



NextGRID Accounting and Billing Profile

Editors:	Justin Ferris	IT Innovation

Date	Author	Comments	Version	Status
07/01/2008	Justin Ferris	Initial Version	V0.1	Draft
28/02/2008	Justin Ferris	Reviewers comments addressed	V0.2	Draft
05/03/2008	Justin Ferris	Change tracking and comments removed	V1.0	Final

1	INTRODUCTION	3
1.1	Profile Overview	3
1.2	Relationships to Other Generalised Specifications	3
1.3	Notational Conventions	3
1.4	Profile Identification and Versioning	4
2	PROFILE CONFORMANCE	4
2.1	Conformance Targets	4
2.2	Claiming Conformance	4
3	REPORTING AND QUERYING RESOURCE USAGE	5
3.1	Reporting Usage	5
3.2	Querying Usage	7
3.3	Addressing	9
3.4	Micro Charges and Billing	9
3.5	WSDL for Usage Reporting and Query	10
4	REFERENCES	14

1 Introduction

This document defines the NextGRID Accounting and Billing Profile 1.0 (hereafter, "this Profile"). It consists of a set of Web services specifications, along with clarifications, refinements, interpretations and amplifications of those specifications that promote interoperability among implementations of those specifications within the context of the NextGRID architecture.

This Profile forms part of the NextGRID Generalized Specifications, which aims at capturing NextGRID architectural concepts in a set of profiles that may be composed together. These profiles are specified in such a way that they could be implemented in terms of other well known specifications. While overall consistency is achieved at the conceptual level, and captured through the motivating Use Cases accompanying each Generalized Specification, system implementations, which may be based on other specifications, may not be consistent with this profile. Thus, each profile defines a realisation of the underlying concept that can be implemented. However, implementers of the NextGRID architecture may need to support multiple underlying specifications.

Section 1 introduces this Profile and its relationships with other NextGRID Generalized Specifications.

Section 2, "Profile Conformance", explains what it means to be conformant to this Profile.

Each subsequent section addresses a component of this Profile.

1.1 Profile Overview

This Profile is concerned with reporting and querying resource usage under the context of a Service Level Agreement (SLA), in line with the concepts of the NextGRID project [1]. A component implementation that uses Accounting and Billing in a manner conformant with the Profile may be said to be an "implementation of the NextGRID Accounting and Billing Profile 1.0".

The primary issues addressed in this Profile are as follows:

- *Reporting and aggregation of resource usage under the context of an SLA.* This Profile defines Web service operations for reporting resource use for an SLA [2].
- *Query of resource usage for an SLA.* This Profile defines Web service operations for querying the resource usage that have been reported for an SLA [2].

1.2 Relationships to Other Generalised Specifications

This Profile extends the NextGRID Basic Profile [3] to define compliant Web service operations for reporting and querying resource usage under the context of an SLA.

1.3 Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [4].

Normative statements of requirements in this Profile are presented in the manner detailed in the Conformance Requirements section of WS-I Basic Profile 1.1 [5].

Both requirement statements and extensibility statements can be considered namespace-qualified.

This Profile uses a number of namespace prefixes throughout; their associated URIs are listed below. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

Table 1 Namespaces used by NextGRID Accounting and Billing Profile 1.0

Prefix	Namespace
wSDL	http://schemas.xmlsoap.org/wSDL/
tns	http://www.nextgrid.org/accounting-and-billing/v1
xsd	http://www.w3.org/2001/XMLSchema

1.4 Profile Identification and Versioning

Profile identification and versioning uses the style described in WS-I Basic Profile 1.1 [5] and abides by the normative descriptions contained therein. The name of this Profile is “NextGRID Accounting and Billing Profile” and the version number is “1.0”.

2 Profile Conformance

Conformance to this Profile is defined normatively in WS-I Basic Profile 1.1 [5]. This Profile abides by those definitions.

