer Notification

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:ident="http://www.ogf.org/infod"
            targetNamespace="http://www.ogf.org/infod/INFODNotify">

  <xsd:element name="ConsumerNotification">
    <xsd:annotation>
       <xsd:documentation>
           Notification of Consumers
       </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="SubscriptionReference"
                     type="wsa:EndpointReferenceType"
                     minOccurs="1" maxOccurs="1"/>
           <xsd:element name="PublisherEntryReference"
                       type="wsa:EndpointReferenceType"
                       minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

6 Appendix II – WSDL 1.1

```
<wsdl:definitions name="infodBaseNotification"
targetNamespace="http://www.ggf.org/INFOD"
xmlns:tns="http://www.ggf.org/INFOD"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsrf-rw="http://docs.oasis-open.org/wsrf/rw-2"
xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:infod="http://www.ggf.org/INFOD"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">

<wsdl:import namespace="http://docs.oasis-open.org/wsrf/rw-2"
location="http://docs.oasis-open.org/wsrf/rw-2.wsdl" />


<wsdl:types>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
targetNamespace="http://www.ggf.org/INFOD"
xmlns:infodxsd="http://www.ggf.org/INFOD/infodTypes.xsd">
<xsd:complexType name="Notification">
<xsd:simpleContent>
<xsd:extension base="xsd:boolean"></xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:element name="CreatePublisherEntryMissingRequiredParameterFault"
type="xsd:string"></xsd:element>
<xsd:element name="DropPublisherEntryExecutionModeFault"
type="xsd:string"></xsd:element>
<xsd:element name="DropSubscriberEntryDropResourceAuthorizationFailure"
      type="xsd:string"></xsd:element>
<xsd:element name="CreateConsumerEntryMissingRequiredParameterFault"
type="xsd:string"></xsd:element>
<xsd:element name="DropConsumerEntryMissingRequiredParameterFault"
type="xsd:string"></xsd:element>
<xsd:element name="CreateSubscriptionMissingRequiredParameterFault"
      type="xsd:string"></xsd:element>
<xsd:element name="DropSubscriptionMissigRequiredParameterFault"
type="xsd:string"></xsd:element>
</xsd:schema>
<xsd:schema>

<xsd:import namespace="http://docs.oasis-open.org/wsn/b-2"
schemaLocation="http://docs.oasis-open.org/wsn/b-2.xsd"     />
<xsd:import namespace="http://www.ggf.org/INFOD"
schemaLocation="infodTypes.xsd">
</xsd:import>
</xsd:schema>

</wsdl:types>

