Editors:
J. Ainsworth, ESNW
S. Newhouse, OMII
J. MacLaren, CCT
August, 2005

Resource Usage Service (RUS) based on WS-I Basic Profile 1.0

Status of this Memo

This document defines the Resource Usage Service (RUS) specification, which uses only Web Services standards contained in the WS-I Basic Profile 1.0 [WSIBP10]. Distribution of this document is unlimited. This is a DRAFT document and continues to be revised.

Copyright Notice

Abstract

This document describes the Resource Usage Service (RUS) specification as developed by the Market for Computational Services project within the UK e-Science programme. This proposed specification only uses established Web Service specifications contained within the WS-I Basic Profile 1.0 [WSIBP10], and does not require specifications in the OGSI or WSRF families. Therefore it is a ‘plain web services’ version of the of the original RUS specification [OGSI-RUS], which was dependent on OGSI. Consequently the port types are largely unchanged, whilst the service data elements no longer exist. WSDL and XML schemas are attached as appendices. The RUS uploads (and provides) record of resource usage through an XML document using a format developed by the Usage Record Working Group (UR-WG) in their ‘Usage Record XML Format’ document [UR-SPEC].

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

Contents

Joel Replogle 10/4/05 5:19 PM
Deleted: 17

Joel Replogle 10/4/05 5:19 PM
Deleted: 17

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

Joel Replogle 10/4/05 5:03 PM
Deleted: ~~19/08/2005~~

1 Introduction

This document describes the service interface and behaviour of a WS-I Basic Profile 1.0 compatible Resource Usage Service (RUS) that enables the recording and retrieval of consumed resource information. This service is needed to provide information about service use to a variety of Grid entities:

- The service manager who wishes to examine utilisation across their resources.
- A service that wishes to charge for the use of the consumed resources.
- A virtual organisation that wishes to monitor resource activity within their Grid.

This document is based on the OGSI RUS specification [OGSI-RUS], and the original text has been retained where ever possible.

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

2 Overview

2.1 Architecture

The Resource Usage Service's primary requirement is to store records relating to the consumption of resources as described through the Usage Records specification [UR-SPEC]. While the Usage Records specification and the current deployment of the RUS has focussed around recording the resources consumed within computational batch jobs, there is no reason why the Usage Records schema could not be extended to record information relating to the invocation of a Web Service. However, its primary goal is to collect resource usage information within a virtual organisation, by transferring information from the collection (client) point to the storage (service) point using the Usage Record specification, e.g. as within the UK e-Science Grid. There are therefore two primary functions: the upload of resource usage information into, and the extraction of resource usage data from the service.

A RUS implementation is therefore composed of two principal components:

- A web service that implements the RUS interface and provides for the storage, retrieval and management of records.
- A mechanism which provides for the persistent storage of records and query functionality across the stored data (e.g. an XML database)

It is the former that is specified in this document as the latter is implementation dependent and never exposed through the service interface.

2.2 Definitions and Notational Conventions

Throughout this document we will use the term 'user' as a generic term for a client to a RUS which may be an interactive client or a service instance interacting with a RUS instance.

This specification uses namespace prefixes throughout; they are listed in *Table 1*. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

Joel Replogle 10/4/05 5:19 PM
Deleted: Table 1

Table 1: Prefixes and namespaces used in this specification.

Prefix	Namespace
wsdl	"http://schemas.xmlsoap.org/wsdl/"
http	"http://www.w3.org/2002/06/wsdl/http"
xsd	"http://www.w3.org/2001/XMLSchema"
xsi	"http://www.w3.org/2001/XMLSchema-instance"
urwg	"http://www.gridforum.org/2003/ur-wg"
rus	"http://www.gridforum.org/2005/rus-wg/types"

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

2.3 Scope

In the remainder of this document we describe the Web Service interface to the RUS.

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

3 Configuration

3.1 Users and Authorisation

The security model (see Section 6) defined for the RUS requires the identification of 'resource managers' who are permitted to contribute records and view the records that they have contributed and 'administrators' who can view all records. As the identities of these individuals are not exposed through the service interface there is no need to specify how this information is stored.

3.1.1 Resource Manager

The resource manager has the right to view all activity relating to the resource for which they have managerial responsibility. Responsibility may be defined by machine name, project name, submit host or domain name (e.g., "*.cfs.ac.uk" for machines in the ".cfs.ac.uk" domain).

3.1.2 Administrator

The administrator has full read and write permission on all the records stored within the RUS. This is not restricted any further by a resource identifier.

3.1.3 Other Users

An implementation may define other classes of users that have other degrees of access to the records.

3.2 Mandatory Usage Record Elements

The Usage Record schema [UR-SPEC], which is used by the RUS to describe the record of consumed resources being stored within the service, defines nearly all elements as optional. To ensure consistency and traceability across a set of Usage Records, some elements must be present. For example, it is useful to know who consumed what resources and when. To this end, the RUS can be configured to ensure that a set of elements is present (or mandatory) in a Usage Record. If a record is received with out these elements the RUS will reject it as invalid. The XML schema fragment for recording the mandatory Usage Record elements is shown below.

```
    <xsd:complexType name="MandatoryElementsType">
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
             <xsd:element ref="urwg:MachineName"/>
             <xsd:element ref="urwg:ProjectName"/>
             <xsd:element ref="urwg:SubmitHost"/>
                 … full list from ur-schema.xsd ...
        </xsd:choice>
    </xsd:complexType>
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

4 Resource Usage Record Format

The record of resource consumption that moves in or out of the RUS is based around the Usage Record XML schema [UR-SPEC] and should be consulted in reading this section. The internal storage within the RUS of these records is an implementation issue.

Each usage record retrieved from the RUS is encapsulated within a rus:RUSUsageRecord element. This element contains:

1. A history list with 1 or more entries each encapsulated within a rus:RecordHistory element, which identifies who modified a record and when. The first entry in this list has the following form:
   a. A StoredBy element, containing a XMLDSIG KeyInfo element containing the X509SubjectName of the entity which stored the record.
   b. A TimeStamp element, simply a xsd:dateTime, stating when the record was stored.

   Subsequent entries have the form:
   a. A ModifiedBy element, containing a XMLDSIG KeyInfo element containing the X509SubjectName of the entity which modified the record.
   b. A TimeStamp element, simply a xsd:dateTime, stating when the modified record was stored.
2. A RUSRecordId element, an xsd:unsigned-long uniquely identifying the RUSUsageRecord element *within that RUS*.
3. A single UsageRecord element.

When you query a service for a record, you retrieve the RUSUsageRecord element described below, which encapsulates the original UsageRecord element. This means that the RecordHistory and RUSRecordId can be retrieved along with the associated Usage Record. An example of such an XML document is shown here:

```
<rus:RUSUsageRecord xmlns:rus="http://www.gridforum.org/2005/rus-
wg/types">
 <rus:RecordHistory>
  <rus:StoredBy>
   <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:X509Data>
     <ds:X509SubjectName>
      CN=john ainsworth, L=MC, OU=Manchester, O=eScience, C=UK
     </ds:X509SubjectName>
    </ds:X509Data>
   </ds:KeyInfo>
  </rus:StoredBy>
  <rus:TimeStamp>13 January 2005 14:28:12 GMT</rus:TimeStamp>
 </rus:RecordHistory>
 <rus:RUSRecordId>19870</rus:RUSRecordId>
 <UsageRecord xmlns=http://www.gridforum.org/2003/ur-wg
xmlns:urwg="http://www.gridforum.org/2003/ur-wg"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RecordIdentity urwg:recordId="JSS-UNIQUE-ID"
urwg:createTime="2003-08-13T18:56:56Z" />
   <JobIdentity>
    <GlobalJobId>green147989</GlobalJobId>
    <LocalJobId>147989</LocalJobId>
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
   </JobIdentity>
  <UserIdentity>
   <LocalUserId>wwmarko</LocalUserId>
   <ds:KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <X509Data>
      <X509SubjectName>
        CN=jon maclaren, L=MC, OU=Manchester, O=eScience, C=UK
      </X509SubjectName>
    </X509Data>
   </ds:KeyInfo>
  </UserIdentity>
  <JobName>-------</JobName>
  <Status>completed</Status>
  <TimeDuration urwg:type="cpuTimeRequested">PT1800S</TimeDuration>
  <TimeDuration urwg:type="wallTimeRequested">PT1800S</TimeDuration>
  <TimeInstant urwg:type="timeSubmitted">2004-11-
29T06:47:30</TimeInstant>
  <Processors>1</Processors>
  <ProjectName>cs5015</ProjectName>
  <Host>green</Host>
  <CpuDuration>PT0.0S</CpuDuration>
  <WallDuration>PT1S</WallDuration>
  <StartTime>2004-11-29T06:48:33</StartTime>
  <EndTime>2004-11-29T06:48:34</EndTime>
  <MachineName>green</MachineName>
  <SubmitHost>wren</SubmitHost>
  <Queue>normal</Queue>
  <Resource urwg:description="quoteReference">contract1234</Resource>
  <Resource urwg:description="contractNumber">escience</Resource>
 </UsageRecord>
</rus:RUSUsageRecord>
```

The schema for the `rus:RUSUsageRecord` is attached in Section *13.1*.

Joel Replogle 10/4/05 5:19 PM
Deleted: 11.1

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

5 Service Interface Definition

The RUS has two main functionalities to: upload and retrieve information. The information moved into or out of the RUS is contained within an XML document the syntax of which was described in the previous section.

5.1 General

5.1.1 operationResult Element

Whenever records are stored, retrieved, updated or deleted, there can be failures. In a correctly functioning RUS, three failures are possible: records might not be processed due to permission problems, a non-existent RUSRecordId may be used, a submitted Usage Record maybe invalid.

In order to allow the RUS to inform the client of what has happened, we define a rus:operationResult element which contains:

1. A mandatory “TotalSuccess” element (xsd:boolean). Only true if there were no records not processed (i.e. returned/inserted/modified/removed) due to problems.
2. Optional “Processed” element (xsd:unsigned-long). The number of records successfully processed.
3. Optional “PermissionDenied” element (xsd:unsigned-long). The number of records not processed due to permission problems.
4. Optional “NonExistent” element (xsd:unsigned-long). The number of records not processed due to non-existent RUSRecordIds being used.
5. Optional “Invalid” element (xsd:unsigned-long). The number of records that were rejected as not conforming to either the Usage Record schema or the mandatory elements list.
6. Optional “Duplicate” element (xsd:unsigned-long). The number of records that were rejected as duplicates of records already stored in the RUS.

Items 2, 3, 4, 5 and 6 are optional because a RUS may not want to say why something has failed, e.g. the RUS might not want the user to be able to distinguish a non-existent record with one which the user has not got permission to see. The XML schema fragment is shown below:

```
<xsd:element name="OperationResult">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="Status" type="xsd:boolean"
minOccurs="1" maxOccurs="1"/>
            <xsd:element name="Processed" type="xsd:unsignedLong"
minOccurs="0" maxOccurs="1"/>
            <xsd:element name="PermissionDenied"
type="xsd:unsignedLong" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="NonExistent" type="xsd:unsignedLong"
minOccurs="0" maxOccurs="1"/>
            <xsd:element name="Invalid"
type="xsd:unsignedLong" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="Duplicate" type="xsd:unsignedLong"
minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
</xsd:element>
```

5.1.2 RUSRecordIdList Element

For insert, replace, modify and update operations, the RUS returns a list of xsd:long which has to be the same length as the supplied list of elements (to allow a client to determine which records were stored, and which failed). If the record is operated on successfully, its RUSRecordId is returned in this list. If the operation is unsuccessful for the record, then a return code is placed into the RUSRecordIdList to indicate the cause of failure. The order of the RUSRecordId elements in the RUSRecordIdList MUST correspond to the order of the records in the initial input operation.

These are defined as:

RUSRecordId	>0	Any positive non-zero integer indicates a valid RUS Record Id.
Unspecified	0	Used when the cause of the failure is to be kept private
PermissionDenied	-1	User is not authorised to perform the requested operation on this record
NonExistent	-2	The RUSRecordId does not exists
Invalid	-3	The supplied Usage Record is invalid when checked against the schema or has missing mandatory elements
Duplicate	-4	The record already exists in the RUS

The structure of the XML element is defined by the following schema fragment:

```
<xsd:complexType name="RUSRecordIdListType">
    <xsd:sequence>
        <xsd:element name="RUSRecordId" type="xsd:long" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

5.2 Faults

Three faults may be raised by the RUS:

- RUSProcessingFault – an internal error has occurred
- RUSInputFault – the parameters supplied when an operation was invoked are incorrect. For example, this fault is raised if the supplied XML is invalid when checked against the schema
- RUSUserNotAuthorisedFault – the user does not have permission to invoke the requested operation on the RUS

5.3 Permission Model

The Distinguished Name from the client's certificate must be known to the RUS's access control mechanism (see 3.1) otherwise the RUSUserNotAuthorisedFault message is returned. Authorisation for the user to perform the requested operation is determined for each individual record.

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

5.4 Storage

5.4.1 RUS::insertUsageRecords

The `insertUsageRecords` places new usage records into the RUS.

Input

- *Mandatory: List of Usage Record elements*: A list of UsageRecord elements as defined in [UR-SPEC].

Output

- *Mandatory: OperationResult* element: As defined in section 5.1.1.
- *Mandatory: RecordIdList* element: As defined in section 5.1.2.

The following XML schema fragment defines the combined output for this port type.

```
<xsd:element name="RecordListOperationResult" >
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="rus:OperationResult"/>
            <xsd:element ref="rus:RUSRecordIdList"/>
         </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

Faults

- RUSProcessingFault
- RUSUserNotAuthorised

The `insertUsageRecords` operation differs from the `replaceUsageRecord` operation in that it will insert a new usage record into the RUS. The success or failure to insert a UsageRecord is recorded in the output RUSRecordIdList element.

5.5 Extraction

5.5.1 RUS::extractRUSUsageRecords

Enable the client to find all usage records relating to a more complicated set of requirements.

Input

- *Mandatory: searchTerm (1)*: All resource usage records relating to the search criteria that meet the specified access rules should be returned to the client. The search term should be specified as part of an XPath/XQuery string.

Output

- *Mandatory:* Zero or more *RUSUsageRecord* elements.
- *Mandatory: OperationResult.*

These two outputs are combined into a single extractRecordsResponse, as in the following XML schema fragment.

```
<xsd:element name="extractRUSUsageRecordsResponse">
      <xsd:complexType>
          <xsd:sequence>
            <xsd:element ref="rus:OperationResult"/>
            <xsd:element ref="rus:RUSUsageRecord" minOccurs="0"
maxOccurs="unbounded"/>
          </xsd:sequence>
      </xsd:complexType>
</xsd:element>
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

Faults
- RUSProcessingFault
- RUSUserNotAuthorised

5.5.2 RUS::extractRUSRecordIds
Enable the client to find all usage records relating to a more complicated set of requirements, returning just the record identifiers and not the full usage records.

