

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 © Open Grid Forum (2004-2007). All Rights Reserved

Abstract

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

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

Table of Contents

| | |
|---|-----------|
| TABLE OF CONTENTS | 2 |
| 1 INTRODUCTION | 4 |
| 1.1 The Registry | 8 |
| 1.1.1 Resources | 8 |
| 1.1.1.1 Publisher Entry, Consumer Entry and Subscriber Entry | 8 |
| 1.1.1.2 Data Vocabularies and Data Source Entries | 8 |
| 1.1.1.3 Property Vocabularies and Property Vocabulary Instances | 9 |
| 1.1.1.4 Subscriptions and Constraints | 9 |
| 1.1.2 Dependencies | 9 |
| 1.1.3 Matching Publishers to Subscriptions | 10 |
| 1.2 Security | 10 |
| 1.3 Lifetime Management | 11 |
| 1.4 Summary of key aspects of INFOD | 11 |
| 1.5 Glossary | 11 |
| 1.6 Terminology | 13 |
| 1.7 Namespaces | 14 |
| 1.8 Fault Definitions | 15 |
| 2 THE BASE INFOD REGISTRY INTERFACE | 16 |
| 2.1 Managing Publisher Entries | 17 |
| 2.1.1 <i>CreatePublisherEntry</i> | 17 |
| 2.1.2 <i>ReplacePublisherEntry</i> | 19 |
| 2.1.3 <i>DropPublisherEntry</i> | 21 |
| 2.2 Managing Subscriber Entries | 22 |
| 2.2.1 <i>CreateSubscriberEntry</i> | 22 |
| 2.2.2 <i>ReplaceSubscriberEntry</i> | 24 |
| 2.2.3 <i>DropSubscriberEntry</i> | 26 |
| 2.3 Managing Consumer Entries | 27 |
| 2.3.1 <i>CreateConsumerEntry</i> | 27 |
| 2.3.2 <i>ReplaceConsumerEntry</i> | 29 |
| 2.3.3 <i>DropConsumerEntry</i> | 30 |
| 2.4 Managing Subscriptions | 32 |
| 2.4.1 <i>CreateSubscription</i> | 32 |
| 2.4.2 <i>ReplaceSubscription</i> | 34 |
| 2.4.3 <i>DropSubscription</i> | 36 |
| 2.5 Managing Vocabularies | 37 |
| 2.5.1 <i>CreatePropertyVocabulary</i> | 38 |
| 2.5.2 <i>DropPropertyVocabulary</i> | 39 |
| 2.5.3 <i>CreatePropertyVocabularyInstance</i> | 40 |
| 2.5.4 <i>DropPropertyVocabularyInstance</i> | 41 |
| 2.5.5 <i>CreateDataVocabulary</i> | 42 |

| | | |
|----------|--|-----------|
| 2.5.6 | <i>DropDataVocabulary</i> | 44 |
| 2.6 | Data Source Entries | 45 |
| 2.6.1 | <i>CreateDataSourceEntry</i> | 45 |
| 2.6.2 | <i>DropDataSourceEntry</i> | 47 |
| 2.7 | The GetMetaData Operation | 48 |
| 3 | BASE INFOD NOTIFICATION INTERFACES | 50 |
| 3.1 | Notifications from Publishers to Consumers | 50 |
| 3.2 | Notification from the Registry | 52 |
| 3.2.1 | <i>Notification of Publishers</i> | 52 |
| 3.2.2 | <i>Notification of Subscribers</i> | 53 |
| 3.2.3 | <i>Notification of Consumers</i> | 54 |
| 4 | SECURITY CONSIDERATIONS | 55 |
| 4.1 | Message Encryption and Data Privacy Requirements | 55 |
| 4.2 | Integration with Authorization Model..... | 56 |
| 5 | APPENDIX I – XML SCHEMA | 57 |
| 5.1 | Publisher Entry | 58 |
| 5.2 | Subscriber Entry | 59 |
| 5.3 | Consumer Entry | 59 |
| 5.4 | Subscription | 60 |
| 5.5 | Property Vocabulary..... | 60 |
| 5.6 | Property Vocabulary Instance | 61 |
| 5.7 | Data Vocabulary | 61 |
| 5.8 | Data Source Entry | 62 |
| 5.9 | Error Messages | 62 |
| 5.10 | INFOD Notification | 64 |
| 5.11 | INFOD Publisher Notification | 65 |
| 5.12 | INFOD Subscriber Notification | 65 |
| 5.13 | INFOD Consumer Notification..... | 66 |
| 6 | APPENDIX II – WSDL 1.1 | 67 |
| 7 | AUTHOR INFORMATION | 84 |
| 8 | INTELLECTUAL PROPERTY STATEMENT | 86 |
| 9 | FULL COPYRIGHT NOTICE | 87 |

1 Introduction

Having the most up-to-date information available is becoming increasingly important. Rick Hayes-Roth uses the term VIRT (Valuable Information at the Right Time) to capture this requirement^{1,2}. The core idea of VIRT is that consumers of information should receive the information that is relevant to them as soon as it is available or whenever it is needed. COI (Conditions Of Interest) determine which information is needed when and by whom. Information Dissemination³ (INFOD) provides core technology to support the VIRT objectives for a wide range of applications.

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

This basic model has been extended by INFOD with:

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

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

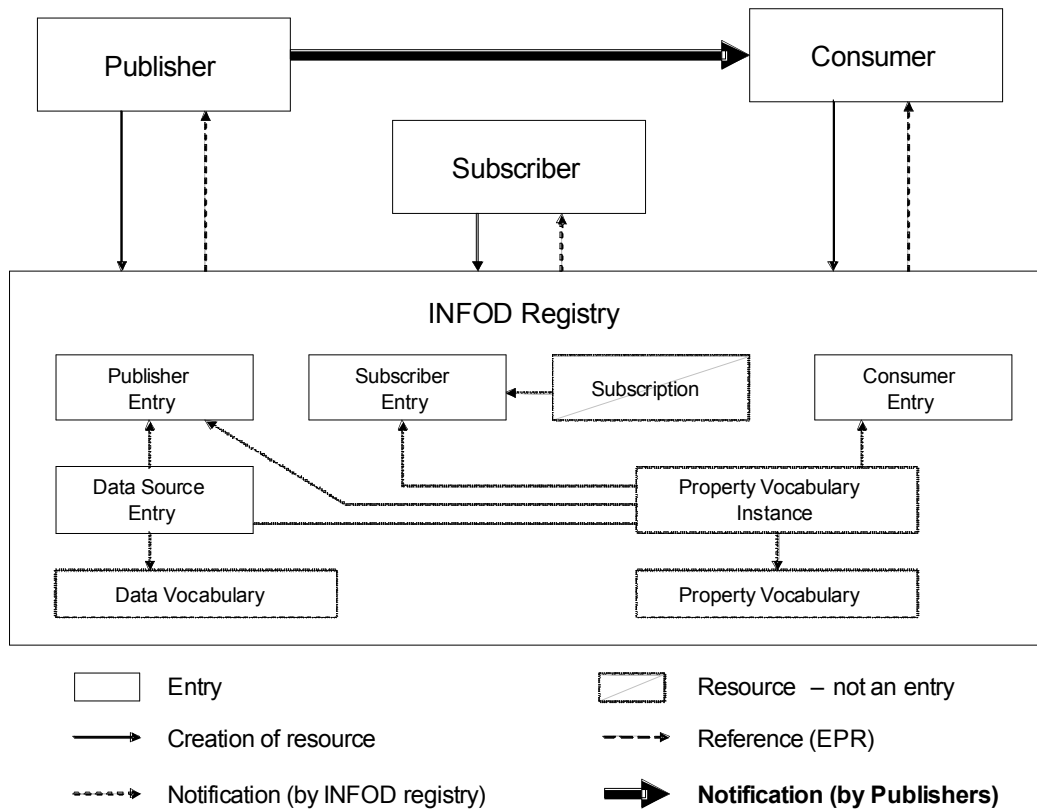
² Event Processing in the Global Information Grid (GIG) (http://complexevents.com/wp-content/uploads/2007/01/Event_Processing_GIG_RHR.ppt)

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

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

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

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



53

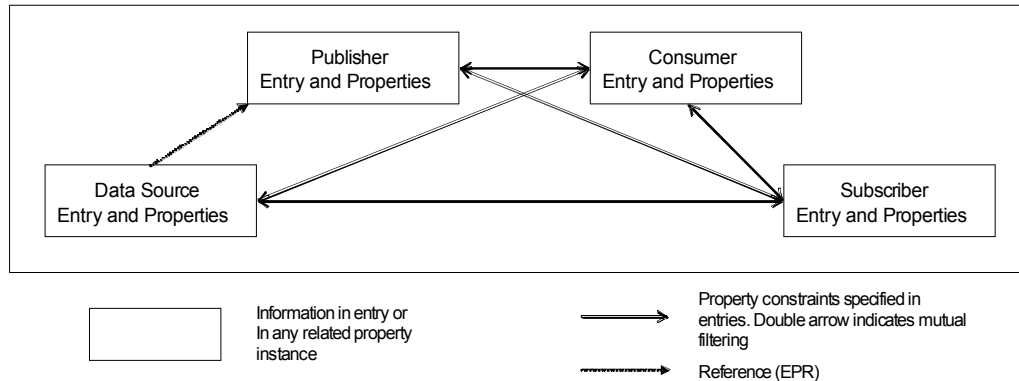
54

Figure 1: INFOD Resources

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

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

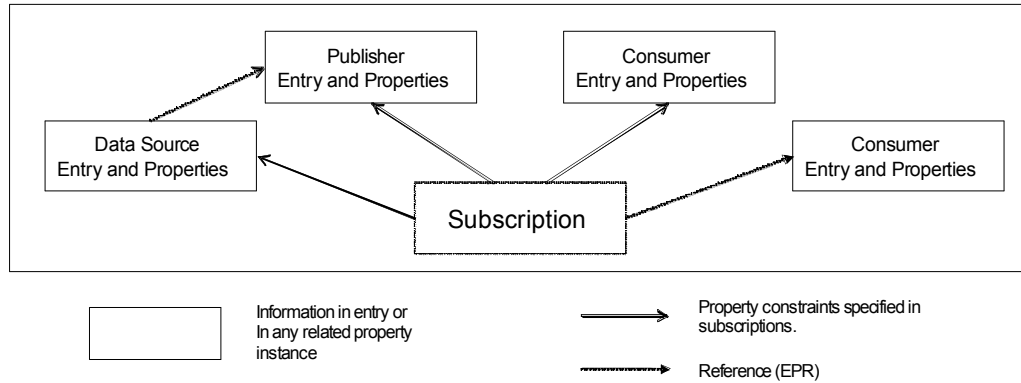
- 60 • **Property Constraints and Mutual Filtering:** Each entry can specify a set of property
 61 constraints referencing information related to other entries; a property constraint is an
 62 XQuery or XPath expression referencing entries and property vocabulary instances. For
 63 example, a car buyer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



70 **Figure 2: Property Constraints**

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

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



84

85

Figure 3: Property Constraints in Subscriptions

86

87

88

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

89

90

91

92

93

94

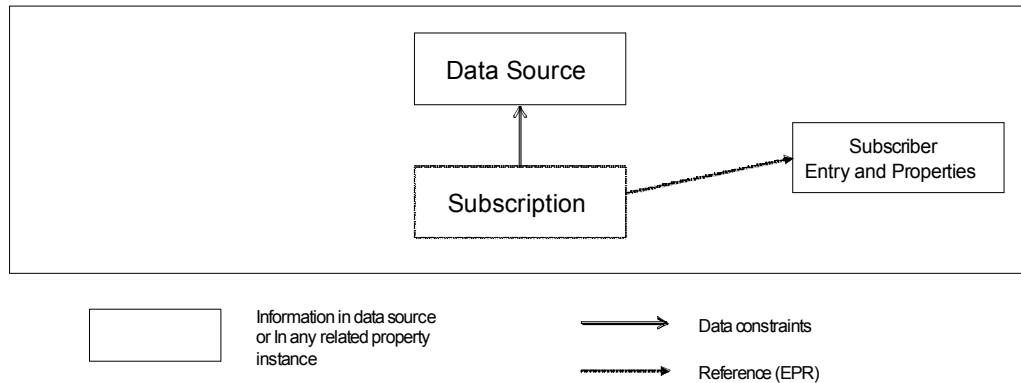
- **Data Constraints (in Subscription):** A data constraint is a query supported by a data vocabulary. Data constraints are used to specify which information is of interest. To make the information *as valuable as possible* subscribers can specify what an event is by defining conditions or patterns on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

95

96

97

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



98

99

Figure 4: Data Constraints in Subscriptions

100

101

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

102

103

104

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

105 **1.1 The Registry**

106 **1.1.1 Resources**

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

110 **1.1.1.1 Publisher Entry, Consumer Entry and Subscriber Entry**

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

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

121 The replace operation (for example ReplacePublisher in Section 2.1.2) takes the EPR that was
122 returned by the create operation as an additional parameter and keeps only the identity of the
123 entry: all the data associated with it by the create operation is replaced by new data however all
124 relations established after the original entry was created are preserved as the identity of the entry
125 remains unchanged. The drop operation (for example DropPublisher in Section 2.1.3) takes the
126 EPR of the entry and makes the stored entry unavailable and so makes subsequent use of the
127 EPR invalid. The drop operation is not allowed to make the system inconsistent (see Section
128 1.1.2) so, by default, an error will be reported if an attempt is made to drop an entry which is still
129 referenced. There is an optional flag which can be set to "DISABLE NEW REFERENCES" which
130 results in the entry being dropped when the last reference to the entry has been removed and
131 "CASCADE", which also drops (recursively) all entries referencing that entry.

132 **1.1.1.2 Data Vocabularies and Data Source Entries**

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

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

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

147 **1.1.1.3 Property Vocabularies and Property Vocabulary Instances**

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

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

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

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

161 **1.1.1.4 Subscriptions and Constraints**

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

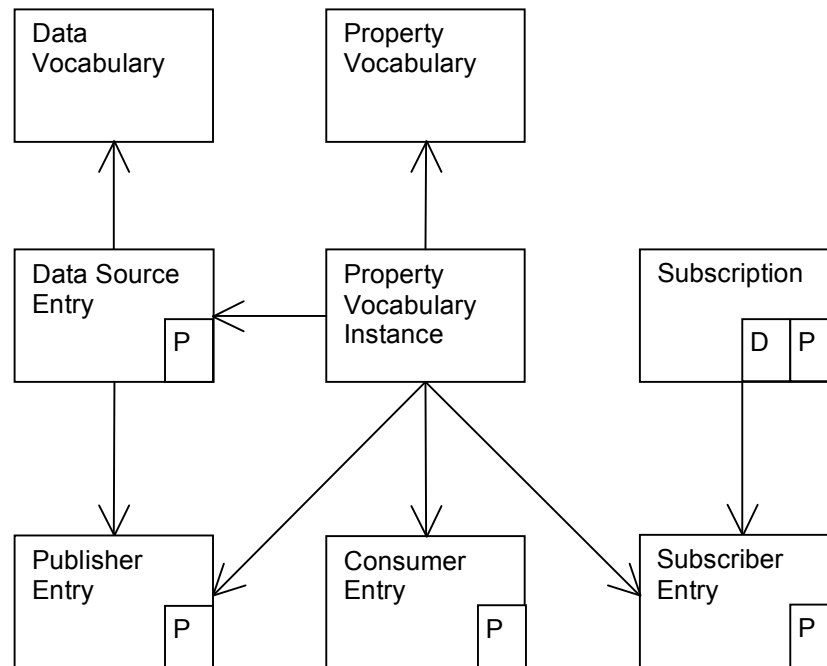
171 In addition a subscription may include dynamic consumer constraints. These are constraints
172 which are evaluated by the consumer rather than the registry by looking at the contents of a
173 potential message.

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

176 **1.1.2 Dependencies**

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

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



182 **Figure 5: Relation between INFOD resources**

183 1.1.3 Matching Publishers to Subscriptions

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

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

190 1.2 Security

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

194 1.3 Lifetime Management

195 The INFOD specification does not contain any specific resource lifetime management other than
 196 the facilities to remove INFOD resources, for example *DropSubscription* etc. However, to ensure
 197 that in cases where a client becomes disconnected from the INFOD Registry and is unable or
 198 unwilling to destroy obsolete INFOD resources, some form of lifetime management should be
 199 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
 200 destroyed after a period of time unless the scheduled termination time is extended.
 201

202 1.4 Summary of key aspects of INFOD

203 The INFOD base specification may be summarized:

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

215 1.5 Glossary

| | | |
|---|----------------------------|---|
| 216 217 218 219 220 221 222 | <u>Constraint</u> | Constraints are used to specify which conditions must be satisfied to be eligible for an interaction. Constraints must be formulated in the constraint language(s) that are associated to the vocabularies, which are used to structure the referenced data. Most constraints are evaluated by the registry but dynamic consumer constraints are dealt with by the consumer. The absence of constraints shows that the interaction is unrestricted. |
| 223 224 | <u>Constraint Language</u> | The grammar of the constraints specification associated to a type system. |
| 225 226 227 | <u>Entry</u> | An entry is the information about an external object that is stored in the registry. There of four types of entry: publisher, consumer, subscriber and data source. |
| 228 229 230 | <u>Data Vocabulary</u> | A data vocabulary defines the structure of the data associated to a data source. Data vocabularies can be specified using any type system. |
| 231 232 233 234 | <u>Consumer</u> | A consumer is able to receive messages delivered by publishers. Property vocabularies can be used to extend the description of consumers; consumers can limit the flow of messages by defining constraints. |
| 235 | <u>Consumer Entry</u> | Information about a consumer stored in a registry |

| | | |
|-----|-------------------------------------|---|
| 236 | <u>EPR</u> | An EPR (Endpoint Reference) is an XML structure encapsulating information useful for addressing a message to a Web service. |
| 237 | | |
| 238 | | This includes the destination address of the message, any |
| 239 | | additional parameters (called reference parameters) necessary to |
| 240 | | route the message to the destination, and optional metadata about |
| 241 | | the service. |
| 242 | <u>Event</u> | An event is a view at a state transition specified by a publisher or a |
| 243 | | subscriber. Publishers may allow subscribers to reference events |
| 244 | | (those defined by publishers) to create messages or to define |
| 245 | | events by referencing state transitions. |
| 246 | | In many cases, publishers do not provide access to events but |
| 247 | | allow only access to (and selection of) messages. In this case the |
| 248 | | state and event definitions are hidden to subscribers. |
| 249 | <u>Message</u> | A message is used to deliver data from publishers to consumers. A |
| 250 | | message normally contains information about an event that is |
| 251 | | observed by a publisher. |
| 252 | <u>Property Vocabulary</u> | A property vocabulary specifies the structure of properties |
| 253 | | associated to entries. |
| 254 | <u>Property Vocabulary Instance</u> | A property vocabulary instance represents the (values of) |
| 255 | | properties that are associated to specific entries. A property |
| 256 | | vocabulary instance has to be structured according to a property |
| 257 | | vocabulary. |
| 258 | <u>Publisher</u> | A publisher is able to create and deliver data in the form of |
| 259 | | messages to consumers. Property vocabularies can be used to |
| 260 | | extend the description of publishers; publishers can limit |
| 261 | | subscriptions requests by defining constraints. |
| 262 | | Publishers may create and deliver messages unconditionally or |
| 263 | | make the delivery of messages dependent on subscriptions. |
| 264 | | Publisher may allow subscribers (using subscriptions) to specify |
| 265 | | which messages should be created in response to which events; |
| 266 | | events maybe pre-defined or based on (subscriptions) |
| 267 | | specifications referencing state changes. |
| 268 | <u>Publisher Entry</u> | Information about a publisher that is stored in a registry as a |
| 269 | | resource |
| 270 | <u>Registry</u> | A repository of INFOD resources able to deliver notifications |
| 271 | <u>Resource</u> | A resource is an elementary object in the registry that may be |
| 272 | | created, replaced or dropped. |
| 273 | <u>Subscriber</u> | A subscriber specifies subscriptions. Subscriptions are the primary |
| 274 | | means of specifying the message flow from publishers to |
| 275 | | consumers. |
| 276 | <u>Subscriber Entry</u> | Information about a subscriber that is stored in a registry as a |
| 277 | | resource. |
| 278 | <u>Subscription</u> | A subscription defines which information has to be delivered by |
| 279 | | which publishers to which consumers. The information is selected |
| 280 | | by constraint specifications; publishers and consumers are |
| 281 | | identified through explicit references (EPR's) or constraints on |
| 282 | | property vocabularies. |

| | | |
|-----|--------------------------|---|
| 283 | <u>Type System</u> | A type system is an enumeration that defines the list of acceptable value domains, their value ranges and binary representation in a digital system. |
| 284 | | |
| 285 | | |
| 286 | <u>Vocabulary</u> | 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. |
| 287 | | |
| 288 | | |
| 289 | | |
| 290 | <u>WSN</u> | Web Service Notification is a pattern-based approach to allow Web services to disseminate information to one another |
| 291 | | |
| 292 | <u>Data Source entry</u> | A data source entry specifies that data structured with the referenced vocabulary (and constraint language) is offered by the referenced publisher. |
| 293 | | |
| 294 | | |
| 295 | | |

296 **1.6 Terminology**

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

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

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

- 305 • '?' denotes optionality (i.e. zero or one occurrences),
- 306 • '*' denotes zero or more occurrences,
- 307 • '+' one or more occurrences,
- 308 • '[' and ']' are used to form groups,
- 309 • '|' represents choice.