```

```
<wsdl:message name="CreatePublisherEntryRequest">
<wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
<wsdl:part name="PublisherName"     type="xsd:string"></wsdl:part>
<wsdl:part name="PublisherDescription" type="xsd:string"></wsdl:part>
<wsdl:part name="PropertyConstraints"
element="infod:PropertyConstraints">
</wsdl:part>
<wsdl:part name="Notification" type="xsd:boolean"></wsdl:part>
</wsdl:message>

<wsdl:message name="CreatePublisherEntryResponse">
<wsdl:part name="INFODResourceReference"
element="infod:EndPointReference">
</wsdl:part>
</wsdl:message>


<wsdl:message name="ReplacePublisherEntryRequest">
<wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
<wsdl:part name="PublisherEntryReference"
element="infod:EndPointReference">
</wsdl:part>
<wsdl:part name="PublisherName"     type="xsd:string"></wsdl:part>
<wsdl:part name="PublisherDescription" type="xsd:string"></wsdl:part>
<wsdl:part name="PropertyConstraints"
element="wsinfod:PropertyConstraints">
</wsdl:part>
<wsdl:part name="Notification" type="xsd:boolean"></wsdl:part>
</wsdl:message>

<wsdl:message name="ReplacePublisherEntryResponse">
<wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
</wsdl:message>

<wsdl:message name="DropPublisherEntryRequest">
<wsdl:part name="PublisherEntryReference"
element="infod:EndPointReference">
</wsdl:part>
<wsdl:part name="ExecutionMode"
      element="infod:ExecutionMode"></wsdl:part>
</wsdl:message>

<wsdl:message name="DropPublisherEntryResponse">
<wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
</wsdl:message>

<wsdl:message name="CreateSubscriberEntryRequest">
<wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
<wsdl:part name="SubscriberName" type="xsd:string"></wsdl:part>
<wsdl:part name="SubscriberDescription"   type="xsd:string"></wsdl:part>
<wsdl:part name="PropertyConstraints"
element="infod:PropertyConstraints">
</wsdl:part>
<wsdl:part name="Notification" type="xsd:boolean"></wsdl:part>
```

```
</wsdl:message>

<wsdl:message name="CreateSubscriberEntryResponse">
<wsdl:part name="INFODResourceReference"
element="infod:EndPointReference">
</wsdl:part>
</wsdl:message>

<wsdl:message name="ReplaceSubscriberEntryRequest">
<wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
<wsdl:part name="SubscriberEntryReference"
element="infod:EndPointReference">
</wsdl:part>
<wsdl:part name="SubscriberName" type="xsd:string"></wsdl:part>
<wsdl:part name="SubscriberDescription"   type="xsd:string"></wsdl:part>
<wsdl:part name="PropertyConstraints"
element="infod:PropertyConstraints">
</wsdl:part>
<wsdl:part name="Notification" type="xsd:boolean"></wsdl:part>
</wsdl:message>

<wsdl:message name="ReplaceSubscriberEntryResponse">
<wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
</wsdl:message>

<wsdl:message name="DropSubscriberEntryRequest">
<wsdl:part name="INFODResourceReference"
element="infod:EndPointReference">
</wsdl:part>
<wsdl:part name="ExecutionMode"
      element="infod:ExecutionMode"></wsdl:part>
</wsdl:message>

<wsdl:message name="DropSubscriberEntryResponse">
<wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
</wsdl:message>


<wsdl:message name="CreateConsumerEntryRequest">
<wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
<wsdl:part name="INFODResourceReference"
element="infod:EndPointReference">
</wsdl:part>
<wsdl:part name="ConsumerName" type="xsd:string"></wsdl:part>
<wsdl:part name="ConsumerDescription" type="xsd:string"></wsdl:part>
<wsdl:part name="PropertyConstraints"
element="infod:PropertyConstraints">
</wsdl:part>
<wsdl:part name="Notification" type="xsd:boolean"></wsdl:part>
</wsdl:message>

<wsdl:message name="CreateConsumerEntryResponse">
<wsdl:part name="INFODResourceReference"
element="infod:EndPointReference">
</wsdl:part>
```

```
</wsdl:message>

<wsdl:message name="ReplaceConsumerEntryRequest">
<wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
<wsdl:part name="INFODResourceReference"
element="infod:EndPointReference">
</wsdl:part>
<wsdl:part name="ConsumerName" type="xsd:string"></wsdl:part>
<wsdl:part name="ConsumerDescription" type="xsd:string"></wsdl:part>
<wsdl:part name="PropertyConstraints"
element="infod:PropertyConstraints">
</wsdl:part>
<wsdl:part name="Notification" type="xsd:boolean"></wsdl:part>
</wsdl:message>

<wsdl:message name="ReplaceConsumerEntryResponse">
<wsdl:part name="Status" element="infod:status"></wsdl:part>
</wsdl:message>

<wsdl:message name="DropConsumerEntryRequest">
<wsdl:part name="INFODResourceReference"
element="infod:EndPointReference">
</wsdl:part>