Input
- *Mandatory: searchTerm*: the identifiers of all resource usage records relating to the search criteria that meet the specified access rules should be returned to the client. The search term should be specified as part of an XPath/XQuery string.

Output
- *Mandatory: RUSRecordIdList* element..
- *Mandatory: OperationResult* element.

These two outputs are combined into a single extractRUSRecordIdsResponse, as shown in the following XML schema fragment.

```
<xsd:element name="extractRUSRecordIdsResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="rus:OperationResult"/>
            <xsd:element ref="rus:RUSRecordIdList"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

Faults
- RUSProcessingFault
- RUSUserNotAuthorised

5.5.3 RUS::extractSpecifiedRUSUsageRecords
Enable the client to retrieve records with the specified RUSRecordIds.

Input
- *Mandatory: RUSRecordIdList element*: This is a RUSRecordIdList elements which contains the RUSRecordIds of the records to be retrieved.

Output
- *Mandatory:* Zero or more *RUSUsageRecord* elements.
- *Mandatory: OperationResult.*

These two outputs are combined into a single extractRUSUsageRecordsResponse, as shown in section 5.5.1. If some of the specified RUSRecordIds are not found then the total number of RUSUsageRecords returned by the operation will not match the number of requested RUSRecordIds. The RUSRecordIds of the successfully returned elements can be discovered from the RUSUsageRecord.

Faults
- RUSProcessingFault
- RUSUserNotAuthorised

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

5.5.4 RUS::extractRUSUsageRecordByGlobalJobID

Enable the client to find all usage records relating to the specified GlobalJobID.

Input

- *Mandatory: GlobalJobID*: All resource usage records relating to this GlobalJobID that meet the specified access rules should be returned to the client.

Output

- *Mandatory*: Zero or more *RUSUsageRecord* elements.
- *Mandatory: OperationResult.*

These two outputs are combined into a single extractRUSUsageRecordsResponse, as shown above.

Faults

- RUSProcessingFault
- RUSUserNotAuthorised

5.5.5 RUS::extractRUSUsageRecordByGlobalUserID

Enable the client to find all usage records relating to the specified X509SubjectName, as located within the XMLDSIG KeyInfo component of a UsageRecord's UserIdentity element.

Input

- *Mandatory: GlobalUserID* : All resource usage records relating to this GlobalUserID that meet the specified access rules should be returned to the client. This is specified as an XMLDSIG X509SubjectName element.

Output

- *Mandatory:* Zero or more *RUSUsageRecord* elements.
- *Mandatory: OperationResult.*

These two outputs are combined into a single extractRUSUsageRecordsResponse, as shown above.

Faults

- RUSProcessingFault
- RUSUserNotAuthorised

5.5.6 RUS::extractRUSUsageRecordByMachineName

Enable the client to find all usage records relating to the specified MachineName.

Input

- *Mandatory: MachineName*: All resource usage records relating to this MachineName that meet the specified access rules should be returned to the client.

Output

- *Mandatory:* Zero or more *RUSUsageRecord* elements.
- *Mandatory: OperationResult.*

These two outputs are combined into a single extractRUSUsageRecordsResponse, as shown above.

Faults

- RUSProcessingFault
- RUSUserNotAuthorised

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

5.5.7 RUS::extractRUSUsageRecordBySubmitHost

Enable the client to find all usage records relating to the specified SubmitHost.

Input

- *Mandatory: SubmitHost*: All RUS Usage Records relating to this SubmitHost that meet the specified access rules should be returned to the client.

Output

- *Mandatory:* Zero or more *RUSUsageRecord* elements.
- *Mandatory: OperationResult.*

These two outputs are combined into a single extractRUSUsageRecordsResponse, as shown above.

Faults

- RUSProcessingFault
- RUSUserNotAuthorised

5.6 Modification

These operations all modify an existing RUS record in some way. The rus:RecordHistory MUST be appended with a ModifiedBy element by these operations.

5.6.1 RUS::incrementUsageRecordPart

This operation will add a numeric increment to an element in a single record. This gives a safe way for multiple writers to update the same record.

Input

- *Mandatory: RUSRecordId element*: The record to be updated
- *Mandatory: ElementPath element (xsd:string – the XPath expression)*: This expression should identify a single element within the record which is of some numerical type. The query expression is applied to the UsageRecord element, not the surrounding RUSUsageRecord element, to prevent the alteration of a RUSRecordId.
- *Mandatory: Increment element (xsd:long)*: The increment to be applied to the identified element

Output

- *Mandatory: OperationResult element*: In this case, this refers to the successful location of an accessible record, not the overall success of the operation.
- *Mandatory: Modified element (xsd:boolean)*: This will be true if only if the record was updated correctly, which implies that the record was found, permissions were OK, *and* the XPath identified a single numeric element

Faults

- RUSProcessingFault
- RUSUserNotAuthorised

5.6.2 RUS::modifyUsageRecordPart

This operation will modify a Usage Record according to an XUpdate expression. Note that this expression will be applied to a single UsageRecord element, NOT the surrounding RUSUsageRecord element, thus preventing the modification of the RUSRecordId, time of storage, etc.

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

It is possible to create races between multiple writers depending upon the XUpdate expressions used.

Input
- *Mandatory: RUSRecordId element*: The element to be updated
- *Mandatory: XUpdate element*: This element will dictate how the usage record will be updated.

Output
- *Mandatory: OperationResult element*: In this case, this refers to the successful location of an accessible record, not the overall success of the operation.
- *Optional: XUpdate results*: If the RUS record was found, this part MUST be present.

Faults
- RUSProcessingFault
- RUSUserNotAuthorised

5.6.3 RUS::replaceUsageRecords

The `replaceUsageRecords` replaces records held in the RUS.

We define a ReplacementRecord element, which contains a RUSRecordId element, and a UsageRecord element, as defined in [UR-SPEC].

Input
- *Mandatory: List of ReplacementRecord elements*: A list of elements, each being a paired RUSRecordId and usage record.

The following XML schema fragment defines this input:
```
<xsd:complexType name="ReplacementRecordType">
    <xsd:sequence>
        <xsd:element name="RUSRecordId" type="xsd:unsignedLong"
minOccurs="1" maxOccurs="1"/>
        <xsd:element ref=”urwg:Usage" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="ReplacementRecords">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="List" type="ReplacementRecordType"
minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```
Output
- *Mandatory: RUSRecordIdList element*: This is a list of elements which MUST be the same length and order as the inputted list. Each element in this list will be a RUSRecordId (an xsd:long).
- *Mandatory: OperationResult element*

The following XML schema fragment defines the combined output for this port type.
```
  <xsd:element name="RecordListOperationResult" >
      <xsd:complexType>
          <xsd:sequence>
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
               <xsd:element ref="rus:OperationResult"/>
               <xsd:element ref="rus:RUSRecordIdList"/>
          </xsd:sequence>
     </xsd:complexType>
 </xsd:element>
```

Faults

- RUSProcessingFault
- RUSUserNotAuthorised

5.7 Deletion

5.7.1 RUS::deleteRecords

Enable the client to delete all usage records that match the specified criteria.

Input

- *Mandatory: searchTerm (1)*: All resource usage records relating to the search criteria that meet the specified access rules should be returned to the client. Defined in the same way as *searchTerm* in extractRecords, Section 5.5.1.

Output

- *Mandatory: RUSRecordIdList* element: As defined in section 5.1.2. This will provide the RUSRecordIds of the records that have been deleted and a reason as to why the record has not been deleted.
- *Mandatory: OperationResult element*

Faults

- RUSProcessingFault
- RUSUserNotAuthorised

5.7.2 RUS::deleteSpecificRecords

Enable the client to delete all usage records with the specified RUSRecordIds

Input

- *Mandatory:RUSRecordIdList element*: This is a list of RUSRecordId elements which contain the RUSRecordIds of the records to be deleted.