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

318

```

319 <!--sample pseudo-schema -->
320 <element
321     required_attribute_of_type_QName="xs:QName"
322     optional_attribute_of_type_string="xs:string"? >
323     <required_element />
324     <optional_element /> ?
325     <one_or_more_of_these_elements /> +
326     [ <choice_1 /> | <choice_2 /> ] *
327 </element>

```

328

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

334 1.7 Namespaces

335 The following namespaces are used in this document:

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

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

337 **1.8 Fault Definitions**

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

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

342 <http://www.ogf.org/infod/fault>.

343

2 The Base INFOD Registry Interface

344

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

345

346

The Base INFOD Registry Interface:

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

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

347 2.1 Managing Publisher Entries

348 These operations are used to manage publishers

- 349 • CreatePublisherEntry (section 2.1.1)
- 350 • ReplacePublisherEntry (section 2.1.2)
- 351 • DropPublisherEntry (section 2.1.3)

352 2.1.1 CreatePublisherEntry

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

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

```

357 <infod:CreatePublisherEntry>
358   <infod:WSReference>
359     wsa:EndPointReferenceType
360   </infod:WSReference> ?
361   <infod:PublisherName> xsd:string </infod:PublisherName> ?
362   <infod:PublisherDescription>
363     xsd:string
364   </infod:PublisherDescription> ?
365   <infod:PropertyConstraint>
366     xsd:any
367   </infod:PropertyConstraint> *
368   <infod:Notification>

```

```

369     xsd:boolean default "FALSE"
370   </infod:Notification> ?
371 </infod:CreatePublisherEntry>

```

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

373 /infod:WSReference

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

377 /infod:PublisherName

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

379 /infod:PublisherDescription

380 A string representing a description of the publisher.

381 /infod:PropertyConstraint

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

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

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

391 /infod:Notification

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

394 For further details see section 3.2.1

395 A WS-Addressing Action header with the value

396 <http://www.ogf.org/infod/INFODRegistry/CreatePublisherEntry> MUST accompany the message.

397 **INFOD Registry Response**

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

```

401 <infod:CreatePublisherEntryResponse>
402   <infod:PublisherEntryReference>
403     wsa:EndPointReferenceType
404   </infod:PublisherEntryReference>
405 </infod:CreatePublisherEntryResponse>

```

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

407 /infod:PublisherEntryReference

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

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

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

415 The message MUST be structured according to the WS-Base Faults specification. For examples using
416 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).
417

418 Example SOAP Encoding of the Create Publisher Message

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

```

420 <s:Envelope ... >
421   <s:Header>
422     <wsa:Action>
423       http://www.ogf.org/infod/INFODRegistry/CreatePublisherEntry
424     </wsa:Action>
425     ...
426   </s:Header>
427   <s:Body>
428     <infod:CreatePublisherEntry>
429       <infod:WSReference>
430         <wsa:Address>
431           http://www.example.org/SomePublisher
432         </wsa:Address>
433       </infod:WSReference>
434       <infod:PublisherName>
435         SomePublisher
436       </infod:PublisherName>
437       <infod:PublisherDescription>
438         This publisher can publish some information
439       </infod:PublisherDescription>
440       <infod:PropertyConstraints>
441         fn:doc("INFODRegistry.xml")/Consumers/infodConsumer
442           [fn:contains(ConsumerName,"Ronny")]
443       </infod:PropertyConstraints>
444       <infod:Notification>
445         TRUE
446       </infod:Notification>
447     </infod:CreatePublisherEntry>
448   </s:Body>
449 </s:Envelope>

```

450 2.1.2 ReplacePublisherEntry

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

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

```

459 <infod:ReplacePublisherEntry>

```

```

460 <infod:WSReference>
461   wsa:EndPointReferenceType
462 </infod:WSReference> ?
463 <infod:PublisherEntryReference>
464   wsa:EndPointReferenceType
465 </infod:PublisherEntryReference>
466 <infod:PublisherName> xsd:string </infod:PublisherName> ?
467 <infod:PublisherDescription>
468   xsd:string
469 </infod:PublisherDescription> ?
470 <infod:PropertyConstraint>
471   xsd:any
472 </infod:PropertyConstraint> *
473 <infod:Notification>
474   xsd:boolean "FALSE"
475 </infod:Notification> ?
476 </infod:ReplacePublisherEntry>

```

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

478 /infod:WSReference

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

482 /infod:PublisherEntryReference

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

485 /infod:PublisherName

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

487 /infod:PublisherDescription

488 A string representing a description of the publisher.

489 /infod:PropertyConstraint

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

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

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

499 /infod:Notification

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

502 For further details see section 3.2.1

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

505 **INFOD Registry Response**

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

```
509 <infod:ReplacePublisherEntryResponse>
510 <infod:Status>
511   xsd:string default "COMPLETED"
512 </infod:Status>
513 </infod:ReplacePublisherEntryResponse>
```

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

515 /infod:Status

516 An indication that the request has been successfully executed.

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

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

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

526 **2.1.3 DropPublisherEntry**

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

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

```
529 <infod:DropPublisherEntry>
530 <infod:PublisherEntryReference>
531   wsa:EndPointReferenceType
532 </infod:PublisherEntryReference>
533 <infod:ExecutionMode> xsd:string </infod:ExecutionMode> ?
534 </infod:DropPublisherEntry>
```

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

536 /infod:PublisherEntryReference

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

539 /infod:ExecutionMode

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

541 "IF UNUSED" The drop request will execute only if the resource is unreferenced

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

544 “CASCADE” The drop request will execute immediately and all references to the
 545 resource will be removed recursively

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

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

549 INFOD Registry Response

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

```
553 <infod:DropPublisherEntryResponse>
554   <infod:Status>
555     xsd:string default "COMPLETED"
556   </infod:Status>
557 </infod:DropPublisherEntryResponse>
```

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

559 /infod:Status

560 An indication that the request has been successfully executed.

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

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

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

570 2.2 Managing Subscriber Entries

571 The following operations are used to manage subscribers:

- 572 • CreateSubscriberEntry (section 2.2.1)
- 573 • ReplaceSubscriberEntry (section 2.2.2)
- 574 • DropSubscriberEntry (section 2.2.3)

575 2.2.1 CreateSubscriberEntry

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

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

```

580 <infod:CreateSubscriberEntry>
581   <infod:WSReference>
582     wsa:EndPointReferenceType
583   </infod:WSReference> ?
584   <infod:SubscriberName> xsd:string </infod:SubscriberName> ?
585   <infod:SubscriberDescription>
586     xsd:string
587   </infod:SubscriberDescription> ?
588   <infod:PropertyConstraint>
589     xsd:any
590   </infod:PropertyConstraint> *
591   <infod:Notification>
592     xsd:Boolean default "FALSE"
593   </infod:Notification> ?
594 </infod:CreateSubscriber>

```

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

596 /infod:WSReference

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

600 /infod:SubscriberName

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

602 /infod:SubscriberDescription

603 A string representing a description of the subscriber.

604 /infod:PropertyConstraint

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

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

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

614 infod:Notification

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

617 For further details see section 3.2.2.

618 A WS-Addressing Action header with the value

619 <http://www.ogf.org/infod/INFODRegistry/CreateSubscriberEntry> MUST accompany the message

620 **INFOD Registry Response**

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

```

624 <infod:CreateSubscriberEntryResponse>
625   <infod:SubscriberEntityReference>
626     wsa:EndPointReferenceType
627   </infod:SubscriberEntityReference>
628 </infod:CreateSubscriberEntryResponse>

```

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

630 /infod:SubscriberEntityReference

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

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

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

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

640 2.2.2 ReplaceSubscriberEntry

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

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

```

649 <infod:ReplaceSubscriberEntryEntry>
650   <infod:WSReference>
651     wsa:EndPointReferenceType
652   </infod:WSReference> ?
653   <infod:SubscriberEntryReference>
654     wsa:EndPointReferenceType
655   </infod:SubscriberEntryReference>
656   <infod:SubscriberName> xsd:string </infod:SubscriberName> ?
657   <infod:SubscriberDescription>
658     xsd:string
659   </infod:SubscriberDescription> ?
660   <infod:PropertyConstraint>
661     xsd:any
662   </infod:PropertyConstraint> *
663   <infod:Notification>
664     xsd:Boolean default "FALSE"
665   </infod:Notification> ?
666 </infod:ReplaceSubscriberEntry>

```

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

668 /infod:WSReference

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

672 /infod:SubscriberEntryReference

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

675 /infod:SubscriberName

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

677 /infod:SubscriberDescription

678 A string representing a description of the subscriber.

679 /infod:PropertyConstraint

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

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

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

689 infod:Notification

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

692 For further details see section 3.2.2.

693 A WS-Addressing Action header with the value

694 <http://www.ogf.org/infod/INFODRegistry/ReplaceSubscriberEntry> MUST accompany the message

695 **INFOD Registry Response**

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

```
699 <infod:ReplaceSubscriberEntryResponse>
700   <infod:Status>
701     xsd:string default "COMPLETED"
702   </infod:Status>
703 </infod:ReplaceSubscriberEntryResponse>
```

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

705 /infod:Status

706 An indication that the request has been successfully executed. .

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

- 708 • ReplaceResourceAuthorizationFault: User not authorized to replace the INFOD resource
- 709 at this INFOD registry

- 710 • UnknownResourceReferenceFault: An resource has been referenced that is unknown to
- 711 the INFOD registry

- 712 • MissingRequiredParameterFault: A required parameter was not specified

- 713 • UnsupportedXQueryFault: The XQuery specified could not be parsed correctly

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

716 **2.2.3 DropSubscriberEntry**

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

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

```
719 <infod:DropSubscriberEntry>  
720 <infod:SubscriberEntryReference>  
721 wsa:EndPointReferenceType  
722 </infod:SubscriberEntryReference>  
723 <infod:ExecutionMode> xsd:string </infod:ExecutionMode>  
724 </infod:DropSubscriberEntry>
```

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

726 /infod:ResourceReference

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

729 /infod:ExecutionMode

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

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

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

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

739 **INFOD Registry Response**

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

```
743 <infod:DropSubscriberEntryResponse>  
744 <infod:Status>  
745 xsd:string default "COMPLETED"  
746 </infod:Status>  
747 </infod:DropSubscriberEntryResponse>
```

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

749 /infod:Status

750 An indication that the request has been successfully executed.

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

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

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

760 2.3 Managing Consumer Entries

761 The following operations are used to manage consumers:

- 762 • CreateConsumerEntry (section 2.3.1)
- 763 • ReplaceConsumerEntry (section 2.3.2)
- 764 • DropConsumerEntry (section 2.3.3)

765 2.3.1 CreateConsumerEntry

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

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

```

770 <infod:CreateConsumerEntry>
771   <infod:WSReference>
772     wsa:EndPointReferenceType
773   </infod:WSReference>
774   <infod:ConsumerName> xsd:string </infod:ConsumerName> ?
775   <infod:ConsumerDescription>
776     xsd:string
777   </infod:ConsumerDescription> ?
778   <infod:PropertyConstraint>
779     xsd:any
780   </infod:PropertyConstraint> *
781   <infod:Notification>
782     xsd:Boolean default "FALSE"
783   </infod:Notification> ?
784 </infod:CreateConsumerEntry>

```

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

786 /infod:WSReference

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

790 /infod:ConsumerName

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

792 /infod:ConsumerDescription

793 A string representing a description of the consumer

794 /infod:PropertyConstraint

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

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

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

804 infod:Notification

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

807 For further details see section 3.2.3.

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

810 **INFOD Registry Response**

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

```
814 <infod:CreateConsumerEntryResponse>
815 <infod:ConsumerEntryReference>
816   wsa:EndPointReferenceType
817 </infod:ConsumerEntryReference>
818 </infod:CreateConsumerEntryResponse>
```

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

820 /infod:ConsumerEntryReference

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

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

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

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

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

830 2.3.2 ReplaceConsumerEntry

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

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

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

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

857 /infod:WSReference

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

861 /infod:ConsumerEntryReference

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

864 /infod:ConsumerName

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

866 /infod:ConsumerDescription

867 A string representing a description of the consumer

868 /infod:PropertyConstraint

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

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

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

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

878 infod:Notification

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

881 For further details see section 3.2.3.

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

884 INFOD Registry Response

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

```
888 <infod:ReplaceConsumerEntryResponse>
889   <infod:Status>
890     xsd:string default "COMPLETED"
891   </infod:Status>
892 </infod:ReplaceConsumerEntryResponse>
```

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

894 /infod:Status

895 An indication that the request has been successfully executed.

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

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

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

905 2.3.3 DropConsumerEntry

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

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

```
908 <infod:DropConsumerEntry>
909   <infod:ConsumerEntryReference>
910     wsa:EndPointReferenceType
911   </infod:ConsumerEntryReference>
```

```

912     <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
913 </infod:DropConsumerEntry>

```

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

915 /infod:ConsumerEntryReference

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

918 /infod:ExecutionMode

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

920 "IF UNUSED" The drop request will execute only if the resource is unreferenced

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

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

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

926 A WS-Addressing Action header with the value

927 <http://www.ogf.org/infod/INFODRegistry/DropConsumerEntry> MUST accompany the message

928 INFOD Registry Response

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

```

932 <infod:DropConsumerEntryResponse>
933 <infod:Status>
934   xsd:string default "COMPLETED"
935 </infod:Status>
936 </infod:DropConsumerEntryResponse>

```

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

938 /infod:Status

939 An indication that the request has been successfully executed.

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

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

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

949 2.4 Managing Subscriptions

950 The following operations are used to manage subscriptions:

- 951 • CreateSubscription (section 2.4.1)
- 952 • ReplaceSubscription (section 2.4.2)
- 953 • DropSubscription (section 2.4.3)

954 2.4.1 CreateSubscription

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

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

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

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

```

963 <infod:CreateSubscription>
964 <infod:SubscriptionName> xsd:string </infod:SubscriptionName> ?
965 <infod:SubscriptionDescription>
966   xsd:string
967 </infod:SubscriptionDescription> ?
968 <infod:SubscriberEntryReference>
969   wsa:EndPointReferenceType
970 </infod:SubscriberEntryReference>
971 <infod:DataConstraint >
972   xsd:anyType
973 </infod:DataConstraint> *
974 <infod:PropertyConstraint>
975   xsd:any
976 </infod:PropertyConstraint> *
977 <infod:DynamicConsumerConstraint>
978   xsd:anyType
979 </infod:DynamicConsumerConstraint> *
980 </infod:CreateSubscription>

```

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

982 /infod:SubscriptionName

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

984 /infod:SubscriptionDescription

985 A string representing a description of the subscription.

986 /infod:SubscriberEntryReference

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

989 /infod:DataConstraint

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

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

994 /infod:PropertyConstraint

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

999 For example, a subscription identifies the set of publishers that are eligible to react to this
 1000 subscription.

1001 Note that the XQuery statement MUST be encoded correctly, i.e. characters such as ">"
 1002 would be represented as ">,"

1003 /infod:DynamicConsumerConstraint

1004 An element specifying which consumers receive a specific message. The constraint(s)
 1005 language(s) is/are implicitly defined through the reference of the vocabulary EPR.

1006 These Constraints are designed to determine the consumers of each message based on its
 1007 content; i.e., a Dynamic Consumer Constraint cannot be applied by the INFOD registry and is
 1008 processed by the publishers.

1009 infod:PropertyConstraint should be used to specify consumer constraints if all messages
 1010 created in response to the subscription are published to the same set of consumers.

1011 For example, a message representing a bill should be *published* to the payee.

1012 A WS-Addressing Action header with the value

1013 <http://www.ogf.org/infod/INFODRegistry/CreateSubscription> MUST accompany the message

1014 **INFOD Registry Response**

1015 If the INFOD registry accepts the CreateSubscription message, it MUST respond to the WS endpoint
 1016 specified in the request message with a CreateSubscriptionResponse message. The
 1017 CreateSubscription response message is a message of the following form:

```
1018 <infod:CreateSubscriptionResponse>
1019   <infod:SubscriptionReference>
1020     wsa:EndPointReferenceType
1021   </infod:SubscriptionReference>
1022 </infod:CreateSubscriptionResponse>
```

1023 The elements of the CreateSubscriptionResponse message are further described as follows:

1024 /infod:SubscriptionReference

1025 An endpoint reference element, as defined by WS-Addressing, used to identify the newly
 1026 created subscription in the INFOD registry.

1027 One of the following faults MUST be sent if the operation fails:

- 1028 • CreateResourceAuthorizationFault: User not authorized to create the INFOD resource at this
 1029 INFOD registry

- 1030 • UnknownResourceReferenceFault: An resource has been referenced that is unknown to the
1031 INFOD registry
- 1032 • MissingRequiredParameterFault: A required parameter was not specified
- 1033 • UnsupportedXQueryFault: The XQuery specified could not be parsed correctly

1034 The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP
1035 v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

1036 2.4.2 ReplaceSubscription

1037 As part of the processing of a ReplaceSubscription request message, the INFOD registry MUST
1038 replace the entire INFOD metadata for the resource representing the subscription. All previously
1039 defined values MUST be deleted. The ReplaceSubscription differs from the CreateSubscription
1040 interface in that it replaces an existing subscription resource and assigns the original EPR to the
1041 replaced subscription.

