

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

Copyright © Global Grid Forum (2004-2006). All Rights Reserved

Abstract

Notification-based interaction is very common as noted by the authors of the WS-Notification family of specifications. WS-Notification offers a basic topic-based publish/subscribe pattern complemented by a notify operation.

INFOD (Information Dissemination) uses the same notify operation. Additionally, INFOD 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 allows the characterization of publishers, consumers and various other components using vocabularies that are meaningful to members of the communities they belong to.

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 which messages should be created in response to events.

Table of Contents

TABLE OF CONTENTS	2
1 INTRODUCTION	4
1.1 The Registry	5
1.1.1 <i>Entities</i>	5
1.1.2 <i>Data Vocabularies</i>	5
1.1.3 <i>Property Vocabularies</i>	6
1.1.4 <i>Vocabulary Associations</i>	6
1.1.5 <i>Dependencies</i>	6
1.1.6 <i>Matching Publishers to Subscriptions</i>	7
1.2 Security	7
1.3 Lifetime Management	7
1.4 Summary of key aspects of INFOD	8
1.5 Glossary	8
1.6 Terminology	10
1.7 Namespaces	11
1.8 Fault Definitions	11
1.9 Interfaces	11
2 THE BASE INFOD REGISTRY INTERFACE	13
2.1 Managing Publishers	14
2.1.1 <i>CreatePublisher</i>	14
2.1.2 <i>ReplacePublisher</i>	17
2.1.3 <i>DropPublisher</i>	19
2.2 Managing Subscribers	20
2.2.1 <i>CreateSubscriber</i>	20
2.2.2 <i>ReplaceSubscriber</i>	22
2.2.3 <i>DropSubscriber</i>	23
2.3 Managing Consumers	25
2.3.1 <i>CreateConsumer</i>	25
2.3.2 <i>ReplaceConsumer</i>	26
2.3.3 <i>DropConsumer</i>	28
2.4 Managing Subscriptions	29
2.4.1 <i>CreateSubscription</i>	30

2.4.2	<i>ReplaceSubscription</i>	32
2.4.3	<i>DropSubscription</i>	34
2.5	Managing Vocabularies	35
2.5.1	<i>RegisterPropertyVocabulary</i>	36
2.5.2	<i>CreatePropertyVocabularyInstance</i>	37
2.5.3	<i>DropPropertyVocabularyInstance</i>	39
2.5.4	<i>RegisterDataVocabulary</i>	40
2.5.5	<i>UnregisterVocabulary</i>	42
2.6	Managing Associations.....	43
2.6.1	<i>AssociateVocabulary</i>	43
2.6.2	<i>DisassociateVocabulary</i>	45
2.7	The GetMetaData Operation	46
3	THE BASE INFOD NOTIFY INTERFACE.....	49
3.1.1	<i>Notify</i>	49
3.1.2	<i>Notification of Publishers</i>	51
4	SECURITY CONSIDERATIONS.....	53
4.1	Message Encryption and Data Privacy Requirements	53
4.2	Integration with authorization model	54
5	APPENDIX I – XML SCHEMA	55
5.1	Publishers	56
5.2	Subscribers.....	57
5.3	Consumers	57
5.4	Subscriptions	58
5.5	Property Vocabularies.....	58
5.6	Property Vocabulary Instances	59
5.7	Data Vocabularies.....	59
5.8	Associations.....	60
5.9	Error Messages	60
5.10	INFOD Notification	62
5.11	INFOD Publisher Notification	63
6	APPENDIX II – WSDL 1.1	65
7	AUTHOR INFORMATION	76
8	INTELLECTUAL PROPERTY STATEMENT.....	78
9	FULL COPYRIGHT NOTICE.....	79

1 Introduction

Information Dissemination¹ (INFOD) allows data sources to make data available to consumers by a notification mechanism. State changes of data may generate events; events may generate messages; messages are created by *publishers* and sent to *consumers*.

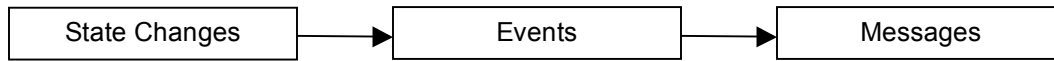
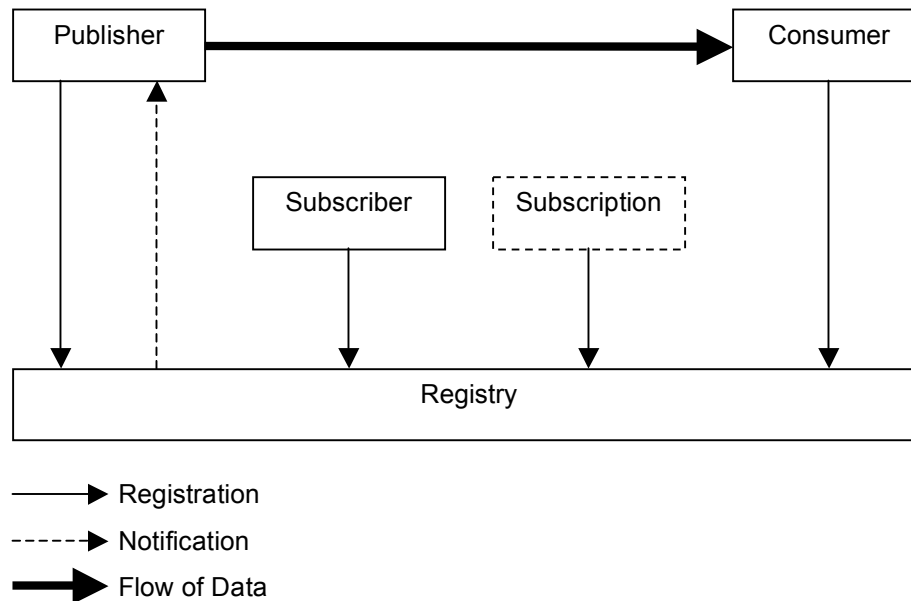


Figure 1: INFOD Message Provenance

Publishers publish messages for consumption by consumers. Subscribers create subscriptions to match publishers' data with consumers' needs. If needed, subscribers are able to specify what is considered to be an event, what message gets published in response to which event and which consumer should receive a message. Messages are sent directly from publishers to consumers.

The information about INFOD objects such as publishers, consumers, subscribers, subscriptions and vocabularies are contained in an INFOD Registry. This allows the registry to match publishers and consumers and notify the publisher, which can then send a message to the consumer. Instead of identifying publishers and consumers explicitly, subscribers only need to specify the properties of publishers and consumers.



¹ 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

Figure 2: A typical INFOD interaction

Figure 2 shows a typical interaction. The publisher, subscriber and consumer must register. They might register themselves or it may be done on their behalf. Registering an object involves the creation and storage, by the registry, of a corresponding entity. There is an important distinction between, for example, a consumer, which is a web service with its own end point reference, and the Consumer Entity, which is information about the consumer service, that is stored in the INFOD Registry and which also has its own end point reference. The subscription is different in that it only exists in the registry, which is why it is shown with a broken line in the diagram. When the registry detects that a subscription has indicated that a publisher should send a message to a consumer it notifies the publisher, which in turn constructs and sends the message directly to the consumer. This notification could be triggered by the arrival of a new consumer, the passing of time or any other change in state detected by the registry. The publisher in Figure 2 must also be a web service to receive the notification. If the publisher is not a web service, or if notifications are not desired, the publisher may poll the registry to search for suitable consumers.

1.1 The Registry

The resources managed by the registry are: entities, data vocabularies and instances, and property vocabularies and their associations.

1.1.1 Entities

An entity is the information about an external object that is stored in the registry. Each is identified in the registry by a unique EPR and may have a reference to the EPR of the corresponding resource outside the registry. Operations are provided to create, replace and drop entities. The act of creation involves storing information and returning the EPR of the entity. 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 entities. 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 entity: all the data associated with it by the create operation is replaced by new data however all relations established after the original entity was created are preserved as the identity of the entity remains unchanged. The drop operation (for example DropPublisher in Section 2.1.3) takes the EPR of the entity and makes the stored entity unavailable and so makes subsequent use of the EPR invalid. The drop operation is not allowed to make the system inconsistent (see Section 1.1.5) so, by default, an error will be reported if an attempt is made to drop an entity which is still referenced. There is an optional flag which can be set to "DISABLE NEW REFERENCES" which results in the entity being dropped when the last reference to the entity has been removed and "CASCADE", which also drops (recursively) all entities referencing that entity. As already mentioned the subscription is a special entity as it has no counterpart outside the registry. Each 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. There is then an optional property constraint expressed in terms of a property vocabulary that is used to ensure that entities are mutually acceptable.

The subscription entity also has an optional data constraint expressed in terms of a data vocabulary. The data constraints in subscriptions specify which data are of interest and therefore trigger the flow of messages to consumers. INFOD is agnostic to whether these constraints reference messages, events or states; this information is derived from the data vocabulary and the associated constraints.

1.1.2 Data Vocabularies

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 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 RegisterDataVocabulary (Section 2.5.4) to store information about the data vocabulary in the registry.

1.1.3 Property Vocabularies

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 store actual values in the registry for each entity.

Constraints identifying which other entities 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 extensions mechanism may be used to formalize some properties; e.g., operational characteristics.

1.1.4 Vocabulary Associations

The INFOD registry manages associations between data vocabularies and entities. One especially important use for this is to associate a publisher entity with one or more data vocabularies thereby identifying the publisher as a source of some specific type of information..

Associations, like entities have their own EPR and an optional name and description and they have the EPR of the two things they are relating.

1.1.5 Dependencies

There is a basic dependency rule that governs the creation, modification and removal of resources within the INFOD registry: only Resources that are registered in the INFOD registry SHALL be referenced.

Figure 3 shows the relations between the various INFOD resources. The open triangle shows generalization/inheritance and the arrows show the direction of reference.

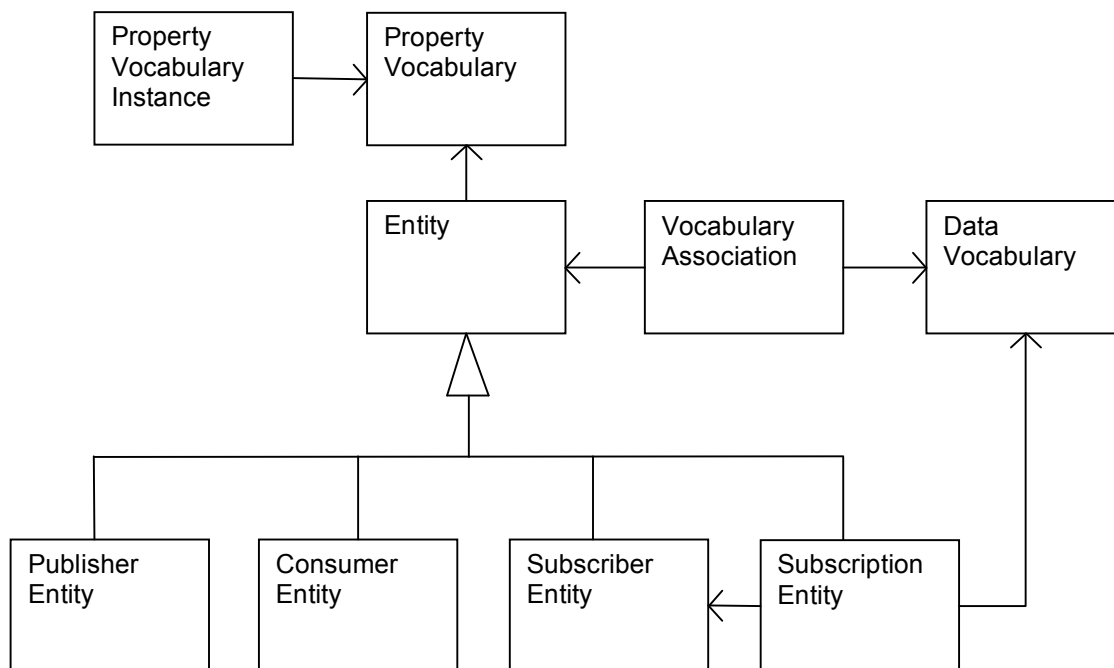


Figure 3: Relation between INFOD resources

Vocabularies MAY be referenced through Property or Data Constraints or in the specification of other vocabularies. They MUST be referenced by their infod:EPR. (The namespace infod: refers to the namespace newly defined for schemas in this specification document.)