<wsdl:part name="ExecutionMode"
      element="infod:ExecutionMode"></wsdl:part>
</wsdl:message>

<wsdl:message name="DropConsumerEntryResponse">
<wsdl:part name="Status" element="infod:status"></wsdl:part>
</wsdl:message>

<wsdl:message name="CreateSubscriptionRequest">
<wsdl:part name="SubscriptionName" type="xsd:string"></wsdl:part>
<wsdl:part name="SubscriptionDescription" type="xsd:string"></wsdl:part>
<wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
<wsdl:part name="DataConstraints"
element="infod:DataConstraints">
</wsdl:part>
<wsdl:part name="PropertyConstraints"
element="infod:PropertyConstraints">
</wsdl:part>
<wsdl:part name="DynamicConsumerConstraints"
element="infod:DynamicConsumerConstraints">
</wsdl:part>
</wsdl:message>

<wsdl:message name="CreateSubscriptionResponse">
<wsdl:part name="INFODResourceReference"
element="wsa:EndPointReference">
</wsdl:part>
</wsdl:message>

<wsdl:message name="ReplaceSubscriptionRequest">
<wsdl:part name="INFODResourceReference"
element="wsa:EndPointReference">
```

```
</wsdl:part>
<wsdl:part name="SubscriptionName" type="xsd:string"></wsdl:part>
<wsdl:part name="SubscriptionDescription" type="xsd:string"></wsdl:part>
<wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
<wsdl:part name="DataConstraints"
element="infod:DataConstraints">
</wsdl:part>
<wsdl:part name="PropertyConstraints"
element="infod:PropertyConstraints">
</wsdl:part>
<wsdl:part name="DynamicConsumerConstraints"
element="infod:DynamicConsumerConstraints">
</wsdl:part>
</wsdl:message>

<wsdl:message name="ReplaceSubscriptionResponse">
<wsdl:part name="Status" element="infod:status"></wsdl:part>
</wsdl:message>

<wsdl:message name="DropSubscriptionRequest">
<wsdl:part name="INFODResourceReference"
element="wsa:EndPointReference">
</wsdl:part>
<wsdl:part name="ExecutionMode"
      element="infod:ExecutionMode"></wsdl:part>
</wsdl:message>

<wsdl:message name="DropSubscriptionResponse">
<wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
</wsdl:message>

<wsdl:message name="CreatePropertyVocabularyRequest">
<wsdl:part name="VocabularyName" type="xsd:string"></wsdl:part>
<wsdl:part name="VocabularyDescription"   type="xsd:string"></wsdl:part>
<wsdl:part name="VocabularyBody" type="xsd:anyType"></wsdl:part>
</wsdl:message>

<wsdl:message name="CreatePropertyVocabularyResponse">
<wsdl:part name="INFODVocabularyReference"
element="infod:EndPointReference">
</wsdl:part>
</wsdl:message>

<wsdl:message name="CreatePropertyVocabularyInstanceRequest">
<wsdl:part name="VocabularyInstanceReference"
element="wsa:EndPointReference">
</wsdl:part>
<wsdl:part name="VocabularyInstanceVocabularyReference"
element="wsa:EndPointReference">
</wsdl:part>
<wsdl:part name="VocabularyInstanceVocabularyBody"
type="xsd:anyType">
</wsdl:part>
</wsdl:message>
```

```
<wsdl:message name="CreatePropertyVocabularyInstanceResponse">
<wsdl:part name="INFODVocabularyInstanceReference"
element="wsa:EndPointReference">
</wsdl:part>
</wsdl:message>

<wsdl:message name="DropPropertyVocabularyInstanceRequest">
<wsdl:part name="VocabularyInstanceReference"
element="wsa:EndPointReference">
</wsdl:part>
<wsdl:part name="ExecutionMode"
      element="infod:ExecutionMode"></wsdl:part>
</wsdl:message>

<wsdl:message name="DropPropertyVocabularyInstanceResponse">
<wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
</wsdl:message>


<wsdl:message name="CreateDataVocabularyRequest">
<wsdl:part name="VocabularyName" type="xsd:string"></wsdl:part>
<wsdl:part name="VocabularyDescription"   type="xsd:string"></wsdl:part>
<wsdl:part name="VocabularyLanguage" type="xsd:anyURI"></wsdl:part>
<wsdl:part name="LanguageUsageDescription" type="xsd:anyType"></wsdl:part>
<wsdl:part name="VocabularyInstanceVocabularyBody"
type="xsd:anyType">
</wsdl:part>
</wsdl:message>


<wsdl:message name="CreateDataVocabularyResponse">
<wsdl:part name="DataVocabularyReference"
element="wsa:EndPointReference">
</wsdl:part>
</wsdl:message>

<wsdl:message name="DropDataVocabularyRequest">
<wsdl:part name="INFODVocabularyReference"
element="wsa:EndPointReference">
</wsdl:part>
<wsdl:part name="ExecutionMode"
      element="infod:ExecutionMode"></wsdl:part>
</wsdl:message>

<wsdl:message name="DropDataVocabularyResponse">
<wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
</wsdl:message>

<wsdl:message name="CreateDataSourceEntryRequest">
<wsdl:part name="CreateDataSourceEntryName"
      type="xsd:string"></wsdl:part>
<wsdl:part name="CreateDataSourceEntryDescription"
type="xsd:string">
</wsdl:part>