1042 The format of the request message for a ReplaceSubscription operation is also based on the schema
1043 definition provided in Appendix I – XML Schema for an INFOD resource. Details are as follows:

```

1044 <infod:ReplaceSubscription>
1045   <infod:SubscriptionReference>
1046     wsa:EndPointReferenceType
1047   </infod:SubscriptionReference>
1048   <infod:SubscriptionName> xsd:string </infod:SubscriptionName> ?
1049   <infod:SubscriptionDescription>
1050     xsd:string
1051   </infod:SubscriptionDescription> ?
1052   <infod:SubscriberReference>
1053     wsa:EndPointReferenceType
1054   </infod:SubscriberReference>
1055   <infod:DataConstraint>
1056     xsd:anyType
1057   </infod:DataConstraint> *
1058   <infod:PropertyConstraint>
1059     xsd:any
1060   </infod:PropertyConstraint> *
1061   <infod:DynamicConsumerConstraint>
1062     xsd:anyType
1063   </infod:DynamicConsumerConstraint> *
1064 </infod:ReplaceSubscription>

```

1065 The elements of the ReplaceSubscription message are further described as follows:

1066 /infod:SubscriptionReference

1067 An endpoint reference element, as defined by WS-Addressing, used to identify the
1068 subscription resource in the INFOD registry that will be replaced.

1069 /infod:SubscriptionName

1070 A string representing the name of the subscription. This name MAY NOT be unique.

1071 /infod:SubscriptionDescription

1072 A string representing a description of the subscription.

1073 /infod:SubscriberEntryReference

1074 An endpoint reference element to the INFOD EPR, as defined by WS-Addressing, used to
1075 identify the subscriber entry responsible for the subscription.

1076 /infod:DataConstraint

1077 DataConstraint specifies which information is of interest to consumers. The constraint(s)
1078 language(s) is/are implicitly defined through the reference of the vocabulary EPR. Data
1079 Constraints are not applied by the INFOD registry but by the publishers.

1080 See 2.5 for more details on how to define a vocabulary referenced by such constraints.

1081 Note: If no data constraint is specified all messages published by publishers are of interest.

1082 /infod:PropertyConstraint

1083 Property constraints are used to specify which conditions must be satisfied by entries
1084 (publishers, data sources, and consumers) to be eligible for this subscription. A property
1085 constraint MUST be formulated as an XQuery. The INFOD Base Use Case Scenarios (see
1086 <http://forge.gridforum.org/sf/go/doc13626?nav=1>) provide examples of XQueries.

1087 For example, a subscription identifies the set of publishers that are eligible to react to this
1088 subscription.

1089 Note that the XQuery statement MUST be encoded correctly, i.e. characters such as ">"
1090 would be represented as ">,"

1091 /infod:DynamicConsumerConstraint

1092 An element specifying which consumers receive a specific message. The constraint(s)
1093 language(s) is/are implicitly defined through the reference of the vocabulary EPR.

1094 These Constraints are designed to determine the consumers of each message based on its
1095 content; i.e., a Dynamic Consumer Constraint cannot be applied by the INFOD registry and is
1096 processed by the publishers.

1097 infod:PropertyConstraint should be used to specify consumer constraints if all messages
1098 created in response to the subscription are disseminated to the same set of consumers.

1099 For example, a message representing a bill should be disseminated to the payee.

1100 A WS-Addressing Action header with the value

1101 <http://www.ogf.org/infod/INFODRegistry/ReplaceSubscription> MUST accompany the message

1102 INFOD Registry Response

1103 If the INFOD registry accepts the ReplaceSubscriptionRequest, it MUST respond to the WS endpoint
1104 specified in the request message with a ReplaceSubscription message. The ReplaceSubscription
1105 response message is a message of the following form:

```
1106 <infod:ReplaceSubscriptionResponse>
1107   <infod:Status>
1108     xsd:string default "COMPLETED"
1109   </infod:Status>
1110 </infod:ReplaceSubscriptionResponse>
```

1111 The elements of the ReplaceSubscriptionResponse message are further described as follows:

1112 /infod:SubscriptionReference

1113 An endpoint reference element, as defined by WS-Addressing, used to identify the
1114 subscription resource in the INFOD registry to replace.

1115 One of the following faults MUST be sent if the operation fails:

- 1116 • ReplaceResourceAuthorizationFault: User not authorized to replace the INFOD resource
1117 at this INFOD registry
- 1118 • UnknownResourceReferenceFault: An resource has been referenced that is unknown to
1119 the INFOD registry
- 1120 • MissingRequiredParameterFault: A required parameter was not specified
- 1121 • UnsupportedXQueryFault: The XQuery specified could not be parsed correctly

1122 The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP
1123 v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

1124 2.4.3 DropSubscription

1125 The DropSubscription operation MUST be used to remove an INFOD subscription resource from an
1126 INFOD registry.

1127 The format of the request message for a DropSubscription operation is:

```
1128 <infod:DropSubscription>
1129   <infod:SubscriptionReference>
1130     wsa:EndPointReferenceType
1131   </infod:SubscriptionReference>
1132   <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
1133 </infod:DropSubscription>
```

1134 The elements of the DropSubscription message are further described as follows:

1135 /infod:SubscriptionReference

1136 An endpoint reference element, as defined by WS-Addressing, used to identify the INFOD
1137 subscription resource in the registry to drop.

1138 /infod:ExecutionMode

1139 An optional parameter indicating the mode of execution of the drop request. Possible values
1140 are:

- 1141 "IF UNUSED" The drop request will execute only if the resource is unreferenced
- 1142 "DISABLE NEW" No new references are possible for the resource. The resource will
1143 be dropped when the last reference to this resource is gone
- 1144 "CASCADE" The drop request will execute immediately and all references to the
1145 resource will be removed recursively

1146 If this parameter is not specified, the default value "IF UNUSED" MUST be used.

1147 A WS-Addressing Action header with the value
1148 <http://www.ogf.org/infod/INFODRegistry/DropSubscription> MUST accompany the message

1149 INFOD Registry Response

1150 If the INFOD registry accepts the DropSubscription request, it MUST respond to the WS endpoint
1151 specified in the request message with a DropSubscriptionResponse message. The
1152 DropSubscriptionResponse message is a message of the following form:

```
1153 <infod:DropSubscriptionResponse>
1154   <infod:Status>
1155     xsd:string default "COMPLETED"
1156   </infod:Status>
```

1157 `</infod:DropSubscriptionResponse>`

1158 The elements of the ReplaceSubscriptionResponse message are further described as follows:

1159 `/infod:Status`

1160 An indication that the request has been successfully executed.

1161 One of the following faults MUST be sent if the operation fails:

- 1162 • DropResourceAuthorizationFailure: User not authorized to drop the INFOD resource at this
1163 INFOD registry
- 1164 • UnknownElementReferenceFault: An element has been referenced that is unknown to the
1165 INFOD registry
- 1166 • MissingRequiredParameterFault: A required parameter was not specified
- 1167 • ExecutionModeFault: Cannot use ExecutionMode provided

1168 The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP
1169 v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

1170 2.5 Managing Vocabularies

1171 INFOD has a set of predefined vocabularies. These are REQUIRED vocabularies for the INFOD
1172 registry:

- 1173 • INFOD PublisherEntry Vocabulary
- 1174 • INFOD SubscriberEntry Vocabulary
- 1175 • INFOD ConsumerEntry Vocabulary
- 1176 • INFOD Subscription Vocabulary
- 1177 • INFOD DataSourceEntry Vocabulary

1178 These vocabularies are used by the INFOD registry to match publishers with consumers through
1179 subscriptions and ensure that property constraints and data constraints are validated. All of these
1180 vocabularies are described in xml and detailed in section 5

1181 Users MAY also define two additional types of vocabularies:

1182 **Property Vocabularies:** Entries may specify properties that define their characteristics. They
1183 do that using a property vocabulary that may be queried. If two or more entries share the
1184 same property vocabulary, they can specify constraints on each other. The INFOD registry
1185 MAY manage constraints on these property vocabularies in addition to constraints formulated
1186 in the INFOD vocabularies. Property Vocabularies MUST be defined in xml.

1187 **Data Vocabularies:** In order to tell publishers which messages a subscription is interested in,
1188 they MUST agree on the data vocabulary. The data vocabulary is referenced in the
1189 *DataConstraints* component of a subscription resource, which allows INFOD subscribers to
1190 describe the structure of the published data/data of interest to them.

1191 Data constraints' definitions MUST point to an existing data vocabulary and thus are simply
1192 equivalent to defining operations on top of an existing vocabulary (i.e. selection criteria, etc.
1193 on top of published data). Data Vocabularies are not limited to xml.

1194 This section describes how these two types of vocabulary are created and dropped from an INFOD
 1195 registry. It also includes operations for creating and dropping instances of a registered property
 1196 vocabulary.

1197 **2.5.1 CreatePropertyVocabulary**

1198 The CreatePropertyVocabulary creates a property vocabulary in an INFOD registry. The Property
 1199 Vocabulary is an XML schema. As part of the processing of a CreatePropertyVocabulary request
 1200 message, the INFOD registry MUST create a new resource for that vocabulary.

1201 The format of the request message for CreatePropertyVocabulary operation is as follows:

```
1202 <infod:CreatePropertyVocabulary>
1203   <infod:PropertyVocabularyName>
1204     xsd:string
1205   </infod:PropertyVocabularyName> ?
1206   <infod:PropertyVocabularyDescription>
1207     xsd:string
1208   </infod:PropertyVocabularyDescription> ?
1209   <infod:PropertyVocabularyBody>
1210     xsd:schema
1211   </infod:PropertyVocabularyBody>
1212 </infod:CreatePropertyVocabulary>
```

1213 The elements of the CreatePropertyVocabulary message are further described as follows:

1214 /infod:PropertyVocabularyName

1215 A string representing a name that is local to the INFOD registry where the
 1216 CreatePropertyVocabulary operation takes place. This name MAY NOT be unique.

1217 Names MUST NOT start with \$\$infod.

1218 /infod:PropertyVocabularyDescription

1219 A string representing a description of the vocabulary.

1220 /infod:PropertyVocabularyBody

1221 An element defining an XML Schema. This is an extensibility mechanism to allow XML
 1222 elements to be specified for the defined property vocabulary.

1223 A WS-Addressing Action header with the value

1224 <http://www.ogf.org/infod/INFODRegistry/CreatePropertyVocabulary> MUST accompany the message.

1225 **INFOD Registry Response**

1226 If the INFOD registry accepts the CreatePropertyVocabulary request, it MUST respond to the WS
 1227 endpoint specified in the request message with a CreateVocabularyResponse message.

1228 In case of a successful registration, the CreateVocabularyResponse message is a message of the
 1229 following form:

```
1230 <infod:CreatePropertyVocabularyResponse>
1231   <infod:PropertyVocabularyReference>
1232     wsa:EndPointReferenceType
1233   </infod:PropertyVocabularyReference>
1234 </infod:CreateVocabularyResponse>
```

1235 The elements of the CreateVocabularyResponse message are further described as follows:

1236 /infod:PropertyVocabularyReference

1237 An endpoint reference element, as defined by WS-Addressing, used to identify the newly
1238 created property vocabulary.

1239 One of the following faults MUST be sent if the operation fails::

- 1240 • CreateResourceAuthorizationFault: User not authorized to create a resource at this
1241 INFOD registry
- 1242 • MissingRequiredParameterFault: A required parameter was not specified
- 1243 • UnSupportedVocabularyFault: Vocabulary Language not supported

1244 The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP
1245 v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

1246 2.5.2 DropPropertyVocabulary

1247 The DropPropertyVocabulary operation drops a particular property vocabulary from an INFOD
1248 registry.

1249 The format of the request message for an DropPropertyVocabulary operation is:

```
1250 <infod:DropPropertyVocabulary>
1251   <infod:PropertyVocabularyReference>
1252     wsa:EndPointReferenceType
1253   </infod:PropertyVocabularyReference>
1254   <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
1255 </infod:DropPropertyVocabulary>
```

1256 The elements of the DropPropertyVocabulary message are further described as follows:

1257 /infod:PropertyVocabularyReference

1258 An endpoint reference element, as defined by WS-Addressing, used to identify the vocabulary
1259 to drop from the Registry.

1260 /infod:ExecutionMode

1261 A parameter indicating the mode of execution of the drop request. Possible values are:

- 1262 “IF UNUSED” The drop request will execute only if the resource is unreferenced
- 1263 “DISABLE NEW” No new references are possible for the resource. The resource will
1264 be dropped when the last reference to this resource is gone
- 1265 “CASCADE” The drop request will execute immediately and all references to the
1266 resource will be removed recursively

1267 If this parameter is not specified, the default value “IF UNUSED” MUST be used.

1268 A WS-Addressing Action header with the value

1269 <http://www.ogf.org/infod/INFODRegistry/DropPropertyVocabulary> MUST accompany the message

1270 INFOD Registry Response

1271 If the INFOD registry accepts the DropPropertyVocabulary request, it MUST respond to the WS
1272 endpoint specified in the request message with an DropPropertyVocabularyResponse message. The
1273 DropPropertyVocabulary response message is a message of the following form:

```
1274 <infod:DropPropertyVocabularyResponse>
1275   <infod:Status>
1276     xsd:string default "COMPLETED"
```

```

1277     </infod:Status>
1278 </infod:DropPropertyVocabularyResponse>

```

1279 The elements of the DropPropertyVocabularyResponse message are further described as follows:

1280 /infod:Status

1281 An indication that the request has been successfully executed.

1282 One of the following faults MUST be sent if the operation fails:

- 1283 • DropResourceAuthorizationFailure: User not authorized to drop the resource at this
1284 INFOD registry
- 1285 • UnknownResourceReferenceFault: An element has been referenced that is unknown to
1286 the INFOD registry
- 1287 • MissingRequiredParameterFault: A required parameter was not specified
- 1288 • ExecutionModeFault: Cannot use ExecutionMode provided

1289 The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP
1290 v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

1291 2.5.3 CreatePropertyVocabularyInstance

1292 The CreatePropertyVocabularyInstance operation creates a new instance of a particular property
1293 vocabulary previously created in the INFOD registry. An instance of a property vocabulary fills in
1294 values into the vocabulary structure defined by the Property Vocabulary (section 2.5.1) and relates a
1295 particular INFOD entry to the instance. The referenced entry is now identified to use the property
1296 vocabulary.

1297 As part of the processing of a CreatePropertyVocabularyInstance request message, the INFOD
1298 registry MUST create a new instance for that vocabulary.

1299 The format of the request message for CreatePropertyVocabularyInstance operation is as follows:

```

1300 <infod:CreatePropertyVocabularyInstance>
1301   <infod:EntryReference>
1302     wsa:EndPointReferenceType
1303   </infod:EntryReference>
1304   <infod:PropertyVocabularyReference>
1305     wsa:EndPointReferenceType
1306   </infod:PropertyVocabularyReference>
1307   <infod:PropertyVocabularyInstanceBody>
1308     {xsd:anyType} ?
1309   </infod:PropertyVocabularyInstanceBody>
1310 </infod:CreatePropertyVocabularyInstance>

```

1311 The elements of the CreatePropertyVocabularyInstance message are further described as follows:

1312 /infod:EntryReference

1313 EPR of the INFOD entry that the instance of the property vocabulary will be identified with.

1314 /infod:PropertyVocabularyReference

1315 EPR of a vocabulary that will be referenced to the INFOD resource.

1316 /infod:PropertyVocabularyInstanceBody

1317 An element that contains specific instance information that needs to match the structure of the
1318 vocabulary defined in VocabularyReference.

1319 A WS-Addressing Action header with the value
1320 <http://www.ogf.org/infod/INFODRegistry/CreatePropertyVocabularyInstance> MUST accompany the
1321 message.

1322 **INFOD Registry Response**

1323 If the INFOD registry accepts the CreatePropertyVocabularyInstance request, it MUST respond to the
1324 WS endpoint specified in the request message with a CreatePropertyVocabularyInstance response
1325 message.

1326 The CreatePropertyVocabularyInstanceResponse message is a message of the following form:

```
1327 <infod:CreatePropertyVocabularyInstanceResponse>
1328 <infod:PropertyVocabularyInstanceReference>
1329   wsa:EndPointReferenceType
1330 </infod:PropertyVocabularyInstanceReference>
1331 </infod:CreatePropertyVocabularyInstanceResponse>
```

1332 The elements of the CreatePropertyVocabularyInstanceResponse message are further described as
1333 follows:

1334 /infod:PropertyVocabularyInstanceReference

1335 An endpoint reference element, as defined by WS-Addressing, used to identify the newly
1336 created vocabulary instance.

1337 One of the following faults MUST be sent if the operation fails:

- 1338 • CreateResourceAuthorizationFault: User not authorized to create the INFOD resource at
1339 this INFOD registry
- 1340 • UnknownResourceReferenceFault: An resource has been referenced that is unknown to
1341 the INFOD registry
- 1342 • MissingRequiredParameterFault: A required parameter was not specified
- 1343 • UnSupportedVocabularyFault: Vocabulary Language not supported

1344 The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP
1345 v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

1346 **2.5.4 DropPropertyVocabularyInstance**

1347 The DropPropertyVocabularyInstance operation drops an existing instance of a particular property
1348 vocabulary previously created in the INFOD registry.

1349 The format of the request message for a DropPropertyVocabularyInstance operation is:

```
1350 <infod:DropPropertyVocabularyInstance>
1351 <infod:PropertyVocabularyInstanceReference>
1352   wsa:EndPointReferenceType
1353 </infod:PropertyVocabularyInstanceReference>
1354 <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
1355 </infod:DropPropertyVocabularyInstance>
```

1356 The elements of the DropPropertyVocabularyInstance message are further described as follows:

1357 /infod:PropertyVocabularyInstanceReference

1358 An endpoint reference element, as defined by WS-Addressing, used to identify the property
1359 vocabulary instance to drop from the Registry.

1360 /infod:ExecutionMode

1361 A parameter indicating the mode of execution of the drop request. Possible values are:

1362 "IF UNUSED" The drop request will execute only if the resource is unreferenced

1363 "DISABLE NEW" No new references are possible for the resource. The resource will
1364 be dropped when the last reference to this resource is gone

1365 "CASCADE" The drop request will execute immediately and all references to the
1366 resource will be removed recursively

1367 If this parameter is not specified, the default value "IF UNUSED" MUST be used.

1368 A WS-Addressing Action header with the value
1369 <http://www.ogf.org/infod/INFODRegistry/DropVocabularyInstance> MUST accompany the message.

1370 **INFOD Registry Response**

1371 If the INFOD registry accepts the DropPropertyVocabularyInstance request, it MUST respond to the
1372 WS endpoint specified in the request message with a DropPropertyVocabularyInstanceResponse
1373 message in the following form:

```
1374 <infod:DropPropertyVocabularyInstanceResponse>
1375   <infod:Status>
1376     xsd:string default "COMPLETED"
1377   </infod:Status>
1378 </infod:DropPropertyVocabularyInstanceResponse>
```

1379 The elements of the DropPropertyVocabularyInstanceResponse message are further described as
1380 follows:

1381 /infod:Status

1382 An indication that the request has been successfully executed.

1383 One of the following faults MUST be sent if the operation fails:

- 1384 • DropResourceAuthorizationFailure: User not authorized to drop the resource at this
1385 INFOD registry
- 1386 • UnknownResourceReferenceFault: An resource has been referenced that is unknown to
1387 the INFOD registry
- 1388 • MissingRequiredParameterFault: A required parameter was not specified
- 1389 • ExecutionModeFault: Cannot use ExecutionMode provided

1390 The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP
1391 v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

1392 **2.5.5 CreateDataVocabulary**

1393 As part of the processing of a CreateDataVocabulary request message, the INFOD registry MUST
1394 create a new resource for that vocabulary.