Vocabulary Instances MAY be referenced in Vocabulary Associations or by entity properties. They MUST be referenced by their infod:EPR.

1.1.6 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 to look up entities, vocabularies and associations 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 by WSN 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 entities, 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 entities, some form of lifetime management should be employed such as WS-ResourceLifetime (see <http://docs.oasis-open.org/wsr/2004/06/wsr-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 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, via WSN, of changes that it considers relevant.

The INFOD registry can apply constraints simultaneously.

1.5 Glossary

<u>Association</u>	See Vocabulary Association.
<u>Constraint</u>	A constraint is used to specify which instances of data are of interest to an entity. A constraint can reference single instances or a set of instances, which may represent entities, property vocabulary instances, messages, events or state transitions. Constraints must be formulated in the constraint language(s) that are associated to the vocabularies, which are used to structure the referenced data.
<u>Constraint Language</u>	The grammar of the constraints specification associated to a type system.
<u>Entity</u>	An entity is the information about an external object that is stored in the registry. There are four types of entities: Publisher, consumer, subscriber and subscription.
<u>Data Vocabulary</u>	A data vocabulary defines the structure of the data associated to a vocabulary association. Data vocabularies can be specified using any types system.
<u>Consumer</u>	A consumer is an entity that 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.
<u>Event</u>	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.

<u>Message</u>	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.
<u>Property Vocabulary</u>	A property vocabulary specifies the structure of properties associated to entities or vocabulary associations. Data vocabularies must be specified using XML as the type system.
<u>Property Vocabulary Instance</u>	A property vocabulary instance represents the (values of) properties that are associated to specific entities or vocabulary associations. A property vocabulary instance has to be structures according to a property vocabulary.
<u>Publisher</u>	<p>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.</p> <p>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.</p>
<u>Registry</u>	A repository of object designed to match for information providers and information consumers. The INFOD registry contains entities, vocabularies, vocabulary instance, and vocabulary associations.
<u>Resource</u>	A resource is an entry in the registry. This includes entities, data vocabularies and instances, and property vocabularies and their associations.
<u>State Change</u>	<p>A data source as specified by vocabulary association may expose states; e.g., the values of all its entities at a specific time. A state change happens if at least one value of one entity changes and becomes visible.</p> <p>State changes can only be referenced if the grammar of the constraint language exposes state transitions.</p> <p>In many cases, the time is an (implicit) part of some or all entities; consequently, the change of time implies a state change.</p>
<u>Subscriber</u>	A subscriber is an entity specifying subscriptions. Subscriptions are the primary means of specifying the message flow from publishers to consumers.
<u>Subscription</u>	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.

Vocabulary Association

A vocabulary association specifies that data structured with the referenced vocabulary (and constraint language) is offered by the referenced publisher.

1.6 Terminology

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 [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.

```
<!--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
S	http://schemas.xmlsoap.org/soap/envelope/ OR http://www.w3.org/2003/05/soap-envelope
xsd	http://www.w3.org/2001/XMLSchema
wsa	http://www.w3.org/2005/08/addressing
wsrf-rp	http://docs.oasis-open.org/wsrp/rp-2
wsrf-bf	http://docs.oasis-open.org/wsrp/bf-2
wsnt	http://docs.oasis-open.org/wsn/b-2
wsntw	http://docs.oasis-open.org/wsn/bw-2
wstop	http://docs.oasis-open.org/wsn/t-1
infod	http://www.ggf.org/infod

For extensibility the 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/wsrp/wsrp-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.ggf.org/infod/fault>.

1.9 Interfaces

This specification defines two Web Service interfaces: a registry interface with a large number of operations grouped in this document according to the entity or other element resource they are managing and a notify interface with just the one operation.

The interfaces MUST be called using a request-reply message. The response mechanism to these messages depends on the definition of the reply-to address of the request message:

1. If the Reply-To address is present in the header message, it MUST be obeyed. Otherwise,
2. If the sender is identified by whatever protocol the request used (e.g. SOAP which has a built-in reply address) it MUST be used as a reply-to address. Otherwise,

3. The EPR identified in the `wsa:address` property of the EPR in the 'Reply-To' except when the value is <http://www.w3.org/2005/08/addressing/anonymous> when (1) also applies

For information to the WS-Addressing, please refer to section 3.4 of the WS-Addressing 1.0 Core spec (see <http://www.w3.org/Submission/ws-addressing/>). For information on the SOAP addressing, please refer to the WS Addressing 1.0 SOAP Binding spec (see <http://www.w3.org/TR/2006/PR-ws-addr-soap-20060321/#anonaddress>).

Also, note that the WSDL has been written in accordance to the conventions described in the Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts (see <http://dev.w3.org/cvsweb/~checkout~/2002/ws/desc/wsd120/wsd120-adjuncts.html>)

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 Publishers	CreatePublisher	This operation defines how to create a new Publisher entity in an INFOD Registry.	2.1.1
	ReplacePublisher	This operation defines how to replace the metadata associated with a particular Publisher entity in an INFOD Registry.	2.1.2
	DropPublisher	This operation defines how to drop an existing Publisher entity from an INFOD Registry.	2.1.3
Managing Subscribers	CreateSubscriber	This operation defines how to create a new Subscriber entity in an INFOD Registry.	2.2.1
	ReplaceSubscriber	This operation defines how to replace the metadata associated with a particular Subscriber entity in an INFOD Registry.	2.2.2
	DropSubscriber	This operation defines how to drop an existing Subscriber entity from an INFOD Registry.	2.2.3
Managing Consumers	CreateConsumer	This operation defines how to create a new Consumer entity in an INFOD Registry.	2.3.1
	ReplaceConsumer	This operation defines how to replace the metadata associated with a particular Consumer entity in an INFOD Registry.	2.3.2
	DropConsumer	This operation defines how to drop an existing Consumer entity from an INFOD Registry.	2.3.3
Managing Subscriptions	CreateSubscription	This operation defines how to create a new Subscription entity in an INFOD Registry.	2.4.1
	ReplaceSubscription	This operation defines how to replace the metadata associated with a particular Subscription entity in an INFOD Registry.	2.4.2

	DropSubscription	This operation defines how to drop an existing Subscription entity from an INFOD Registry.	2.4.3
Managing Vocabularies	RegisterPropertyVocabulary	This operation defines how to register a property vocabulary to an INFOD Registry	2.5.1
	CreatePropertyVocabularyInstance	This operation creates a new instance of a property vocabulary that is already registered in an INFOD Registry.	2.5.2
	DropPropertyVocabularyInstance	This operation drops an existing instance of a particular property vocabulary registered in an INFOD Registry.	2.5.3
	RegisterDataVocabulary	This operation defines how to register a data vocabulary to an INFOD Registry	2.5.4
	UnregisterVocabulary	This operation defines how to un-register a vocabulary from an INFOD Registry.	2.5.5
Managing Associations	AssociateVocabulary	This operation defines how to associate a data vocabulary with a particular entity in an INFOD Registry.	2.6.1
	DisassociateVocabulary	This operation defines how to drop an existing association of a data vocabulary with a particular entity from an INFOD Registry.	2.6.2
	GetMetaData	This operation queries the metadata of entities defined in a particular INFOD Registry.	2.7

2.1 Managing Publishers

These operations **MUST** be used to manage publishers

- CreatePublisher (section 2.1.1)
- ReplacePublisher (section 2.1.2)
- DropPublisher (section 2.1.3)

2.1.1 CreatePublisher

The CreatePublisher operation **MUST** be used to create an INFOD publisher entity in an INFOD Registry. The publisher is responsible for publishing data to consumers using subscriptions. The publisher has to identify subscriptions and consumers that match the entity property constraints specified by the involved entities. The publisher **MAY** also specify property constraints on entities it will communicate with. When the registry matches consumers with publishers through a subscription, it takes into consideration all the property constraints specified by each involved entity. Specifying a property constraint requires the identification of the publisher with the corresponding Property

Vocabulary Instance (Section 2.5.2). The publisher MAY also specify what data vocabulary it is using for its publication. This MUST be done by associating a Data Vocabulary (Section 2.6.1)

As part of the processing of a CreatePublisher request message, the INFOD Registry MUST create an INFOD entity and an EPR representing the publisher.

The format of the request message for CreatePublisher operation is based on the schema provided in Appendix I – XML Schema (XML schema definition of an ‘infodEntity’), as follows:

```
<infod:CreatePublisher>
  <infod:WSEntityReference>
    wsa:EndPointReferenceType
  </infod:WSEntityReference> ?
  <infod:PublisherName> xsd:string </infod:PublisherName> ?
  <infod:PublisherDescription>
    xsd:string
  </infod:PublisherDescription> ?
  <infod:PropertyConstraints>
    <xsd:complexType>
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element>tns:PropertyConstraint</xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </infod:PropertyConstraints> *
  <infod:Notification>
    xsd:boolean default "FALSE"
  </infod:Notification> ?
</infod:CreatePublisher>
```

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

/infod:WSEntityReference

An endpoint reference element, as defined by WS-Addressing, used to identify the WS endpoint for the entity. 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 entity.

/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:PropertyConstraints

An element specifying which other INFOD entities are qualified to interact with the publisher entity. The property constraints MAY reference attributes related to other INFOD entities (consumers, subscribers and subscriptions), associations and property vocabulary instances. The Property Constraints 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.

Constraints can not reference publishers.

For example, a publisher identifies in that manner the set of consumers that are not authorized to receive data.

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. `infod:WSEntityReference` MUST be specified.

For further details see section 3.1.2

WS-Addressing of the action MUST contain the URI
<http://www.ggf.org/infod/INFODRegistry/CreatePublisher>

INFOD Registry Response

If the INFOD Registry accepts the `CreatePublisher` message, it MUST respond to the WS endpoint specified in the request message with a `CreatePublisherResponse` message. The `CreatePublisherResponse` message is a message of the following form:

```
<infod:CreatePublisherResponse>
  <infod:INFODEntityReference>
    wsa:EndPointReferenceType
  </infod:INFODEntityReference>
</infod:CreatePublisherResponse>
```

The elements of the `CreatePublisherResponse` message are further described as follows:

/infod:INFODEntityReference

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

Instead of the `CreatePublisherResponse` message, the INFOD Registry MAY send the following faults in response to a `CreatePublisher` message:

- `CreateEntityAuthorizationFailure`: User not authorized to create the INFOD entity at this Registry
- `UnknownElementReferenceFault`: An element has been referenced that is unknown to the 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 `CreatePublisher` request message using SOAP:

```
<s:Envelope ... >
  <s:Header>
    <wsa:Action>
      http://www.ggf.org/infod/INFODRegistry/CreatePublisher
    </wsa:Action>
    ...
  </s:Header>
  <s:Body>
    <infod:CreatePublisher>
      <infod:WSEntityReference>
        <wsa:Address>
          http://www.example.org/SomePublisher
        </wsa:Address>
      </infod:WSEntityReference>
    </infod:CreatePublisher>
  </s:Body>
</s:Envelope>
```



```

    </wsa:Address>
  </infod:WSEntityReference>
  <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:CreatePublisher>
</s:Body>
</s:Envelope>

```

2.1.2 ReplacePublisher

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

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

```

<infod:ReplacePublisher>
  <infod:WSEntityReference>
    wsa:EndPointReferenceType
  </infod:WSEntityReference> ?
  <infod:INFODEntityReference>
    wsa:EndPointReferenceType
  </infod:INFODEntityReference>
  <infod:PublisherName> xsd:string </infod:PublisherName> ?
  <infod:PublisherDescription>
    xsd:string
  </infod:PublisherDescription> ?
  <infod:PropertyConstraints>
    <xsd:complexType>
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element>tns:PropertyConstraint</xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </infod:PropertyConstraints> *
  <infod:Notification>
    xsd:boolean "FALSE"
  </infod:Notification> ?
</infod:ReplacePublisher>

```

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

/infod:WSEntityReference

An endpoint reference element, as defined by WS-Addressing, used to identify the WS endpoint for the entity. 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 entity.

/infod:INFODEntityReference

An endpoint reference element, as defined by WS-Addressing, used to identify the publisher entity 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:PropertyConstraints

An element specifying which other INFOD entities are qualified to interact with the publisher entity. The property constraints MAY reference attributes related to other INFOD entities (consumers, subscribers and subscriptions), associations and property vocabulary instances. The Property Constraints 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 in that manner the set of consumers that are not authorized to receive data.

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. `infod:WSEntityReference` MUST be specified.

For further details see section 3.1.2

WS-Addressing of the action MUST contain the URI
<http://www.ggf.org/infod/INFODRegistry/ReplacePublisher>

INFOD Registry Response

If the INFOD Registry accepts the `ReplacePublisher` message, it MUST respond to the WS endpoint specified in the request message with a `ReplacePublisherResponse` message. The `ReplacePublisherResponse` message is a message of the following form:

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

The elements of the `ReplacePublisherResponse` message are further described as follows:

/infod:Status

An indication that the request has been successfully executed.

Instead of the `ReplacePublisherResponse` message, the INFOD Registry MAY send the following faults in response to a `ReplacePublisher` request message:

- `ReplaceEntityAuthorizationFailure`: User not authorized to replace the INFOD entity at this Registry

- **UnknownElementReferenceFault:** An element has been referenced that is unknown to the 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 DropPublisher

The DropPublisher operation **MUST** be used to remove an INFOD publisher entity from an INFOD Registry.

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

```
<infod:DropPublisher>
  <infod:INFODEntityReference>
    wsa:EndPointReferenceType
  </infod:INFODEntityReference>
  <infod:ExecutionMode> xsd:string </infod:ExecutionMode> ?
</infod:DropPublisher>
```

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

/infod:INFODEntityReference

An endpoint reference element, as defined by WS-Addressing, used to identify the INFOD entity 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 entity is unused and unreferenced |
| “DISABLE NEW” | No new references are possible for the entity. The entity will be dropped when the last reference to this entity is gone |
| “CASCADE” | The drop request will execute immediately and all references to the entity will be removed recursively |

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

WS-Addressing of the action **MUST** contain the URI
<http://www.ggf.org/infod/INFODRegistry/DropPublisher>

INFOD Registry Response

If the INFOD Registry accepts the DropPublisher message, it **MUST** respond to the WS endpoint specified in the request message with a DropPublisherResponse message. The DropPublisherResponse message is a message of the following form:

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

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

/infod:Status

An indication that the request has been successfully executed.

Instead of the DropPublisherResponse message, the INFOD Registry MAY send the following faults in response to a DropPublisher message:

- DropEntityAuthorizationFailure: User not authorized to drop the INFOD entity at this Registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the 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 Subscribers

The following operations MUST be used to manage subscribers:

- CreateSubscriber (section 2.2.1)
- ReplaceSubscriber (section 2.2.2)
- DropSubscriber (section 2.2.3)

2.2.1 CreateSubscriber

The CreateSubscriber operation MUST be used to create an INFOD subscriber entity in an INFOD Registry. This subscriber entity acts as a mechanism to create, replace and drop subscription. Only subscribers MAY create subscriptions. The subscriber MAY also specify property constraints on entities it will communicate with. Specifying a property constraint requires the identification of the publisher with the corresponding Property Vocabulary Instance (section 2.5.2). Any subscription that is created MUST reference the subscriber that was used to generate the subscription (section 2.4.1).

As part of the processing of a CreateSubscriber request message, the INFOD Registry implementing the BaseSubscriberManager interface MUST create an INFOD resource representing the subscriber.

The format of the request message for CreateSubscriber operation is based on the schema provided in Appendix I – XML Schema (XML schema definition of an 'infodEntity'), as follows:

```
<infod:CreateSubscriber>
  <infod:WSEntityReference>
    wsa:EndPointReferenceType
  </infod:WSEntityReference> ?
  <infod:SubscriberName> xsd:string </infod:SubscriberName> ?
  <infod:SubscriberDescription>
    xsd:string
  </infod:SubscriberDescription> ?
  <infod:PropertyConstraints>
    <xsd:complexType>
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element>tns:PropertyConstraint</xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </infod:PropertyConstraints> *
```

```
</infod:CreateSubscriber>
```

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

/infod:WSEntityReference

An endpoint reference element, as defined by WS-Addressing, used to identify the WS endpoint for the entity. 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 entity.

/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:PropertyConstraints

An element that specifies which other INFOD entities are qualified to interact with the subscriber entity. The property constraints MAY reference attributes related to other INFOD entities (publishers and consumers), associations and property vocabulary instances. The Property Constraints 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 MAY identify the set of publishers who are not eligible to be selected for any publication.

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

WS-Addressing of the action MUST contain the URI
<http://www.ggf.org/infod/INFODRegistry/CreateSubscriber>

INFOD Registry Response

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

```
<infod:CreateSubscriberResponse>
  <infod:INFODEntityReference>
    wsa:EndPointReferenceType
  </infod:INFODEntityReference>
</infod:CreateSubscriberResponse>
```

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

/infod:INFODEntityReference

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

Instead of the CreateSubscriberResponse message, the INFOD Registry MAY send the following faults in response to a CreateSubscriber message:

- CreateEntityAuthorizationFailure: User not authorized to create the INFOD entity at this Registry

- **UnknownElementReferenceFault:** An element has been referenced that is unknown to the 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 ReplaceSubscriber

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

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

```
<infod:ReplaceSubscriber>
  <infod:WSEntityReference>
    wsa:EndPointReferenceType
  </infod:WSEntityReference> ?
  <infod:INFODEntityReference>
    wsa:EndPointReferenceType
  </infod:INFODEntityReference>
  <infod:SubscriberName> xsd:string </infod:SubscriberName> ?
  <infod:SubscriberDescription>
    xsd:string
  </infod:SubscriberDescription> ?
  <infod:PropertyConstraints>
    <xsd:complexType>
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element>tns:PropertyConstraint</xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </infod:PropertyConstraints> *
</infod:ReplaceSubscriber>
```

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

/infod:WSEntityReference

An endpoint reference element, as defined by WS-Addressing, used to identify the WS endpoint for the entity. 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 entity.

/infod:INFODEntityReference

An endpoint reference element, as defined by WS-Addressing, used to identify the subscriber entity 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:PropertyConstraints

An element that specifies the other INFOD entities that are qualified to interact with the subscriber entity. The property constraints MAY reference attributes related to other INFOD entities (publishers and consumers), associations and property vocabulary instances. The Property Constraints 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 MAY identify the set of publishers who are not eligible to be selected for any publication.

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

WS-Addressing of the action MUST contain the URI
<http://www.ggf.org/infod/INFODRegistry/ReplaceSubscriber>

INFOD Registry Response

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

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

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

/infod:Status

An indication that the request has been successfully executed. .

Instead of the ReplaceSubscriberResponse message, the INFOD Registry MAY send the following faults in response to a ReplaceSubscriber message:

- ReplaceEntityAuthorizationFailure: User not authorized to replace the INFOD entity at this Registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the 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 DropSubscriber

The DropSubscriber operation MUST be used to remove an INFOD subscriber entity from an INFOD Registry.

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

```
<infod:DropSubscriber>
```

```

<infod:INFODEntityReference>
  wsa:EndPointReferenceType
</infod:INFODEntityReference>
<infod:ExecutionMode> xsd:string </infod:ExecutionMode>
</infod:DropSubscriber>

```

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

/infod:INFODEntityReference

An endpoint reference element, as defined by WS-Addressing, used to identify the INFOD entity 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 entity is unused and unreferenced |
| “DISABLE NEW” | No new references are possible for the entity. The entity will be dropped when the last reference to this entity is gone |
| “CASCADE” | The drop request will execute immediately and all references to the entity will be removed recursively |

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

WS-Addressing of the action MUST contain the URI
<http://www.ggf.org/infod/INFODRegistry/DropSubscriber>

INFOD Registry Response

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

```

<infod:DropSubscriberResponse>
  <infod:Status>
    xsd:string default "COMPLETED"
  </infod:Status>
</infod:DropSubscriberResponse>

```

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

/infod:Status

An indication that the request has been successfully executed.

Instead of the DropSubscriberResponse message, the INFOD Registry MAY send the following faults in response to a DropSubscriber message:

- DropEntityAuthorizationFailure: User not authorized to drop the INFOD entity at this Registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the 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/wsr/wsr/ws_base_faults-1.2-spec-os.pdf).

2.3 Managing Consumers

The following operations MUST be used to manage consumers:

- CreateConsumer (section 2.3.1)
- ReplaceConsumer (section 2.3.2)
- DropConsumer (section 2.3.3)

2.3.1 CreateConsumer

The CreateConsumer operation MUST be used to create an INFOD consumer entity in an INFOD Registry. This consumer entity receives the messages from publishers identified by the corresponding subscription. The consumer MAY also specify property constraints on entities it will communicate with. When the registry matches consumers with publishers using a subscription, it takes into consideration all the property constraints that are specified by each involved entity. Specifying a property constraint requires the identification of the consumer with the corresponding Property Vocabulary Instance (Section 2.5.2).

As part of the processing of a CreateConsumer request message, the INFOD Registry MUST create an INFOD resource representing the consumer.

The format of the request message for CreateConsumer operation is based on the schema provided in Appendix I – XML Schema (XML schema definition of an 'infodEntity'), as follows:

```
<infod:CreateConsumer>
  <infod:WSEntityReference>
    wsa:EndPointReferenceType
  </infod:WSEntityReference>
  <infod:ConsumerName> xsd:string </infod:ConsumerName> ?
  <infod:ConsumerDescription>
    xsd:string
  </infod:ConsumerDescription> ?
  <infod:PropertyConstraints>
    <xsd:complexType>
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element>tns:PropertyConstraint</xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </infod:PropertyConstraints> *
</infod:CreateConsumer>
```

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

/infod:WSEntityReference

An endpoint reference element, as defined by WS-Addressing, used to identify the WS endpoint for the entity. 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 entity.

/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:PropertyConstraints

An element specifying which other INFOD entities are qualified to interact with this consumer entity. The property constraints MAY reference attributes related to other INFOD entities (publishers, subscribers and subscriptions), associations and property vocabulary instances. The Property Constraints 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 MAY identify the set of publishers who are not authorized to send messages.

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

WS-Addressing of the action MUST contain the URI

<http://www.ggf.org/infod/INFODRegistry/CreateConsumer>

INFOD Registry Response

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

```
<infod:CreateConsumerResponse>
  <infod:INFODEntityReference>
    wsa:EndPointReferenceType
  </infod:INFODEntityReference>
</infod:CreateConsumerResponse>
```

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

/infod:INFODEntityReference

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

Instead of the CreateConsumerResponse message, the INFOD Registry MAY send the following faults in response to a CreateConsumer message:

- CreateEntityAuthorizationFailure: User not authorized to create the INFOD entity at this Registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the 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).

2.3.2 ReplaceConsumer

The ReplaceConsumer operation MUST be used to replace an INFOD consumer entity's metadata information at a given INFOD Registry. As part of the processing of a ReplaceConsumer request message, the INFOD Registry MUST replace the entire INFOD metadata for the entity representing the consumer. All previously defined values MUST be deleted. The ReplaceConsumer differs from

the CreateConsumer interface in that it replaces an existing consumer entity and assigns the original EPR to the replaced consumer.

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

```
<infod:ReplaceConsumer>
  <infod:WSEntityReference>
    wsa:EndPointReferenceType
  </infod:WSEntityReference>
  <infod:INFODEntityReference>
    wsa:EndPointReferenceType
  </infod:INFODEntityReference>
  <infod:ConsumerName> xsd:string </infod:ConsumerName> ?
  <infod:ConsumerDescription>
    xsd:string
  </infod:ConsumerDescription> ?
  <infod:PropertyConstraints>
    <xsd:complexType>
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element>tns:PropertyConstraint</xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </infod:PropertyConstraints> *
</infod:ReplaceConsumer>
```

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

/infod:WSEntityReference

A REQUIRED endpoint reference element, as defined by WS-Addressing, used to identify the WS endpoint for the entity. 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 entity.

/infod:INFODEntityReference

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

/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:PropertyConstraints

An element specifying which other INFOD entities are qualified to interact with this consumer entity. The property constraints MAY reference attributes related to other INFOD entities (publishers, subscribers and subscriptions), associations and property vocabulary instances. The Property Constraints 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 MAY identify the set of publishers who are not authorized to send messages.

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

WS-Addressing of the action **MUST** contain the URI
<http://www.ggf.org/infod/INFODRegistry/ReplaceConsumer>

INFOD Registry Response

If the INFOD Registry accepts the ReplaceConsumer message, it **MUST** respond to the WS endpoint specified in the request message with a ReplaceConsumerResponse message. The ReplaceConsumer response message is a message of the following form:

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

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

/infod:Status

An indication that the request has been successfully executed.

Instead of the ReplaceConsumerResponse message, the INFOD Registry **MAY** send the following faults in response to a ReplaceConsumer message:

- ReplaceEntityAuthorizationFailure: User not authorized to replace the INFOD entity at this Registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the 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 DropConsumer

The DropConsumer operation **MUST** be used to remove an INFOD consumer entity from an INFOD Registry.

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

```
<infod:DropConsumer>
  <infod:INFODEntityReference>
    wsa:EndPointReferenceType
  </infod:INFODEntityReference>
  <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
</infod:DropConsumer>
```

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

/infod:INFODEntityReference

An endpoint reference element, as defined by WS-Addressing, used to identify the INFOD entity 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 entity is unused and unreferenced
“DISABLE NEW”	No new references are possible for the entity. The entity will be dropped when the last reference to this entity is gone
“CASCADE”	The drop request will execute immediately and all references to the entity will be removed recursively

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

WS-Addressing of the action MUST contain the URI
<http://www.ggf.org/infod/INFODRegistry/DropConsumer>

INFOD Registry Response

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

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

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

/infod:Status

An indication that the request has been successfully executed.

Instead of the DropConsumerResponse message, the INFOD Registry MAY send the following faults in response to a DropConsumer message:

- DropEntityAuthorizationFailure: User not authorized to drop the INFOD entity at this Registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the 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/wsrp-ws_base_faults-1.2-spec-os.pdf).

2.4 Managing Subscriptions

The following operations MUST be 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 **MUST** be used by an INFOD client, referred to as a subscriber, to create an INFOD subscription in an INFOD Registry.

This subscription entity is responsible to describe the condition of interests of potential consumers for potential publishers. The registry acts on subscriptions by finding matching publishers and consumers. Using the property constraints of publishers, consumers, subscribers and subscriptions, the INFOD Registry finds the consumers and publishers matching the subscription.

A subscription **MAY** include property constraints on entities that are linked to the subscription. When the registry matches consumers with publishers through a subscription, it **MUST** take into consideration all the property constraints specified by each involved entity.

Subscriptions are the only entities that **MAY** specify data constraints and dynamic consumer constraints. Messages being sent from the publisher **MUST** satisfy the data and dynamic consumer constraints.

Any subscription that is created **MUST** reference the subscriber that was used to generate the subscription (section 2.4.1).

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 (XML schema definition of an 'infodEntity'), as follows:

```
<infod:CreateSubscription>
  <infod:SubscriptionName> xsd:string </infod:SubscriptionName> ?
  <infod:SubscriptionDescription>
    xsd:string
  </infod:SubscriptionDescription> ?
  <infod:SubscriberReference>
    wsa:EndPointReferenceType
  </infod:SubscriberReference>
  <infod:DataConstraints >
    xsd:anyType
  </infod:DataConstraints> *
  <infod:PropertyConstraints>
    <xsd:complexType>
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element>tns:PropertyConstraint</xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </infod:PropertyConstraints> *
  <infod:DynamicConsumerConstraints>
    xsd:anyType
  </infod:DynamicConsumerConstraints> *
</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:SubscriberReference

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

/infod:DataConstraints

DataConstraints specify which information is of interest to the subscriber. The constraint(s) language(s) is/are implicitly defined through the reference of the vocabulary EPR. The Data Constraints are not applied by the INFOD Registry but are handled by the publisher.

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:PropertyConstraints

An element specifying which other INFOD entities are qualified to interact with the subscription entity. The property constraints MAY reference attributes related to other INFOD entities (publishers, consumers), associations or property vocabulary instances. The Property Constraints 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.

The PropertyConstraints are the only constraints that are acted upon by the registry.

For example, a subscription MAY limit eligible publishers.

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

/infod:DynamicConsumerConstraints

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., Dynamic Consumer Constraints can not be applied by the INFOD Registry and are processed by the publishers.

infod:PropertyConstraints 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.

WS-Addressing of the action MUST contain the URI

<http://www.ggf.org/infod/INFODRegistry/CreateSubscription>

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

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

/infod:INFODEntityReference

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

Instead of the CreateSubscriptionResponse message, the INFOD Registry MAY send the following faults in response to a CreateSubscription message:

- CreateEntityAuthorizationFailure: User not authorized to create the INFOD entity at this Registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the 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/wsrfl-ws_base_faults-1.2-spec-os.pdf).

2.4.2 ReplaceSubscription

The ReplaceSubscription operation MUST be used to replace an INFOD subscriber entity's metadata information at a given INFOD Registry. As part of the processing of a ReplaceSubscription request message, the INFOD Registry MUST replace the entire INFOD metadata for the entity representing the subscription. All previously defined values MUST be deleted. The ReplaceSubscription differs from the CreateSubscription interface in that it replaces an existing subscription entity 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 entity. Details are as follows:

```
<infod:ReplaceSubscription>
  <infod:INFODEntityReference>
    wsa:EndPointReferenceType
  </infod:INFODEntityReference>
  <infod:SubscriptionName> xsd:string </infod:SubscriptionName> ?
  <infod:SubscriptionDescription>
    xsd:string
  </infod:SubscriptionDescription> ?
  <infod:SubscriberReference>
    wsa:EndPointReferenceType
  </infod:SubscriberReference>
  <infod:DataConstraints>
    xsd:anyType
  </infod:DataConstraints> *
  <infod:PropertyConstraints>
    <xsd:complexType>
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element>tns:PropertyConstraint</xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </infod:PropertyConstraints> *
  <infod:DynamicConsumerConstraints>
    xsd:anyType
  </infod:DynamicConsumerConstraints> *
</infod:ReplaceSubscription>
```

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

/infod:INFODEntityReference

An endpoint reference element, as defined by WS-Addressing, used to identify the subscription entity 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:SubscriberReference

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

/infod:DataConstraints

DataConstraints specify which information is of interest to the subscriber. The constraint(s) language(s) is/are implicitly defined through the reference of the vocabulary EPR. The Data Constraints are not applied by the INFOD Registry but are handled by the publisher.

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:PropertyConstraints

An element specifying which other INFOD entities are qualified to interact with the subscription entity. The property constraints MAY reference attributes related to other INFOD entities (publishers, consumers), associations or property vocabulary instances. The Property Constraints 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.

The PropertyConstraints are the only constraints that are acted upon by the registry.

For example, a subscription MAY limit eligible publishers.

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

/infod:DynamicConsumerConstraints

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., Dynamic Consumer Constraints can not be applied by the INFOD Registry and are processed by the publishers.

infod:PropertyConstraints 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.

WS-Addressing of the action MUST contain the URI
<http://www.ggf.org/infod/INFODRegistry/ReplaceSubscription>

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:INFODEntityReference

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

Instead of the ReplaceSubscription Response message, the INFOD Registry MAY send the following faults in response to a ReplaceSubscription message:

- ReplaceEntityAuthorizationFailure: User not authorized to replace the INFOD entity at this Registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the 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 entity from an INFOD Registry.

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

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

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

/infod:INFODEntityReference

An endpoint reference element, as defined by WS-Addressing, used to identify the INFOD subscription entity 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 entity is unused and unreferenced

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

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

WS-Addressing of the action MUST contain the URI
<http://www.ggf.org/infod/INFODRegistry/DropSubscription>

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.

Instead of the DropSubscriptionResponse message, the INFOD Registry MAY send the following faults in response to a DropSubscription message:

- DropEntityAuthorizationFailure: User not authorized to drop the INFOD entity at this Registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the 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/wsrp-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 Publisher Vocabulary
- INFOD Subscriber Vocabulary
- INFOD Consumer Vocabulary
- INFOD Subscription Vocabulary
- INFOD Association 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: Entities and Associations MAY specify properties that define their characteristics. They do that using a property vocabulary that MAY be queried. If two or more entities 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 entity, 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 vocabularies are registered in and un-registered from an INFOD Registry. It also includes operations for creating and dropping instances of a registered vocabulary.

The following operations MUST be used to manage vocabularies:

- RegisterPropertyVocabulary operation (section 2.5.1)
- CreatePropertyVocabularyInstance (section 2.5.2)
- DropPropertyVocabularyInstance (section 2.5.3)
- RegisterDataVocabulary operation (section 2.5.4)
- UnregisterVocabulary operation (section 2.5.5)

2.5.1 RegisterPropertyVocabulary

The RegisterPropertyVocabulary operation MUST be used to register a property vocabulary in an INFOD Registry. The Property Vocabulary is an XML schema.