Output

- *Mandatory: RUSRecordIdList* element. As defined in section 5.1.2. This will provide the RUSRecordIds of the records that have been deleted and a reason as to why the record has not been deleted.
- *Mandatory: OperationResult element*

Faults

- RUSProcessingFault
- RUSUserNotAuthorised

5.8 Configuration

5.8.1 RUS::listMandatoryUsageRecordElements

Enable the client to retrieve a list of the usage record elements required by this RUS implementation.

Input

- *None*

Joel Replogle 10/4/05 5:03 PM
Deleted: ~~19/08/2005~~

Output

- *Optional: MandatoryElements element*
- *Mandatory: OperationResult element*

Faults

- RUSProcessingFault
- RUSUserNotAuthorised

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

6 Security Considerations

As the RUS is a system on which financial transactions may ultimately depend, security is of the utmost importance. The following measures are required to ensure that service consumers trust service providers to record the usage correctly and that fraudulent use is prevented or traceable. This is the usual trust model employed when consumption is metered.

6.1 Authentication

Underpinning much of the following discussion is the ability to authenticate users. A RUS implementation must accept messages transported in a way that allows user authentication (e.g. mutual TLS or signed messages using WS-Security). If confidentiality is required then the messages between the RUS client and the RUS can be encrypted (e.g. TLS or MLS within WS-Security).

6.2 Authorisation

The authentication data is used to implement a role-based access control model to govern access to the RUS on a per record basis thereby ensuring the privacy of the stored data. A minimal model, defining administrators and resource managers, is defined in section 3.1. For example, a resource manager must be able to only retrieve usage information relating to their resources, and no one else's. Likewise, write access must be restricted to those entities that own the metered resources.

6.3 Audit

The data stored in the RUS must, from the perspective of the RUS, be non-repudiatable. It is not the responsibility of the RUS to determine if the data it is storing is correct, but it must be able to demonstrate the source of this data and how it may have been altered by subsequent operations if it is challenged. This can be achieved by ensuring message integrity between RUS client and RUS, and by maintaining an audit trail of each operation on the RUS that changes the usage data. The audit trail should include the message requesting the operation, the identity of the entity performing the operation, a digital signature of the message and the date and time the operation was requested. The message and its signature are not part of the rus:RUSUsageRecord but should be recorded by a RUS so that they are available for audits and resolving disputes.

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

7 Editor Information

John Ainsworth
e-Science NorthWest
Kilburn Building
University of Manchester
Manchester M13 9PL
United Kingdom
Email: john.ainsworth@manchester.ac.uk

Steven Newhouse
Open Middleware Infrastructure Institute
Suite 6005, Faraday Building (B21), Highfield Campus,
Southampton University, Highfield, Southampton, SO17 1BJ
United Kingdom
Email: s.newhouse@omii.ac.uk

Jon MacLaren
Centre for Computation and Technology (CCT)
352 Johnston Hall
Louisiana State University
Baton Rouge, LA 70803
United States of America
Email: maclaren@cct.lsu.edu

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

8 Contributors

The editors would like to acknowledge the contribution of Anthony Mayer (LeSC) and Asif Saleem (LeSC) for discussions relating to this document.

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

9 Acknowledgements

The creation of this document has been supported by the Computational Markets project funded under the UK e-Science Core programme by the Department of Trade and Industry and the Engineering and Physical Sciences Research Council (http://www.lesc.ic.ac.uk/markets/). Additional support to attend Global Grid Forum meetings has been provided by the UK e-Science GridNet programme.

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

10 Intellectual Property Statement

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

11 Full Copyright Notice

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

12 References

12.1 Normative References

[RFC 2119]
Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, Author. Internet Engineering Task Force, RFC 2119, March 1997. Available at http://www.ietf.org/rfc/rfc2119.txt

[WSDL 1.2]
Web Services Description Language (WSDL) Version 1.2, Published W3C Working Draft, World Wide Web Consortium. Available at http://www.w3.org/TR/wsdl12/

[WSDL 1.2 DRAFT]
Web Services Description Language (WSDL) Version 1.2, W3C Working Draft 3 March 2003, World Wide Web Consortium. Available at http://www.w3.org/TR/2003/WD-wsdl12-20030303

[WSIBP10]
Web Services Interoperability Basic Profile Verion 1.0, Web Services Interoperability Organization. Available at http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html

[UR-SPEC]
Usage Record – XML Format, Draft GGF Recommendation. Available from: http://www.psc.edu/~lfm/Grid/UR-WG/

12.2 Informative References

[OGSI-RUS]
S. Newhouse and J. MacLaren, "Resource Usage Service RUS" Global Grid Forum Resource Usage Service Working Group draft-ggf-rus-service-4. Available online at https://forge.gridforum.org/projects/rus-wg/document/draft-ggf-rus-service-4-public/en/1

[JAX-RPC]
Java^{TM} API for XML-Based RPC (JAX-RPC).
http://java.sun.com/xml/jaxrpc/docs.html

[Web Services Book]
Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI, s. Graham, S. Simeonov, T. Boubez, G. Daniels, D. Davis, Y. Nakamura, R. Neyama. Sams, 2001.

[WSIF]
Welcome to WSIF: Web Services Invocation Framework, http://www.apache.org/wsif/

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

13 XSD and WSDL Specifications

This section contains the full XSD and WSDL definitions for everything described in this document. The definitions in this section MUST be considered definitive, if there are any discrepancies between the definitions in this section and those portions described in other sections above.