<wsdl:part name="AssociationEntryReference"
```

```
element="wsa:EndPointReference">
</wsdl:part>
<wsdl:part name="VocabularyReference"
element="wsa:EndPointReference">
</wsdl:part>
<wsdl:part name="PropertyConstraints"
element="infod:PropertyConstraints">
</wsdl:part>
</wsdl:message>

<wsdl:message name="CreateDataSourceEntryResponse">
<wsdl:part name="INFODAssociationReference"
element="wsa:EndPointReference">
</wsdl:part>
</wsdl:message>

<wsdl:message name="DropDataSourceEntryRequest">
<wsdl:part name="DataSourceEntryReference"
element="wsa:EndPointReference">
</wsdl:part>
<wsdl:part name="ExecutionMode"
      element="infod:ExecutionMode"></wsdl:part>
</wsdl:message>

<wsdl:message name="DropDataSourceEntryResponse">
<wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
</wsdl:message>

<wsdl:message name="GetMetaDataRequest">
<wsdl:part name="MetaDataQueryExpression" type="xsd:anyType"></wsdl:part>
</wsdl:message>

<wsdl:message name="GetMetaDataResponse">
<wsdl:part name="MetaDataQueryResult" type="xsd:anyType"></wsdl:part>
</wsdl:message>

<wsdl:message name="NotifyRequest"></wsdl:message>


<wsdl:message name="CreateResourceAuthorizationFailureErrorMessage">
<wsdl:part name="err"
element="infod:CreateEntryAuthorizationFailure">
</wsdl:part>
</wsdl:message>
<wsdl:message name="UnknownResourceReferenceFaultErrorMessage">
<wsdl:part name="err"
element="infod:UnknownResourceReferenceFault">
</wsdl:part>
</wsdl:message>
<wsdl:message name="MissingRequiredParameterFaultErrorMessage">
<wsdl:part name="err"
element="infod:MissingRequiredParameterFault">
</wsdl:part>
</wsdl:message>
<wsdl:message name="UnsupportedXQueryFaultErrorMesage">
```

```
<wsdl:part name="err" element="infod:UnsupportedXQueryFault"></wsdl:part>
</wsdl:message>
<wsdl:message name="ReplaceResourceAuthorizationFaultErrorMessage">
<wsdl:part name="err"
element="infod:ReplaceEntryAuthorizationFailure">
</wsdl:part>
</wsdl:message>
<wsdl:message name="DropResourceAuthorizationFaultErrorMessage">
<wsdl:part name="err"
element="infod:DropEntryAuthorizationFailure">
</wsdl:part>
</wsdl:message>
<wsdl:message name="ExecutionModeFaultErrorMessage">
<wsdl:part name="err" element="infod:ExecutionModeFault"></wsdl:part>
</wsdl:message>

<wsdl:message name="UnSupportedVocabularyFaultErrorMessage">
<wsdl:part name="err"
element="infod:UnSupportedVocabularyFault">
</wsdl:part>
</wsdl:message>

<wsdl:message
name="CreateAssociationAuthorizationFailureErrorMessage">
<wsdl:part name="err"
element="infod:CreateAssociationAuthorizationFailure">
</wsdl:part>
</wsdl:message>
<wsdl:message name="DropPropertyVocabularyResponse">
<wsdl:part name="DropPropertyVocabularyResponse"
type="xsd:string">
</wsdl:part>
</wsdl:message>
<wsdl:message name="DropPropertyVocabularyRequest">
<wsdl:part name="PropertyVocabularyReference"
element="wsa:EndPointRefernceType"></wsdl:part>
<wsdl:part name="ExecutionMode"     element="infod:ExecutionMode">
</wsdl:part>
<wsdl:part name="ExecutionMode"     element="infod:ExecutionMode">
</wsdl:part>
</wsdl:message>
<wsdl:portType name="infodRegistry">
<wsdl:operation   name="CreatePublisherEntry">
<wsdl:input message="tns:CreatePublisherEntryRequest"
name="createPublisherMessage">
</wsdl:input>
<wsdl:output message="tns:CreatePublisherEntryResponse"></wsdl:output>
<wsdl:fault
message="tns:CreateResourceAuthorizationFailureErrorMessage"
name="CreateResourceAuthorizationFault">
</wsdl:fault>
<wsdl:fault
message="infod:MissingRequiredParameterFaultErrorMessage"
name="UnsupportedXQueryFault">
</wsdl:fault>
```

```
<wsdl:fault
message="infod:UnsupportedXQueryFaultErrorMesage"
name="UnsupportedXQueryFault">
</wsdl:fault>
<wsdl:fault name="MissingRequiredParameterFault"
message="tns:MissingRequiredParameterFaultErrorMessage">
</wsdl:fault>
</wsdl:operation>

<wsdl:operation   name="ReplacePublisherEntry">
<wsdl:input message="tns:ReplacePublisherEntryRequest"
name="ReplacePublisherMessage">
</wsdl:input>
<wsdl:output message="tns:ReplacePublisherEntryResponse"
name="ReplacePublisherResponse">
</wsdl:output>
<wsdl:fault
message="infod:ReplaceResourceAuthorizationFaultErrorMessage"
name="ReplaceResourceAuthorizationFault">
</wsdl:fault>
<wsdl:fault
message="tns:UnknownResourceReferenceFaultErrorMessage"
name="UnknownResourceReferenceFault">
</wsdl:fault>
<wsdl:fault
message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingrequiredParameterFault">
</wsdl:fault>
<wsdl:fault
message="infod:UnsupportedXQueryFaultErrorMesage"
name="UnsupportedXQueryFault">
</wsdl:fault>
</wsdl:operation>

