

## Usage Record – Format Recommendation

### Status of This Memo

This memo provides information to the Grid community in the area of usage records and accounting. Distribution is unlimited.

### Copyright Notice

Copyright © Open Grid Forum (2006-2007). All Rights Reserved.

### **Abstract**

For resources to be shared, sites must be able to exchange basic accounting and usage data in a common format. This format must encompass both job level accounting and aggregate accounting. This document describes a common format with which to exchange basic accounting and usage data over a grid instantiation. This record format is intended to facilitate the sharing of usage information among grid sites, particularly in the area of job accounting. This document describes the requirements in natural language form for a Usage Record standard. The usage record is then represented in an XML format.

This document does not address how these records should be used, nor does it attempt to dictate the format in which the accounting records are stored at a local site; instead, it defines a common exchange format. Furthermore, nothing is said regarding the communications mechanisms employed to exchange the records, i.e. transport layer, framing, authentication, integrity, etc.

### **Table of Contents**

Usage Record – Format Recommendation.....	1
Abstract.....	1
Table of Contents.....	1
1. Introduction.....	4
2. Conventions Used in this Document.....	5
2.1 Key Words.....	5
2.2 Meta Properties.....	5
2.3 Supported Data Types.....	6
3. Base Properties.....	7
3.1 RecordIdentity.....	7
3.2 GlobalJobId.....	7
3.3 LocalJobId.....	7
3.4 ProcessId.....	7
3.5 LocalUserId:.....	8
3.6 GlobalUsername.....	8
3.7 JobName.....	8

3.8	Charge .....	8
3.9	Status .....	9
3.10	WallDuration .....	9
3.11	CpuDuration .....	9
3.12	EndTime .....	9
3.14	MachineName .....	10
3.15	Host .....	10
3.16	SubmitHost .....	10
3.17	Queue .....	10
3.18	ProjectName .....	11
4.	Differentiated Properties .....	11
4.1	Network .....	11
4.2	Disk .....	11
4.3	Memory .....	11
4.4	Swap .....	12
4.5	NodeCount .....	12
4.6	Processors .....	12
4.7	TimeDuration .....	13
4.8	TimeInstant .....	13
4.9	ServiceLevel .....	13
4.10	Extension .....	13
5.	Job Level Accounting .....	13
6.	Aggregate Accounting .....	14
7.	XML Overview .....	14
7.1	Encoding .....	14
7.2	XML Conventions .....	14
7.3	Schema .....	16
8.	Global Element Definitions .....	17
8.1	UsageRecord Element .....	18
8.2	JobUsageRecord Element .....	18
8.3	UsageRecords Element .....	18
9.	Global Attribute and Attribute Group Definitions .....	18
9.1	Description .....	19
9.2	storageUnit .....	19
9.3	phaseUnit .....	20
9.4	intervallicVolume Attribute Group: .....	20
9.5	metric .....	20
9.6	type .....	21
9.7	unit .....	21
10.	Global Type Definitions .....	21
10.1	UsageRecordType Complex Type .....	21
10.2	domainNameType Simple Type .....	23
11.	Usage Record Common Properties .....	23
11.1	RecordIdentity Element .....	23
11.2	JobIdentity .....	25
11.3	UserIdentity .....	25
11.4	JobName .....	27

11.5	Charge .....	27
11.6	Status .....	28
11.7	WallDuration .....	28
11.8	CpuDuration .....	29
11.9	EndTime .....	29
11.10	StartTime .....	29
11.11	MachineName .....	30
11.12	Host .....	30
11.13	SubmitHost .....	30
11.14	Queue .....	31
11.15	ProjectName .....	31
12.	Differentiated Properties .....	31
12.1	Network .....	32
12.2	Disk: .....	32
12.3	Memory .....	33
12.4	Swap .....	33
12.5	NodeCount .....	34
12.6	Processors .....	34
12.7	TimeDuration .....	35
12.8	TimeInstant .....	35
12.9	ServiceLevel .....	36
13.	Extension Framework .....	36
13.1	Type Definitions .....	37
13.2	Element Definitions .....	37
14.	Example Usage Records .....	38
14.1	Sample Usage Record 1 .....	38
14.2	Sample Usage Record 2 .....	39
15.	Security Considerations .....	40
16.	Document Credits .....	41
16.1	Author Information .....	41
16.2	Contributor Information .....	41
17.	Intellectual Property Statement .....	42
18.	Full Copyright Notice .....	42
19.	References .....	43
	Appendix A – Current Practices Survey .....	44
A-1	Survey Background .....	44
A-2	Survey Results .....	44
A-3	Datatypes .....	49
A-4	Common Usage Record Fields .....	50
	Appendix B – Common Usage Record Properties .....	51
	Appendix C – Units of Measure for Volume .....	52
	Appendix D – Usage Record XML Schema .....	53