2.1 Conformance Targets

The following conformance targets are used in this Profile:

- **REPORTING-OPERATIONS** – Web service operations for reporting resource usage for an SLA.
- **QUERY-OPERATIONS** – Web service operations for querying usage information aggregated under an SLA.
- **INSTANCE** – software that implements a wsdl:port (from WS-I Basic Profile 1.1, without “bindingTemplate” from the namespace urn:uddi-org:api_v2) and implements the REPORTING-OPERATIONS and QUERY-OPERATIONS defined in this Profile.
- **OPERATIONAL-MANAGEMENT-COMPONENT** – software that reports resource usage using the REPORTING-OPERATIONS.
- **CUSTOMER-COMPONENT** – software component that consumes usage information aggregated under an SLA using the QUERY-OPERATIONS.
- **ENDPOINTREFERENCE** – the serialization of the wsa:EndpointReference element and its content (from OGSA WSRF Basic Profile)

2.2 Claiming Conformance

Claims of conformance to this Profile and are the same as normatively described in WS-I Basic Profile 1.1 [5].

The conformance claim URI for this Profile is “<http://www.nextgrid.org/accounting-and-billing/v1>”.

3 Reporting and Querying Resource Usage

This Profile defines an interface comprising of REPORTING-OPERATIONS and QUERY-OPERATIONS that are Web service operations for reporting and querying, respectively, resource usage for an SLA.

Informally (in Java-like syntax), the set of operations in the interface are given below:

```
void reportAbsoluteUsageAtInstantByMetric(long instant, string metric,
    string msg, decimal value)

void reportDeltaUsageAtInstantByMetric(long instant, string metric,
    string msg, decimal value)

decimal getUsageForMetricAtInstant(string metric, long instant)

UsageRecord[] getUsageRecordsForMetricInPeriod(string metric,
    long startInstant, long endInstant)
```

The first two methods, `reportAbsoluteUsageAtInstantByMetric` and `reportDeltaUsageAtInstantByMetric` are used to report resource usage for an SLA at an instant in time for a particular metric. Usage can be reported as an absolute value using the first of these methods or as a delta value using the latter. Reporting delta usage allows OPERATIONAL-MANAGEMENT-COMPONENTs to report changes in usage, instead of an absolute value for resource usage at an instant in time. In some cases this will be more convenient and help to reduce the amount of state management and processing that OPERATIONAL-MANAGEMENT-COMPONENTs must perform when monitoring resource use.

The final two methods, `getUsageForMetricAtInstant` and `getUsageRecordsForMetricInPeriod` are used to query previously reported resource usage, for an SLA and a metric. `getUsageForMetricAtInstant` is used to obtain the usage value for a metric at an instant in time, whereas `getUsageRecordsForMetricInPeriod` is used to obtain a view of the usage reported for a particular SLA and metric in a time range, by returning a set of `UsageRecords`.

Each of these operations, whether for reporting or querying usage, must be invoked in the context of an SLA (Section 3.3).

Note that the full WSDL is provided later in this document, after first describing in more detail each Web service operation.

3.1 Reporting Usage

Resource usage data are reported by an OPERATIONAL-MANAGEMENT-COMPONENT in the context of an SLA (Section 3.3) and a metric at an instant in time. Here, a metric is defined simply as a characteristic that can be measured.

Usage can be reported by an OPERATIONAL-MANAGEMENT-COMPONENT as an absolute value or as a delta value. Absolute usage values are straightforward and, intuitively, they must be interpreted by the INSTANCE as being the usage value at the instant that the report is made. For example, an OPERATIONAL-MANAGEMENT-COMPONENT could report that 2 CPUs are being used under a particular SLA at an instant in time.

In contrast, delta usage values can be reported by an OPERATIONAL-MANAGEMENT-COMPONENT to report changes in usage instead of absolute usage values. An INSTANCE MUST interpret a delta usage value as a change in resource usage for the given metric and time instant. For example, an OPERATIONAL-MANAGEMENT-COMPONENT could report that 1 more CPU is being used for a particular SLA and instant in time.