<wsdl:operation   name="DropPublisherEntry">
<wsdl:input message="tns:DropPublisherEntryRequest"
name="DropPublisherRequestMessage">
</wsdl:input>
<wsdl:output message="tns:DropPublisherEntryResponse"
name="DropPublisherResponseMessage">
</wsdl:output>
<wsdl:fault
message="infod:DropResourceAuthorizationFaultErrorMessage"
name="DropResourceAuthorizationFailure">
</wsdl:fault>
<wsdl:fault
message="tns:UnknownResourceReferenceFaultErrorMessage"
name="UnknownResourceReferenceFault">
</wsdl:fault>
<wsdl:fault message="infod:ExecutionModeFaultErrorMessage"
name="ExecutionModeFault">
</wsdl:fault>
<wsdl:fault name="ExecutionModeFault"
message="tns:ExecutionModeFaultErrorMessage">
</wsdl:fault>
```

```
</wsdl:operation>

<wsdl:operation   name="CreateSubscriberEntry">
<wsdl:input message="tns:CreateSubscriberEntryRequest"></wsdl:input>
<wsdl:output message="tns:CreateSubscriberEntryResponse"></wsdl:output>
<wsdl:fault
message="tns:CreateResourceAuthorizationFailureErrorMessage"
name="CreateResourceAuthorizationFailure">
</wsdl:fault>
<wsdl:fault
message="tns:UnknownResourceReferenceFaultErrorMessage"
name="UnknownResourceReferenceFault">
</wsdl:fault>
<wsdl:fault
message="infod:MissingRequiredParameterFaultErrorMessage"
name="UnsupportedXQueryFault">
</wsdl:fault>
<wsdl:fault
message="infod:UnsupportedXQueryFaultErrorMesage"
name="UnsupportedXQueryFault">
</wsdl:fault>
</wsdl:operation>

<wsdl:operation   name="ReplaceSubscriberEntry">
<wsdl:input message="tns:ReplaceSubscriberEntryRequest"></wsdl:input>
<wsdl:output message="tns:ReplaceSubscriberEntryResponse"></wsdl:output>
<wsdl:fault
message="infod:ReplaceResourceAuthorizationFaultErrorMessage"
name="ReplaceResourceAuthorizationFault">
</wsdl:fault>
<wsdl:fault
message="tns:UnknownResourceReferenceFaultErrorMessage"
name="UnknownResourceReferenceFault">
</wsdl:fault>
<wsdl:fault
message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingrequiredParameterFault">
</wsdl:fault>
<wsdl:fault
message="infod:UnsupportedXQueryFaultErrorMesage"
name="UnsupportedXQueryFault">
</wsdl:fault>
</wsdl:operation>

<wsdl:operation   name="DropSubscriberEntry">
<wsdl:input message="tns:DropSubscriberEntryRequest"></wsdl:input>
<wsdl:output message="tns:DropSubscriberEntryResponse"></wsdl:output>
<wsdl:fault
message="infod:DropResourceAuthorizationFaultErrorMessage"
name="DropEntryAuthorizationFailure">
</wsdl:fault>
<wsdl:fault
message="tns:UnknownResourceReferenceFaultErrorMessage"
name="UnknownResourceReferenceFault">
</wsdl:fault>
```

```
<wsdl:fault message="infod:ExecutionModeFaultErrorMessage"
name="ExecutionModeFault">
</wsdl:fault>
</wsdl:operation>

<wsdl:operation   name="CreateConsumerEntry">
<wsdl:input message="tns:CreateConsumerEntryRequest"></wsdl:input>
<wsdl:output message="tns:CreateConsumerEntryResponse"></wsdl:output>
<wsdl:fault
message="tns:CreateResourceAuthorizationFailureErrorMessage"
name="CreateResourceAuthorizationFailure">
</wsdl:fault>
<wsdl:fault
message="infod:MissingRequiredParameterFaultErrorMessage"
name="UnsupportedXQueryFault">
</wsdl:fault>
<wsdl:fault
message="infod:UnsupportedXQueryFaultErrorMesage"
name="UnsupportedXQueryFault">
</wsdl:fault>
<wsdl:fault name="MissingRequiredParameterFault"
message="tns:MissingRequiredParameterFaultErrorMessage">
</wsdl:fault>
</wsdl:operation>

<wsdl:operation   name="ReplaceConsumerEntry">
<wsdl:input message="tns:ReplaceConsumerEntryRequest"></wsdl:input>
<wsdl:output message="tns:ReplaceConsumerEntryResponse"></wsdl:output>
<wsdl:fault
message="infod:ReplaceResourceAuthorizationFaultErrorMessage"
name="ReplaceResourceAuthorizationFault">
</wsdl:fault>
<wsdl:fault
message="tns:UnknownResourceReferenceFaultErrorMessage"
name="UnknownResourceReferenceFault">
</wsdl:fault>
<wsdl:fault
message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingrequiredParameterFault">
</wsdl:fault>
<wsdl:fault
message="infod:UnsupportedXQueryFaultErrorMesage"
name="UnsupportedXQueryFault">
</wsdl:fault>
</wsdl:operation>

<wsdl:operation   name="DropConsumerEntry">
<wsdl:input message="tns:DropConsumerEntryRequest"></wsdl:input>
<wsdl:output message="tns:DropConsumerEntryResponse"></wsdl:output>
<wsdl:fault
message="infod:DropResourceAuthorizationFaultErrorMessage"
name="DropResourceAuthorizationFailure">
</wsdl:fault>
<wsdl:fault
message="infod:UnknownResourceReferenceFaultErrorMessage"
```

```
name="UnknownResourceReferenceFault">
</wsdl:fault>
<wsdl:fault message="infod:ExecutionModeFaultErrorMessage"
name="ExecutionModeFault">
</wsdl:fault>
<wsdl:fault name="MissingRequiredParameterFault"