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

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

```
<infod:RegisterPropertyVocabulary>
  <infod:VocabularyName> xsd:string </infod:VocabularyName> ?
  <infod:VocabularyDescription>
    xsd:string
  </infod:VocabularyDescription> ?
  <infod:VocabularyBody>
    xsd:anyType
  </infod:VocabularyBody>
</infod:RegisterPropertyVocabulary>
```

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

/infod:VocabularyName

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

The following names MUST NOT be used: Publishers, Subscribers, Consumers, Subscriptions, PropertyVocabularies, DataVocabularies and Associations.

The following names are reserved for future use and also MUST NOT be used: Disseminators, POBoxes and any name that starts with infod.

/infod:VocabularyDescription

A string representing a description of the vocabulary.

/infod:VocabularyBody

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

WS-Addressing of the action MUST contain the URI <http://www.ggf.org/infod/INFODRegistry/RegisterPropertyVocabulary>.

INFOD Registry Response

If the INFOD Registry accepts the RegisterPropertyVocabulary request, it MUST respond to the WS endpoint specified in the request message with a RegisterVocabularyResponse message.

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

```
<infod:RegisterPropertyVocabularyResponse>
  <infod:INFODVocabularyReference>
    wsa:EndPointReferenceType
  </infod:INFODVocabularyReference>
</infod:RegisterVocabularyResponse>
```

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

/infod:INFODVocabularyReference

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

In case of an error, the INFOD Registry MAY send the following faults in response to a RegisterPropertyVocabulary request message:

- RegisterVocabularyAuthorizationFailure: User not authorized to register a vocabulary at this Registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the 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 CreatePropertyVocabularyInstance

The CreatePropertyVocabularyInstance operation MUST be used to create a new instance of a particular property vocabulary previously registered at 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 references a particular INFOD entity to the instance. The referenced entity 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:VocabularyInstanceEntityReference>
    wsa:EndPointReferenceType
  </infod:VocabularyInstanceEntityReference>
  <infod:VocabularyInstanceVocabularyReference>
    wsa:EndPointReferenceType
  </infod:VocabularyInstanceVocabularyReference>
  <infod:VocabularyInstanceVocabularyBody>
    {xsd:anyType} ?
  </infod:VocabularyInstanceBody>
</infod:CreatePropertyVocabularyInstance>
```

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

/infod:VocabularyInstanceEntityReference

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

/infod:VocabularyInstanceVocabularyReference

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

/infod:VocabularyInstanceBody

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

WS-Addressing of the action **MUST** contain the URI

<http://www.ggf.org/infod/INFODRegistry/CreatePropertyVocabularyInstance>.

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

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

/infod:INFODVocabularyInstanceReference

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

In case of an error, the INFOD Registry MAY send the following faults in response to a CreatePropertyVocabularyInstance message:

- CreatePropertyVocabularyInstanceAuthorizationFailure: User not authorized to create the INFOD property vocabulary instance at this Registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the 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).

2.5.3 DropPropertyVocabularyInstance

The DropPropertyVocabularyInstance operation MUST be used to drop an existing instance of a particular property vocabulary previously created at the INFOD Registry.

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

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

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

/infod:INFODVocabularyInstanceReference

An endpoint reference element, as defined by WS-Addressing, used to identify the 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 entity is unused and unreferenced |
| “DISABLE NEW” | No new references are possible for the entity. The entity will be dropped when the last reference to this entity is gone |
| “CASCADE” | The drop request will execute immediately and all references to the entity will be removed recursively |

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

WS-Addressing of the action MUST contain the URI <http://www.ggf.org/infod/INFODRegistry/DropPropertyVocabularyInstance>.

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.

In case of an error, the INFOD Registry MAY send the following faults in response to a DropPropertyVocabularyInstance message:

- DropPropertyVocabularyInstanceAuthorizationFailure: User not authorized to drop the INFOD property vocabulary instance at this Registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the 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).

2.5.4 RegisterDataVocabulary

The RegisterDataVocabulary operation MUST be used to register a data vocabulary in an INFOD Registry. The Data Vocabulary defines the structure of the vocabulary and MAY be used to define constraints on messages. In order to have a publisher entity use a data vocabulary, it MUST create an association to a data vocabulary (section 2.6.1).

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

The format of the request message for RegisterDataVocabulary operation is:

```

<infod:RegisterDataVocabulary>
  <infod:VocabularyName> xsd:string </infod:VocabularyName> ?
  <infod:VocabularyDescription>
    xsd:string
  </infod:VocabularyDescription> ?
  <infod:VocabularyLanguage>
    {anyURI} (Namespace/URI of DataFormat)
  </infod:VocabularyLanguage>
  <infod:LanguageUsageDescription>
    xsd:string
  </infod:LanguageUsageDescription> ?
  <infod:VocabularyBody>
    xsd:anyType
  </infod:VocabularyBody>
</infod:RegisterDataVocabulary>

```

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

/infod:VocabularyName

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

The following names MUST NOT be used: Publishers, Subscribers, Consumers, Subscriptions, PropertyVocabularies, DataVocabularies and Associations.

The following names are reserved for future use and also MUST NOT be used: Disseminators, POBoxes and any name that starts with infod.

/infod:VocabularyDescription

A string representing a description of the vocabulary.

/infod:VocabularyLanguage

A URI defining the format of the data vocabulary.

/infod:LanguageUsageDescription

This string MAY be used to express how the vocabulary language MAY be interpreted (Example: "the xml in this vocabulary may be used with the following interpretation")

/infod:VocabularyBody

A string representing a data vocabulary.

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

The INFOD specification allows communities of interest to form around accepted VocabularyLanguage definitions and Subscribers, Publishers, and Consumers will be able to interoperate. The INFOD Registry operation does not impede this from happening.

WS-Addressing of the action MUST contain the URI
<http://www.ggf.org/infod/INFODRegistry/RegisterDataVocabulary>.

INFOD Registry Response

If the INFOD Registry accepts the RegisterDataVocabulary request, it MUST respond to the WS endpoint specified in the request message with a RegisterVocabularyResponse message.

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

```
<infod:RegisterVocabularyResponse>
  <infod:INFODVocabularyReference>
    wsa:EndPointReferenceType
  </infod:INFODVocabularyReference>
</infod:RegisterVocabularyResponse>
```

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

/infod:INFODVocabularyReference

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

In case of an error, the INFOD Registry MAY send the following faults in response to a RegisterPropertyVocabulary request message:

- RegisterVocabularyAuthorizationFailure: User not authorized to register a vocabulary at this Registry

- **UnknownElementReferenceFault:** An element has been referenced that is unknown to the 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).

2.5.5 UnregisterVocabulary

The UnregisterVocabulary operation **MUST** be used to un-register a particular data or property vocabulary from an INFOD Registry.

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

```
<infod:UnregisterVocabulary>
  <infod:INFODVocabularyReference>
    wsa:EndPointReferenceType
  </infod:INFODVocabularyReference>
  <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
</infod:UnregisterVocabulary>
```

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

/infod:INFODVocabularyReference

An endpoint reference element, as defined by WS-Addressing, used to identify the vocabulary to un-register 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 entity is unused and unreferenced |
| “DISABLE NEW” | No new references are possible for the entity. The entity will be dropped when the last reference to this entity is gone |
| “CASCADE” | The drop request will execute immediately and all references to the entity will be removed recursively |

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

WS-Addressing of the action **MUST** contain the URI

<http://www.ggf.org/infod/INFODRegistry/UnregisterVocabulary>

INFOD Registry Response

If the INFOD Registry accepts the UnregisterVocabulary request, it **MUST** respond to the WS endpoint specified in the request message with an UnregisterVocabularyResponse message. The UnregisterVocabulary response message is a message of the following form:

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

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

/infod:Status

An indication that the request has been successfully executed.

In case of an error, the INFOD Registry MAY send the following faults in response to a UnregisterVocabulary request message:

- UnregisterVocabularyAuthorizationFailure: User not authorized to un-register the INFOD vocabulary instance at this Registry
- CreateEntityAuthorizationFailure: User not authorized to create the INFOD entity at this Registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the 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 Managing Associations

An INFOD entity MAY be associated with a data vocabulary. This type of association is herein referred to as 'vocabulary association'. Associating a data vocabulary with an INFOD entity is used to enable the use of Data Vocabulary constraints by a subscription (section 2.2.1). In order for publishers to send out messages to consumers, they MUST agree on a data vocabulary. This is done by the publisher associating itself to a data vocabulary a subscription is referencing.

The following operations MAY be used to manage associations:

- AssociateVocabulary (section 2.6.1)
- DisassociateVocabulary (section 2.6.2)

2.6.1 AssociateVocabulary

The AssociateVocabulary operation MUST be used to create a relation between an INFOD entity and a data vocabulary at the INFOD Registry.

As part of the processing of an AssociateVocabulary operation message, the INFOD Registry MUST create an INFOD vocabulary association resource.

The format of the request message for an AssociateVocabulary operation is similar to the request message for an AssociateEntity operation.

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

```
<infod:AssociateVocabulary>
  <infod:AssociateVocabularyName> ?
    xsd:string
  </infod:AssociateVocabularyName>
  <infod:AssociateVocabularyDescription>
    xsd:string
  </infod:AssociateVocabularyDescription> ?
  <infod:AssociationEntityReference>
    wsa:EndPointReferenceType
  </infod:AssociationEntityReference>
```

```

<infod:VocabularyReference>
  wsa:EndPointReferenceType
</infod:VocabularyReference> +
<infod:PropertyConstraints>
  <xsd:complexType>
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
      <xsd:element>tns:PropertyConstraint</xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</infod:PropertyConstraints> ?
</infod:AssociateVocabulary>

```

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

/infod:AssociateVocabularyName

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

/infod:AssociateVocabularyDescription

A string representing a description of the vocabulary association..

/infod:AssociationEntityReference

The EPR of the INFOD entity for which a vocabulary association is created.

/infod:VocabularyReference

One or more EPR(s) of registered vocabulary with which to associate the primary INFOD entity as defined in AssociationEntityReference.

/infod:PropertyConstraints

An element specifying which other INFOD entities that are qualified to interact with the publisher entity through this association. The property constraints MAY reference attributes related to INFOD entities (publishers, subscriber, consumers, and subscribers) or property vocabulary instances. The Property Constraints MUST be formulated as an XQuery. For examples of XQueries see the INFOD Base Use Case Scenarios (see <http://forge.gridforum.org/sf/go/doc13626?nav=1>).

While the constraints defined for publishers apply to all interaction of that publisher with other entities the constraints specified in the Association only apply to this association. So this is a specialization of the more generic publisher PropertyConstraints attribute.

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

WS-Addressing of the action MUST contain the URI
<http://www.ggf.org/infod/INFODRegistry/AssociateVocabulary>.

INFOD Registry Response

If the INFOD Registry accepts the AssociateVocabulary request, it MUST respond to the WS endpoint specified in the request message with an AssociateVocabularyResponse message.

The AssociateVocabularyResponse message is a message of the following form:

```

<infod:AssociateVocabularyResponse>
  <infod:INFODAssociationReference>
    wsa:EndPointReferenceType
  </infod:INFODAssociationReference>

```

```
</infod:AssociateVocabularyResponse>
```

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

/infod:INFODAssociationReference

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

Instead of the AssociateVocabularyResponse message, the INFOD Registry MAY send the following faults in response to an AssociateVocabulary message:

- CreateAssociationAuthorizationFailure: User not authorized to create the association at this Registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the 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/wsr/wsr/ws_base_faults-1.2-spec-os.pdf).

2.6.2 DisassociateVocabulary

The DisassociateVocabulary operation MUST be used to drop an association from an INFOD Registry.

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