1395 The format of the request message for CreateDataVocabulary operation is:

```
1396 <infod:CreateDataVocabulary>
```

```

1397 <infod:DataVocabularyName> xsd:string </infod:DataVocabularyName> ?
1398 <infod:DataVocabularyDescription>
1399   xsd:string
1400 </infod:DataVocabularyDescription> ?
1401 <infod:DataVocabularyLanguage>
1402   {anyURI} (Namespace/URI of DataFormat)
1403 </infod:DataVocabularyLanguage>
1404 <infod:LanguageUsageDescription>
1405   xsd:string
1406 </infod:LanguageUsageDescription> ?
1407 <infod:DataVocabularyBody>
1408   xsd:anyType
1409 </infod:DataVocabularyBody>
1410 </infod:CreateDataVocabulary>

```

1411 The elements of the CreateDataVocabulary message are further described as follows:

1412 /infod:DataVocabularyName

1413 A string representing a name in the INFOD registry where the CreateDataVocabulary
1414 operation takes place. This name MAY NOT be unique.

1415 Names MUST NOT start with \$\$infod.

1416 /infod:DataVocabularyDescription

1417 A string representing a description of the vocabulary.

1418 /infod:DataVocabularyLanguage

1419 A URI defining the format of the data vocabulary.

1420 /infod:DataVocabularyBody

1421 A string representing a data vocabulary.

1422 This embedded string represents the vocabulary and MUST be encoded correctly as defined
1423 through the DataVocabularyLanguage definition (escape characters etc.)

1424 A WS-Addressing Action header with the value

1425 <http://www.ogf.org/infod/INFODRegistry/CreateDataVocabulary> MUST accompany the message.

1426 **INFOD Registry Response**

1427 If the INFOD registry accepts the CreateDataVocabulary request, it MUST respond to the WS
1428 endpoint specified in the request message with a CreateVocabularyResponse message.

1429 In case of a successful registration, the CreateVocabularyResponse message is a message of the
1430 following form:

```

1431 <infod:CreateDataVocabularyResponse>
1432 <infod:DataVocabularyReference>
1433   wsa:EndPointReferenceType
1434 </infod:DataVocabularyReference>
1435 </infod:CreateDataVocabularyResponse>

```

1436 The elements of the CreateVocabularyResponse message are further described as follows:

1437 /infod:DataVocabularyReference

1438 An endpoint reference element, as defined by WS-Addressing, used to identify the newly
1439 created vocabulary.

1440 One of the following faults MUST be sent if the operation fails:

- 1441 • CreateResourceAuthorizationFault: User not authorized to create a resource at this
1442 INFOD registry
- 1443 • UnknownResourceReferenceFault: An resource has been referenced that is unknown to
1444 the INFOD registry
- 1445 • MissingRequiredParameterFault: A required parameter was not specified
- 1446 • UnSupportedVocabularyFault: Vocabulary Language not supported

1447 The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP
1448 v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

1449 2.5.6 DropDataVocabulary

1450 The DropDataVocabulary operation drops a particular data vocabulary from an INFOD registry.

1451 The format of the request message for an DropDataVocabulary operation is:

```
1452 <infod:DropDataVocabulary>
1453   <infod:DataVocabularyReference>
1454     wsa:EndPointReferenceType
1455   </infod:DataVocabularyReference>
1456   <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
1457 </infod:DropDataVocabulary>
```

1458 The elements of the DropDataVocabulary message are further described as follows:

1459 /infod:DataVocabularyReference

1460 An endpoint reference element, as defined by WS-Addressing, used to identify the vocabulary
1461 to drop from the Registry.

1462 /infod:ExecutionMode

1463 A parameter indicating the mode of execution of the drop request. Possible values are:

- 1464 "IF UNUSED" The drop request will execute only if the resource is unreferenced
- 1465 "DISABLE NEW" No new references are possible for the resource. The resource will
1466 be dropped when the last reference to this resource is gone
- 1467 "CASCADE" The drop request will execute immediately and all references to the
1468 resource will be removed recursively

1469 If this parameter is not specified, the default value "IF UNUSED" MUST be used.

1470 A WS-Addressing Action header with the value

1471 <http://www.ogf.org/infod/INFODRegistry/DropDataVocabulary> MUST accompany the message

1472 INFOD Registry Response

1473 If the INFOD registry accepts the DropDataVocabulary request, it MUST respond to the WS endpoint
1474 specified in the request message with a DropDataVocabularyResponse message. The
1475 DropDataVocabulary response message is a message of the following form:

```
1476 <infod:DropDataVocabularyResponse>
1477   <infod:Status>
1478     xsd:string default "COMPLETED"
1479   </infod:Status>
```

1480 `</infod:DropDataVocabularyResponse>`

1481 The elements of the DropDataVocabularyResponse message are further described as follows:

1482 `/infod:Status`

1483 An indication that the request has been successfully executed.

1484 One of the following faults MUST be sent if the operation fails:

- 1485 • DropResourceAuthorizationFailure: User not authorized to drop the resource at this
1486 INFOD registry
- 1487 • UnknownResourceReferenceFault: An element has been referenced that is unknown to
1488 the INFOD registry
- 1489 • MissingRequiredParameterFault: A required parameter was not specified
- 1490 • ExecutionModeFault: Cannot use ExecutionMode provided

1491 The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP
1492 v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

1493 2.6 Data Source Entries

1494 A Data Source Entry relates a publisher entry with a data vocabulary.

1495 The following operations are used to manage data sources:

- 1496 • CreateDataSourceEntry (section 2.6.1)
- 1497 • DropDataSourceEntry (section 2.6.2)

1498 2.6.1 CreateDataSourceEntry

1499 The CreateDataSourceEntry operation creates a relation between an INFOD publisher entry and a
1500 data vocabulary at the INFOD registry. As part of the processing of an CreateDataSourceEntry
1501 operation message, the INFOD registry MUST create an INFOD vocabulary association resource.

1502 The format of the request message for an CreateDataSourceEntry operation is:

```

1503 <infod:CreateDataSourceEntry>
1504   <infod:DataSourceEntryName> ?
1505     xsd:string
1506   </infod:DataSourceEntryName>
1507   <infod:DataSourceEntryDescription>
1508     xsd:string
1509   </infod:DataSourceEntryDescription> ?
1510   <infod:PublisherEntryReference>
1511     wsa:EndPointReferenceType
1512   </infod:PublisherEntryReference>
1513   <infod:DataVocabularyReference>
1514     wsa:EndPointReferenceType
1515   </infod:DataVocabularyReference> +
1516   <infod:PropertyConstraint>
1517     xsd:any
1518   </infod:PropertyConstraint> *
1519 </infod:CreateDataSourceEntry>

```

1520 The elements of the CreateDataSourceEntry message are further described as follows:

- 1521 /infod:DataSourceEntryName
- 1522 A string representing the name of the data source entry. This name MAY NOT be unique.
- 1523 /infod:DataSourceEntryDescription
- 1524 A string representing a description of the data source entry.
- 1525 /infod:PublisherEntryReference
- 1526 The EPR of the publisher entry for which a data source entry is created.
- 1527 /infod:DataVocabularyReference
- 1528 The EPR(s) of a vocabulary with which to associate the publisher entry.
- 1529 /infod:PropertyConstraint
- 1530 Property constraints are used to specify which conditions must be satisfied by entries
- 1531 (subscribers and consumers) to be eligible to receive data from this data source. A property
- 1532 constraint MUST be formulated as an XQuery. The INFOD Base Use Case Scenarios (see
- 1533 <http://forge.gridforum.org/sf/go/doc13626?nav=1>) provide examples of XQueries.
- 1534 For example, a data sources identifies the set of consumers that are eligible to receive data
- 1535 from this data source.
- 1536 Note that the XQuery statement MUST be encoded correctly, i.e. characters such as ">"
- 1537 would be represented as ">".
- 1538 A WS-Addressing Action header with the value
- 1539 <http://www.ogf.org/infod/INFODRegistry/CreateDataSourceEntry> MUST accompany the message.
- 1540 **INFOD Registry Response**
- 1541 If the INFOD registry accepts the CreateDataSourceEntry request, it MUST respond to the WS
- 1542 endpoint specified in the request message with a CreateDataSourceEntryResponse message.
- 1543 The CreateDataSourceEntryResponse message is a message of the following form:
- 1544

```
<infod:CreateDataSourceEntryResponse>
```
- 1545

```
<infod:DataSourceEntryReference>
```
- 1546

```
  wsa:EndPointReferenceType
```
- 1547

```
</infod:DataSourceEntryReference>
```
- 1548

```
</infod:CreateDataSourceEntryResponse>
```
- 1549 The elements of the response message are further described as follows:
- 1550 /infod:DataSourceEntryReference
- 1551 An endpoint reference element, as defined by WS-Addressing, used to identify the newly
- 1552 created vocabulary association.
- 1553 One of the following faults MUST be sent if the operation fails:
- 1554 • CreateResourceAuthorizationFault: User not authorized to create the resource at this
- 1555 INFOD registry
- 1556 • UnknownResourceReferenceFault: An resource has been referenced that is unknown to
- 1557 the INFOD registry
- 1558 • MissingRequiredParameterFault: A required parameter was not specified
- 1559 • UnsupportedXQueryFault: The XQuery specified could not be parsed correctly

1560 The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP
1561 v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

1562 2.6.2 DropDataSourceEntry

1563 The DropDataSourceEntry operation drops a data source entry from an INFOD registry.

1564 The format of the request message for a DropDataSourceEntry operation is:

```
1565 <infod:DropDataSourceEntry>
1566   <infod:DataSourceEntryReference>
1567     wsa:EndPointReferenceType
1568   </infod:DataSourceEntryReference>
1569   <infod:ExecutionMode> xsd:string </infod:ExecutionMode>
1570 </infod:DropDataSourceEntry>
```

1571 The elements of the DropDataSourceEntry message are further described as follows:

1572 /infod:DataSourceEntryReference

1573 An endpoint reference element, as defined by WS-Addressing, used to identify the association
1574 to drop from the Registry.

1575 /infod:ExecutionMode

1576 A parameter indicating the mode of execution of the drop request. Possible values are:

1577 "IF UNUSED" The drop request will execute only if the resource is unreferenced

1578 "DISABLE NEW" No new references are possible for the resource. The resource will
1579 be dropped when the last reference to this resource is gone

1580 "CASCADE" The drop request will execute immediately and all references to the
1581 resource will be removed recursively

1582 If this parameter is not specified, the default value "IF UNUSED" MUST be used.

1583 A WS-Addressing Action header with the value

1584 <http://www.ogf.org/infod/INFODRegistry/DropDataSourceEntry> MUST accompany the message.

1585 INFOD Registry Response

1586 If the INFOD registry accepts the DropDataSourceEntry request, it MUST respond to the WS endpoint
1587 specified in the request message with a DropDataSourceEntryResponse message. The
1588 DisCreateDataSourceEntryResponse message is a message of the following form:

```
1589 <infod:DropDataSourceEntryResponse>
1590   <infod:Status>
1591     xsd:string default "COMPLETED"
1592   </infod:Status>
1593 </infod:DropDataSourceEntryResponse>
```

1594 The elements of the DropDataSourceEntryResponse message are further described as follows:

1595 /infod:Status

1596 An indication that the request has been successfully executed.

1597 One of the following faults MUST be sent if the operation fails:

- 1598 • DropResourceAuthorizationFailure: User not authorized to drop the resource at this
1599 INFOD registry

- 1600 • UnknownResourceReferenceFault: An resource has been referenced that is unknown to
1601 the INFOD registry
- 1602 • MissingRequiredParameterFault: A required parameter was not specified
- 1603 • ExecutionModeFault: Cannot use ExecutionMode provided

1604 The message MUST be sent using the WS-Base Faults. For examples using SOAP, see the SOAP
1605 v1.2. Base Fault Spec (see http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf).

1606 2.7 The GetMetaData Operation

1607 The Base Meta Data Access interface provides access to data contained in an INFOD registry. The
1608 request is formulated as an XQuery and the result is returned according to the specification in the
1609 return clause of the XQuery.

1610 The format of the request message for a GetMetadata operation is:

```
1611 <infod:GetMetaData>
1612   <infod:MetaDataQueryExpression>
1613     {xsd:anyType}
1614   </infod:MetaDataQueryExpression>
1615 </infod:GetMetadata>
```

1616 The elements of the GetMetadata message are further described as follows:

1617 /infod:MetaDataQueryExpression

1618 The element MUST be a valid XQuery or an XPath expression for the INFOD registry.

1619 The INFOD registry is fully qualified by an INFOD registry service name appended to the
1620 string "INFODRegistry.xml". A fully qualified name allows the registry instance to be
1621 referenced uniquely.

1622 An example for a fully qualified INFOD registry is:

1623 <http://www.w3c.org/OGF/INFOD/Instance/INFODRegistry.xml>.

1624 The INFOD registry service name need not be hard coded into the XQuery fn:doc but could
1625 be specified by setting the base-URI to be the service name e.g. declare base-URI
1626 "<http://www.w3c.org/OGF/INFOD/Instance/INFODRegistry.xml>". This indirection allows us to
1627 specify specific a registry amongst many in a given environment.

1628

1629 **Default XPath expressions:**

1630 In addition to supporting user defined Xpath/XQuery expressions, INFOD reserves the following paths
1631 and mandates their implementation.

- 1632 • **All publishers** - fn:doc('INFODRegistry.xml')/publishers/\$\$infodPublisher
- 1633 • **All subscribers** - fn:doc('INFODRegistry.xml')/subscribers/\$\$infodSubscriber
- 1634 • **All consumers** - fn:doc('INFODRegistry.xml')/consumers/\$\$infodConsumer
- 1635 • **All subscriptions** - fn:doc('INFODRegistry.xml')/subscriptions/\$\$infodSubscription
- 1636 • **All property vocabularies** -
1637 fn:doc('INFODRegistry.xml')/propertyvocabularies/\$\$infodPropertyVocabulary

- 1638 • **All property vocabulary instances** -
- 1639 fn:doc('INFODRegistry.xml')/propertyvocabularyinstances/\$\$infodPropertyVocabularyInstance
- 1640 • **All data vocabulary** - fn:doc('INFODRegistry.xml')/datavocabularies/\$\$infodDataVocabulary

1641 A WS-Addressing Action header with the value <http://www.ogf.org/infod/INFODRegistry/GetMetaData>
 1642 MUST accompany the message.

1643 **INFOD Registry Response**

1644 The response of the INFOD registry is:

```

1645 <infod:GetMetaDataQueryResponse>
1646   <infod:MetaDataQueryResult>
1647     {xsd:anyType}
1648     <infod:GetMetaDataQueryResult>
1649   </infod:MetaDataQueryResult>
  </infod:GetMetaDataQueryResponse>

```

1650 The content of infod:GetMetaDataQueryResult MUST be structured according to the return
 1651 specification in the GetMetaData request.

1652 One of the following faults MUST be sent if the operation fails:

- 1653 • **GetMetaDataAuthorizationFailure:** User not authorized to use the operation at this INFOD
 1654 registry
- 1655 • **MissingRequiredParameterFault:** A required parameter was not specified
- 1656 • **UnsupportedXQueryFault:** The XQuery specified could not be parsed correctly

1657 The message MUST be structured according to the WS-Base Faults specification. For examples using
 1658 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).
 1659

3 Base INFOD Notification Interfaces

1660

1661 We divide notifications between INFOD components into two major categories: notifications from
 1662 publishers to consumers which carry the actual data, and notifications from the registry to publishers,
 1663 subscribers, and consumers which contain information about relevant state changes in the registry.
 1664 INFOD does not use the WSN notify interface due to different header requirements.

3.1 Notifications from Publishers to Consumers

1666 An INFOD publisher uses a Notify operation similar to that defined by WS-Notification to send
 1667 messages to an INFOD consumer (see <http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.3-draft-01.pdf>).
 1668

1669

1670 The following xml describes the format of an INFOD Notify message:

```

1671 <infod:Notify>
1672   <infod:NotificationMessage>
1673     <infod:SubscriptionReference>
1674       wsa:EndpointReferenceType
1675     </infod:SubscriptionReference> ?
1676     <infod:Topic Dialect="xsd:anyURI">
1677       {any} ?
1678     </infod:Topic>?
1679     <infod:PublisherReference>
1680       wsa:EndpointReferenceType
1681     </infod:PublisherReference> ?
1682     <infod:Message>
1683       {any}
1684     </infod:Message>
1685   </infod:NotificationMessage> +
1686   {any} *
1687 </infod:Notify>
  
```

1688 The components of the Notify message are further described as follows:

1689 /infod:Notify

1690 Contains a collection of one or more Notifications.

1691 /infod:NotificationMessage

1692 Contains a Notification payload.

1693 /infod:SubscriptionReference

1694 An endpoint reference to the Subscription that is associated with the Notify message.

1695 /infod:Topic

1696 An endpoint reference to the VocabularyAssociation representing the source of the payload.

1697 /infod:Topic/@Dialect

1698 An endpoint reference to the vocabulary that was used to structure the payload.

1699 /infod:ProducerReference

1700 An endpoint reference to the Publisher that produced the Notification.

1701 /infod:Message

1702 The actual Notification payload.

1703 /infod:Notify/{any}

1704 The Notify message also allows for open content, in order to accommodate elements that may
 1705 be needed by extensions built on the WSN BaseNotification (see [http://docs.oasis-
 1707 open.org/wsn/2004/06/wsn-WS-BaseNotification-1.3-draft-01.pdf](http://docs.oasis-

 1706 open.org/wsn/2004/06/wsn-WS-BaseNotification-1.3-draft-01.pdf)), including those providing
 additional filtering mechanisms.

1708 A WS-Addressing Action header with the value <http://www.ogf.org/infod/INFODNotify/Notify> MUST
 1709 accompany the message

1710 INFOD Registry Response

1711 No response is expected from the INFOD consumer upon receipt of this message.

1712 Example SOAP Encoding of the Notify Message

1713 The following is a non-normative example of a Notify request message using SOAP:

```

1714 <s:Envelope ... >
1715   <s:Header>
1716     <wsa:Action>
1717       http://www.ogf.org/infod/INFODNotify/Notify
1718     </wsa:Action>
1719     ...
1720   </s:Header>
1721   <s:Body>
1722     <infod:Notify>
1723       <infod:NotificationMessage>
1724         <infod:SubscriptionReference>
1725           <wsa:Address>
1726             http://www.example.org/SomeSubscripition
1727           </wsa:Address>
1728         </infod:SubscriptionReference>
1729         <infod:Topic Dialect=
1730           "http://www.myinfodregistry.com/infod/MyDataVocabularyEPR">
1731           infod:DatavocabularyEPR
1732         </infod:Topic>
1733         <infod:ProducerReference>
1734           <wsa:Address>
1735             http://www.example.org/Publisher
1736           </wsa:Address>
1737         </infod:ProducerReference>
1738         <infod:Message>
1739           <MyDataVocabulary:MessageContent>MessageDataContent</MyDataVocabul
1740 ary:MessageContent>
1741         </infod:Message>
1742       </infod:NotificationMessage>
1743     </infod:Notify>
1744   </s:Body>
1745 </s:Envelope>
  
```

1747 3.2 Notification from the Registry

1748 The registry sends notifications to those publishers, subscribers and consumers that have registered
 1749 for them. Changes of state within the registry lead to generation of events. The specifics of the
 1750 payload and the condition under which a notification MUST be sent are described in the following
 1751 section:

- 1752 • Notification of publishers (section 3.2.1)
- 1753 • Notification of subscribers (section 3.2.2)
- 1754 • Notification of consumers (section 3.2.3)

1755 3.2.1 Notification of Publishers

1756 The INFOD registry will inform publishers that need to react to changes in the INFOD registry. The
 1757 notification is conditional on the information in the publisher entry. Publishers SHOULD react
 1758 immediately to these notifications.