13.1 RUS Base Types schema

```
<xsd:schema
      targetNamespace="http://www.gridforum.org/2005/rus-wg/types"
      elementFormDefault="qualified"
      xmlns="http://www.gridforum.org/2005/rus-wg/types"
      xmlns:urwg="http://www.gridforum.org/2003/ur-wg"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">

      <xsd:import namespace="http://www.gridforum.org/2003/ur-wg"
schemaLocation="./urwg-schema-11.xsd" />

      <xsd:complexType name="MandatoryElementsType">
            <xsd:choice minOccurs="1" maxOccurs="unbounded">
              <xsd:element ref="urwg:MachineName"/>
              <xsd:element ref="urwg:ProjectName"/>
              <xsd:element ref="urwg:SubmitHost"/>
              <xsd:element ref="urwg:Processors"/>
              <xsd:element ref="urwg:Memory"/>
              <xsd:element ref="urwg:Disk"/>
              <xsd:element ref="urwg:Network"/>
              <xsd:element ref="urwg:TimeDuration"/>
              <xsd:element ref="urwg:CpuDuration"/>
              <xsd:element ref="urwg:StartTime"/>
              <xsd:element ref="urwg:EndTime"/>
              <xsd:element ref="urwg:NodeCount"/>
              <xsd:element ref="urwg:Queue"/>
              <xsd:element ref="urwg:ServiceLevel"/>
              <xsd:element ref="urwg:WallDuration"/>
              <xsd:element ref="urwg:UserIdentity"/>
              <xsd:element ref="urwg:JobIdentity"/>
            </xsd:choice>
      </xsd:complexType>

      <xsd:element name="RUSUsageRecords" type="RUSUsageRecordsType"
/>

        <xsd:element name="OperationResult">
             <xsd:complexType>
              <xsd:sequence>
                  <xsd:element name="Status" type="xsd:boolean"
minOccurs="1" maxOccurs="1"/>
                  <xsd:element name="Processed"
type="xsd:unsignedLong" minOccurs="0" maxOccurs="1"/>
                  <xsd:element name="PermissionDenied"
type="xsd:unsignedLong" minOccurs="0" maxOccurs="1"/>
                  <xsd:element name="NonExistent"
type="xsd:unsignedLong" minOccurs="0" maxOccurs="1"/>
                  <xsd:element name="Invalid" type="xsd:unsignedLong"
minOccurs="0" maxOccurs="1"/>
                  <xsd:element name="Duplicate"
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
type="xsd:unsignedLong" minOccurs="0" maxOccurs="1"/>
              </xsd:sequence>
           </xsd:complexType>
        </xsd:element>

      <xsd:complexType name="WhoAndWhenType">
            <xsd:sequence>
                  <xsd:element ref="ds:KeyInfo"/>
                  <xsd:element name="TimeStamp" type="xsd:dateTime"
maxOccurs="1" minOccurs="0"/>
            </xsd:sequence>
      </xsd:complexType>

      <xsd:complexType name="RecordHistoryType">
            <xsd:sequence>
                  <xsd:element name="StoredBy"
type="WhoAndWhenType"/>
                  <xsd:element name="ModifiedBy"
type="WhoAndWhenType" maxOccurs="unbounded" minOccurs="0"/>
            </xsd:sequence>
      </xsd:complexType>

      <xsd:complexType name="RUSUsageRecordType">
            <xsd:sequence>
                  <xsd:element name="RecordHistory"
type="RecordHistoryType"/>
                  <xsd:element name="RUSRecordId" type="xsd:long"/>
                  <xsd:element ref="urwg:Usage"/>
            </xsd:sequence>
      </xsd:complexType>


      <xsd:complexType name="RUSUsageRecordsType">
              <xsd:sequence>
                  <xsd:element name="RUSUsageRecord"
type="RUSUsageRecordType" minOccurs="0" maxOccurs="unbounded"/>
              </xsd:sequence>
      </xsd:complexType>


        <xsd:complexType name="ReplacementRecordType">
                <xsd:sequence>
                <xsd:element name="RUSRecordId" type="xsd:long"
minOccurs="1" maxOccurs="1"/>
                    <xsd:element ref="urwg:Usage" minOccurs="1"
maxOccurs="1"/>
                </xsd:sequence>
        </xsd:complexType>

        <xsd:element name="ReplacementRecords">
                <xsd:complexType>
                <xsd:sequence>
                <xsd:element name="List" type="ReplacementRecordType"
minOccurs="1" maxOccurs="unbounded"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>

        <xsd:element name="RUSRecordIdList">
            <xsd:complexType>
                <xsd:sequence>
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
                    <xsd:element name="RUSRecordId" type="xsd:long"
minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>


            <xsd:complexType name="RUSRecordIdListType">
                <xsd:sequence>
                    <xsd:element name="RUSRecordId" type="xsd:long"
minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
            </xsd:complexType>
</xsd:schema>
```