message="tns:MissingRequiredParameterFaultErrorMessage">
</wsdl:fault>
</wsdl:operation>

<wsdl:operation   name="CreateSubscription">
<wsdl:input message="infod:CreateSubscriptionRequest"></wsdl:input>
<wsdl:output message="infod:CreateSubscriptionResponse"></wsdl:output>
<wsdl:fault
message="tns:CreateResourceAuthorizationFailureErrorMessage"
name="CreateResourceAuthorizationFailure">
</wsdl:fault>
<wsdl:fault
message="infod:UnknownResourceReferenceFaultErrorMessage"
name="UnknownResourceReferenceFault">
</wsdl:fault>
<wsdl:fault
message="infod:MissingRequiredParameterFaultErrorMessage"
name="UnsupportedXQueryFault">
</wsdl:fault>
<wsdl:fault
message="infod:UnsupportedXQueryFaultErrorMesage"
name="UnsupportedXQueryFault">
</wsdl:fault>
<wsdl:fault name="MissingRequiredParameterFault"
message="tns:MissingRequiredParameterFaultErrorMessage">
</wsdl:fault>
</wsdl:operation>

<wsdl:operation   name="ReplaceSubscription">
<wsdl:input message="infod:ReplaceSubscriptionRequest"></wsdl:input>
<wsdl:output message="infod:ReplaceSubscriptionResponse"></wsdl:output>
<wsdl:fault
message="infod:ReplaceResourceAuthorizationFaultErrorMessage"
name="ReplaceResourceAuthorizationFault">
</wsdl:fault>
<wsdl:fault
message="infod:UnknownResourceReferenceFaultErrorMessage"
name="UnknownResourceReferenceFault">
</wsdl:fault>
<wsdl:fault
message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingrequiredParameterFault">
</wsdl:fault>
<wsdl:fault
message="infod:UnsupportedXQueryFaultErrorMesage"
name="UnsupportedXQueryFault">
</wsdl:fault>
</wsdl:operation>
```

```
<wsdl:operation   name="DropSubscription">
<wsdl:input message="infod:DropSubscriptionRequest"></wsdl:input>
<wsdl:output message="infod:DropSubscriptionResponse"></wsdl:output>
<wsdl:fault
message="infod:DropResourceAuthorizationFaultErrorMessage"
name="DropResourceAuthorizationFailure">
</wsdl:fault>
<wsdl:fault
message="infod:UnknownResourceReferenceFaultErrorMessage"
name="UnknownResourceReferenceFault">
</wsdl:fault>
<wsdl:fault message="infod:ExecutionModeFaultErrorMessage"
name="ExecutionModeFault">
</wsdl:fault>
<wsdl:fault name="MissigRequiredParameterFault"
message="tns:MissingRequiredParameterFaultErrorMessage">
</wsdl:fault>
</wsdl:operation>

</wsdl:portType>
<wsdl:portType name="infodVocabulary">
<wsdl:operation   name="CreatePropertyVocabulary">
<wsdl:input
message="tns:CreatePropertyVocabularyRequest">
</wsdl:input>
<wsdl:output
message="tns:CreatePropertyVocabularyResponse">
</wsdl:output>
<wsdl:fault
message="tns:CreateResourceAuthorizationFailureErrorMessage"
name="CreateResourceAuthorizationFailure">
</wsdl:fault>
<wsdl:fault
message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingRequiredParameterFault">
</wsdl:fault>
<wsdl:fault
message="infod:UnSupportedVocabularyFaultErrorMessage"
name="UnSupportedVocabularyFault">
</wsdl:fault>
</wsdl:operation>
<wsdl:operation   name="CreatePropertyVocabularyInstance">
<wsdl:input
message="infod:CreatePropertyVocabularyInstanceRequest">
</wsdl:input>
<wsdl:output
message="infod:CreatePropertyVocabularyInstanceResponse">
</wsdl:output>
<wsdl:fault
message="tns:CreateResourceAuthorizationFailureErrorMessage"
name="CreateResourceAuthorizationFailure">
</wsdl:fault>
<wsdl:fault
message="infod:UnknownResourceReferenceFaultErrorMessage"
name="UnknownResourceReferenceFault">
```

```
</wsdl:fault>
<wsdl:fault
message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingRequiredParameterFault">
</wsdl:fault>
<wsdl:fault
message="infod:UnSupportedVocabularyFaultErrorMessage"
name="UnSupportedVocabularyFault">
</wsdl:fault>
</wsdl:operation>
<wsdl:operation   name="DropPropertyVocabularyInstance">
<wsdl:input
message="infod:DropPropertyVocabularyInstanceRequest"
name="DropPropertyVocabularyInstanceRequest">
</wsdl:input>
<wsdl:output
message="infod:DropPropertyVocabularyInstanceResponse"
name="DropPropertyVocabularyInstanceResponse">
</wsdl:output>
<wsdl:fault
message="infod:DropVocabularyAuthorizationFailureErrorMessage"
name="DropResourceAuthorizationFailure">
</wsdl:fault>
<wsdl:fault
message="infod:UnknownResourceReferenceFaultErrorMessage"
name="UnknownResourceReferenceFault">
</wsdl:fault>
<wsdl:fault
message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingRequiredParameterFault">
</wsdl:fault>
<wsdl:fault name="ExecutionModeFault"
message="tns:ExecutionModeFaultErrorMessage">
</wsdl:fault>
</wsdl:operation>
<wsdl:operation   name="CreateDataVocabulary">
<wsdl:input message="tns:CreateDataVocabularyRequest"></wsdl:input>