1759 A new publisher MUST be informed about each subscription that requires⁴ this publisher to send
 1760 messages; there will be one notification per subscription.

1761 For existing publishers notifications MUST be sent about those subscriptions that mandate different
 1762 messages or mandate messages to be sent to different consumers. An empty list of static and
 1763 dynamic consumers indicates that a publisher MUST stop publishing for the referenced subscription

1764 Notifications are determined by processing the property constraints and the vocabulary reference in
 1765 the data constraints.

1766 The notification contains the following message body:

```

1767 <infod:PublisherNotification>
1768   <infod:SubscriptionReference>
1769     wsa:EndPointReferenceType
1770   </infod:SubscriptionReference>
1771   <infod:ConsumerEntryReference>
1772     wsa:EndPointReferenceType
1773   <infod:ConsumerEntryReference> *
1774   <infod:DynamicConsumerConstraint>
1775     {xsd:anyType}
1776   <infod:DynamicConsumerConstraint> *
1777   <infod>DataConstraint>
1778     {xsd:anyType}
1779   <infod>DataConstraint> *
1780 </infod:PublisherNotification>
  
```

1781 The message content is further described as follows:

1782 /infod:SubscriptionReference

1783 This is the EPR of the subscription for which the information is provided.

1784 If all other parameters are omitted the publisher does not need to process this subscription
 1785 any longer. This EPR is not valid after the subscription is dropped. However, the no longer
 1786 valid EPR is propagated, as some of the publishers may be using the EPR for their internal
 1787 references.

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

- 1788 /infod:ConsumerEntryReference
- 1789 This is a list of 0 to n EPR references of consumer entries. The list of consumers is computed
1790 by the INFOD Registry and given to each publisher.
- 1791 /infod:DynamicConsumerConstraint
- 1792 This is an expression that directs the publisher to determine the consumer(s) based on the
1793 listed expressions. Each expression references data that are created by the publishers, e.g.
1794 messages to be published, and references properties of INFOD Registry resources.
- 1795 The subscription should be discarded if there is no entry for StaticConsumers and for
1796 DynamicConsumerConstraint.
- 1797 /infod:DataConstraint
- 1798 These are the data constraints as specified in the referenced subscription.
- 1799 WS-Addressing of the action MUST contain the URI
1800 <http://www.ogf.org/infod/INFODNotify/SubscriptionNotification>.

1801 3.2.2 Notification of Subscribers

- 1802 The INFOD registry MUST inform subscriber that need to know the impact of changes in the INFOD
1803 registry on their subscriptions; e.g., subscription with an EPR pointing to them. The notification is
1804 conditional on the information in the subscriber entry.
- 1805 In reaction to a newly created or replaced subscription the subscriber MUST be informed which
1806 publishers send and consumers receive messages based on that subscription.
- 1807 In reaction to any other change in the INFOD registry the subscriber MUST be informed about those
1808 subscription for which the list of publishers or consumers has changed.
- 1809 Notifications are determined by processing the property constraints and the vocabulary reference in
1810 the data constraints.

1811 The notification contains the following message body:

```

1812 <infod:SubscriberNotification.
1813 <infodSubscriptionReference>
1814   wsa:endPointReferenceType
1815 </infodSubscriptionReference>
1816 <infod:PublisherEntryReference>
1817   wsa:endPointReferenceType
1818 </infod:PublisherEntryReference> *
1819 <infod:ConsumerEntryReference>
1820   wsa:endPointReferenceType
1821 </infod:ConsumerEntryReference> *
1822 <infod:SubscriberNotification

```

1823 The message content is further described as follows:

- 1824 /infod:SubscriptionReference
- 1825 This is the EPR of the subscription for which the information is provided
- 1826 /infod:PublisherEntryReference
- 1827 This is a list of 0 to n EPR references of publisher entries. The list of publisher entries is
1828 computed by the INFOD registry.
- 1829 Infod:ConsumerEntryReference

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

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

1834 3.2.3 Notification of Consumers

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

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

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

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

1846 The notification will not be send to dynamic consumers.

1847 The notification contains the following message body:

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

1856 The message content is further described as follows:

1857 /infod:SubscriptionReference

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

1859 /infod:PublisherEntryReference

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

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

1864 4 Security Considerations

1865 An INFOD operating environment consists of a set of publishers, consumers and registries. All the
1866 above service components operate in different security domains and require “long-term” secure
1867 communication of messages. Additionally, as the INFOD services operate in a web services
1868 environment, SOAP may be used as the base communication protocol. SOAP based communication
1869 between services can be secured by using the mechanisms described by the *WS security*
1870 specification (see [http://www.oasis-open.org/committees/download.php/5531/oasis-200401-wsssoap-
1872 message-security-1.0.pdf](http://www.oasis-open.org/committees/download.php/5531/oasis-200401-wsssoap-
1871 message-security-1.0.pdf)). Although, the use of WS-Security provides the mechanisms to
1873 accommodate multiple security tokens and encryption technologies, it remains limited to providing a
1874 secured point-to-point communication mechanism on a message level. However, INFOD services
1875 need to build upon this security mechanism to describe the security context under which they could
1876 sustain long running exchanges of messages. A communication session between the two parties such
1877 as publisher and consumer serves as the basis for establishing the security context. Establishing a
1878 security context between system entries allows secured messaging on the session level and reduces
1879 the synchronization overheads required to obtain it on per-message basis. *WS-Secure Conversation*
1880 (see <ftp://www6.software.ibm.com/software/developer/library/ws-secureconversation.pdf>) provides the
mechanism for maintaining such long-term contexts for message exchange.

1881 The INFOD model **RECOMMENDS** the establishment of the following contexts:

- 1882 • Publisher – Registry secured context, with Registry as the context security token creator.
- 1883 • Consumer – Registry secured context, with Registry as the context security token creator.
- 1884 • Subscriber – Registry secured context, with Registry as the context security token creator.
- 1885 • Publisher – Consumer secured context, with Publisher as the context security token creator. It
1886 may be possible to support registry mediated delegation, where the registry mediates the
1887 establishment of trust between producer and consumer.

1888 Authentication remains a crucial aspect of formation of a secured conversation. Hence, the
1889 specification identifies the objects that create the secured context. It is envisaged that an INFOD-
1890 Registry will provide services to multiple publishers/consumers/subscriptions and controls the access
1891 to this shared state. Hence, it is imperative to have the INFOD-Registry act as the authenticator for
1892 other services. Similarly, a publisher controls the dissemination of the messages and hence is
1893 deemed responsible for establishing the context with the consumers. In the future, it is envisaged that
1894 in later versions INFOD may introduce mechanisms for mutual authentication based on trust
1895 mechanisms. An example, is that future authentication of consumers by the publishers could be
1896 mediated by the registry.

1897 4.1 Message Encryption and Data Privacy Requirements

1898 INFOD advocates the use of mutual filtering techniques to provide smart dissemination of the
1899 messages. Mutual filtering requires the publishers and consumers to be able to interpret the contents
1900 of the messages being routed. As INFOD isolates a publisher from a consumer and does not require
1901 either the publishers or the consumers to authenticate each other, secured point-to-point
1902 communication becomes a non-issue for the base specification. It is assumed that publishers are able
1903 to authenticate the consumers based on their EPR references.

1904 INFOD system provides non-repudiation of transmitted messages. It is recommended that the
1905 publisher signs its message and also provides its public key for subsequent verification by the
1906 recipients. It is suggested that the public key of each publisher is registered with the INFOD registry

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

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

1913 **4.2 Integration with Authorization Model**

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

- 1919 • Access model for INFOD resources
- 1920 • Access model for INFOD messages

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

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

1932 5 Appendix I – XML Schema

1933 This section includes the following xml schema:

- 1934 • Publisher Entry (section 5.1)
- 1935 • Subscriber Entry (section 5.2)
- 1936 • Consumer Entry (section 5.3)
- 1937 • Subscription (section 5.4)
- 1938 • Property Vocabulary (section 5.5)
- 1939 • Property Vocabulary Instance (section 5.6)
- 1940 • Data Vocabulary (section 5.7)
- 1941 • Data Source Entry (section 5.8)
- 1942 • INFOD Error Messages (section 5.9)
- 1943 • INFOD Notification (section 5.10)
- 1944 • Publishers Notification (section 5.11)
- 1945 • Subscriber Notification (section 5.12)
- 1946 • Consumer Notification (section 5.13)

1947 The following graphic depicts the XML schema relations for the www.ogf.org/infod/INFODRegistry
1948 Namespace. The circles in the first row represent the operations of the INFOD registry. The circles in
1949 the second row show the vocabularies that are managed by the vocabulary operations. The boxes in
1950 the third row represent the resources, data entries and property vocabulary instances. Within the xml
1951 schema of those boxes, there are reference pointers to other entries or vocabularies, represented by
1952 EPRs. The honeycombs represent the external web services EPRs that are associated to the
1953 resources. Note that the same Web Service EPR can be associated to multiple INFOD resources.

1954

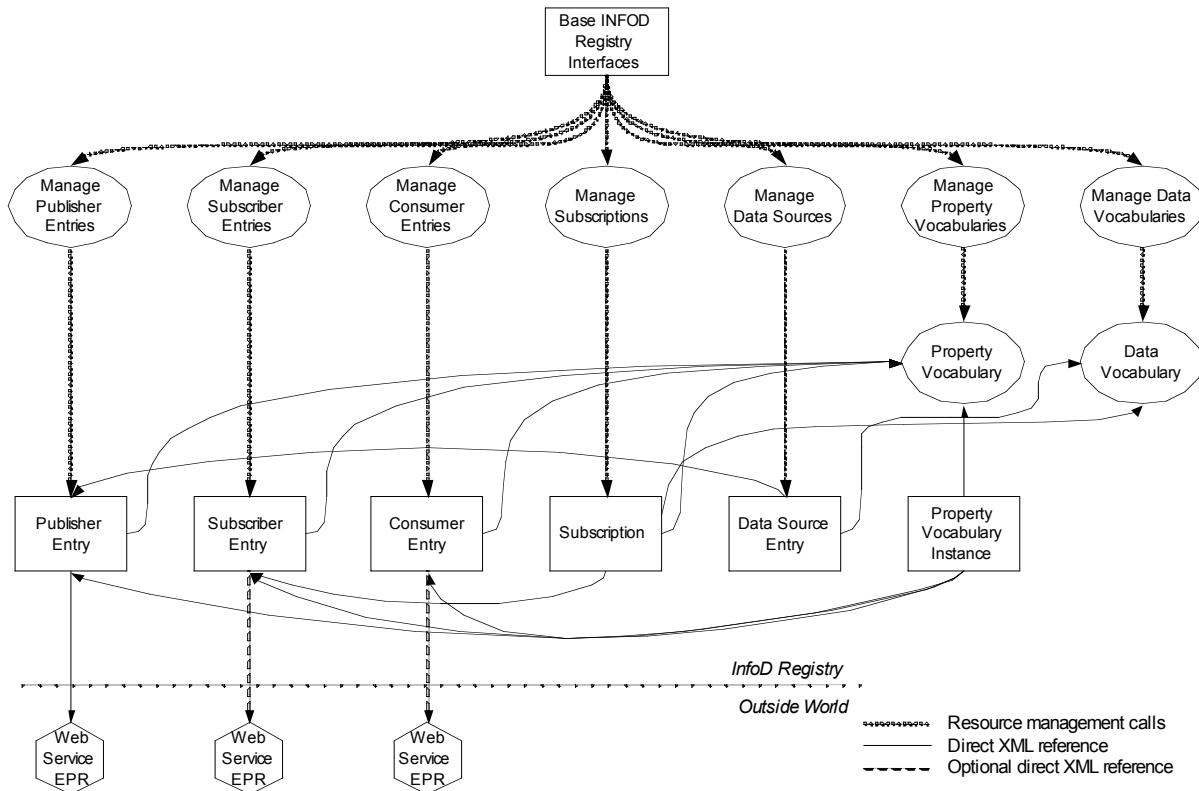


Figure 6: XML schema relations of INFODRegistry namespace

1955
1956

1957 Figure 6, provides a schematic representation of internals of INFOD registry. Web Service EPRs from
1958 publisher, subscriber and consumers are associated with publisher, subscriber and consumer entries
1959 respectively. Associations between entries with data and property vocabulary instances are also
1960 highlighted.

1961

1962 5.1 Publisher Entry

```

1963 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1964           xmlns:ident="http://www.ogf.org/infod"
1965           targetNamespace="http://www.ogf.org/infod/INFODRegistry">
1966
1967   <xsd:element name="infodPublisherEntry">
1968     <xsd:annotation>
1969       <xsd:documentation>
1970         Description of Publisher Entries
1971       </xsd:documentation>
1972     </xsd:annotation>
1973     <xsd:complexType>
1974       <xsd:sequence>
1975         <xsd:element name="WSReference"
1976                     type="wsa:EndpointReferenceType"
1977                     minOccurs="0" maxOccurs="1"/>
1978         <xsd:element name="PublisherName" type="xsd:string"
1979                     minOccurs="0" maxOccurs="1"/>
1980         <xsd:element name="PublisherDescription" type="xsd:string"
1981                     minOccurs="0" maxOccurs="1"/>
1982         <xsd:element name="PropertyConstraint" type="xsd:any"
  
```

```

1983         minOccurs="0" maxOccurs="unbounded"/>
1984     <xsd:element name="Notification" type="xsd:boolean"
1985         nillable="true"
1986         minOccurs="0" maxOccurs="1"/>
1987     </xsd:sequence>
1988 </xsd:complexType>
1989 </xsd:element>
1990 </xsd:schema>

```

1991 5.2 Subscriber Entry

```

1992 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1993     xmlns:infod="http://www.ogf.org/infod"
1994     targetNamespace="http://www.ogf.org/infod/INFODRegistry">
1995
1996     <xsd:element name="infodSubscriberEntry">
1997         <xsd:annotation>
1998             <xsd:documentation>
1999                 Description of Subscriber Entries
2000             </xsd:documentation>
2001         </xsd:annotation>
2002         <xsd:complexType>
2003             <xsd:sequence>
2004                 <xsd:element name="WSReference"
2005                     type="wsa:EndpointReferenceType"
2006                     minOccurs="0" maxOccurs="1"/>
2007                 <xsd:element name="SubscriberName" type="xsd:string"
2008                     minOccurs="0" maxOccurs="1"/>
2009                 <xsd:element name="SubscriberDescription" type="xsd:string"
2010                     minOccurs="0" maxOccurs="1"/>
2011                 <xsd:element name="PropertyConstraint" type="xsd:any"
2012                     minOccurs="0" maxOccurs="unbounded"/>
2013                 <xsd:element name="Notification" type="xsd:boolean"
2014                     nillable="true" minOccurs="0" maxOccurs="1"/>
2015             </xsd:sequence>
2016         </xsd:complexType>
2017     </xsd:element>
2018 </xsd:schema>

```

2019 5.3 Consumer Entry

```

2020 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2021     xmlns:infod="http://www.ogf.org/infod"
2022     targetNamespace="http://www.ogf.org/infod/INFODRegistry">
2023
2024     <xsd:element name="infodConsumerEntry">
2025         <xsd:annotation>
2026             <xsd:documentation>
2027                 Description of Consumer Entries
2028             </xsd:documentation>
2029         </xsd:annotation>
2030         <xsd:complexType>
2031             <xsd:sequence>
2032                 <xsd:element name="WSReference"
2033                     type="wsa:EndpointReferenceType"
2034                     minOccurs="0" maxOccurs="1"/>
2035                 <xsd:element name="ConsumerrName" type="xsd:string"
2036                     minOccurs="0" maxOccurs="1"/>
2037                 <xsd:element name="ConsumerDescription" type="xsd:string"

```

```

2038         minOccurs="0" maxOccurs="1"/>
2039     <xsd:element name="PropertyConstraint" type="xsd:any"
2040         minOccurs="0" maxOccurs="unbounded"/>
2041     <xsd:element name="Notification" type="xsd:boolean"
2042         nillable="true" minOccurs="0" maxOccurs="1"/>
2043     </xsd:sequence>
2044 </xsd:complexType>
2045 </xsd:element>
2046 </xsd:schema>

```

2047 5.4 Subscription

```

2048 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2049     xmlns:ident="http://www.ogf.org/infod"
2050     targetNamespace="http://www.ogf.org/infod/INFODRegistry">
2051
2052     <xsd:element name="infodSubscription">
2053         <xsd:annotation>
2054             <xsd:documentation>
2055                 Description of Subscriptions
2056             </xsd:documentation>
2057         </xsd:annotation>
2058         <xsd:complexType>
2059             <xsd:sequence>
2060                 <xsd:element name="SubscriptionName" type="xsd:string"
2061                     minOccurs="0" maxOccurs="1"/>
2062                 <xsd:element name="SubscriptionDescription" type="xsd:string"
2063                     minOccurs="1" maxOccurs="1"/>
2064                 <xsd:element name="SubscriberEntryReference"
2065                     type="wsa:EndpointReferenceType"
2066                     minOccurs="0" maxOccurs="1"/>
2067                 <xsd:element name="DataConstraint" type="xsd:any"
2068                     minOccurs="0" maxOccurs="1"/>
2069                 <xsd:element name="PropertyConstraint" type="xsd:any"
2070                     minOccurs="0" maxOccurs="unbounded"/>
2071                 <xsd:element name="DynamicConsumerConstraint" type="xsd:any"
2072                     minOccurs="0" maxOccurs="unbounded"/>
2073             </xsd:sequence>
2074         </xsd:complexType>
2075     </xsd:element>
2076 </xsd:schema>

```

2077 5.5 Property Vocabulary

```

2078 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2079     xmlns:ident="http://www.ogf.org/infod"
2080     targetNamespace="http://www.ogf.org/infod/INFODRegistry">
2081
2082     <xsd:element name="infodPropertyVocabulary">
2083         <xsd:annotation>
2084             <xsd:documentation>
2085                 Description of a Property Vocabulary
2086             </xsd:documentation>
2087         </xsd:annotation>
2088         <xsd:complexType>
2089             <xsd:sequence>
2090                 <xsd:element name="PropertyVocabularyName" type="xsd:string"
2091                     minOccurs="0" maxOccurs="1"/>
2092                 <xsd:element name="PropertyVocabularyDescription" type="xsd:string"

```

```

2093         minOccurs="0" maxOccurs="1"/>
2094         <xsd:element name="PropertyVocabularyBody" type="xsd:any"
2095             minOccurs="1" maxOccurs="1"/>
2096     </xsd:sequence>
2097 </xsd:complexType>
2098 </xsd:element>
2099 </xsd:schema>