13.2 Service Interface WSDL specifications

13.2.1 resource-usage-porttype.wsdl

```
<wsdl:definitions name="resourceusage"
      targetNamespace="http://www.gridforum.org/2005/rus-wg/service"
      xmlns="http://schemas.xmlsoap.org/wsdl/"
      xmlns:intf="http://www.gridforum.org/2005/rus-wg/service"
      xmlns:types="http://www.gridforum.org/2005/rus-
wg/service/types"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:urwg="http://www.gridforum.org/2003/ur-wg"
      xmlns:rus="http://www.gridforum.org/2005/rus-wg/types"
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
      xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <types>
        <xsd:schema
            targetNamespace="http://www.gridforum.org/2005/rus-
wg/service/types"
            elementFormDefault="qualified"
            attributeFormDefault="qualified"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
          xmlns:rus="http://www.gridforum.org/2005/rus-wg/types"
          xmlns:urwg="http://www.gridforum.org/2003/ur-wg"
            xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
          xmlns="http://www.w3.org/2001/XMLSchema">

            <xsd:import namespace="http://www.gridforum.org/2003/ur-
wg" schemaLocation="./urwg-schema-11.xsd" />
            <xsd:import namespace="http://www.gridforum.org/2005/rus-
wg/types" schemaLocation="./RUS.xsd" />

            <xsd:element name="RusInputFault" type="xsd:string"/>
            <xsd:element name="RusProcessingFault"
type="xsd:string"/>
            <xsd:element name="RusUserNotAuthorisedFault"
type="xsd:string"/>

            <xsd:element name="recordListOperationResponse" >
                <xsd:complexType>
                    <xsd:sequence>
                  <xsd:element ref="rus:OperationResult"/>
                  <xsd:element ref="rus:RUSRecordIdList"/>
                    </xsd:sequence>
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

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

<xsd:element name="insertUsageRecordsRequest">
            <xsd:complexType>
                  <xsd:sequence>
                        <xsd:element name="records"
type="urwg:UsageRecordType" minOccurs="1" maxOccurs="unbounded"/>
                  </xsd:sequence>
            </xsd:complexType>
          </xsd:element>


           <xsd:simpleType name="xPathQuery" type="xsd:string"/>

            <xsd:element name="extractRecordsResponse">□-A
            <xsd:complexType>
                <xsd:sequence>
                  <xsd:element ref="rus:OperationResult"/>
                  <xsd:element name="Records"
type="rus:RUSUsageRecordType" minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
            </xsd:complexType>
          </xsd:element>

            <xsd:element name="extractRUSRecordIdsResponse">
            <xsd:complexType>
                <xsd:sequence>
                  <xsd:element ref="rus:OperationResult"/>
                  <xsd:element ref="rus:RUSRecordIdList"
minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
            </xsd:complexType>
          </xsd:element>

         <xsd:element
name="extractRUSUsageRecordsBySubmitHostRequest">
           <xsd:complexType>
                <xsd:sequence>
                  <xsd:element ref="urwg:SubmitHost"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>

          <xsd:element
name="extractRUSUsageRecordsByGlobalJobIDRequest" type="xsd:string"/>

          <xsd:element
name="extractRUSUsageRecordsByMachineNameRequest" >
            <xsd:complexType>
              <xsd:sequence>
                    <xsd:element ref="urwg:MachineName"/>
              </xsd:sequence>
            </xsd:complexType>
        </xsd:element>

          <xsd:element
name="extractRUSUsageRecordsByGlobalUserIDRequest"
type="xsd:string"/>

          <xsd:element name="replaceUsageRecordsRequest" >
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
            <xsd:complexType>
                  <xsd:sequence>
                        <xsd:element ref="rus:ReplacementRecords"/>
                  </xsd:sequence>
            </xsd:complexType>
          </xsd:element>

            <xsd:element name="deleteRecordsRequest"
type="types:xPathQuery"/>

            <xsd:element name="extractRecordsRequest"
type="types:xPathQuery"/>
            <xsd:element name="extractRUSRecordIdsRequest"
type="types:xPathQuery"/>
            <xsd:element name="deleteSpecificRecordsRequest"
type="rus:RUSRecordIdListType"/>
            <xsd:element name="extractSpecificRUSRecordsRequest"
type="rus:RUSRecordIdListType"/>

            <xsd:element name="incrementUsageRecordPartRequest">
            <xsd:complexType>
                  <xsd:sequence>
                        <xsd:element name="RUSRecordId"
type="xsd:long"/>
                                <xsd:element name="path"
type="types:xPathQuery"/>
                        <xsd:element name="increment"
type="xsd:long"/>
                  </xsd:sequence>
            </xsd:complexType>
          </xsd:element>

            <xsd:element name="incrementUsageRecordPartResponse">
            <xsd:complexType>
                  <xsd:sequence>
                        <xsd:element ref="rus:OperationResult"/>
                                <xsd:element name="modified"
type="xsd:boolean" minOccurs="0" />
                  </xsd:sequence>
            </xsd:complexType>
          </xsd:element>

            <xsd:element name="modifyUsageRecordPartRequest">
            <xsd:complexType>
                  <xsd:sequence>
                        <xsd:element name="RUSRecordId"
type="xsd:long"/>
                        <xsd:element name="XUpdate"
type="types:xPathQuery"/>
                  </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
            <xsd:element
name="listMandatoryUsageRecordElementsResponse">
            <xsd:complexType>
                  <xsd:sequence>
                        <xsd:element ref="rus:OperationResult"/>
                                <xsd:element name="MandatoryElements"
type="rus:MandatoryElementsType" minOccurs="0"/>
                  </xsd:sequence>
            </xsd:complexType>
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
          </xsd:element>
            <xsd:element name="modifyUsageRecordPartResponse">
            <xsd:complexType>
                  <xsd:sequence>
                        <xsd:element ref="rus:OperationResult"/>
                                <xsd:element name="XUpdateResults"
type="xsd:boolean" minOccurs="0"/>
                  </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
           <xsd:element
name="listMandatoryUsageRecordElementsRequest">
            <xsd:complexType>
                  <xsd:sequence>
                                <xsd:element name="request"
type="xsd:boolean" minOccurs="0"/>
                  </xsd:sequence>
            </xsd:complexType>
          </xsd:element>

       </xsd:schema>
   </types>

    <wsdl:message name="RusUserNotAuthorisedFaultMessage">
      <part name="FaultDiagnostic"
element="types:RusUserNotAuthorisedFault"/>
    </wsdl:message>

    <wsdl:message name="RusInputFaultMessage">
      <part name="FaultDiagnostic" element="types:RusInputFault"/>
    </wsdl:message>

    <wsdl:message name="RusProcessingFaultMessage">
      <part name="FaultDiagnostic"
element="types:RusProcessingFault"/>
    </wsdl:message>

    <wsdl:message name="listMandatoryUsageRecordElementsInput"/>
    <wsdl:message name="listMandatoryUsageRecordElementsOutput">
        <wsdl:part name="output"
element="types:listMandatoryUsageRecordElementsResponse"/>
    </wsdl:message>

    <wsdl:message name="insertUsageRecordsOutput">
        <wsdl:part name="output"
element="types:recordListOperationResponse"/>
    </wsdl:message>

    <wsdl:message name="deleteRecordsOutput">
        <wsdl:part name="output"
element="types:recordListOperationResponse"/>
    </wsdl:message>

    <wsdl:message name="deleteSpecificRecordsOutput">
        <wsdl:part name="output"
element="types:recordListOperationResponse"/>
    </wsdl:message>

    <wsdl:message name="extractRUSRecordIdsOutput">
        <wsdl:part name="output"
element="types:extractRUSRecordIdsResponse"/>
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
    </wsdl:message>

    <wsdl:message name="extractRecordsOutput">
        <wsdl:part name="output"
element="types:extractRecordsResponse"/>
    </wsdl:message>

    <wsdl:message name="extractSpecificRUSRecordsOutput">
        <wsdl:part name="output"
element="types:extractRecordsResponse"/>
    </wsdl:message>

    <wsdl:message name="extractRUSUsageRecordsBySubmitHostOutput">
        <wsdl:part name="output"
element="types:extractRecordsResponse" />
    </wsdl:message>

    <wsdl:message name="extractRUSUsageRecordsByGlobalJobIDOutput">
        <wsdl:part name="parameters"
element="types:extractRecordsResponse" />
    </wsdl:message>

    <wsdl:message name="extractRUSUsageRecordsByGlobalUserIDOutput">
        <wsdl:part name="parameters"
element="types:extractRecordsResponse" />
    </wsdl:message>