<wsdl:output
message="tns:CreateDataVocabularyResponse">
</wsdl:output>
<wsdl:fault
message="tns:CreateResourceAuthorizationFailureErrorMessage"
name="CreateResourceAuthorizationFailure">
</wsdl:fault>
<wsdl:fault
message="infod:UnknownResourceReferenceFaultErrorMessage"
name="UnknownResourceReferenceFault">
</wsdl:fault>
<wsdl:fault
message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingRequiredParameterFault">
</wsdl:fault>
<wsdl:fault
message="infod:UnSupportedVocabularyFaultErrorMessage"
name="UnSupportedVocabularyFault">
```

```
</wsdl:fault>
</wsdl:operation>
<wsdl:operation   name="DropDataVocabulary">
<wsdl:input message="infod:DropDataVocabularyRequest"></wsdl:input>
<wsdl:output message="infod:DropDataVocabularyResponse"></wsdl:output>
<wsdl:fault
message="tns:DropResourceAuthorizationFaultErrorMessage"
name="DropResourceAuthorizationFailure">
</wsdl:fault>
<wsdl:fault
message="infod:UnknownResourceReferenceFaultErrorMessage"
name="UnknownResourceReferenceFault">
</wsdl:fault>
<wsdl:fault
message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingRequiredParameterFault">
</wsdl:fault>
<wsdl:fault
message="infod:UnSupportedVocabularyFaultErrorMessage"
name="UnSupportedVocabularyFault">
</wsdl:fault>
</wsdl:operation>
<wsdl:operation   name="CreateDataSourceEntry">
<wsdl:input message="infod:CreateDataSourceEntryRequest"></wsdl:input>
<wsdl:output
message="infod:CreateDataSourceEntryResponse">
</wsdl:output>
<wsdl:fault
message="infod:CreateAssociationAuthorizationFailureErrorMessage"
name="CreateAssociationAuthorizationFailure">
</wsdl:fault>
<wsdl:fault
message="infod:UnknownResourceReferenceFaultErrorMessage"
name="UnknownResourceReferenceFault">
</wsdl:fault>
<wsdl:fault
message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingRequiredParameterFault">
</wsdl:fault>
<wsdl:fault
message="infod:UnSupportedVocabularyFaultErrorMessage"
name="UnSupportedVocabularyFault">
</wsdl:fault>
</wsdl:operation>
<wsdl:operation   name="DropDataSourceEntry">
<wsdl:input
message="tns:DropDataSourceEntryRequest">
</wsdl:input>
<wsdl:output
message="tns:DropDataSourceEntryResponse">
</wsdl:output>
<wsdl:fault
message="tns:DropResourceAuthorizationFaultErrorMessage"
name="DropResourceAuthorizationFailure">
</wsdl:fault>
```

```
<wsdl:fault
message="infod:UnknownResourceReferenceFaultErrorMessage"
name="UnknownResourceReferenceFault">
</wsdl:fault>
<wsdl:fault
message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingRequiredParameterFault">
</wsdl:fault>
<wsdl:fault name="ExecutionModeFault"
message="tns:ExecutionModeFaultErrorMessage">
</wsdl:fault>
</wsdl:operation>
<wsdl:operation   name="DropPropertyVocabulary">
<wsdl:input message="tns:DropPropertyVocabularyRequest"></wsdl:input>
<wsdl:output message="tns:DropPropertyVocabularyResponse"></wsdl:output>
<wsdl:fault name="DropResourceAuthorizationFailure"
message="tns:DropResourceAuthorizationFaultErrorMessage">
</wsdl:fault>
<wsdl:fault name="UnknownResourceReferenceFault"
message="tns:UnknownResourceReferenceFaultErrorMessage">
</wsdl:fault>
<wsdl:fault name="MissingRequiredParameterFault"
message="tns:MissingRequiredParameterFaultErrorMessage">
</wsdl:fault>
<wsdl:fault name="ExecutionModeFault"
message="tns:ExecutionModeFaultErrorMessage">
</wsdl:fault>
</wsdl:operation>
</wsdl:portType>
<wsdl:portType name="infodMetaData">
<wsdl:operation   name="GetMetaData">
<wsdl:input message="infod:GetMetaDataQueryExpression"></wsdl:input>
<wsdl:output message="infod:GetMetaDataQueryResponse"></wsdl:output>
<wsdl:fault name="GetMetaDataAuthorizationFailure"
message="tns:GetMetaDataAuthorizationFailureErrorMessage">
</wsdl:fault>
<wsdl:fault name="MissingRequiredParameterFault"
message="tns:MissingRequiredParameterFaultErrorMessage">
</wsdl:fault>
<wsdl:fault name="UnsupportedXQueryFault"
message="tns:UnsupportedXQueryFaultErrorMessage">
</wsdl:fault>
</wsdl:operation>
</wsdl:portType>
<wsdl:portType name="infodNotify">
<wsdl:operation   name="Notify">
<wsdl:input message="infod:NotificationMessage"></wsdl:input>
</wsdl:operation>
<wsdl:operation   name="PublisherNotification">
<wsdl:input message="infod:PublisherNotification"></wsdl:input>
</wsdl:operation>
<wsdl:operation   name="SubscriberNotification">
<wsdl:input message="infod:SubscriberNotification"></wsdl:input>
</wsdl:operation>
<wsdl:operation   name="ConsumerNotification">
```

```
<wsdl:input message="infod:ConsumerNotification"></wsdl:input>
</wsdl:operation>
</wsdl:portType>
<wsdl:service name="INFOD"></wsdl:service>
</wsdl:definitions>
```