```
<infod:DisassociateVocabulary>
  <infod:INFODAssociationReference>
    wsa:EndPointReferenceType
  </infod:INFODAssociationReference>
  <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
</infod:DisassociateVocabulary>
```

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

/infod: INFODAssociationReference

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

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

WS-Addressing of the action MUST contain the URI
<http://www.ggf.org/infod/INFODRegistry/DisassociateVocabulary>.

INFOD Registry Response

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

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

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

/infod:Status

An indication that the request has been successfully executed.

Instead of the DisassociateVocabularyResponse message, the INFOD Registry MAY send the following faults in response to a DisassociateVocabulary message:

- DisAssociationAuthorizationFailure: User not authorized to drop the association at this Registry
- UnknownElementReferenceFault: An element has been referenced that is unknown to the 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 MUST be formulated as an XQuery, 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 for the INFOD registry.

The INFOD registry is fully qualified by an INFOD registry service name appended to the string "INFODRegistry.xml".

An example for a fully qualified INFOD registry is: <file://c:/data/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 "file:///c:/data/xml/". This indirection allows us to specify multiple registries in a given environment.

The XQuery MUST be re-written by the INFOD registry to include relevant property constraints that are defined in the INFOD registry.

Examples:

fn:doc('INFODRegistry.xml')/consumers/infodConsumer identifies all consumers of a registry

fn:doc('INFODRegistry.xml')/vocabularies/infodvocabulary/[infodfn:epr = EPR] identifies all instances related to the vocabulary with the specified EPR.

More examples for GetMetaData can be found in the INFOD Base Use Case Scenarios (see <http://forge.gridforum.org/sf/go/doc13626?nav=1>).

Conventions:

- **All publishers** - fn:doc('INFODRegistry.xml')/pubsihers/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 vocababulary instances:**
fn:doc('INFODRegistry.xml')/propertyvocabulariyinstances/infodPropertyVocabularyInstanc
- **All data vocabulary** - fn:doc('INFODRegistry.xml')/datavocabularies/infodDataVocabulary

WS-Addressing of the action MUST contain the URI
<http://www.ggf.org/infod/INFODRegistry/GetMetaData>.

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.

Instead of the GetMetaDataQueryResult message, the INFOD Registry MAY send the following faults in response to a GetMetaData message:

- **GetMetaDataAuthorizationFailure:** User not authorized to create the INFOD entity at this 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/wsrfl-ws_base_faults-1.2-spec-os.pdf).

3 The Base INFOD Notify Interface

The table below lists the operations of the base INFOD notify interface and the section that describes it in detail.

The Base INFOD Notify Interface:

Operation	Description	Section
Notify	<p>This operation defines how an INFOD message is consumed by an INFOD consumer.</p> <p>Note: INFOD Messages MAY consist of data and/or metadata (in other words, some INFOD messages only deliver INFOD metadata information).</p>	3.1.1

3.1.1 Notify

An INFOD publisher uses the Notify operation as 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:

```
<wsnt:Notify>
  <wsnt:NotificationMessage>
    <wsnt:SubscriptionReference>
      wsa:EndpointReferenceType
    </wsnt:SubscriptionReference> ?
    <wsnt:Topic Dialect="xsd:anyURI">
      {any} ?
    </wsnt:Topic>?
    <wsnt:ProducerReference>
      wsa:EndpointReferenceType
    </wsnt:ProducerReference> ?
    <wsnt:Message>
      {any}
    </wsnt:Message>
  </wsnt:NotificationMessage> +
  {any} *
</wsnt: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 EndpointReference to the Subscription that is associated with the Notify message.

wsnt:Topic

An EndpointReference to the VocabularyAssociation representing the source of the payload.

wsnt:Topic/@Dialect

An Endpoint Reference to the vocabulary that was used to structure the payload.

/infod:ProducerReference

An EndpointReference to the Publisher that produced the Notification.

/infod:Message

A copy of 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.

WS-Addressing of the action MUST contain the URI <http://www.ggf.org/infod/INFODNotify/Notify>.

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

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

3.1.2 Notification of Publishers

The INFOD registry will send one message to each publisher who has to react to a (matching) subscription and who has requested to be notified..

There are several cases to consider:

- A subscriber is new or has been replaced – notifications about all matching subscriptions will be sent to the publisher
- A subscription is new or has been replaced – notifications about will be sent to all (matching) publishers
- A subscription has been dropped – notifications will be sent to all previously (matching) publishers
- Any other change to the INFOD registry:
 - If there is a new match between a subscription and a publisher – the publisher will be notified
 - If there was a match and there is none any more between a publisher and a subscription – the publisher will be notified –
 - If there is no match any more between a publisher and a subscription – the notification will only have a reference to an EPR the old notification with the same subscription EPR should not be used any more

The rules for publisher are very simple: if a notification arrives publish immediately according to this notification.

If specified, INFOD will inform publishers if changes in the INFOD registry resulted in adjustment of those subscriptions that each publisher MUST react to. Publishers MUST react immediately to these notifications.

The notification contains the following message body:

```
<infod:SubscriptionNotification>
  <infod:SubscriptionReference>
    wsa:EndPointReferenceType
  </infod:SubscriptionReference>
  <infod:StaticConsumers>
    wsa:EndPointReferenceType
  <infod:StaticConsumers> *
  <infod:DynamicConsumerConstraints>
    {xsd:anyType}
  <infod:DynamicConsumerConstraints> *
  <infod>DataConstraint>
    {xsd:anyType}
  <infod>DataConstraint> *
</infod:SubscriptionNotification> +
```

The message content is further described as follows:

/infod:SubscriptionReference

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

`/infod:StaticConsumers`

This is a list of 0 to n EPR references of consumers. The list of consumers is computed by the INFOD Registry and given to each publisher, and hence is therefore static.

`/infod:DynamicConsumerConstraints`

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 entities.

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

Please refer to the INFOD Base Use Case Scenarios (see <http://forge.gridforum.org/sf/go/doc13626?nav=1>) for use cases.

`/infod:DataConstraint`

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

WS-Addressing of the action MUST contain the URI
<http://www.ggf.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 it is imperative that such communications use SOAP 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, INFO-D 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 entities such as publisher and consumer serves as the basis for establishing the security context. Establishing a security context between system entities 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.
- 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 future, it is envisaged that in the latter 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 the either, 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 allows formation of dissemination networks that allow publishers to propagate messages to an unspecified list of recipients. In such cases it is imperative that the 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 for retrieval by the network entities. However, if the disseminator enriches the message, the new contents need to be signed by the disseminator.

In some cases, INFO-D publishers can determine the list of consumers and can provide messages for consumption by a single or a group of consumers. No present security mechanism supports such communication pattern without the establishment of a shared key between the group of consumers and the publisher. As INFOD does not support conversational pattern, but is limited to support one-way flow of messages, it is therefore restricted in its ability to establish secured communication in the above-mentioned pattern.

4.2 Integration with authorization model

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

- Access model for INFOD resources
- Access model for INFOD messages

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

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

5 Appendix I – XML Schema

This section includes the following xml schema:

- INFOD Publishers (section 5.1)
- INFOD Subscribers (section 0)
- INFOD Consumers (section 5.3)
- INFOD Subscriptions (section 5.4)
- INFOD Property Vocabularies (section 5.5)
- INFOD Property Vocabulary Instances (section 5.6)
- INFOD Data Vocabularies (section 5.7)
- INFOD Associations(section 5.8)
- INFOD Error Messages (section 5.9)
- INFOD Notification (section 5.10)
- INFOD Notification of Publishers (section 5.11)

The following graphic depicts the XML schema relations for the www.ggf.org/infod/INFODRegistry Namespace. The green circles represent the operations of the INFODRegistry. The red circles show the vocabularies that are managed by the vocabulary operations. The light blue boxes represent the entities, associations and property vocabulary instances. Within the xml schema of those boxes, there are reference pointers to other entities or vocabularies, represented by EPRs. The yellow honeycombs represent the external web services EPRs that are associated to the entities. Note that the same Web Service EPR can be associated to multiple INFOD entities.

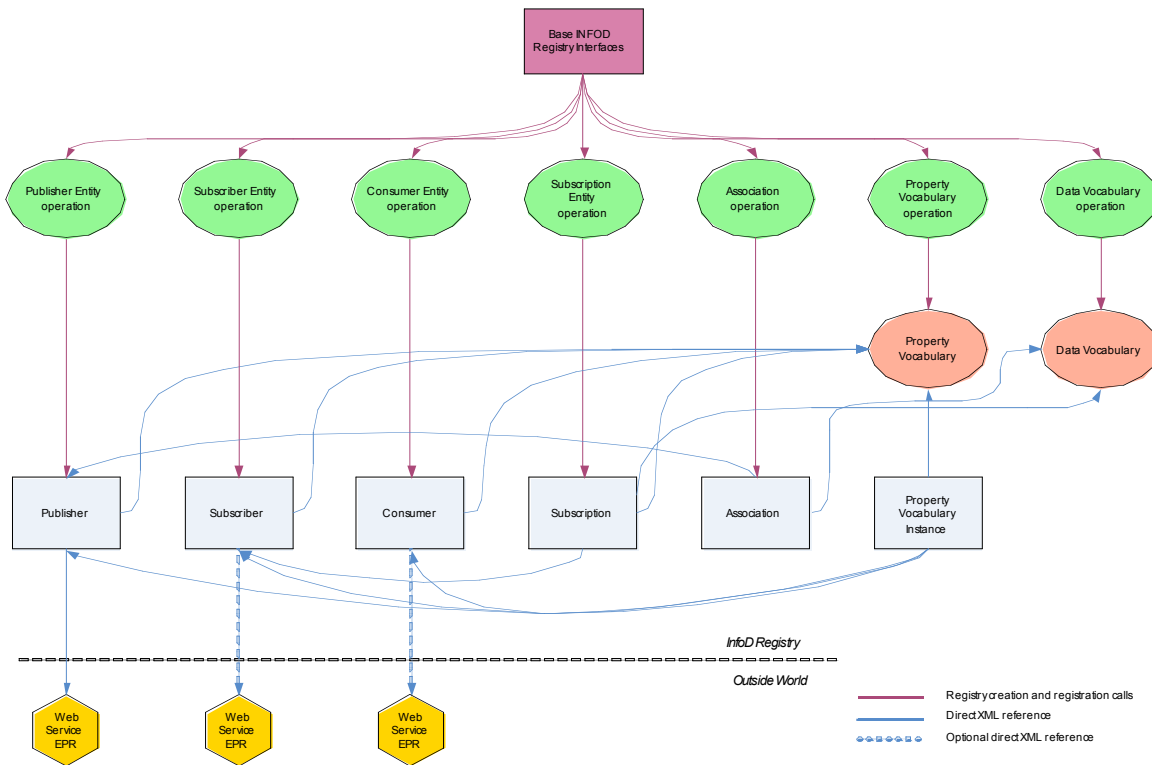


Figure 4: XML schema relations of INFODRegistry namespace

5.1 Publishers

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:ident="http://www.ggf.org/infod"
            targetNamespace="http://www.ggf.org/infod/INFODRegistry">
  <xsd:annotation>
    <xsd:documentation> Publishers Description </xsd:documentation>
  </xsd:annotation>
  <xsd:element name="infodPublisher">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="WSEntityReference"
                    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="PropertyConstraints" 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 Subscribers