```

2100 5.6 Property Vocabulary Instance

```

2101 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2102             xmlns:ident="http://www.ogf.org/infod"
2103             targetNamespace="http://www.ogf.org/infod/INFODRegistry">
2104
2105     <xsd:element name="infodPropertyVocabularyInstance">
2106         <xsd:annotation>
2107             <xsd:documentation>
2108                 Description of Property Vocabulary Instance
2109             </xsd:documentation>
2110         </xsd:annotation>
2111         <xsd:complexType>
2112             <xsd:sequence>
2113                 <xsd:element name="EntryReference"
2114                     type="wsa:EndpointReferenceType"
2115                     minOccurs="1" maxOccurs="1"/>
2116                 <xsd:element name="PropertyVocabularyReference"
2117                     type="wsa:EndpointReferenceType"
2118                     minOccurs="1" maxOccurs="1"/>
2119                 <xsd:element name="PropertyVocabularyInstanceBody"
2120                     type="xsd:schema"
2121                     minOccurs="1" maxOccurs="1"/>
2122             </xsd:sequence>
2123         </xsd:complexType>
2124     </xsd:element>
2125 </xsd:schema>

```

2126 5.7 Data Vocabulary

```

2127 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2128             xmlns:ident="http://www.ogf.org/infod"
2129             targetNamespace="http://www.ogf.org/infod/INFODRegistry">
2130
2131     <xsd:element name="infodDataVocabulary">
2132         <xsd:annotation>
2133             <xsd:documentation>
2134                 Description of Data Vocabulary
2135             </xsd:documentation>
2136         </xsd:annotation>
2137         <xsd:complexType>
2138             <xsd:sequence>
2139                 <xsd:element name="DataVocabularyName" type="xsd:string"
2140                     minOccurs="0" maxOccurs="1"/>
2141                 <xsd:element name="DataVocabularyDescription" type="xsd:string"
2142                     minOccurs="0" maxOccurs="1"/>
2143                 <xsd:element name="DataVocabularyLanguage" type="xsd:string"
2144                     minOccurs="1" maxOccurs="1"/>
2145                 <xsd:element name="DataVocabularyBody" type="xsd:any"
2146                     minOccurs="1" maxOccurs="1"/>
2147             </xsd:sequence>

```

```

2148     </xsd:complexType>
2149 </xsd:element>
2150 </xsd:schema>

```

2151 5.8 Data Source Entry

```

2152 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2153           xmlns:ident="http://www.ogf.org/infod"
2154           targetNamespace="http://www.ogf.org/infod/INFODRegistry">
2155
2156   <xsd:element name="infodDataSourceEntry">
2157     <xsd:annotation>
2158       <xsd:documentation>
2159         Description of Data Source Entries
2160       </xsd:documentation>
2161     </xsd:annotation>
2162     <xsd:complexType>
2163       <xsd:sequence>
2164         <xsd:element name="CreateDataSourceEntryName" type="xsd:string"
2165           minOccurs="0" maxOccurs="1"/>
2166         <xsd:element name="DataSourceEntryDescription"
2167           type="xsd:string"
2168           minOccurs="0" maxOccurs="1"/>
2169         <xsd:element name="PublisherEntryReference"
2170           type="wsa:EndpointReferenceType"
2171           minOccurs="1" maxOccurs="1"/>
2172         <xsd:element name="DataVocabularyReference"
2173           type="wsa:EndpointReferenceType"
2174           minOccurs="1" maxOccurs="1"/>
2175         <xsd:element name="PropertyConstraint" type="xsd:any"
2176           minOccurs="0" maxOccurs="unbounded"/>
2177       </xsd:sequence>
2178     </xsd:complexType>
2179   </xsd:element>
2180 </xsd:schema>

```

2181 5.9 Error Messages

```

2182 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2183           xmlns:wsrf-bf="http://www.ogf.org/infod/fault"
2184           xmlns:ident="http://www.ogf.org/infod"
2185           targetNamespace="http://www.ogf.org/infod/INFODRegistry">
2186
2187   <xsd:complexType name="CreateResourceAuthorizationFaultType">
2188     <xsd:complexContent>
2189       <xsd:extension base="wsrf-bf:BaseFaultType"/>
2190     </xsd:complexContent>
2191   </xsd:complexType>
2192   <xsd:element name="CreateResourceAuthorizationFault"
2193     type="infod:CreateResourceAuthorizationFaultType"/>
2194
2195   <xsd:complexType name="ReplaceResourceAuthorizationFaultType">
2196     <xsd:complexContent>
2197       <xsd:extension base="wsrf-bf:BaseFaultType"/>
2198     </xsd:complexContent>
2199   </xsd:complexType>
2200   <xsd:element name="ReplaceResourceAuthorizationFault"
2201     type="infod:ReplaceResourceAuthorizationFaultType"/>
2202

```

```

2203 <xsd:complexType name="DropResourceAuthorizationFailureType">
2204   <xsd:complexContent>
2205     <xsd:extension base="wsrf-bf:BaseFaultType"/>
2206   </xsd:complexContent>
2207 </xsd:complexType>
2208 <xsd:element name="DropResourceAuthorizationFailure"
2209   type="infod:DropResourceAuthorizationFailureType"/>
2210
2211 <xsd:complexType name="ExecutionModeFaultType">
2212   <xsd:complexContent>
2213     <xsd:extension base="wsrf-bf:BaseFaultType"/>
2214   </xsd:complexContent>
2215 </xsd:complexType>
2216 <xsd:element name="ExecutionModeFault"
2217   type="infod:ExecutionModeFaultType"/>
2218
2219 <xsd:complexType name="UnsupportedVocabularyFaultType">
2220   <xsd:complexContent>
2221     <xsd:extension base="wsrf-bf:BaseFaultType"/>
2222   </xsd:complexContent>
2223 </xsd:complexType>
2224 <xsd:element name="UnsupportedVocabularyFault"
2225   type="infod:UnsupportedVocabularyFaultType"/>
2226
2227 <xsd:complexType name="UnsupportedXQueryFaultType">
2228   <xsd:complexContent>
2229     <xsd:extension base="wsrf-bf:BaseFaultType"/>
2230   </xsd:complexContent>
2231 </xsd:complexType>
2232 <xsd:element name="UnsupportedXQueryFault"
2233   type="infod:UnsupportedXQueryFaultType"/>
2234
2235 <xsd:complexType name="GetMetaDataAuthorizationFailureType">
2236   <xsd:complexContent>
2237     <xsd:extension base="wsrf-bf:BaseFaultType"/>
2238   </xsd:complexContent>
2239 </xsd:complexType>
2240 <xsd:element name="GetMetaDataAuthorizationFailure"
2241   type="infod:GetMetaDataAuthorizationFailureType"/>
2242
2243 <xsd:complexType name="UnknownResourceReferenceFaultType">
2244   <xsd:complexContent>
2245     <xsd:extension base="wsrf-bf:BaseFaultType"/>
2246   </xsd:complexContent>
2247 </xsd:complexType>
2248 <xsd:element name="UnknownResourceReferenceFault"
2249   type="infod:UnknownElementReferenceFaultType"/>
2250
2251 <xsd:complexType name="MissingRequiredParameterFaultType">
2252   <xsd:complexContent>
2253     <xsd:extension base="wsrf-bf:BaseFaultType"/>
2254   </xsd:complexContent>
2255 </xsd:complexType>
2256 <xsd:element name="MissingRequiredParameterFault"
2257   type="infod:MissingRequiredParameterFaultType"/>
2258
2259 <xsd:complexType name="UnknownFaultType">
2260   <xsd:complexContent>
2261     <xsd:extension base="wsrf-bf:BaseFaultType"/>
2262   </xsd:complexContent>
2263 </xsd:complexType>

```

```

2264 <xsd:element name="UnknownFault"
2265           type="infod:UnknownFaultType"/>
2266
2267 </xsd:schema>

```

2268 5.10 INFOD Notification

```

2269 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2270           xmlns:ident="http://www.ogf.org/infod"
2271           targetNamespace="http://www.ogf.org/infod/INFODNotify">
2272
2273 <!-- ===== Notification Metadata ===== -->
2274
2275 <xsd:element name="SubscriptionReference"
2276           type="wsa:EndpointReferenceType"/>
2277 <xsd:element name="Topic"
2278           type="infod:TopicExpressionType"/>
2279 <xsd:element name="PublisherReference"
2280           type="wsa:EndpointReferenceType"/>
2281
2282 <!-- ===== Message Helper Types ===== -->
2283
2284 <xsd:complexType name="TopicExpressionType" mixed="true">
2285   <xsd:sequence>
2286     <xsd:any minOccurs="0" maxOccurs="1" processContents="lax"/>
2287   </xsd:sequence>
2288   <xsd:attribute name="Dialect" type="xsd:anyURI" use="required"/>
2289   <xsd:anyAttribute/>
2290 </xsd:complexType>
2291
2292 <xsd:complexType name="NotificationMessageHolderType">
2293   <xsd:sequence>
2294     <xsd:element ref="infod:SubscriptionReference"
2295               minOccurs="1" maxOccurs="1"/>
2296     <xsd:element ref="infod:Topic"
2297               minOccurs="0" maxOccurs="1"/>
2298     <xsd:element ref="infod:PublisherReference"
2299               minOccurs="0" maxOccurs="1"/>
2300     <xsd:element name="Message">
2301       <xsd:complexType>
2302         <xsd:sequence>
2303           <xsd:any namespace="##any" processContents="lax"
2304                 minOccurs="1" maxOccurs="1"/>
2305         </xsd:sequence>
2306       </xsd:complexType>
2307     </xsd:element>
2308   </xsd:sequence>
2309 </xsd:complexType>
2310 <xsd:element name="NotificationMessage"
2311           type="infod:NotificationMessageHolderType"/>
2312
2313 <!-- ===== Message Types for Consumer Notification by Publishers ===== -->
2314
2315 <xsd:element name="Notify">
2316   <xsd:annotation>
2317     <xsd:documentation> Notification of Consumers by Publishers
2318   </xsd:documentation>
2319   </xsd:annotation>
2320 <xsd:complexType>
2321   <xsd:sequence>

```



```

2322     <xsd:element ref="infod:NotificationMessage"
2323             minOccurs="1" maxOccurs="unbounded"/>
2324     <xsd:any namespace="##other" processContents="lax"
2325             minOccurs="0" maxOccurs="unbounded"/>
2326     </xsd:sequence>
2327 </xsd:complexType>
2328 </xsd:element>
2329
2330 </xsd:schema>

```

2331 5.11 INFOD Publisher Notification

```

2332 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2333             xmlns:ident="http://www.ogf.org/infod"
2334             targetNamespace="http://www.ogf.org/infod/INFODNotify">
2335
2336     <xsd:element name="PublisherNotification">
2337         <xsd:annotation>
2338             <xsd:documentation>
2339                 Notification of Publishers
2340             </xsd:documentation>
2341         </xsd:annotation>
2342         <xsd:complexType>
2343             <xsd:sequence>
2344                 <xsd:element name="SubscriptionReference"
2345                             type="wsa:EndpointReferenceType"
2346                             minOccurs="1" maxOccurs="1"/>
2347                 <xsd:choice minOccurs="1" maxOccurs="1">
2348                     <xsd:sequence>
2349                         <xsd:element name="ConsumerEntryReference"
2350                                     type="xsd:any"
2351                                     minOccurs="1" maxOccurs="unbounded"/>
2352                         <xsd:element name="DynamicConsumerConstraint"
2353                                     type="xsd:any"
2354                                     minOccurs="0" maxOccurs="unbounded"/>
2355                     </xsd:sequence>
2356                     <xsd:sequence>
2357                         <xsd:element name="ConsumerEntryReference"
2358                                     type="xsd:any"
2359                                     minOccurs="0" maxOccurs="unbounded"/>
2360                         <xsd:element name="DynamicConsumerConstraint"
2361                                     type="xsd:any"
2362                                     minOccurs="1" maxOccurs="unbounded"/>
2363                     </xsd:sequence>
2364                 </xsd:choice>
2365                 <xsd:element name="DataConstraint"
2366                             type="xsd:any"
2367                             minOccurs="1" maxOccurs="1"/>
2368             </xsd:sequence>
2369         </xsd:complexType>
2370     </xsd:element>
2371 </xsd:schema>

```

2372 5.12 INFOD Subscriber Notification

```

2373 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2374             xmlns:ident="http://www.ogf.org/infod"
2375             targetNamespace="http://www.ogf.org/infod/INFODNotify">
2376

```

```

2377 <xsd:element name="SubscriberNotification">
2378   <xsd:annotation>
2379     <xsd:documentation>
2380       Notification of Subscribers
2381     </xsd:documentation>
2382   </xsd:annotation>
2383   <xsd:complexType>
2384     <xsd:sequence>
2385       <xsd:element name="SubscriptionReference"
2386         type="wsa:EndpointReferenceType"
2387         minOccurs="1" maxOccurs="1"/>
2388       <xsd:element name="ConsumerEntryReference"
2389         type="wsa:EndpointReferenceType"
2390         minOccurs="0" maxOccurs="unbounded"/>
2391       <xsd:element name="PublisherEntryReference"
2392         type="wsa:EndpointReferenceType"
2393         minOccurs="0" maxOccurs="unbounded"/>
2394     </xsd:sequence>
2395   </xsd:complexType>
2396 </xsd:element>
2397 </xsd:schema>

```

2398 5.13 INFOD Consumer Notification

```

2399 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2400   xmlns:ident="http://www.ogf.org/infod"
2401   targetNamespace="http://www.ogf.org/infod/INFODNotify">
2402
2403   <xsd:element name="ConsumerNotification">
2404     <xsd:annotation>
2405       <xsd:documentation>
2406         Notification of Consumers
2407       </xsd:documentation>
2408     </xsd:annotation>
2409     <xsd:complexType>
2410       <xsd:sequence>
2411         <xsd:element name="SubscriptionReference"
2412           type="wsa:EndpointReferenceType"
2413           minOccurs="1" maxOccurs="1"/>
2414         <xsd:element name="PublisherEntryReference"
2415           type="wsa:EndpointReferenceType"
2416           minOccurs="0" maxOccurs="unbounded"/>
2417       </xsd:sequence>
2418     </xsd:complexType>
2419   </xsd:element>
2420 </xsd:schema>

```

2421

6 Appendix II – WSDL 1.1

```

2422 <wsdl:definitions name="infodBaseNotification"
2423 targetNamespace="http://www.ggf.org/INFOD"
2424 xmlns:tns="http://www.ggf.org/INFOD"
2425 xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
2426 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2427 xmlns:wsrw="http://docs.oasis-open.org/wsrw/rw-2"
2428 xmlns:wsa="http://www.w3.org/2005/08/addressing"
2429 xmlns:infod="http://www.ggf.org/INFOD"
2430 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
2431
2432 <wsdl:import namespace="http://docs.oasis-open.org/wsrw/rw-2"
2433 location="http://docs.oasis-open.org/wsrw/rw-2.wsdl" />
2434
2435
2436 <wsdl:types>
2437 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2438 elementFormDefault="qualified"
2439 targetNamespace="http://www.ggf.org/INFOD"
2440 xmlns:infodxsd="http://www.ggf.org/INFOD/infodTypes.xsd">
2441 <xsd:complexType name="Notification">
2442 <xsd:simpleContent>
2443 <xsd:extension base="xsd:boolean"></xsd:extension>
2444 </xsd:simpleContent>
2445 </xsd:complexType>
2446 <xsd:element name="CreatePublisherEntryMissingRequiredParameterFault"
2447 type="xsd:string"></xsd:element>
2448 <xsd:element name="DropPublisherEntryExecutionModeFault"
2449 type="xsd:string"></xsd:element>
2450 <xsd:element name="DropSubscriberEntryDropResourceAuthorizationFailure"
2451 type="xsd:string"></xsd:element>
2452 <xsd:element name="CreateConsumerEntryMissingRequiredParameterFault"
2453 type="xsd:string"></xsd:element>
2454 <xsd:element name="DropConsumerEntryMissingRequiredParameterFault"
2455 type="xsd:string"></xsd:element>
2456 <xsd:element name="CreateSubscriptionMissingRequiredParameterFault"
2457 type="xsd:string"></xsd:element>
2458 <xsd:element name="DropSubscriptionMissigRequiredParameterFault"
2459 type="xsd:string"></xsd:element>
2460 </xsd:schema>
2461 <xsd:schema>
2462
2463 <xsd:import namespace="http://docs.oasis-open.org/wsn/b-2"
2464 schemaLocation="http://docs.oasis-open.org/wsn/b-2.xsd" />
2465 <xsd:import namespace="http://www.ggf.org/INFOD"
2466 schemaLocation="infodTypes.xsd">
2467 </xsd:import>
2468 </xsd:schema>
2469
2470 </wsdl:types>
2471