The XSD schema for the SOAP message request for reporting absolute usage is provided below.

```
<element name="reportAbsoluteUsageAtInstantByMetric">
  <complexType>
    <sequence>
      <element name="instant" type="xsd:long"/>
      <element name="metric" type="xsd:string"/>
      <element name="msg" type="xsd:string"/>
      <element name="value" type="xsd:decimal"/>
    </sequence>
  </complexType>
</element>
```

The reportAbsoluteUsageAtInstantByMetric element contains the following sub-elements:

- instant - the current time in milliseconds, calculated as the difference between the current time and midnight, January 1, 1970 UTC.
- metric - a string defining the characteristic that was measured.
- msg - an arbitrary string that can be used to record a human readable message.
- value - a decimal representing the absolute value of resource in use.

The XSD schema for the SOAP message request for reporting delta usage is very similar to the message for reporting absolute usage and is provided below.

```
<element name="reportDeltaUsageAtInstantByMetric">
  <complexType>
    <sequence>
      <element name="instant" type="xsd:long"/>
      <element name="metric" type="xsd:string"/>
      <element name="msg" type="xsd:string"/>
      <element name="value" type="xsd:decimal"/>
    </sequence>
  </complexType>
</element>
```

The reportDeltaUsageAtInstantByMetric element contains the following sub-elements:

- instant - the current time in milliseconds, calculated as the difference between the current time and midnight, January 1, 1970 UTC.
- metric - a string defining the characteristic that was measured.
- msg - an arbitrary string that can be used to record a human readable message.
- value - a decimal representing a change in resource use for the metric at instant.

3.2 Querying Usage

A CUSTOMER-COMPONENT queries resource usage in the context of an SLA (Section 3.3) and a metric. There are two operations for querying usage.

The first operation is used to obtain the usage value for a metric at an instant in time. The XSD schema for the SOAP message request for getting usage for a metric at an instant in time is given below.

```
<element name="getUsageForMetricAtInstant">
  <complexType>
    <sequence>
      <element name="metric" type="xsd:string"/>
      <element name="instant" type="xsd:long"/>
    </sequence>
  </complexType>
</element>
```

The `getUsageForMetricAtInstant` element has the following sub elements:

- `metric` - a string defining the characteristic that was measured.
- `instant` - the current time in milliseconds, calculated as the difference between the current time and midnight, January 1, 1970 UTC.

The INSTANCE responds with the absolute value for usage for the metric and the instant specified in the request. The XSD schema for the response message is given below.

```
<element name="getUsageForMetricAtInstantResponse">
  <complexType>
    <sequence>
      <element name="getUsageForMetricAtInstantReturn" type="xsd:decimal"/>
    </sequence>
  </complexType>
</element>
```

The `getUsageForMetricAtInstantResponse` element has a single sub element:

- `getUsageForMetricAtInstantReturn` – the usage, as a decimal value, for the metric at the instant in time specified in the request.

The second operation for querying resource usage allows a CUSTOMER-COMPONENT to obtain a view over reported usage for a metric and SLA over a period of time. The INSTANCE responds with a data structure that represents a view over reported usage.

The XSD schema for the SOAP message for getting a view of reported usage for a metric over time period is given below.

```
<element name="getUsageRecordsForMetricInPeriod">
  <complexType>
    <sequence>
      <element name="metric" type="xsd:string"/>
      <element name="startInstant" type="xsd:long"/>
      <element name="endInstant" type="xsd:long"/>
    </sequence>
  </complexType>
```

```
</element>
```

The `getUsageRecordsForMetricInPeriod` element has the following sub elements:

- `metric` - a string defining the characteristic that was measured.
- `startInstant` - the time in milliseconds, calculated as the difference between the current time and midnight, January 1, 1970 UTC, for the start of the period of interest.
- `endInstant` - the time in milliseconds, calculated as the difference between the current time and midnight, January 1, 1970 UTC, for the end of the period of interest.