## 1. Introduction

In order for resources to be shared, sites must be able to exchange basic accounting and usage data in a common format. This document focuses on the representation of resource consumption data. The document then goes on to describe an XML-based format for usage records. The record format is intended to be specific enough to facilitate information sharing among grid sites, yet general enough that the usage data can be used for a variety of purposes: traditional usage accounting, charging, service usage monitoring, performance tuning, etc. The purpose of this document is to outline the basic building blocks of the accounting record, and how to properly represent them. All other tangential concerns such as the use, transport mechanism, and security are out of scope for this representation layer.

### 1.1 Context

To comprehend the structure of the schema presented in this document, it is important to understand the context in which this specification has been developed. The specification begins with the premises that

- The fundamental component of a grid is a resource;
- The fundamental consumer of a resource is a job on that resource;
- Jobs may be batch (i.e. queued) or interactive.

Hence, the usage record schema is designed to collect consumption data from individual grid resources at the fundamental, or atomic level.

#### 1.1.1 History

The schema was originally designed around traditional high performance computing practices, which tend towards batch processes on computational components. Leading edge grid and high performance computing sites provided input on their best practices for usage tracking on their resources. From that input, a natural language description of accounting requirements was developed. This description included provisions for non-computational resources and interactive job usage tracking. From the initial best practices survey and follow-up development of the natural language “specification” came the premises for this current specification.

While grids and grid technology have evolved since the initial work was done on this specification, the need for simple job accountability in the form of usage tracking has not changed. Many leading edge grid implementations have adopted and adapted this early usage record format to suite their needs, in both development and production environments. Future revisions to this specification will require a level of backward compatibility or mechanisms for transition so that these early adopters are not left behind.

#### 1.1.2 What This Document Is Not

This document and specification do not attempt to define a comprehensive “grid accounting” standard. As with all accounting implementations, there is no one-size-fits-all solution, that will meet the needs of all grid projects and resource providers. In defining the properties of a brick, this document does not presume to mandate what the house should look like. This document does not address aggregation, summary records, “grid job” records, consolidated records, or anything other than an atomic resource consumption instantiation. Sufficient resource, site, job, and user information is collected to allow for effective and appropriate levels of aggregation, consolidation, and summarization, but the details of how sites implement these features (i.e. what grids do with the atomic data) are beyond the scope of this document.

## 1.2 Format of the Record Specification

- 1.2.1 **Base Properties:** Base properties are those items that all or most sites deem critical for accurately recording the usage on their resources. They include job and user identification, as well as most of the common types of resources that sites need to measure.
- 1.2.2 **Differentiated Properties:** Usage for a particular job may be measured by more than one metric. To accommodate this, the specification allows for differentiated properties, which can occur more than once in a single record.
- 1.2.3 **Extensions:** Provisions have been made in the schema to allow for site and grid-specific extensions. The only requirement for extending the specification is that all resources in the grid agree on the format and semantics of the extension.

Many of the elements include an `<xsd:anyAttribute>` extension point to allow arbitrary XML attributes to be added to the construct without the need for an explicit schema extension. This allows additional fields to be added as needed to supply additional parameters to be used, for example, to track per-host memory on shared nodes. Such extensions **MUST NOT** add local (non-namespace-qualified) XML attributes or XML attributes qualified by a URWG-defined namespace to an element or any element derivation; such attributes are reserved for future maintenance and enhancement of the specification itself.

- 1.2.4 **Aggregation:** Most sites have indicated that job-level aggregation is sufficient for their current needs. Where sites aggregate or decompose to other levels of usage tracking, provisions are available within the schema to accommodate. As with extensions, the parties exchanging aggregated data should agree on the level of aggregation to be reported and reflect that agreement in their usage of this schema.

## 2. Conventions Used in this Document

### 2.1 Key Words

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this specification are to be interpreted as described in IETF RFC 2119 [RFC2119]:

*“they MUST only be used where it is actually required for interoperation or to limit behavior which has potential for causing harm (e.g. limiting retransmission)”*

These key words are thus capitalized when used to unambiguously specify requirements over features and behavior that affect interoperability and security of implementations.

### 2.2 Meta Properties

Meta properties are associated with individual base properties to provide additional information and semantic meaning of the value for a base property. The meta properties outlined below are commonly encountered and should be supported for the indicated base properties.