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

  <xsd:element name="infodSubscriber">
    <xsd:annotation>
      <xsd:documentation>
        Description of Subscribers
      </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="WSEntityReference"
          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="PropertyConstraints" type="xsd:any"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

5.3 Consumers

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

  <xsd:element name="infodConsumer">
    <xsd:annotation>
      <xsd:documentation>
        Description of Consumers
      </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="WSEntityReference"
          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="PropertyConstraints" type="xsd:any"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

5.4 Subscriptions

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:infod="http://www.ggf.org/infod"
            targetNamespace="http://www.ggf.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="SubscriberReference"
                    type="wsa:EndpointReferenceType"
                    minOccurs="0" maxOccurs="1"/>
        <xsd:element name="DataConstraints" type="xsd:any"
                    minOccurs="0" maxOccurs="1"/>
        <xsd:element name="PropertyConstraints" type="xsd:any"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="DynamicConsumerConstraints" type="xsd:any"
                    minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

5.5 Property Vocabularies

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:infod="http://www.ggf.org/infod"
            targetNamespace="http://www.ggf.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="VocabularyName" type="xsd:string"
                    minOccurs="0" maxOccurs="1"/>
        <xsd:element name="VocabularyDescription" type="xsd:string"
                    minOccurs="0" maxOccurs="1"/>
        <xsd:element name="VocabularyBody" type="xsd:any"
                    minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

5.6 Property Vocabulary Instances

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:infod="http://www.ggf.org/infod"
            targetNamespace="http://www.ggf.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="VocabularyInstanceEntityReference"
                    type="wsa:EndpointReferenceType"
                    minOccurs="1" maxOccurs="1"/>
        <xsd:element name="VocabularyInstanceVocabularyReference"
                    type="wsa:EndpointReferenceType"
                    minOccurs="1" maxOccurs="1"/>
        <xsd:element name="VocabularyInstanceBody"
                    type="xsd:any"
                    minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

5.7 Data Vocabularies

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:infod="http://www.ggf.org/infod"
            targetNamespace="http://www.ggf.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="VocabularyName" type="xsd:string"
                    minOccurs="0" maxOccurs="1"/>
        <xsd:element name="VocabularyDescription" type="xsd:string"
                    minOccurs="0" maxOccurs="1"/>
        <xsd:element name="VocabularyLanguage" type="xsd:string"
                    minOccurs="1" maxOccurs="1"/>
        <xsd:element name="LanguageUsageDescription"
                    type="xsd:string"
                    minOccurs="0" maxOccurs="1"/>
        <xsd:element name="VocabularyBody" type="xsd:any"
                    minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

5.8 Associations

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

  <xsd:element name="infodAssociation">
    <xsd:annotation>
      <xsd:documentation>
        Description of Associations
      </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="AssociateVocabularyName" type="xsd:string"
                    minOccurs="0" maxOccurs="1"/>
        <xsd:element name="AssociateVocabulryDescription"
                    type="xsd:string"
                    minOccurs="0" maxOccurs="1"/>
        <xsd:element name="AssociationEntityReference"
                    type="wsa:EndpointReferenceType"
                    minOccurs="1" maxOccurs="1"/>
        <xsd:element name="AssociationVocabularyReference"
                    type="wsa:EndpointReferenceType"
                    minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PropertyConstraints" 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:wsr-f-bf="http://www.ggf.org/infod/fault"
            xmlns:ident="http://www.ggf.org/infod"
            targetNamespace="http://www.ggf.org/infod/INFODRegistry">

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

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

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

```

<xsd:element name="DropEntityAuthorizationFailure"
             type="infod:DropEntityAuthorizationFailureType"/>

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

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

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

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

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

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

<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="UnknownElementReferenceFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="UnknownElementReferenceFault"
  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.ggf.org/infod"
  targetNamespace="http://www.ggf.org/infod/INFODNotify">

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

```

```

<xsd:element name="SubscriptionReference"
  type="wsa:EndpointReferenceType"/>
<xsd:element name="Topic"
  type="wsnt: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="wsnt: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 NotificationConsumer ===== -->

<xsd:element name="Notify">
  <xsd:annotation>
    <xsd:documentation> Notification to 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.ggf.org/infod"

```

```
targetNamespace="http://www.ggf.org/infod/INFODNotify">
<xsd:element name="SubscriptionNotification">
  <xsd:annotation>
    <xsd:documentation> Notification to 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="StaticConsumers"
            type="xsd:any"
            minOccurs="1" maxOccurs="unbounded"/>
          <xsd:element name="DynamicConsumers"
            type="xsd:any"
            minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
          <xsd:element name="StaticConsumers"
            type="xsd:any"
            minOccurs="0" maxOccurs="unbounded"/>
          <xsd:element name="DynamicConsumerConstraints"
            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>
```


6 Appendix II – WSDL 1.1

```

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

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

<wsdl:types>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" targetNamespace="http://www.w3.org/INFOD"
xmlns:infodxsd="http://www.w3.org/INFOD"></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.w3.org/INFOD"
schemaLocation="infodTypes.xsd"></xsd:import>
</xsd:schema>

</wsdl:types>

<wsdl:message name="CreatePublisherRequest">
<wsdl:part name="WSEntityReference"
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:message>

<wsdl:message name="CreatePublisherResponse">
<wsdl:part name="INFODEntityReference"
element="infod:EndPointReference"></wsdl:part>
</wsdl:message>

<wsdl:message name="ReplacePublisherRequest">
<wsdl:part name="WSEntityReference"
element="wsa:EndPointReference"></wsdl:part>
<wsdl:part name="INFODEntityReference"
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="ReplacePublisherResponse">

```

```

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

<wsdl:message name="DropPublisherRequest">
<wsdl:part name="INFODEntityReference"
element="infod:EndPointReference"></wsdl:part>
<wsdl:part name="ExecutionMode" element="infod:ExecutionMode"></wsdl:part>
</wsdl:message>

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

<wsdl:message name="CreateSubscriberRequest">
<wsdl:part name="WSEntityReference"
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:message>

<wsdl:message name="CreateSubscriberResponse">
<wsdl:part name="INFODEntityReference"
element="infod:EndPointReference"></wsdl:part>
</wsdl:message>

<wsdl:message name="ReplaceSubscriberRequest">
<wsdl:part name="WSEntityReference"
element="wsa:EndPointReference"></wsdl:part>
<wsdl:part name="INFODEntityReference"
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:message>

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

<wsdl:message name="DropSubscriberRequest">
<wsdl:part name="INFODEntityReference"
element="infod:EndPointReference"></wsdl:part>
<wsdl:part name="ExecutionMode" element="infod:ExecutionMode"></wsdl:part>
</wsdl:message>

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

<wsdl:message name="CreateConsumerRequest">
<wsdl:part name="WSEntityReference"
element="wsa:EndPointReference"></wsdl:part>
<wsdl:part name="INFODEntityReference"
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:message>

<wsdl:message name="CreateConsumerResponse">
<wsdl:part name="INFODEntityReference"
element="infod:EndPointReference"></wsdl:part>
</wsdl:message>

<wsdl:message name="ReplaceConsumerRequest">
<wsdl:part name="WSEntityReference"
element="wsa:EndPointReference"></wsdl:part>
<wsdl:part name="INFODEntityReference"
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:message>

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

<wsdl:message name="DropConsumerRequest">
<wsdl:part name="INFODEntityReference"
element="infod:EndPointReference"></wsdl:part>
<wsdl:part name="ExecutionMode" element="infod:ExecutionMode"></wsdl:part>
</wsdl:message>

<wsdl:message name="DropConsumerResponse">
<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="WSEntityReference"
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="INFODEntityReference"
element="wsa:EndPointReference"></wsdl:part>
</wsdl:message>

<wsdl:message name="ReplaceSubscriptionRequest">
<wsdl:part name="INFODEntityReference"
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="WSEntityReference"
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="INFODEntityReference"
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="RegisterPropertyVocabularyRequest">
<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="RegisterPropertyVocabularyResponse">
<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:message>

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

<wsdl:message name="RegisterDataVocabularyRequest">
<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="RegisterDataVocabularyResponse">
<wsdl:part name="INFODVocabularyReference"
element="wsa:EndPointReference"></wsdl:part>
</wsdl:message>

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

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

<wsdl:message name="AssociateVocabularyRequest">
<wsdl:part name="AssociateVocabularyName" type="xsd:string"></wsdl:part>
<wsdl:part name="AssociateVocabularyDescription"
type="xsd:string"></wsdl:part>
<wsdl:part name="AssociationEntityReference"
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="AssociateVocabularyResponse">
<wsdl:part name="INFODAssociationReference"
element="wsa:EndPointReference"></wsdl:part>
</wsdl:message>

<wsdl:message name="DisassociateVocabularyRequest">
<wsdl:part name="INFODAssociationReference"
element="wsa:EndPointReference"></wsdl:part>
<wsdl:part name="ExecutionMode" element="infod:ExecutionMode"></wsdl:part>
</wsdl:message>

<wsdl:message name="DisassociateVocabularyResponse">
<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="CreateEntityAuthorizationFailureErrorMessage">

```

```

<wsdl:part name="err"
element="infod:CreateEntityAuthorizationFailure"></wsdl:part>
</wsdl:message>
<wsdl:message name="UnknownElementReferenceFaultErrorMessage">
<wsdl:part name="err"
element="infod:UnknownElementReferenceFault"></wsdl:part>
</wsdl:message>
<wsdl:message name="MissingRequiredParameterFaultErrorMessage">
<wsdl:part name="err"
element="infod:MissingRequiredParameterFault"></wsdl:part>
</wsdl:message>
<wsdl:message name="UnsupportedXQueryFaultErrorMessage">
<wsdl:part name="err" element="infod:UnsupportedXQueryFault"></wsdl:part>
</wsdl:message>
<wsdl:message name="ReplaceEntityAuthorizationFaultErrorMessage">
<wsdl:part name="err"
element="infod:ReplaceEntityAuthorizationFailure"></wsdl:part>
</wsdl:message>
<wsdl:message name="DropEntityAuthorizationFaultErrorMessage">
<wsdl:part name="err"
element="infod:DropEntityAuthorizationFailure"></wsdl:part>
</wsdl:message>
<wsdl:message name="ExecutionModeFaultErrorMessage">
<wsdl:part name="err" element="infod:ExecutionModeFault"></wsdl:part>
</wsdl:message>

<wsdl:message name="RegisterVocabularyAuthorizationFailureErrorMessage">
<wsdl:part name="err"
element="infod:RegisterVocabularyAuthorizationFailure"></wsdl:part>
</wsdl:message>

<wsdl:message name="UnsupportedVocabularyFaultErrorMessage">
<wsdl:part name="err"
element="infod:UnsupportedVocabularyFault"></wsdl:part>
</wsdl:message>

<wsdl:message
name="CreatePropertyVocabularyInstanceAuthorizationFailureErrorMessage">
<wsdl:part name="err"
element="infod:CreatePropertyVocabularyInstanceAuthorizationFailure"></wsdl
:part>
</wsdl:message>

<wsdl:message name="CreateAssociationAuthorizationFailureErrorMessage">
<wsdl:part name="err"
element="infod:CreateAssociationAuthorizationFailure"></wsdl:part>
</wsdl:message>
<wsdl:message name="DisAssociationAuthorizationFailureErrorMessage">
<wsdl:part name="err"
element="infod:DisAssociationAuthorizationFailure"></wsdl:part>
</wsdl:message>
<wsdl:portType name="infodRegistry">
<wsdl:operation name="CreatePublisher">
<wsdl:input message="infod:CreatePublisherRequest"
name="createPublisherMessage"></wsdl:input>
<wsdl:output message="infod:CreatePublisherResponse"></wsdl:output>
<wsdl:fault message="infod:CreateEntityAuthorizationFailureErrorMessage"
name="CreateEntityAuthorizationFailure"></wsdl:fault>