    <wsdl:message name="extractRUSUsageRecordsByMachineNameOutput">
        <wsdl:part name="parameters"
element="types:extractRecordsResponse" />
    </wsdl:message>

    <wsdl:message name="incrementUsageRecordPartOutput">
        <wsdl:part name="parameters"
element="types:incrementUsageRecordPartResponse"/>
    </wsdl:message>

    <wsdl:message name="modifyUsageRecordPartOutput">
        <wsdl:part name="parameters"
element="types:modifyUsageRecordPartResponse"/>
    </wsdl:message>

    <wsdl:message name="replaceUsageRecordsOutput">
        <wsdl:part name="parameters"
element="types:recordListOperationResponse"/>
    </wsdl:message>

    <wsdl:message name="extractRUSUsageRecordsBySubmitHostInput">
        <wsdl:part name="input"
element="types:extractRUSUsageRecordsBySubmitHostRequest"/>
    </wsdl:message>

    <wsdl:message name="extractRUSUsageRecordsByMachineNameInput">
        <wsdl:part name="parameters"
element="types:extractRUSUsageRecordsByMachineNameRequest"/>
    </wsdl:message>

    <wsdl:message name="extractRUSUsageRecordsByGlobalUserIDInput">
        <wsdl:part name="parameters"
element="types:extractRUSUsageRecordsByGlobalUserIDRequest"/>
    </wsdl:message>
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
    <wsdl:message name="extractRUSUsageRecordsByGlobalJobIDInput">
        <wsdl:part name="parameters"
element="types:extractRUSUsageRecordsByGlobalJobIDRequest"/>
    </wsdl:message>

    <wsdl:message name="insertUsageRecordsInput">
        <wsdl:part name="parameters"
element="types:insertUsageRecordsRequest"/>
    </wsdl:message>

    <wsdl:message name="deleteRecordsInput">
        <wsdl:part name="parameters"
element="types:deleteRecordsRequest"/>
    </wsdl:message>

    <wsdl:message name="deleteSpecificRecordsInput">
        <wsdl:part name="parameters"
element="types:deleteSpecificRecordsRequest"/>
    </wsdl:message>

    <wsdl:message name="extractRUSRecordIdsInput">
        <wsdl:part name="parameters"
element="types:extractRUSRecordIdsRequest"/>
    </wsdl:message>

    <wsdl:message name="extractRecordsInput">
        <wsdl:part name="parameters"
element="types:extractRecordsRequest"/>
    </wsdl:message>

    <wsdl:message name="extractSpecificRUSRecordsInput">
        <wsdl:part name="parameters"
element="types:extractSpecificRUSRecordsRequest"/>
    </wsdl:message>

    <wsdl:message name="incrementUsageRecordPartInput">
        <wsdl:part name="parameters"
element="types:incrementUsageRecordPartRequest"/>

    </wsdl:message>

    <wsdl:message name="modifyUsageRecordPartInput">
        <wsdl:part name="parameters"
element="types:modifyUsageRecordPartRequest"/>
    </wsdl:message>

    <wsdl:message name="replaceUsageRecordsInput">
        <wsdl:part name="parameters"
element="types:replaceUsageRecordsRequest"/>
    </wsdl:message>

    <wsdl:portType name="ResourceUsagePortType" >
        <wsdl:operation name="listMandatoryUsageRecordElements" >
            <wsdl:input
message="intf:listMandatoryUsageRecordElementsInput" />
            <wsdl:output
message="intf:listMandatoryUsageRecordElementsOutput" />
            <fault name="RusUserNotAuthorisedFault"
message="intf:RusUserNotAuthorisedFaultMessage"/>
            <fault name="RusProcessingFault"
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
message="intf:RusProcessingFaultMessage"/>
        </wsdl:operation>
        <wsdl:operation name="insertUsageRecords" >
            <wsdl:input message="intf:insertUsageRecordsInput" />
            <wsdl:output message="intf:insertUsageRecordsOutput" />
            <fault name="RusUserNotAuthorisedFault"
message="intf:RusUserNotAuthorisedFaultMessage"/>
            <fault name="RusProcessingFault"
message="intf:RusProcessingFaultMessage"/>
        </wsdl:operation>

        <wsdl:operation name="extractRUSUsageRecordsByGlobalJobID" >
            <wsdl:input
message="intf:extractRUSUsageRecordsByGlobalJobIDInput" />
            <wsdl:output
message="intf:extractRUSUsageRecordsByGlobalJobIDOutput" />
            <fault name="RusUserNotAuthorisedFault"
message="intf:RusUserNotAuthorisedFaultMessage"/>
            <fault name="RusProcessingFault"
message="intf:RusProcessingFaultMessage"/>
        </wsdl:operation>

        <wsdl:operation name="extractRUSUsageRecordsByGlobalUserID" >
            <wsdl:input
message="intf:extractRUSUsageRecordsByGlobalUserIDInput" />
            <wsdl:output
message="intf:extractRUSUsageRecordsByGlobalUserIDOutput"/>
            <fault name="RusUserNotAuthorisedFault"
message="intf:RusUserNotAuthorisedFaultMessage"/>
            <fault name="RusProcessingFault"
message="intf:RusProcessingFaultMessage"/>
        </wsdl:operation>

        <wsdl:operation name="extractRUSUsageRecordsByMachineName" >
            <wsdl:input
message="intf:extractRUSUsageRecordsByMachineNameInput" />
            <wsdl:output
message="intf:extractRUSUsageRecordsByMachineNameOutput" />
            <fault name="RusUserNotAuthorisedFault"
message="intf:RusUserNotAuthorisedFaultMessage"/>
            <fault name="RusProcessingFault"
message="intf:RusProcessingFaultMessage"/>
        </wsdl:operation>

        <wsdl:operation name="extractRUSUsageRecordsBySubmitHost" >
            <wsdl:input
message="intf:extractRUSUsageRecordsBySubmitHostInput" />
            <wsdl:output
message="intf:extractRUSUsageRecordsBySubmitHostOutput" />
            <fault name="RusUserNotAuthorisedFault"
message="intf:RusUserNotAuthorisedFaultMessage"/>
            <fault name="RusProcessingFault"
message="intf:RusProcessingFaultMessage"/>
        </wsdl:operation>

        <wsdl:operation name="extractRecords" >
            <wsdl:input message="intf:extractRecordsInput" />
            <wsdl:output message="intf:extractRecordsOutput" />
            <fault name="RusUserNotAuthorisedFault"
message="intf:RusUserNotAuthorisedFaultMessage"/>
            <fault name="RusProcessingFault"
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
message="intf:RusProcessingFaultMessage"/>
        </wsdl:operation>

        <wsdl:operation name="extractRUSRecordIds" >
            <wsdl:input message="intf:extractRUSRecordIdsInput" />
            <wsdl:output message="intf:extractRUSRecordIdsOutput" />
            <fault name="RusUserNotAuthorisedFault"
message="intf:RusUserNotAuthorisedFaultMessage"/>
            <fault name="RusProcessingFault"
message="intf:RusProcessingFaultMessage"/>
        </wsdl:operation>

        <wsdl:operation name="extractSpecificRUSRecords" >
            <wsdl:input message="intf:extractSpecificRUSRecordsInput"
/>
            <wsdl:output
message="intf:extractSpecificRUSRecordsOutput" />
            <fault name="RusUserNotAuthorisedFault"
message="intf:RusUserNotAuthorisedFaultMessage"/>
            <fault name="RusProcessingFault"
message="intf:RusProcessingFaultMessage"/>
        </wsdl:operation>

        <wsdl:operation name="deleteSpecificRecords" >
            <wsdl:input message="intf:deleteSpecificRecordsInput" />
            <wsdl:output message="intf:deleteSpecificRecordsOutput"
/>
            <fault name="RusUserNotAuthorisedFault"
message="intf:RusUserNotAuthorisedFaultMessage"/>
            <fault name="RusProcessingFault"
message="intf:RusProcessingFaultMessage"/>
        </wsdl:operation>

        <wsdl:operation name="deleteRecords" >
            <wsdl:input message="intf:deleteRecordsInput" />
            <wsdl:output message="intf:deleteRecordsOutput" />
            <fault name="RusUserNotAuthorisedFault"
message="intf:RusUserNotAuthorisedFaultMessage"/>
            <fault name="RusProcessingFault"
message="intf:RusProcessingFaultMessage"/>
        </wsdl:operation>

        <wsdl:operation name="incrementUsageRecordPart" >
            <wsdl:input message="intf:incrementUsageRecordPartInput"
/>
            <wsdl:output
message="intf:incrementUsageRecordPartOutput" />
            <fault name="RusUserNotAuthorisedFault"
message="intf:RusUserNotAuthorisedFaultMessage"/>
            <fault name="RusProcessingFault"
message="intf:RusProcessingFaultMessage"/>
        </wsdl:operation>

        <wsdl:operation name="modifyUsageRecordPart" >
            <wsdl:input message="intf:modifyUsageRecordPartInput" />
            <wsdl:output message="intf:modifyUsageRecordPartOutput"
/>
            <fault name="RusUserNotAuthorisedFault"
message="intf:RusUserNotAuthorisedFaultMessage"/>
            <fault name="RusProcessingFault"
message="intf:RusProcessingFaultMessage"/>
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
            <fault name="RusInputFault"
message="intf:RusInputFaultMessage"/>
        </wsdl:operation>

        <wsdl:operation name="replaceUsageRecords" >
            <wsdl:input message="intf:replaceUsageRecordsInput" />
            <wsdl:output message="intf:replaceUsageRecordsOutput" />
            <fault name="RusUserNotAuthorisedFault"
message="intf:RusUserNotAuthorisedFaultMessage"/>
            <fault name="RusProcessingFault"
message="intf:RusProcessingFaultMessage"/>
        </wsdl:operation>
    </wsdl:portType>
</wsdl:definitions>
```