### 2.2.1 Description

The description provides a mechanism for additional, optional information to be attached to a Usage Record base property. The value of this meta-property MAY provide clues to the semantic context to use while interpreting or examining the value of the owning base property.

### 2.2.2 Units Definition

The following section details the supported units for volume and phase units that apply to the properties that have those units. Some properties MAY have both volume and phase units. For example, megabytes per second. Phase and unit definitions MUST supply both the volume unit, and the phase unit separately.

#### 2.2.2.1 Volume Units

The units for storage and memory use volume units. The legal values for volume units are given in Appendix C. Each value combines a prefix that identifies a multiplier and a suffix that identifies the base quantity. For example, the suffix -B represents volume in bytes and the prefix K- implies a kilo multiplier. Therefore, KB as the designated unit of measure value MUST be interpreted as kilobytes.

#### 2.2.2.2 Phase Units

Phase units represents the duration of time that is relevant to the usage reported. The lexical representation for duration is the [ISO 8601] extended format  $PnYn MnDTnH nMnS$ , where  $nY$  represents the number of years,  $nM$  the number of months,  $nD$  the number of days, 'T' is the date/time separator,  $nH$  the number of hours,  $nM$  the number of minutes and  $nS$  the number of seconds. The number of seconds can include up to 6 decimal digits.

### 2.2.3 Metric

This meta-property identifies the type of measurement used for quantifying the associated resource consumption if there are multiple methods to measure resource usage. As an example, disk usage may be measured as total, average, minimum or maximum usage. However, even if pertinent to the assessed charge, this meta-property does not attempt to differentiate between requested and utilized quantities of resource usage.

### 2.2.4 Time stamps

Time stamps should follow the ISO 8601 standard as well. This include enumerating the time zone, as specified in the standard.

## 2.3 Supported Data Types

2.3.1 **String:** Data of this type has no required restrictions on the length or available characters.

2.3.2 **Integer**

2.3.3 **Positive Integer:** Data of this type must have a value of zero or greater.

2.3.4 **Boolean:** Data of this type may have a value of either true or false.

### 2.3.5 **Float**

2.3.6 **Timestamp**: Data of this type must comply with the UTC time zone format specified in ISO 8601.

2.3.7 **DomainName**: Data of this type must comply with RFC 1034 format for fully qualified domain names. The constraints are a maximum 255 characters long, containing only alphabetic and numeric characters, the “-”, and the “.” characters.

## 3. **Base Properties**

The following is a list of base properties that define common usage record requirements for both job level and aggregate properties. The definitions for the type of data (integer, string) are the base data types, as defined in section §8. Any additional restrictions on the legal values for a specific property are noted within the base property definition. The naming convention and naming style of these properties are implementation specific, and should use the referring name as a guideline for any naming recommendations.

Properties may carry certain meta-properties that can assist in establishing semantic meaning from the actual property value, such as “metric”, “description”, and “units”. Common meta-properties are addressed more fully in section §8.

### 3.1 **RecordIdentity**

A record identity uniquely defines a record in the set of all usage record for the grid implementation.

- This property SHOULD be referred to as recordidentity.
- This property MUST have data of type string.
- This property MUST exist.
- This property MUST be unique.
- Meta-properties
  - Create time of the record MUST be specified.

### 3.2 **GlobalJobId**

The global job identifier as assigned by a metascheduler or federation scheduler.

- This property SHOULD be referred to as globaljobid.
- This property MUST have data of type string.
- This property is optional.
- Meta-properties
  - Description MAY be specified.

### 3.3 **LocalJobId**

The local job identifier as assigned by the batch queue

- This property SHOULD be referred to as localjobid.
- This property MUST have data of type string.
- This property is optional.
- Meta-properties
  - Description MAY be specified.

### 3.4 **ProcessId**

The process id of the jobs (PID).

- This property SHOULD be referred to as processid.
- This property MUST have data of type integer.
- This property is optional.

- Meta-properties
  - Description MAY be specified.

### 3.5 **LocalUserId**

The local identity of the user associated with the resource consumption reported in this Usage Record. This user is often referred to as the requesting user. For example, the value may be the user's login name corresponding to the user's uid in the /etc/passwd file on Unix systems.

- This property SHOULD be referred to as localuserid.
- This property MUST have data of type string.
- This property is optional.

### 3.6 **GlobalUsername**

The global identity of the user associated with the resource consumption reported in this Usage Record. For example, the value may be the distinguished name from the user's certificate.

- This property SHOULD be referred to as globalusername.
- This property MAY have data of type string.
- This property is optional

### 3.7 **JobName**

The job or application name. For example, this could be the name of the executable that ran, or the name of the batch queuing system's name for the job.