The INSTANCE responds with a response message containing a data structure that is a view over the usage reported for the metric in the time period of interest. The XSD schema for the response message is given below.

```
<element name="getUsageRecordsForMetricInPeriodResponse">
  <complexType>
    <sequence>
      <element maxOccurs="unbounded" name="getUsageRecordsForMetricInPeriodReturn"
        type="tns:UsageRecord"/>
    </sequence>
  </complexType>
</element>

<complexType name="UsageRecord">
  <sequence>
    <element name="absValue" nillable="true" type="xsd:decimal"/>
    <element name="absValueSet" type="xsd:boolean"/>
    <element name="deltaValue" nillable="true" type="xsd:decimal"/>
    <element name="id" nillable="true" type="xsd:long"/>
    <element name="instant" nillable="true" type="xsd:long"/>
    <element name="message" nillable="true" type="xsd:string"/>
    <element name="metric" nillable="true" type="xsd:string"/>
    <element name="slaId" nillable="true" type="xsd:string"/>
  </sequence>
</complexType>
```

The `getUsageRecordsForMetricInPeriodResponse` element is an array of elements of type `tns:UsageRecord`. Each element of type `UsageRecord` in the array has the following sub elements:

- `absValue` – the absolute component of reported usage as a decimal.
- `absValueSet` – a boolean indicating whether the `UsageRecord` has an absolute usage component.
- `deltaValue` – the delta component of reported usage as a decimal.
- `id` – an identifier for the `UsageRecord` that is unique within the scope of the INSTANCE.
- `instant` - the current time in milliseconds, calculated as the difference between the current time and midnight, January 1, 1970 UTC.
- `message` - string containing a human readable message.
- `metric` - a string defining the characteristic that was measured.
- `slaId` – a unique identifier for the SLA for which the usage was reported.

Note that the INSTANCE responds with a `UsageRecord` for each instant in time within the period of interest for which usage has been reported for the metric under the SLA. The bounds

of the period of interest specified by startInstant and endInstant in the CUSTOMER-COMPONENT's request message are inclusive. Therefore, UsageRecords with an instant that satisfies the condition startInstant <= instant <= endInstant are included in the response.

Each UsageRecord has two components, one for absolute usage, and the other for delta usage. The absolute usage component, represented by absValue, contains a decimal value for the absolute usage value (if any) at instant. The absValueSet boolean can be used by the CUSTOMER-COMPONENT to determine whether the absValue is meaningful. If absValueSet is true, the absValue reflects the usage reported at instant, otherwise its value is meaningless. The absValueSet flag is intended to prevent the erroneous case of a CUSTOMER-COMPONENT interpreting a null or otherwise unset value of absValue as meaning zero. The value of absValue for a UsageRecord will depend on the last report of absolute usage for the metric at the instant, as reported by the CUSTOMER-COMPONENT in the context of the SLA. Consequently, it is usually an error for a CUSTOMER-COMPONENT to make more than one report of absolute usage for the same SLA, metric and instant, as the previous value will effectively be overwritten. In contrast, the deltaValue component of a UsageRecord corresponds to the sum of delta usage reports made by a CUSTOMER-COMPONENT for the SLA, metric and instant. Delta usage reports are simply aggregated by summing their values to generate a new delta that encompasses all previous reports of delta usage for the same SLA, metric and instant.

Given these rules a CUSTOMER-COMPONENT can process the array of UsageRecords returned by the INSTANCE and, provided the usage value is known at the start of the period of interest (e.g. by the CUSTOMER-COMPONENT having previously called getUsageForMetricAtInstant, or because the first UsageRecord contains an absolute component), determine the usage value for the SLA and metric for each instant represented by a UsageRecord within the period of interest specified in the query. Consequently, a CUSTOMER-COMPONENT would have sufficient information to plot a graph of usage for a metric and SLA over time, for example.