```

```
2472
2473 <wsdl:message name="CreatePublisherEntryRequest">
2474 <wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
2475 <wsdl:part name="PublisherName" type="xsd:string"></wsdl:part>
2476 <wsdl:part name="PublisherDescription" type="xsd:string"></wsdl:part>
2477 <wsdl:part name="PropertyConstraints"
2478 element="infod:PropertyConstraints">
2479 </wsdl:part>
2480 <wsdl:part name="Notification" type="xsd:boolean"></wsdl:part>
2481 </wsdl:message>
2482
2483 <wsdl:message name="CreatePublisherEntryResponse">
2484 <wsdl:part name="INFODResourceReference"
2485 element="infod:EndPointReference">
2486 </wsdl:part>
2487 </wsdl:message>
2488
2489
2490 <wsdl:message name="ReplacePublisherEntryRequest">
2491 <wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
2492 <wsdl:part name="PublisherEntryReference"
2493 element="infod:EndPointReference">
2494 </wsdl:part>
2495 <wsdl:part name="PublisherName" type="xsd:string"></wsdl:part>
2496 <wsdl:part name="PublisherDescription" type="xsd:string"></wsdl:part>
2497 <wsdl:part name="PropertyConstraints"
2498 element="wsinfod:PropertyConstraints">
2499 </wsdl:part>
2500 <wsdl:part name="Notification" type="xsd:boolean"></wsdl:part>
2501 </wsdl:message>
2502
2503 <wsdl:message name="ReplacePublisherEntryResponse">
2504 <wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
2505 </wsdl:message>
2506
2507 <wsdl:message name="DropPublisherEntryRequest">
2508 <wsdl:part name="PublisherEntryReference"
2509 element="infod:EndPointReference">
2510 </wsdl:part>
2511 <wsdl:part name="ExecutionMode"
2512 element="infod:ExecutionMode"></wsdl:part>
2513 </wsdl:message>
2514
2515 <wsdl:message name="DropPublisherEntryResponse">
2516 <wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
2517 </wsdl:message>
2518
2519 <wsdl:message name="CreateSubscriberEntryRequest">
2520 <wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
2521 <wsdl:part name="SubscriberName" type="xsd:string"></wsdl:part>
2522 <wsdl:part name="SubscriberDescription" type="xsd:string"></wsdl:part>
2523 <wsdl:part name="PropertyConstraints"
2524 element="infod:PropertyConstraints">
2525 </wsdl:part>
2526 <wsdl:part name="Notification" type="xsd:boolean"></wsdl:part>
```

```
2527 </wsdl:message>
2528
2529 <wsdl:message name="CreateSubscriberEntryResponse">
2530 <wsdl:part name="INFODResourceReference"
2531 element="infod:EndPointReference">
2532 </wsdl:part>
2533 </wsdl:message>
2534
2535 <wsdl:message name="ReplaceSubscriberEntryRequest">
2536 <wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
2537 <wsdl:part name="SubscriberEntryReference"
2538 element="infod:EndPointReference">
2539 </wsdl:part>
2540 <wsdl:part name="SubscriberName" type="xsd:string"></wsdl:part>
2541 <wsdl:part name="SubscriberDescription" type="xsd:string"></wsdl:part>
2542 <wsdl:part name="PropertyConstraints"
2543 element="infod:PropertyConstraints">
2544 </wsdl:part>
2545 <wsdl:part name="Notification" type="xsd:boolean"></wsdl:part>
2546 </wsdl:message>
2547
2548 <wsdl:message name="ReplaceSubscriberEntryResponse">
2549 <wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
2550 </wsdl:message>
2551
2552 <wsdl:message name="DropSubscriberEntryRequest">
2553 <wsdl:part name="INFODResourceReference"
2554 element="infod:EndPointReference">
2555 </wsdl:part>
2556 <wsdl:part name="ExecutionMode"
2557 element="infod:ExecutionMode"></wsdl:part>
2558 </wsdl:message>
2559
2560 <wsdl:message name="DropSubscriberEntryResponse">
2561 <wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
2562 </wsdl:message>
2563
2564
2565 <wsdl:message name="CreateConsumerEntryRequest">
2566 <wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
2567 <wsdl:part name="INFODResourceReference"
2568 element="infod:EndPointReference">
2569 </wsdl:part>
2570 <wsdl:part name="ConsumerName" type="xsd:string"></wsdl:part>
2571 <wsdl:part name="ConsumerDescription" type="xsd:string"></wsdl:part>
2572 <wsdl:part name="PropertyConstraints"
2573 element="infod:PropertyConstraints">
2574 </wsdl:part>
2575 <wsdl:part name="Notification" type="xsd:boolean"></wsdl:part>
2576 </wsdl:message>
2577
2578 <wsdl:message name="CreateConsumerEntryResponse">
2579 <wsdl:part name="INFODResourceReference"
2580 element="infod:EndPointReference">
2581 </wsdl:part>
```

```
2582 </wsdl:message>
2583
2584 <wsdl:message name="ReplaceConsumerEntryRequest">
2585 <wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
2586 <wsdl:part name="INFODResourceReference"
2587 element="infod:EndPointReference">
2588 </wsdl:part>
2589 <wsdl:part name="ConsumerName" type="xsd:string"></wsdl:part>
2590 <wsdl:part name="ConsumerDescription" type="xsd:string"></wsdl:part>
2591 <wsdl:part name="PropertyConstraints"
2592 element="infod:PropertyConstraints">
2593 </wsdl:part>
2594 <wsdl:part name="Notification" type="xsd:boolean"></wsdl:part>
2595 </wsdl:message>
2596
2597 <wsdl:message name="ReplaceConsumerEntryResponse">
2598 <wsdl:part name="Status" element="infod:status"></wsdl:part>
2599 </wsdl:message>
2600
2601 <wsdl:message name="DropConsumerEntryRequest">
2602 <wsdl:part name="INFODResourceReference"
2603 element="infod:EndPointReference">
2604 </wsdl:part>
2605 <wsdl:part name="ExecutionMode"
2606 element="infod:ExecutionMode"></wsdl:part>
2607 </wsdl:message>
2608
2609 <wsdl:message name="DropConsumerEntryResponse">
2610 <wsdl:part name="Status" element="infod:status"></wsdl:part>
2611 </wsdl:message>
2612
2613 <wsdl:message name="CreateSubscriptionRequest">
2614 <wsdl:part name="SubscriptionName" type="xsd:string"></wsdl:part>
2615 <wsdl:part name="SubscriptionDescription" type="xsd:string"></wsdl:part>
2616 <wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
2617 <wsdl:part name="DataConstraints"
2618 element="infod:DataConstraints">
2619 </wsdl:part>
2620 <wsdl:part name="PropertyConstraints"
2621 element="infod:PropertyConstraints">
2622 </wsdl:part>
2623 <wsdl:part name="DynamicConsumerConstraints"
2624 element="infod:DynamicConsumerConstraints">
2625 </wsdl:part>
2626 </wsdl:message>
2627
2628 <wsdl:message name="CreateSubscriptionResponse">
2629 <wsdl:part name="INFODResourceReference"
2630 element="wsa:EndPointReference">
2631 </wsdl:part>
2632 </wsdl:message>
2633
2634 <wsdl:message name="ReplaceSubscriptionRequest">
2635 <wsdl:part name="INFODResourceReference"
2636 element="wsa:EndPointReference">
```

```
2637 </wsdl:part>
2638 <wsdl:part name="SubscriptionName" type="xsd:string"></wsdl:part>
2639 <wsdl:part name="SubscriptionDescription" type="xsd:string"></wsdl:part>
2640 <wsdl:part name="WSReference" element="wsa:EndPointReference"></wsdl:part>
2641 <wsdl:part name="DataConstraints"
2642 element="infod:DataConstraints">
2643 </wsdl:part>
2644 <wsdl:part name="PropertyConstraints"
2645 element="infod:PropertyConstraints">
2646 </wsdl:part>
2647 <wsdl:part name="DynamicConsumerConstraints"
2648 element="infod:DynamicConsumerConstraints">
2649 </wsdl:part>
2650 </wsdl:message>
2651
2652 <wsdl:message name="ReplaceSubscriptionResponse">
2653 <wsdl:part name="Status" element="infod:status"></wsdl:part>
2654 </wsdl:message>
2655
2656 <wsdl:message name="DropSubscriptionRequest">
2657 <wsdl:part name="INFODResourceReference"
2658 element="wsa:EndPointReference">
2659 </wsdl:part>
2660 <wsdl:part name="ExecutionMode"
2661 element="infod:ExecutionMode"></wsdl:part>
2662 </wsdl:message>
2663
2664 <wsdl:message name="DropSubscriptionResponse">
2665 <wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
2666 </wsdl:message>
2667
2668 <wsdl:message name="CreatePropertyVocabularyRequest">
2669 <wsdl:part name="VocabularyName" type="xsd:string"></wsdl:part>
2670 <wsdl:part name="VocabularyDescription" type="xsd:string"></wsdl:part>
2671 <wsdl:part name="VocabularyBody" type="xsd:anyType"></wsdl:part>
2672 </wsdl:message>
2673
2674 <wsdl:message name="CreatePropertyVocabularyResponse">
2675 <wsdl:part name="INFODVocabularyReference"
2676 element="infod:EndPointReference">
2677 </wsdl:part>
2678 </wsdl:message>
2679
2680 <wsdl:message name="CreatePropertyVocabularyInstanceRequest">
2681 <wsdl:part name="VocabularyInstanceReference"
2682 element="wsa:EndPointReference">
2683 </wsdl:part>
2684 <wsdl:part name="VocabularyInstanceVocabularyReference"
2685 element="wsa:EndPointReference">
2686 </wsdl:part>
2687 <wsdl:part name="VocabularyInstanceVocabularyBody"
2688 type="xsd:anyType">
2689 </wsdl:part>
2690 </wsdl:message>
2691
```

```

2692 <wsdl:message name="CreatePropertyVocabularyInstanceResponse">
2693 <wsdl:part name="INFODVocabularyInstanceReference"
2694 element="wsa:EndPointReference">
2695 </wsdl:part>
2696 </wsdl:message>
2697
2698 <wsdl:message name="DropPropertyVocabularyInstanceRequest">
2699 <wsdl:part name="VocabularyInstanceReference"
2700 element="wsa:EndPointReference">
2701 </wsdl:part>
2702 <wsdl:part name="ExecutionMode"
2703 element="infod:ExecutionMode"></wsdl:part>
2704 </wsdl:message>
2705
2706 <wsdl:message name="DropPropertyVocabularyInstanceResponse">
2707 <wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
2708 </wsdl:message>
2709
2710
2711 <wsdl:message name="CreateDataVocabularyRequest">
2712 <wsdl:part name="VocabularyName" type="xsd:string"></wsdl:part>
2713 <wsdl:part name="VocabularyDescription" type="xsd:string"></wsdl:part>
2714 <wsdl:part name="VocabularyLanguage" type="xsd:anyURI"></wsdl:part>
2715 <wsdl:part name="LanguageUsageDescription" type="xsd:anyType"></wsdl:part>
2716 <wsdl:part name="VocabularyInstanceVocabularyBody"
2717 type="xsd:anyType">
2718 </wsdl:part>
2719 </wsdl:message>
2720
2721
2722 <wsdl:message name="CreateDataVocabularyResponse">
2723 <wsdl:part name="DataVocabularyReference"
2724 element="wsa:EndPointReference">
2725 </wsdl:part>
2726 </wsdl:message>
2727
2728 <wsdl:message name="DropDataVocabularyRequest">
2729 <wsdl:part name="INFODVocabularyReference"
2730 element="wsa:EndPointReference">
2731 </wsdl:part>
2732 <wsdl:part name="ExecutionMode"
2733 element="infod:ExecutionMode"></wsdl:part>
2734 </wsdl:message>
2735
2736 <wsdl:message name="DropDataVocabularyResponse">
2737 <wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
2738 </wsdl:message>
2739
2740 <wsdl:message name="CreateDataSourceEntryRequest">
2741 <wsdl:part name="CreateDataSourceEntryName"
2742 type="xsd:string"></wsdl:part>
2743 <wsdl:part name="CreateDataSourceEntryDescription"
2744 type="xsd:string">
2745 </wsdl:part>
2746 <wsdl:part name="AssociationEntryReference"

```



```

2747 element="wsa:EndPointReference">
2748 </wsdl:part>
2749 <wsdl:part name="VocabularyReference"
2750 element="wsa:EndPointReference">
2751 </wsdl:part>
2752 <wsdl:part name="PropertyConstraints"
2753 element="infod:PropertyConstraints">
2754 </wsdl:part>
2755 </wsdl:message>
2756
2757 <wsdl:message name="CreateDataSourceEntryResponse">
2758 <wsdl:part name="INFODAssociationReference"
2759 element="wsa:EndPointReference">
2760 </wsdl:part>
2761 </wsdl:message>
2762
2763 <wsdl:message name="DropDataSourceEntryRequest">
2764 <wsdl:part name="DataSourceEntryReference"
2765 element="wsa:EndPointReference">
2766 </wsdl:part>
2767 <wsdl:part name="ExecutionMode"
2768     element="infod:ExecutionMode"></wsdl:part>
2769 </wsdl:message>
2770
2771 <wsdl:message name="DropDataSourceEntryResponse">
2772 <wsdl:part name="Status" element="infodxsd:status"></wsdl:part>
2773 </wsdl:message>
2774
2775 <wsdl:message name="GetMetaDataRequest">
2776 <wsdl:part name="MetaDataQueryExpression" type="xsd:anyType"></wsdl:part>
2777 </wsdl:message>
2778
2779 <wsdl:message name="GetMetaDataResponse">
2780 <wsdl:part name="MetaDataQueryResult" type="xsd:anyType"></wsdl:part>
2781 </wsdl:message>
2782
2783 <wsdl:message name="NotifyRequest"></wsdl:message>
2784
2785
2786 <wsdl:message name="CreateResourceAuthorizationFailureErrorMessage">
2787 <wsdl:part name="err"
2788 element="infod:CreateEntryAuthorizationFailure">
2789 </wsdl:part>
2790 </wsdl:message>
2791 <wsdl:message name="UnknownResourceReferenceFaultErrorMessage">
2792 <wsdl:part name="err"
2793 element="infod:UnknownResourceReferenceFault">
2794 </wsdl:part>
2795 </wsdl:message>
2796 <wsdl:message name="MissingRequiredParameterFaultErrorMessage">
2797 <wsdl:part name="err"
2798 element="infod:MissingRequiredParameterFault">
2799 </wsdl:part>
2800 </wsdl:message>
2801 <wsdl:message name="UnsupportedXQueryFaultErrorMesage">

```

```

2802 <wsdl:part name="err" element="infod:UnsupportedXQueryFault"></wsdl:part>
2803 </wsdl:message>
2804 <wsdl:message name="ReplaceResourceAuthorizationFaultErrorMessage">
2805 <wsdl:part name="err"
2806 element="infod:ReplaceEntryAuthorizationFailure">
2807 </wsdl:part>
2808 </wsdl:message>
2809 <wsdl:message name="DropResourceAuthorizationFaultErrorMessage">
2810 <wsdl:part name="err"
2811 element="infod:DropEntryAuthorizationFailure">
2812 </wsdl:part>
2813 </wsdl:message>
2814 <wsdl:message name="ExecutionModeFaultErrorMessage">
2815 <wsdl:part name="err" element="infod:ExecutionModeFault"></wsdl:part>
2816 </wsdl:message>
2817
2818 <wsdl:message name="UnSupportedVocabularyFaultErrorMessage">
2819 <wsdl:part name="err"
2820 element="infod:UnSupportedVocabularyFault">
2821 </wsdl:part>
2822 </wsdl:message>
2823
2824 <wsdl:message
2825 name="CreateAssociationAuthorizationFailureErrorMessage">
2826 <wsdl:part name="err"
2827 element="infod:CreateAssociationAuthorizationFailure">
2828 </wsdl:part>
2829 </wsdl:message>
2830 <wsdl:message name="DropPropertyVocabularyResponse">
2831 <wsdl:part name="DropPropertyVocabularyResponse"
2832 type="xsd:string">
2833 </wsdl:part>
2834 </wsdl:message>
2835 <wsdl:message name="DropPropertyVocabularyRequest">
2836 <wsdl:part name="PropertyVocabularyReference"
2837 element="wsa:EndPointRefernceType"></wsdl:part>
2838 <wsdl:part name="ExecutionMode" element="infod:ExecutionMode">
2839 </wsdl:part>
2840 <wsdl:part name="ExecutionMode" element="infod:ExecutionMode">
2841 </wsdl:part>
2842 </wsdl:message>
2843 <wsdl:portType name="infodRegistry">
2844 <wsdl:operation name="CreatePublisherEntry">
2845 <wsdl:input message="tns:CreatePublisherEntryRequest"
2846 name="createPublisherMessage">
2847 </wsdl:input>
2848 <wsdl:output message="tns:CreatePublisherEntryResponse"></wsdl:output>
2849 <wsdl:fault
2850 message="tns:CreateResourceAuthorizationFailureErrorMessage"
2851 name="CreateResourceAuthorizationFault">
2852 </wsdl:fault>
2853 <wsdl:fault
2854 message="infod:MissingRequiredParameterFaultErrorMessage"
2855 name="UnsupportedXQueryFault">
2856 </wsdl:fault>