7 Author Information

Stephen Davey
National e-Science Centre
15 South College Street
Edinburgh
EH8 9AA
UK

Vijay Dialani
IBM Corporation
Almaden Research Center
650 Harry Road,
San Jose, CA 95120-6099
USA

Ronny Fehling
Oracle Corporation
600 Blvd. de Maisonneuve Ouest
Montreal
Quebec H3A 3J2
Canada

Steve Fisher
Rutherford Appleton Laboratory
Chilton
Didcot
Oxon
OX11 0QX
UK

Dieter Gawlick
Oracle Corporation
500 Oracle Parkway
Redwood Shores
CA 94065
USA

Christopher Kantarjiev
Oracle Corporation
500 Oracle Parkway
Redwood Shores
CA 94065
USA

Cecile Madsen
IBM Silicon Valley Laboratory
555 Bailey Avenue
San Jose, CA 95141
USA

Susan Malaika
IBM Corporation
Silicon Valley Laboratory
555 Bailey Avenue
San Jose, CA 95141
USA

Shailendra Mishra
Oracle Corporation
500 Oracle Parkway
Redwood Shores
CA 94065
USA

Mallikarjun Shankar
Oak Ridge National Laboratory
Oak Ridge
TN 37831
USA

8 Intellectual Property Statement

"Intellectual Property Statement.
The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in
this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.
The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director."

9 Full Copyright Notice