```

```

<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></wsdl:fault>
<wsdl:fault message="infod:MissingRequiredParameterFaultErrorMessage"
name="UnsupportedXQueryFault"></wsdl:fault>
<wsdl:fault message="infod:UnsupportedXQueryFaultErrorMessage"
name="UnsupportedXQueryFault"></wsdl:fault>
</wsdl:operation>

<wsdl:operation name="ReplacePublisher">
<wsdl:input message="infod:ReplacePublisherRequest"
name="ReplacePublisherMessage"></wsdl:input>
<wsdl:output message="infod:ReplacePublisherResponse"
name="ReplacePublisherResponse"></wsdl:output>
<wsdl:fault message="infod:ReplaceEntityAuthorizationFaultErrorMessage"
name="ReplaceEntityAuthorizationFault"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></wsdl:fault>
<wsdl:fault message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingrequiredParameterFault"></wsdl:fault>
<wsdl:fault message="infod:UnsupportedXQueryFaultErrorMessage"
name="UnsupportedXQueryFault"></wsdl:fault>
</wsdl:operation>

<wsdl:operation name="DropPublisher">
<wsdl:input message="infod:DropPublisherRequest"
name="DropPublisherRequestMessage"></wsdl:input>
<wsdl:output message="infod:DropPublisherResponse"
name="DropPublisherResponseMessage"></wsdl:output>
<wsdl:fault message="infod:DropEntityAuthorizationFaultErrorMessage"
name="DropEntityAuthorizationFailure"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></wsdl:fault>
<wsdl:fault message="infod:MissingRequiredParameterFaultErrorMessage"
name="UnsupportedXQueryFault"></wsdl:fault>
<wsdl:fault message="infod:ExecutionModeFaultErrorMessage"
name="ExecutionModeFault"></wsdl:fault>
</wsdl:operation>

<wsdl:operation name="CreateSubscriber">
<wsdl:input message="infod:CreateSubscriberRequest"></wsdl:input>
<wsdl:output message="infod:CreateSubscriberResponse"></wsdl:output>
<wsdl:fault message="infod:CreateEntityAuthorizationFailureErrorMessage"
name="CreateEntityAuthorizationFailure"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></wsdl:fault>
<wsdl:fault message="infod:MissingRequiredParameterFaultErrorMessage"
name="UnsupportedXQueryFault"></wsdl:fault>
<wsdl:fault message="infod:UnsupportedXQueryFaultErrorMessage"
name="UnsupportedXQueryFault"></wsdl:fault>
</wsdl:operation>

<wsdl:operation name="ReplaceSubscriber">
<wsdl:input message="infod:ReplaceSubscriberRequest"></wsdl:input>
<wsdl:output message="infod:ReplaceSubscriberResponse"></wsdl:output>
<wsdl:fault message="infod:ReplaceEntityAuthorizationFaultErrorMessage"
name="ReplaceEntityAuthorizationFault"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></wsdl:fault>
<wsdl:fault message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingrequiredParameterFault"></wsdl:fault>

```

```

<wsdl:fault message="infod:UnsupportedXQueryFaultErrorMessage"
name="UnsupportedXQueryFault"></wsdl:fault>
</wsdl:operation>

<wsdl:operation name="DropSubscriber">
<wsdl:input message="infod:DropSubscriberRequest"></wsdl:input>
<wsdl:output message="infod:DropSubscriberResponse"></wsdl:output>
<wsdl:fault message="infod:DropEntityAuthorizationFaultErrorMessage"
name="DropEntityAuthorizationFailure"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></wsdl:fault>
<wsdl:fault message="infod:MissingRequiredParameterFaultErrorMessage"
name="UnsupportedXQueryFault"></wsdl:fault>
<wsdl:fault message="infod:ExecutionModeFaultErrorMessage"
name="ExecutionModeFault"></wsdl:fault>
</wsdl:operation>

<wsdl:operation name="CreateConsumer">
<wsdl:input message="infod:CreateConsumerRequest"></wsdl:input>
<wsdl:output message="infod:CreateConsumerResponse"></wsdl:output>
<wsdl:fault message="infod:CreateEntityAuthorizationFailureErrorMessage"
name="CreateEntityAuthorizationFailure"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></wsdl:fault>
<wsdl:fault message="infod:MissingRequiredParameterFaultErrorMessage"
name="UnsupportedXQueryFault"></wsdl:fault>
<wsdl:fault message="infod:UnsupportedXQueryFaultErrorMessage"
name="UnsupportedXQueryFault"></wsdl:fault>
</wsdl:operation>

<wsdl:operation name="ReplaceConsumer">
<wsdl:input message="infod:ReplaceConsumerRequest"></wsdl:input>
<wsdl:output message="infod:ReplaceConsumerResponse"></wsdl:output>
<wsdl:fault message="infod:ReplaceEntityAuthorizationFaultErrorMessage"
name="ReplaceEntityAuthorizationFailure"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></wsdl:fault>
<wsdl:fault message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingrequiredParameterFault"></wsdl:fault>
<wsdl:fault message="infod:UnsupportedXQueryFaultErrorMessage"
name="UnsupportedXQueryFault"></wsdl:fault>
</wsdl:operation>

<wsdl:operation name="DropConsumer">
<wsdl:input message="infod:DropConsumerRequest"></wsdl:input>
<wsdl:output message="infod:DropConsumerResponse"></wsdl:output>
<wsdl:fault message="infod:DropEntityAuthorizationFaultErrorMessage"
name="DropEntityAuthorizationFailure"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></wsdl:fault>
<wsdl:fault message="infod:MissingRequiredParameterFaultErrorMessage"
name="UnsupportedXQueryFault"></wsdl:fault>
<wsdl:fault message="infod:ExecutionModeFaultErrorMessage"
name="ExecutionModeFault"></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="infod:CreateEntityAuthorizationFailureErrorMessage"
name="CreateEntityAuthorizationFailure"></wsdl:fault>

```



```

<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></wsdl:fault>
<wsdl:fault message="infod:MissingRequiredParameterFaultErrorMessage"
name="UnsupportedXQueryFault"></wsdl:fault>
<wsdl:fault message="infod:UnsupportedXQueryFaultErrorMessage"
name="UnsupportedXQueryFault"></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:ReplaceEntityAuthorizationFaultErrorMessage"
name="ReplaceEntityAuthorizationFault"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></wsdl:fault>
<wsdl:fault message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingrequiredParameterFault"></wsdl:fault>
<wsdl:fault message="infod:UnsupportedXQueryFaultErrorMessage"
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:DropEntityAuthorizationFaultErrorMessage"
name="DropEntityAuthorizationFailure"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></wsdl:fault>
<wsdl:fault message="infod:MissingRequiredParameterFaultErrorMessage"
name="UnsupportedXQueryFault"></wsdl:fault>
<wsdl:fault message="infod:ExecutionModeFaultErrorMessage"
name="ExecutionModeFault"></wsdl:fault>
</wsdl:operation>

</wsdl:portType>
<wsdl:portType name="infodVocabulary">
<wsdl:operation name="RegisterPropertyVocabulary">
<wsdl:input message="infod:RegisterPropertyVocabularyRequest"></wsdl:input>
<wsdl:output
message="infod:RegisterPropertyVocabularyResponse"></wsdl:output>
<wsdl:fault
message="infod:RegisterVocabularyAuthorizationFailureErrorMessage"
name="RegisterVocabularyAuthorizationFailure"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></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="infod:CreatePropertyVocabularyInstanceAuthorizationFailureErrorMes
sage"
name="CreatePropertyVocabularyInstanceAuthorizationFailure"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></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:operation>
<wsdl:operation name="RegisterDataVocabulary">
<wsdl:input message="infod:RegisterDataVocabularyRequest"></wsdl:input>
<wsdl:output message="infod:RegisterDataVocabularyResponse"></wsdl:output>
<wsdl:fault
message="infod:RegisterVocabularyAuthorizationFailureErrorMessage"
name="RegisterVocabularyAuthorizationFailure"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></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="UnregisterVocabulary">
<wsdl:input message="infod:UnregisterVocabularyRequest"></wsdl:input>
<wsdl:output message="infod:UnregisterVocabularyResponse"></wsdl:output>
<wsdl:fault
message="infod:RegisterVocabularyAuthorizationFailureErrorMessage"
name="RegisterVocabularyAuthorizationFailure"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></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="AssociateVocabulary">
<wsdl:input message="infod:AssociateVocabularyRequest"></wsdl:input>
<wsdl:output message="infod:AssociateVocabularyResponse"></wsdl:output>
<wsdl:fault
message="infod:CreateAssociationAuthorizationFailureErrorMessage"
name="CreateAssociationAuthorizationFailure"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></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="DisassociateVocabulary">
<wsdl:input message="infod:DisassociateVocabularyRequest"></wsdl:input>
<wsdl:output message="infod:DisassociateVocabularyResponse"></wsdl:output>
<wsdl:fault message="infod:DisAssociationAuthorizationFailureErrorMessage"
name="DisAssociationAuthorizationFailure"></wsdl:fault>
<wsdl:fault message="infod:UnknownElementReferenceFaultErrorMessage"
name="UnknownElementReferenceFault"></wsdl:fault>
<wsdl:fault message="infod:MissingRequiredParameterFaultErrorMessage"
name="MissingRequiredParameterFault"></wsdl:fault>
<wsdl:fault message="infod:UnsupportedVocabularyFaultErrorMessage"
name="UnsupportedVocabularyFault"></wsdl:fault>

```

```
</wsdl:operation>
</wsdl:portType>
<wsdl:portType name="infodMetaData">
<wsdl:operation name="GetMetaData">
<wsdl:input message="infod:GetMetaDataRequest"></wsdl:input>
<wsdl:output message="infod:GetMetaDataResponse"></wsdl:output>
</wsdl:operation>
</wsdl:portType>
<wsdl:portType name="infodNotify">
<wsdl:operation name="Notify">
<wsdl:input message="infod:NotifyRequest"></wsdl:input>
</wsdl:operation>
</wsdl:portType>
</wsdl:definitions>
```

7 Author Information

Stephen Davey
National e-Science Centre
15 South College Street
Edinburgh
EH8 9AA
U.K.

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

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

Steve Fisher
Rutherford Appleton Laboratory
Chilton
Didcot
Oxon
OX11 0QX
U.K.

Dieter Gawlick
Oracle Corporation
500 Oracle Parkway
Redwood Shores
CA 94065

Christopher Kantarjiev
Oracle Corporation
500 Oracle Parkway
Redwood Shores
CA 94065

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

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

Shailendra Mishra
Oracle Corporation
500 Oracle Parkway
Redwood Shores
CA 94065

Mallikarjun Shankar
Oak Ridge National Laboratory
Oak Ridge
TN 37831

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

Copyright (C) Global Grid Forum (2004-2006). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works.

However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the GGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the

GGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the GGF or its successors or assignees."