```

```
2857 <wsdl:fault
2858 message="infod:UnsupportedXQueryFaultErrorMessage"
2859 name="UnsupportedXQueryFault">
2860 </wsdl:fault>
2861 <wsdl:fault name="MissingRequiredParameterFault"
2862 message="tns:MissingRequiredParameterFaultErrorMessage">
2863 </wsdl:fault>
2864 </wsdl:operation>
2865
2866 <wsdl:operation name="ReplacePublisherEntry">
2867 <wsdl:input message="tns:ReplacePublisherEntryRequest"
2868 name="ReplacePublisherMessage">
2869 </wsdl:input>
2870 <wsdl:output message="tns:ReplacePublisherEntryResponse"
2871 name="ReplacePublisherResponse">
2872 </wsdl:output>
2873 <wsdl:fault
2874 message="infod:ReplaceResourceAuthorizationFaultErrorMessage"
2875 name="ReplaceResourceAuthorizationFault">
2876 </wsdl:fault>
2877 <wsdl:fault
2878 message="tns:UnknownResourceReferenceFaultErrorMessage"
2879 name="UnknownResourceReferenceFault">
2880 </wsdl:fault>
2881 <wsdl:fault
2882 message="infod:MissingRequiredParameterFaultErrorMessage"
2883 name="MissingrequiredParameterFault">
2884 </wsdl:fault>
2885 <wsdl:fault
2886 message="infod:UnsupportedXQueryFaultErrorMessage"
2887 name="UnsupportedXQueryFault">
2888 </wsdl:fault>
2889 </wsdl:operation>
2890
2891 <wsdl:operation name="DropPublisherEntry">
2892 <wsdl:input message="tns:DropPublisherEntryRequest"
2893 name="DropPublisherRequestMessage">
2894 </wsdl:input>
2895 <wsdl:output message="tns:DropPublisherEntryResponse"
2896 name="DropPublisherResponseMessage">
2897 </wsdl:output>
2898 <wsdl:fault
2899 message="infod:DropResourceAuthorizationFaultErrorMessage"
2900 name="DropResourceAuthorizationFailure">
2901 </wsdl:fault>
2902 <wsdl:fault
2903 message="tns:UnknownResourceReferenceFaultErrorMessage"
2904 name="UnknownResourceReferenceFault">
2905 </wsdl:fault>
2906 <wsdl:fault message="infod:ExecutionModeFaultErrorMessage"
2907 name="ExecutionModeFault">
2908 </wsdl:fault>
2909 <wsdl:fault name="ExecutionModeFault"
2910 message="tns:ExecutionModeFaultErrorMessage">
2911 </wsdl:fault>
```

```
2912 </wsdl:operation>
2913
2914 <wsdl:operation name="CreateSubscriberEntry">
2915 <wsdl:input message="tns:CreateSubscriberEntryRequest"></wsdl:input>
2916 <wsdl:output message="tns:CreateSubscriberEntryResponse"></wsdl:output>
2917 <wsdl:fault
2918 message="tns:CreateResourceAuthorizationFailureErrorMessage"
2919 name="CreateResourceAuthorizationFailure">
2920 </wsdl:fault>
2921 <wsdl:fault
2922 message="tns:UnknownResourceReferenceFaultErrorMessage"
2923 name="UnknownResourceReferenceFault">
2924 </wsdl:fault>
2925 <wsdl:fault
2926 message="infod:MissingRequiredParameterFaultErrorMessage"
2927 name="UnsupportedXQueryFault">
2928 </wsdl:fault>
2929 <wsdl:fault
2930 message="infod:UnsupportedXQueryFaultErrorMessage"
2931 name="UnsupportedXQueryFault">
2932 </wsdl:fault>
2933 </wsdl:operation>
2934
2935 <wsdl:operation name="ReplaceSubscriberEntry">
2936 <wsdl:input message="tns:ReplaceSubscriberEntryRequest"></wsdl:input>
2937 <wsdl:output message="tns:ReplaceSubscriberEntryResponse"></wsdl:output>
2938 <wsdl:fault
2939 message="infod:ReplaceResourceAuthorizationFailureErrorMessage"
2940 name="ReplaceResourceAuthorizationFailure">
2941 </wsdl:fault>
2942 <wsdl:fault
2943 message="tns:UnknownResourceReferenceFaultErrorMessage"
2944 name="UnknownResourceReferenceFault">
2945 </wsdl:fault>
2946 <wsdl:fault
2947 message="infod:MissingRequiredParameterFaultErrorMessage"
2948 name="MissingrequiredParameterFault">
2949 </wsdl:fault>
2950 <wsdl:fault
2951 message="infod:UnsupportedXQueryFaultErrorMessage"
2952 name="UnsupportedXQueryFault">
2953 </wsdl:fault>
2954 </wsdl:operation>
2955
2956 <wsdl:operation name="DropSubscriberEntry">
2957 <wsdl:input message="tns:DropSubscriberEntryRequest"></wsdl:input>
2958 <wsdl:output message="tns:DropSubscriberEntryResponse"></wsdl:output>
2959 <wsdl:fault
2960 message="infod:DropResourceAuthorizationFailureErrorMessage"
2961 name="DropEntryAuthorizationFailure">
2962 </wsdl:fault>
2963 <wsdl:fault
2964 message="tns:UnknownResourceReferenceFaultErrorMessage"
2965 name="UnknownResourceReferenceFault">
2966 </wsdl:fault>
```

```
2967 <wsdl:fault message="infod:ExecutionModeFaultErrorMessage"
2968 name="ExecutionModeFault">
2969 </wsdl:fault>
2970 </wsdl:operation>
2971
2972 <wsdl:operation name="CreateConsumerEntry">
2973 <wsdl:input message="tns:CreateConsumerEntryRequest"></wsdl:input>
2974 <wsdl:output message="tns:CreateConsumerEntryResponse"></wsdl:output>
2975 <wsdl:fault
2976 message="tns:CreateResourceAuthorizationFailureErrorMessage"
2977 name="CreateResourceAuthorizationFailure">
2978 </wsdl:fault>
2979 <wsdl:fault
2980 message="infod:MissingRequiredParameterFaultErrorMessage"
2981 name="UnsupportedXQueryFault">
2982 </wsdl:fault>
2983 <wsdl:fault
2984 message="infod:UnsupportedXQueryFaultErrorMessage"
2985 name="UnsupportedXQueryFault">
2986 </wsdl:fault>
2987 <wsdl:fault name="MissingRequiredParameterFault"
2988 message="tns:MissingRequiredParameterFaultErrorMessage">
2989 </wsdl:fault>
2990 </wsdl:operation>
2991
2992 <wsdl:operation name="ReplaceConsumerEntry">
2993 <wsdl:input message="tns:ReplaceConsumerEntryRequest"></wsdl:input>
2994 <wsdl:output message="tns:ReplaceConsumerEntryResponse"></wsdl:output>
2995 <wsdl:fault
2996 message="infod:ReplaceResourceAuthorizationFaultErrorMessage"
2997 name="ReplaceResourceAuthorizationFault">
2998 </wsdl:fault>
2999 <wsdl:fault
3000 message="tns:UnknownResourceReferenceFaultErrorMessage"
3001 name="UnknownResourceReferenceFault">
3002 </wsdl:fault>
3003 <wsdl:fault
3004 message="infod:MissingRequiredParameterFaultErrorMessage"
3005 name="MissingrequiredParameterFault">
3006 </wsdl:fault>
3007 <wsdl:fault
3008 message="infod:UnsupportedXQueryFaultErrorMessage"
3009 name="UnsupportedXQueryFault">
3010 </wsdl:fault>
3011 </wsdl:operation>
3012
3013 <wsdl:operation name="DropConsumerEntry">
3014 <wsdl:input message="tns:DropConsumerEntryRequest"></wsdl:input>
3015 <wsdl:output message="tns:DropConsumerEntryResponse"></wsdl:output>
3016 <wsdl:fault
3017 message="infod:DropResourceAuthorizationFaultErrorMessage"
3018 name="DropResourceAuthorizationFailure">
3019 </wsdl:fault>
3020 <wsdl:fault
3021 message="infod:UnknownResourceReferenceFaultErrorMessage"
```

```
3022 name="UnknownResourceReferenceFault">
3023 </wsdl:fault>
3024 <wsdl:fault message="infod:ExecutionModeFaultErrorMessage"
3025 name="ExecutionModeFault">
3026 </wsdl:fault>
3027 <wsdl:fault name="MissingRequiredParameterFault"
3028 message="tns:MissingRequiredParameterFaultErrorMessage">
3029 </wsdl:fault>
3030 </wsdl:operation>
3031
3032 <wsdl:operation name="CreateSubscription">
3033 <wsdl:input message="infod:CreateSubscriptionRequest"></wsdl:input>
3034 <wsdl:output message="infod:CreateSubscriptionResponse"></wsdl:output>
3035 <wsdl:fault
3036 message="tns:CreateResourceAuthorizationFailureErrorMessage"
3037 name="CreateResourceAuthorizationFailure">
3038 </wsdl:fault>
3039 <wsdl:fault
3040 message="infod:UnknownResourceReferenceFaultErrorMessage"
3041 name="UnknownResourceReferenceFault">
3042 </wsdl:fault>
3043 <wsdl:fault
3044 message="infod:MissingRequiredParameterFaultErrorMessage"
3045 name="UnsupportedXQueryFault">
3046 </wsdl:fault>
3047 <wsdl:fault
3048 message="infod:UnsupportedXQueryFaultErrorMessage"
3049 name="UnsupportedXQueryFault">
3050 </wsdl:fault>
3051 <wsdl:fault name="MissingRequiredParameterFault"
3052 message="tns:MissingRequiredParameterFaultErrorMessage">
3053 </wsdl:fault>
3054 </wsdl:operation>
3055
3056 <wsdl:operation name="ReplaceSubscription">
3057 <wsdl:input message="infod:ReplaceSubscriptionRequest"></wsdl:input>
3058 <wsdl:output message="infod:ReplaceSubscriptionResponse"></wsdl:output>
3059 <wsdl:fault
3060 message="infod:ReplaceResourceAuthorizationFaultErrorMessage"
3061 name="ReplaceResourceAuthorizationFault">
3062 </wsdl:fault>
3063 <wsdl:fault
3064 message="infod:UnknownResourceReferenceFaultErrorMessage"
3065 name="UnknownResourceReferenceFault">
3066 </wsdl:fault>
3067 <wsdl:fault
3068 message="infod:MissingRequiredParameterFaultErrorMessage"
3069 name="MissingrequiredParameterFault">
3070 </wsdl:fault>
3071 <wsdl:fault
3072 message="infod:UnsupportedXQueryFaultErrorMessage"
3073 name="UnsupportedXQueryFault">
3074 </wsdl:fault>
3075 </wsdl:operation>
3076
```

```
3077 <wsdl:operation name="DropSubscription">
3078 <wsdl:input message="infod:DropSubscriptionRequest"></wsdl:input>
3079 <wsdl:output message="infod:DropSubscriptionResponse"></wsdl:output>
3080 <wsdl:fault
3081 message="infod:DropResourceAuthorizationFaultErrorMessage"
3082 name="DropResourceAuthorizationFailure">
3083 </wsdl:fault>
3084 <wsdl:fault
3085 message="infod:UnknownResourceReferenceFaultErrorMessage"
3086 name="UnknownResourceReferenceFault">
3087 </wsdl:fault>
3088 <wsdl:fault message="infod:ExecutionModeFaultErrorMessage"
3089 name="ExecutionModeFault">
3090 </wsdl:fault>
3091 <wsdl:fault name="MissingRequiredParameterFault"
3092 message="tns:MissingRequiredParameterFaultErrorMessage">
3093 </wsdl:fault>
3094 </wsdl:operation>
3095
3096 </wsdl:portType>
3097 <wsdl:portType name="infodVocabulary">
3098 <wsdl:operation name="CreatePropertyVocabulary">
3099 <wsdl:input
3100 message="tns:CreatePropertyVocabularyRequest">
3101 </wsdl:input>
3102 <wsdl:output
3103 message="tns:CreatePropertyVocabularyResponse">
3104 </wsdl:output>
3105 <wsdl:fault
3106 message="tns:CreateResourceAuthorizationFailureErrorMessage"
3107 name="CreateResourceAuthorizationFailure">
3108 </wsdl:fault>
3109 <wsdl:fault
3110 message="infod:MissingRequiredParameterFaultErrorMessage"
3111 name="MissingRequiredParameterFault">
3112 </wsdl:fault>
3113 <wsdl:fault
3114 message="infod:UnSupportedVocabularyFaultErrorMessage"
3115 name="UnSupportedVocabularyFault">
3116 </wsdl:fault>
3117 </wsdl:operation>
3118 <wsdl:operation name="CreatePropertyVocabularyInstance">
3119 <wsdl:input
3120 message="infod:CreatePropertyVocabularyInstanceRequest">
3121 </wsdl:input>
3122 <wsdl:output
3123 message="infod:CreatePropertyVocabularyInstanceResponse">
3124 </wsdl:output>
3125 <wsdl:fault
3126 message="tns:CreateResourceAuthorizationFailureErrorMessage"
3127 name="CreateResourceAuthorizationFailure">
3128 </wsdl:fault>
3129 <wsdl:fault
3130 message="infod:UnknownResourceReferenceFaultErrorMessage"
3131 name="UnknownResourceReferenceFault">
```

```
3132 </wsdl:fault>
3133 <wsdl:fault
3134 message="infod:MissingRequiredParameterFaultErrorMessage"
3135 name="MissingRequiredParameterFault">
3136 </wsdl:fault>
3137 <wsdl:fault
3138 message="infod:UnSupportedVocabularyFaultErrorMessage"
3139 name="UnSupportedVocabularyFault">
3140 </wsdl:fault>
3141 </wsdl:operation>
3142 <wsdl:operation name="DropPropertyVocabularyInstance">
3143 <wsdl:input
3144 message="infod:DropPropertyVocabularyInstanceRequest"
3145 name="DropPropertyVocabularyInstanceRequest">
3146 </wsdl:input>
3147 <wsdl:output
3148 message="infod:DropPropertyVocabularyInstanceResponse"
3149 name="DropPropertyVocabularyInstanceResponse">
3150 </wsdl:output>
3151 <wsdl:fault
3152 message="infod:DropVocabularyAuthorizationFailureErrorMessage"
3153 name="DropResourceAuthorizationFailure">
3154 </wsdl:fault>
3155 <wsdl:fault
3156 message="infod:UnknownResourceReferenceFaultErrorMessage"
3157 name="UnknownResourceReferenceFault">
3158 </wsdl:fault>
3159 <wsdl:fault
3160 message="infod:MissingRequiredParameterFaultErrorMessage"
3161 name="MissingRequiredParameterFault">
3162 </wsdl:fault>
3163 <wsdl:fault name="ExecutionModeFault"
3164 message="tns:ExecutionModeFaultErrorMessage">
3165 </wsdl:fault>
3166 </wsdl:operation>
3167 <wsdl:operation name="CreateDataVocabulary">
3168 <wsdl:input message="tns:CreateDataVocabularyRequest"></wsdl:input>
3169 <wsdl:output
3170 message="tns:CreateDataVocabularyResponse">
3171 </wsdl:output>
3172 <wsdl:fault
3173 message="tns:CreateResourceAuthorizationFailureErrorMessage"
3174 name="CreateResourceAuthorizationFailure">
3175 </wsdl:fault>
3176 <wsdl:fault
3177 message="infod:UnknownResourceReferenceFaultErrorMessage"
3178 name="UnknownResourceReferenceFault">
3179 </wsdl:fault>
3180 <wsdl:fault
3181 message="infod:MissingRequiredParameterFaultErrorMessage"
3182 name="MissingRequiredParameterFault">
3183 </wsdl:fault>
3184 <wsdl:fault
3185 message="infod:UnSupportedVocabularyFaultErrorMessage"
3186 name="UnSupportedVocabularyFault">
```



```

3187 </wsdl:fault>
3188 </wsdl:operation>
3189 <wsdl:operation name="DropDataVocabulary">
3190 <wsdl:input message="infod:DropDataVocabularyRequest"></wsdl:input>
3191 <wsdl:output message="infod:DropDataVocabularyResponse"></wsdl:output>
3192 <wsdl:fault
3193 message="tns:DropResourceAuthorizationFaultErrorMessage"
3194 name="DropResourceAuthorizationFailure">
3195 </wsdl:fault>
3196 <wsdl:fault
3197 message="infod:UnknownResourceReferenceFaultErrorMessage"
3198 name="UnknownResourceReferenceFault">
3199 </wsdl:fault>
3200 <wsdl:fault
3201 message="infod:MissingRequiredParameterFaultErrorMessage"
3202 name="MissingRequiredParameterFault">
3203 </wsdl:fault>
3204 <wsdl:fault
3205 message="infod:UnsupportedVocabularyFaultErrorMessage"
3206 name="UnsupportedVocabularyFault">
3207 </wsdl:fault>
3208 </wsdl:operation>
3209 <wsdl:operation name="CreateDataSourceEntry">
3210 <wsdl:input message="infod:CreateDataSourceEntryRequest"></wsdl:input>
3211 <wsdl:output
3212 message="infod:CreateDataSourceEntryResponse">
3213 </wsdl:output>
3214 <wsdl:fault
3215 message="infod:CreateAssociationAuthorizationFailureErrorMessage"
3216 name="CreateAssociationAuthorizationFailure">
3217 </wsdl:fault>
3218 <wsdl:fault
3219 message="infod:UnknownResourceReferenceFaultErrorMessage"
3220 name="UnknownResourceReferenceFault">
3221 </wsdl:fault>
3222 <wsdl:fault
3223 message="infod:MissingRequiredParameterFaultErrorMessage"
3224 name="MissingRequiredParameterFault">
3225 </wsdl:fault>
3226 <wsdl:fault
3227 message="infod:UnsupportedVocabularyFaultErrorMessage"
3228 name="UnsupportedVocabularyFault">
3229 </wsdl:fault>
3230 </wsdl:operation>
3231 <wsdl:operation name="DropDataSourceEntry">
3232 <wsdl:input
3233 message="tns:DropDataSourceEntryRequest">
3234 </wsdl:input>
3235 <wsdl:output
3236 message="tns:DropDataSourceEntryResponse">
3237 </wsdl:output>
3238 <wsdl:fault
3239 message="tns:DropResourceAuthorizationFaultErrorMessage"
3240 name="DropResourceAuthorizationFailure">
3241 </wsdl:fault>

```

```

3242 <wsdl:fault
3243 message="infod:UnknownResourceReferenceFaultErrorMessage"
3244 name="UnknownResourceReferenceFault">
3245 </wsdl:fault>
3246 <wsdl:fault
3247 message="infod:MissingRequiredParameterFaultErrorMessage"
3248 name="MissingRequiredParameterFault">
3249 </wsdl:fault>
3250 <wsdl:fault name="ExecutionModeFault"
3251 message="tns:ExecutionModeFaultErrorMessage">
3252 </wsdl:fault>
3253 </wsdl:operation>
3254 <wsdl:operation name="DropPropertyVocabulary">
3255 <wsdl:input message="tns:DropPropertyVocabularyRequest"></wsdl:input>
3256 <wsdl:output message="tns:DropPropertyVocabularyResponse"></wsdl:output>
3257 <wsdl:fault name="DropResourceAuthorizationFailure"
3258 message="tns:DropResourceAuthorizationFaultErrorMessage">
3259 </wsdl:fault>
3260 <wsdl:fault name="UnknownResourceReferenceFault"
3261 message="tns:UnknownResourceReferenceFaultErrorMessage">
3262 </wsdl:fault>
3263 <wsdl:fault name="MissingRequiredParameterFault"
3264 message="tns:MissingRequiredParameterFaultErrorMessage">
3265 </wsdl:fault>
3266 <wsdl:fault name="ExecutionModeFault"
3267 message="tns:ExecutionModeFaultErrorMessage">
3268 </wsdl:fault>
3269 </wsdl:operation>
3270 </wsdl:portType>
3271 <wsdl:portType name="infodMetaData">
3272 <wsdl:operation name="GetMetaData">
3273 <wsdl:input message="infod:GetMetaDataQueryExpression"></wsdl:input>
3274 <wsdl:output message="infod:GetMetaDataQueryResponse"></wsdl:output>
3275 <wsdl:fault name="GetMetaDataAuthorizationFailure"
3276 message="tns:GetMetaDataAuthorizationFailureErrorMessage">
3277 </wsdl:fault>
3278 <wsdl:fault name="MissingRequiredParameterFault"
3279 message="tns:MissingRequiredParameterFaultErrorMessage">
3280 </wsdl:fault>
3281 <wsdl:fault name="UnsupportedXQueryFault"
3282 message="tns:UnsupportedXQueryFaultErrorMessage">
3283 </wsdl:fault>
3284 </wsdl:operation>
3285 </wsdl:portType>
3286 <wsdl:portType name="infodNotify">
3287 <wsdl:operation name="Notify">
3288 <wsdl:input message="infod:NotificationMessage"></wsdl:input>
3289 </wsdl:operation>
3290 <wsdl:operation name="PublisherNotification">
3291 <wsdl:input message="infod:PublisherNotification"></wsdl:input>
3292 </wsdl:operation>
3293 <wsdl:operation name="SubscriberNotification">
3294 <wsdl:input message="infod:SubscriberNotification"></wsdl:input>
3295 </wsdl:operation>
3296 <wsdl:operation name="ConsumerNotification">

```

```
3297 <wsdl:input message="infod:ConsumerNotification"></wsdl:input>  
3298 </wsdl:operation>  
3299 </wsdl:portType>  
3300 <wsdl:service name="INFOD"></wsdl:service>  
3301 </wsdl:definitions>
```

3302 7 Author Information

3303 Stephen Davey
3304 National e-Science Centre
3305 15 South College Street
3306 Edinburgh
3307 EH8 9AA
3308 UK
3309
3310 Vijay Dialani
3311 IBM Corporation
3312 Almaden Research Center
3313 650 Harry Road,
3314 San Jose, CA 95120-6099
3315 USA
3316
3317 Ronny Fehling
3318 Oracle Corporation
3319 600 Blvd. de Maisonneuve Ouest
3320 Montreal
3321 Quebec H3A 3J2
3322 Canada
3323
3324 Steve Fisher
3325 Rutherford Appleton Laboratory
3326 Chilton
3327 Didcot
3328 Oxon
3329 OX11 0QX
3330 UK
3331
3332 Dieter Gawlick
3333 Oracle Corporation
3334 500 Oracle Parkway
3335 Redwood Shores
3336 CA 94065
3337 USA
3338
3339 Christopher Kantarjiev
3340 Oracle Corporation
3341 500 Oracle Parkway
3342 Redwood Shores
3343 CA 94065
3344 USA
3345
3346 Cecile Madsen
3347 IBM Silicon Valley Laboratory
3348 555 Bailey Avenue
3349 San Jose, CA 95141
3350 USA
3351
3352

3353 Susan Malaika
3354 IBM Corporation
3355 Silicon Valley Laboratory
3356 555 Bailey Avenue
3357 San Jose, CA 95141
3358 USA
3359
3360 Shailendra Mishra
3361 Oracle Corporation
3362 500 Oracle Parkway
3363 Redwood Shores
3364 CA 94065
3365 USA
3366
3367 Mallikarjun Shankar
3368 Oak Ridge National Laboratory
3369 Oak Ridge
3370 TN 37831
3371 USA

3372

8 Intellectual Property Statement

3373

“Intellectual Property Statement.

3374

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

3375

3376

this document or the extent to which any license under such rights might or might not be

3377

available; neither does it represent that it has made any effort to identify any such rights. Copies

3378

of claims of rights made available for publication and any assurances of licenses to be made

3379

available, or the result of an attempt made to obtain a general license or permission for the use

3380

of such proprietary rights by implementers or users of this specification can be obtained from the

3381

GGF Secretariat.

3382

The GGF invites any interested party to bring to its attention any copyrights, patents or patent

3383

applications, or other proprietary rights which may cover technology that may be required to

3384

practice this recommendation. Please address the information to the GGF Executive Director.”

3385

3386

9 Full Copyright Notice

3387

Copyright (C) Open Grid Forum (2004-2007). All Rights Reserved.

3388

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.

3389

3390

3391

3392

3393

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

3394

3395

3396

3397

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

3398

3399

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

3400