3.3 Addressing

Request messages sent by a CUSTOMER-COMPONENT or an OPERATIONAL-MANAGEMENT-COMPONENT to an INSTANCE for REPORTING-OPERATIONS and QUERY-OPERATIONS must be addressed to the SLA WS-Resource representing the SLA to which the usage relates. Consequently, the SLA WS-Resource aggregates usage information for the SLA that it represents, and provides the REPORTING-OPERATIONS and QUERY-OPERATIONS described in this Profile. Addressing of messages using an ENDPPOINTREFERENCE must be done according to the mechanisms mandated in the NextGRID Basic Profile [3].

3.4 Micro Charges and Billing

The REPORTING-OPERATIONS and QUERY-OPERATIONS described above can be used for simple accounting and billing.

Micro charges for resource use can be aggregated under an SLA in a straightforward way using the REPORTING-OPERATIONS. Similarly, a CUSTOMER-COMPONENT can retrieve bills for a period of interest using the QUERY-OPERATIONS.

In order to use the usage reporting and query operations in this way, one simply has to define a metric that represents charges for resource use and then use the operations in the normal way and as specified in the previous sections of this Profile.

For example, the metric <http://nextgrid.org/metric/charges> (note that this is simply an example string that could be used to define a metric for monetary charges) could be used to represent monetary charges in a particular currency for resource use. Micro charges for resource use under an SLA can be made by a CUSTOMER-COMPONENT using this metric as context and the REPORTING-OPERATIONS, as described above, to report delta usage. Similarly, a CUSTOMER-COMPONENT can retrieve bills for resource use in a particular period of time using the QUERY-OPERATIONS and this metric also.

3.5 WSDL for Usage Reporting and Query

The full WSDL mark-up for the REPORTING-OPERATIONS and QUERY-OPERATIONS described above is included below for completeness.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://www.nextgrid.org/accounting-and-billing/v1"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.nextgrid.org/accounting-and-billing/v1">
  <wsdl:types>
    <schema elementFormDefault="qualified"
      targetNamespace="http://www.nextgrid.org/accounting-and-billing/v1"
      xmlns="http://www.w3.org/2001/XMLSchema">

      <element name="reportAbsoluteUsageAtInstantByMetric">
        <complexType>
          <sequence>
            <element name="instant" type="xsd:long"/>
            <element name="metric" type="xsd:string"/>
            <element name="msg" type="xsd:string"/>
            <element name="value" type="xsd:decimal"/>
          </sequence>
        </complexType>
      </element>

      <element name="reportAbsoluteUsageAtInstantByMetricResponse">
        <complexType/>
      </element>

      <element name="reportDeltaUsageAtInstantByMetric">
        <complexType>
          <sequence>
            <element name="instant" type="xsd:long"/>
            <element name="metric" type="xsd:string"/>
            <element name="msg" type="xsd:string"/>
            <element name="value" type="xsd:decimal"/>
          </sequence>
        </complexType>
      </element>

      <element name="reportDeltaUsageAtInstantByMetricResponse">
        <complexType/>
      </element>
    </schema>
  </wsdl:types>
</wsdl:definitions>
```

```

<element name="getUsageForMetricAtInstant">
  <complexType>
    <sequence>
      <element name="metric" type="xsd:string"/>
      <element name="instant" type="xsd:long"/>
    </sequence>
  </complexType>
</element>

<element name="getUsageForMetricAtInstantResponse">
  <complexType>
    <sequence>
      <element name="getUsageForMetricAtInstantReturn" type="xsd:decimal"/>
    </sequence>
  </complexType>
</element>

<element name="getUsageRecordsForMetricInPeriod">
  <complexType>
    <sequence>
      <element name="metric" type="xsd:string"/>
      <element name="startInstant" type="xsd:long"/>
      <element name="endInstant" type="xsd:long"/>
    </sequence>
  </complexType>
</element>