- This property SHOULD be referred to as jobname.
- This property MUST contain data of type string.
- This property is optional.
- Meta-properties
  - Description MAY be specified

### 3.8 **Charge**

This property represents the total charge of the job in the system's allocation unit. For example 100, 200, or 3000. The meaning of this charge will be site dependent. The value for this property MAY include premiums or discounts assessed on the actual usage represented within this record. Therefore, the reported charge might not be directly reconstructable from the specific usage reported.

Note that "Charge" does not necessarily refer to a currency-based unit unless that is what members of the grid virtual organization agree to as the definition. If charge denotes a value in currency, standard currency codes should be used to indicate the currency unit being reported.

- This property SHOULD be referred to as charge.
- It MUST contain data of type float.
- This property is optional.
- Meta-properties:
  - Units MAY be specified.
  - Description MAY be specified.
  - Formula MAY be specified that describes how the charge was arrived at. There is no required format for the formula.

### 3.9 **Status**

This property will represent the completion status of the job. For example, this may represent the exit status of an interactive running process or the exit status from the batch queuing system's accounting record. The semantic meaning of status is site dependent.

- This property SHOULD be referred to as status.
- This property MUST contain data of type string.
- This property MUST exist.
- This property MUST support the following values:
  - aborted – A policy or human intervention caused the job to cease execution.
  - completed – The execution completed.
  - failed – Execution halted without external intervention.
  - held – Execution is held at the time this usage record was generated.
  - queued – Execution was queued at the time this usage record was generated.
  - started – Execution started at the time this usage record was generated.
  - suspended – Execution was suspended at the time this usage record was generated.
- This property MAY support other values, as agreed upon within the implementation context.

### 3.10 **WallDuration**

Wall clock time that elapsed while the job was running.

- This property SHOULD be referred to as wallduration.
- This property MUST contain data of type duration.
- This property is optional.
- Meta-properties
  - Description MAY be specified.

### 3.11 **CpuDuration**

CPU time used, summed over all processes in the job.

- This property SHOULD be referred to as cpuduration.
- This property MUST contain data of type duration.
- This property is optional.
- Meta-properties.
  - Description MAY be specified.
  - Type MAY be specified. The types that MUST be supported are:
    - user
    - system

### 3.12 **EndTime**

The time at which the usage consumption (i.e. "job") completed. The value of this property may depend on the selected queue system. For example some systems include time to stage files, others do not.

- This property SHOULD be referred to as endtime.
- This property MUST contain data of type timestamp.
- This property is optional
- Meta-properties
  - Description MAY be specified

### 3.13 **StartTime**

The time at which the usage consumption (i.e. “job”) started. The value of this property may depend on the selected queue system. For example, some systems include time to stage files, others do not.

- This property SHOULD be referred to as starttime.
- This property MUST contain data of type timestamp.
- This property is optional
- Meta-properties
  - Description MAY be specified

### 3.14 **MachineName**

A descriptive name of the machine on which the job ran. This may be a system hostname, or may be a site’s name for a cluster of machines. The identification of the machine by name may assume the context of the grid in which the machine participates; i.e. machine names may be unique within a specific grid, but do not need to be unique across the set of all grids.

- This property SHOULD be referred to as machinename.
- This property MUST contain data of type string.
- This property is optional.
- Meta-properties
  - Description MAY be specified.

### 3.15 **Host**

The system hostname on which the job ran.

- This property SHOULD be referred to as host.
- This property MUST contain data of type domain name.
- This property is optional.
- Meta-properties
  - Description MAY be specified.
  - Primary MAY be specified.
    - This meta-property indicates whether the indicated host acted as the primary host for the execution of the job.
    - The meta-property MUST contain data of type Boolean.
    - The default value for this meta-property is false.

### 3.16 **SubmitHost**

The system hostname from which the job was submitted.

- This property SHOULD be referred to as submithost.
- This property MUST contain data of type domain name.
- This property is optional.
- Meta-properties
  - Description MAY be specified.

### 3.17 **Queue**

The name of the queue from which the job was executed or submitted.

- This property SHOULD be referred to as queue.
- This property MUST have data of type string.
- This property is optional.
- Meta-properties
  - Description MAY be specified.

### 3.18 **ProjectName**

The project associated with the resource usage reported with this record. Some accounting systems define this is the ACID. On some systems, the project is identified with the effective GID under which the job consumed resources.

- This property SHOULD be referred to as projectname.
- This property MUST have data of type string.
- This property is optional.
- There SHOULD be no restriction on the length or available characters.
- Meta-properties
  - Description MAY be specified.