13.2.2 resource-usage-service.wsdl

```
<definitions name="resourceusage"
    targetNamespace="http://www.gridforum.org/2005/rus-wg/service"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://www.gridforum.org/2005/rus-wg/service"
    xmlns:porttype="http://www.gridforum.org/2005/rus-wg/service"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" >

    <import location="./resource-usage-porttype.wsdl"
        namespace="http://www.gridforum.org/2005/rus-wg/service"/>

    <binding name="ResourceUsagePortTypeSOAPBinding"
type="porttype:ResourceUsagePortType">
        <documentation>SOAP Binding for the
ResourceUsagePortType</documentation>
        <soap:binding style="document"
            transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="listMandatoryUsageRecordElements">
            <soap:operation

soapAction="ResourceUsagePortType#listMandatoryUsageRecordElements"/>
          <input>
            <soap:body use="literal"/>
          </input>
            <output>
                  <soap:body use="literal"/>
            </output>
            <fault name="RusUserNotAuthorisedFault">
                  <soap:fault name="RusUserNotAuthorisedFault"
use="literal"/>
            </fault>
            <fault name="RusProcessingFault">
                  <soap:fault name="RusProcessingFault"
use="literal"/>
            </fault>
        </operation>
<operation name="insertUsageRecords">
            <soap:operation

soapAction="ResourceUsagePortType#insertUsageRecords"/>
            <input>
                  <soap:body use="literal"  />
            </input>
            <output>
                  <soap:body use="literal"  />
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
            </output>
            <fault name="RusUserNotAuthorisedFault">
                  <soap:fault name="RusUserNotAuthorisedFault"
use="literal"/>
            </fault>
            <fault name="RusProcessingFault">
                  <soap:fault name="RusProcessingFault"
use="literal"/>
            </fault>
        </operation>
        <operation name="extractRUSUsageRecordsByGlobalJobID">
            <soap:operation

soapAction="ResourceUsagePortType#extractRUSUsageRecordsByGlobalJobID
"/>
            <input>
                  <soap:body use="literal"  />
            </input>
            <output>
                  <soap:body use="literal"  />
            </output>
            <fault name="RusUserNotAuthorisedFault">
                  <soap:fault name="RusUserNotAuthorisedFault"
use="literal"/>
            </fault>
            <fault name="RusProcessingFault">
                  <soap:fault name="RusProcessingFault"
use="literal"/>
            </fault>
        </operation>
        <operation name="extractRUSUsageRecordsByGlobalUserID">
            <soap:operation

soapAction="ResourceUsagePortType#extractRUSUsageRecordsByGlobalUserI
D"/>
            <input>
                  <soap:body use="literal"  />
            </input>
            <output>
                  <soap:body use="literal"  />
            </output>
            <fault name="RusUserNotAuthorisedFault">
                  <soap:fault name="RusUserNotAuthorisedFault"
use="literal"/>
            </fault>
            <fault name="RusProcessingFault">
                  <soap:fault name="RusProcessingFault"
use="literal"/>
            </fault>
        </operation>
        <operation name="extractRUSUsageRecordsByMachineName">
            <soap:operation

soapAction="ResourceUsagePortType#extractRUSUsageRecordsByMachineName
"/>
            <input>
                  <soap:body use="literal"  />
            </input>
            <output>
                  <soap:body use="literal"  />
            </output>
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
            <fault name="RusUserNotAuthorisedFault">
                  <soap:fault name="RusUserNotAuthorisedFault"
use="literal"/>
            </fault>
            <fault name="RusProcessingFault">
                  <soap:fault name="RusProcessingFault"
use="literal"/>
            </fault>
        </operation>
        <operation name="extractRUSUsageRecordsBySubmitHost">
            <soap:operation

soapAction="ResourceUsagePortType#extractRUSUsageRecordsBySubmitHost"
/>
            <input>
                  <soap:body use="literal"  />
            </input>
            <output>
                  <soap:body use="literal"  />
            </output>
            <fault name="RusUserNotAuthorisedFault">
                  <soap:fault name="RusUserNotAuthorisedFault"
use="literal"/>
            </fault>
            <fault name="RusProcessingFault">
                  <soap:fault name="RusProcessingFault"
use="literal"/>
            </fault>
        </operation>
        <operation name="extractRecords">
            <soap:operation
                soapAction="ResourceUsagePortType#extractRecords"/>
            <input>
                  <soap:body use="literal"  />
            </input>
            <output>
                  <soap:body use="literal"  />
            </output>
            <fault name="RusUserNotAuthorisedFault">
                  <soap:fault name="RusUserNotAuthorisedFault"
use="literal"/>
            </fault>
            <fault name="RusProcessingFault">
                  <soap:fault name="RusProcessingFault"
use="literal"/>
            </fault>
        </operation>
        <operation name="extractRUSRecordIds">
            <soap:operation

soapAction="ResourceUsagePortType#extractRUSRecordIds"/>
            <input>
                  <soap:body use="literal"  />
            </input>
            <output>
                  <soap:body use="literal"  />
            </output>
            <fault name="RusUserNotAuthorisedFault">
                  <soap:fault name="RusUserNotAuthorisedFault"
use="literal"/>
            </fault>
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
            <fault name="RusProcessingFault">
                  <soap:fault name="RusProcessingFault"
use="literal"/>
            </fault>
        </operation>
        <operation name="extractSpecificRUSRecords">
            <soap:operation

soapAction="ResourceUsagePortType#extractSpecificRUSRecords"/>
            <input>
                  <soap:body use="literal"  />
            </input>
            <output>
                  <soap:body use="literal"  />
            </output>
            <fault name="RusUserNotAuthorisedFault">
                  <soap:fault name="RusUserNotAuthorisedFault"
use="literal"/>
            </fault>
            <fault name="RusProcessingFault">
                  <soap:fault name="RusProcessingFault"
use="literal"/>
            </fault>
        </operation>
        <operation name="deleteRecords">
            <soap:operation
                soapAction="ResourceUsagePortType#deleteRecords"/>
            <input>
                  <soap:body use="literal"  />
            </input>
            <output>
                  <soap:body use="literal"  />
            </output>
            <fault name="RusUserNotAuthorisedFault">
                  <soap:fault name="RusUserNotAuthorisedFault"
use="literal"/>
            </fault>
            <fault name="RusProcessingFault">
                  <soap:fault name="RusProcessingFault"
use="literal"/>
            </fault>
        </operation>
        <operation name="deleteSpecificRecords">
            <soap:operation

soapAction="ResourceUsagePortType#deleteSpecificRecords"/>
            <input>
                  <soap:body use="literal"  />
            </input>
            <output>
                  <soap:body use="literal"  />
            </output>
            <fault name="RusUserNotAuthorisedFault">
                  <soap:fault name="RusUserNotAuthorisedFault"
use="literal"/>
            </fault>
            <fault name="RusProcessingFault">
                  <soap:fault name="RusProcessingFault"
use="literal"/>
            </fault>
        </operation>
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
        <operation name="incrementUsageRecordPart">
            <soap:operation 

soapAction="ResourceUsagePortType#incrementUsageRecordPart"/>
            <input>
                  <soap:body use="literal"  />
            </input>
            <output>
                  <soap:body use="literal"  />
            </output>
            <fault name="RusUserNotAuthorisedFault">
                  <soap:fault name="RusUserNotAuthorisedFault" 
use="literal"/>
            </fault>
            <fault name="RusProcessingFault">
                  <soap:fault name="RusProcessingFault" 
use="literal"/>
            </fault>
        </operation>
        <operation name="modifyUsageRecordPart">
            <soap:operation 

soapAction="ResourceUsagePortType#modifyUsageRecordPart"/>
            <input>
                  <soap:body use="literal"  />
            </input>
            <output>
                  <soap:body use="literal"  />
            </output>
            <fault name="RusUserNotAuthorisedFault">
                  <soap:fault name="RusUserNotAuthorisedFault" 
use="literal"/>
            </fault>
            <fault name="RusProcessingFault">
                  <soap:fault name="RusProcessingFault" 
use="literal"/>
            </fault>
            <fault name="RusInputFault">
                  <soap:fault name="RusInputFault" use="literal"/>
            </fault>
        </operation>
        <operation name="replaceUsageRecords">
            <soap:operation 

soapAction="ResourceUsagePortType#replaceUsageRecords"/>
            <input>
                  <soap:body use="literal"  />
            </input>
            <output>
                  <soap:body use="literal"  />
            </output>
             <fault name="RusUserNotAuthorisedFault">
                  <soap:fault name="RusUserNotAuthorisedFault" 
use="literal"/>
            </fault>
            <fault name="RusProcessingFault">
                  <soap:fault name="RusProcessingFault" 
use="literal"/>
            </fault>
       </operation>
    </binding>
```

Joel Replogle 10/4/05 5:03 PM
Deleted: 19/08/2005

```
    <service name="ResourceUsageService">
        <port name="ResourceUsagePortTypeSOAPPort"
            binding="tns:ResourceUsagePortTypeSOAPBinding">
            <soap:address location="http://localhost"/>
        </port>
    </service>
</definitions>
```