<element name="getUsageRecordsForMetricInPeriodResponse">
  <complexType>
    <sequence>
      <element maxOccurs="unbounded" name="getUsageRecordsForMetricInPeriodReturn"
        type="tns:UsageRecord"/>
    </sequence>
  </complexType>
</element>

<complexType name="UsageRecord">
  <sequence>
    <element name="absValue" nillable="true" type="xsd:decimal"/>
    <element name="absValueSet" type="xsd:boolean"/>
    <element name="deltaValue" nillable="true" type="xsd:decimal"/>
    <element name="id" nillable="true" type="xsd:long"/>
    <element name="instant" nillable="true" type="xsd:long"/>
    <element name="message" nillable="true" type="xsd:string"/>
    <element name="metric" nillable="true" type="xsd:string"/>
    <element name="slaId" nillable="true" type="xsd:string"/>
  </sequence>
</complexType>

</schema>
</wsdl:types>

<wsdl:message name="reportAbsoluteUsageAtInstantByMetricRequest">
  <wsdl:part

```

```

        element="tns:reportAbsoluteUsageAtInstantByMetric"
        name="parameters" />
</wsdl:message>

<wsdl:message name="reportAbsoluteUsageAtInstantByMetricResponse">
  <wsdl:part
    element="tns:reportAbsoluteUsageAtInstantByMetricResponse"
    name="parameters" />
</wsdl:message>

<wsdl:message name="reportDeltaUsageAtInstantByMetricRequest">
  <wsdl:part
    element="tns:reportDeltaUsageAtInstantByMetric"
    name="parameters" />
</wsdl:message>

<wsdl:message name="reportDeltaUsageAtInstantByMetricResponse">
  <wsdl:part
    element="tns:reportDeltaUsageAtInstantByMetricResponse"
    name="parameters" />
</wsdl:message>

<wsdl:message name="getUsageForMetricAtInstantRequest">
  <wsdl:part element="tns:getUsageForMetricAtInstant" name="parameters" />
</wsdl:message>

<wsdl:message name="getUsageForMetricAtInstantResponse">
  <wsdl:part element="tns:getUsageForMetricAtInstantResponse" name="parameters" />
</wsdl:message>

<wsdl:message name="getUsageRecordsForMetricInPeriodRequest">
  <wsdl:part element="tns:getUsageRecordsForMetricInPeriod" name="parameters" />
</wsdl:message>

<wsdl:message name="getUsageRecordsForMetricInPeriodResponse">
  <wsdl:part
    element="tns:getUsageRecordsForMetricInPeriodResponse" name="parameters" />
</wsdl:message>

<wsdl:portType name="UsageReportingAndQuery">
  <wsdl:operation name="reportAbsoluteUsageAtInstantByMetric">
    <wsdl:input
      message="tns:reportAbsoluteUsageAtInstantByMetricRequest"
      name="reportAbsoluteUsageAtInstantByMetricRequest" />
    <wsdl:output
      message="tns:reportAbsoluteUsageAtInstantByMetricResponse"
      name="reportAbsoluteUsageAtInstantByMetricResponse" />
  </wsdl:operation>

  <wsdl:operation name="reportDeltaUsageAtInstantByMetric">
    <wsdl:input
      message="tns:reportDeltaUsageAtInstantByMetricRequest"
      name="reportDeltaUsageAtInstantByMetricRequest" />
    <wsdl:output
      message="tns:reportDeltaUsageAtInstantByMetricResponse"

```

```

        name="reportDeltaUsageAtInstantByMetricResponse"/>
</wsdl:operation>

<wsdl:operation name="getUsageForMetricAtInstant">
  <wsdl:input message="tns:getUsageForMetricAtInstantRequest"
    name="getUsageForMetricAtInstantRequest"/>
  <wsdl:output message="tns:getUsageForMetricAtInstantResponse"
    name="getUsageForMetricAtInstantResponse"/>
</wsdl:operation>

<wsdl:operation name="getUsageRecordsForMetricInPeriod">
  <wsdl:input message="tns:getUsageRecordsForMetricInPeriodRequest"
    name="getUsageRecordsForMetricInPeriodRequest"/>
  <wsdl:output message="tns:getUsageRecordsForMetricInPeriodResponse"
    name="getUsageRecordsForMetricInPeriodResponse"/>
</wsdl:operation>

</wsdl:portType>

<wsdl:binding name="UsageReportingAndQueryBinding" type="tns:UsageReportingAndQuery">
  <wsdlsoap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>

  <wsdl:operation name="reportAbsoluteUsageAtInstantByMetric">
    <wsdlsoap:operation soapAction="http://www.nextgrid.org/accounting-and-billing/v1/
reportAbsoluteUsageAtInstantByMetric">
      <wsdl:input name="reportAbsoluteUsageAtInstantByMetricRequest">
        <wsdlsoap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="reportAbsoluteUsageAtInstantByMetricResponse">
        <wsdlsoap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>

    <wsdl:operation name="reportDeltaUsageAtInstantByMetric">
      <wsdlsoap:operation soapAction="http://www.nextgrid.org/accounting-and-
billing/v1/reportDeltaUsageAtInstantByMetric">
        <wsdl:input name="reportDeltaUsageAtInstantByMetricRequest">
          <wsdlsoap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="reportDeltaUsageAtInstantByMetricResponse">
          <wsdlsoap:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>

      <wsdl:operation name="getUsageForMetricAtInstant">
        <wsdlsoap:operation soapAction="http://www.nextgrid.org/accounting-and-
billing/v1/getUsageForMetricAtInstant">
          <wsdl:input name="getUsageForMetricAtInstantRequest">
            <wsdlsoap:body use="literal"/>
          </wsdl:input>
          <wsdl:output name="getUsageForMetricAtInstantResponse">
            <wsdlsoap:body use="literal"/>
          </wsdl:output>
        </wsdl:operation>

```

```
<wsdl:operation name="getUsageRecordsForMetricInPeriod">
  <wsdlsoap:operation soapAction="http://www.nextgrid.org/accounting-and-
  billing/v1/getUsageRecordsForMetricInPeriod"/>
  <wsdl:input name="getUsageRecordsForMetricInPeriodRequest">
    <wsdlsoap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="getUsageRecordsForMetricInPeriodResponse">
    <wsdlsoap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

</wsdl:binding>
</wsdl:definitions>
```

4 References

- [1] D. Snelling and M. Fisher and A. Basermann and F. Wray and P. Wieder and M. Surridge (eds.), NextGRID White Paper V5, NextGRID Project, July 25, 2005, http://www.nextgrid.org/download/publications/NextGRID_Architecture_White_Paper.pdf
- [2] J. Ferris (ed.), NextGRID Accounting and Billing Use Cases, V1.0, NextGRID Project, February 2008, http://www.nextgrid.org/GS/management_systems/operational_management/NextGRID_accounting_and_billing_usecases.pdf
- [3] V. Li and D. Snelling (eds.), NextGRID Basic Profile, V1.0, NextGRID Project, September 2006, http://www.nextgrid.org/GS/management_systems/basic/NextGRID_basic_profile.pdf
- [4] S. Bradner (ed.): Key words for use in RFCs to Indicate Requirement Levels, The Internet Engineering Task Force Best Current Practice, March 1997. <http://www.ietf.org/rfc/rfc2119>
- [5] K. Ballinger, D. Ehnebuske, C. Ferris, M. Gudgin, C.K. Liu, M. Nottingham, and P. Yendluri (ed.): Basic Profile Version 1.1, Web Services Interoperability Organization Final Material, 24 August 2004. <http://www.wsi.org/Profiles/BasicProfile-1.1.html>