## 4. **Differentiated Properties**

Differentiated properties are resource consumption measurements that may or may not be required by the grid organization's implementation of this schema. Any, all, or none of these properties may be included in a valid usage record. The properties are generally differentiated by resource type.

### 4.1 **Network**

The amount of network resource used by the job.

- This property SHOULD be referred to as network.
- This property MUST contain data of type positive integer.
- This property is optional.
- Meta-properties
  - Units SHOULD be specified
  - Metric MAY be specified.
  - If metric is used, the metrics that MUST be supported are:
    - average – the average flow rate over the entire usage window.
    - total – volume of data transferred in the specified unit. This is the default.
    - min – minimum flow rate in the specified units.
    - max – maximum flow rate in the specified units.

### 4.2 **Disk**

Disk storage used.

- This property SHOULD be referred to as disk
- This property MUST contain data of type positive integer
- This property is optional
- Meta-properties
  - Units MAY be specified
  - Description MAY be specified
  - Type MAY be specified. The types that MUST be supported are:
    - scratch
    - temp
  - Metric MAY be specified. The metrics that MUST be supported are:
    - average
    - total
    - min
    - max

### 4.3 **Memory**

The amount of memory used by all concurrent processes in the job.

- This property SHOULD be referred to as memory.
- This property MUST contain data of type positive integer.
- This property is optional.
- Meta-properties
  - Units MUST be specified.
  - Description MAY be specified.
  - Metric MAY be specified. The metrics that MUST be supported are:
    - average
    - total
    - min
    - max
  - Type MAY be specified. The types that MUST be supported are:
    - shared
    - physical
    - dedicated

#### 4.4 **Swap**

This property specifies the swap usage

- This property SHOULD be referred to as swap.
- This property MUST have a data of type positive integer.
- This property is optional.
- Meta-properties
  - Description MAY be specified.

#### 4.5 **NodeCount**

Number of nodes used. A node definition may be dependent on the architecture, but typically a node is a physical machine. For example a cluster of 16 physical machines with each machine having one processor each is a 16 “node” machine, each with one “processor”. A 16 processor SMP machine however, is 1 physical node (machine) with 16 processors.

- This property SHOULD be referred to as nodecount.
- This property MUST contain data of type positive integer.
- This property is optional.
- Meta-properties
  - Description MAY be specified.
  - Metric MAY be specified. The metrics that MUST be supported are:
    - average
    - total
    - min
    - max

#### 4.6 **Processors**

The number of processors used or requested. A processor definition may be dependent on the machine architecture. Typically processor is equivalent to the number of physical CPUs used. For example, if a job uses two cluster “nodes”, each node having 16 CPUs each, the total number of processors would be 32.

- This property SHOULD be referred to as processors.
- This property MUST contain data of type positive integer.
- This property is optional.
- Meta-properties
  - Description MAY be specified.

- Metric MAY be specified. The metrics that MUST be supported are:
  - average
  - total
  - min
  - max
- ConsumptionRate MAY be specified.
  - This meta-property defines the consumption rate for this usage.
  - This meta-property MUST contain data of type float.

#### 4.7 **TimeDuration**

This property identifies any additional measure of time duration associated with the resource consumption. For example, it may report the connection time within a multi-tasking queue.

- This property SHOULD be referred to as timeduration.
- This property MUST contain data of type duration.
- This property is optional.

#### 4.8 **TimeInstant**

This property identifies any additionally identified discreet timestamp associated with the resource consumption. For example, it may represent the time the job was queued, rather than the start-time of the job.

- This property SHOULD be referred to as timeinstant.
- This property MUST contain data of type datetime.
- This property is optional.

#### 4.9 **ServiceLevel**

This property identifies the quality of service associated with the resource consumption. For example, service level may represent a priority associated with the usage.

- This property SHOULD be referred to as servicelevel.
- This property MUST contain data of type string.
- This property is optional.

#### 4.10 **Extension**

For sites that may want to exchange data that is not one of the base properties (for example, perhaps grid telescope time) the Extension property can be used to encode any type of usage information. The sites can agree on the meta properties supported for each extension.

- This property SHOULD be referred to as extension.
- This property MUST have data of type string.
- This property is optional.
- Meta-properties
  - Units may be supported.
  - Metric may be supported.
  - Name may be supported.
    - This meta-property must have data of type string.

### 5. **Job Level Accounting**

Job level accounting reports accounting data at the job level. PBS and LoadLeveller, for example, use job level accounting where each job is assigned a unique id and the

accounting for that job is reported for each id. The properties listed in this document are not required to appear within the usage record structure in the order in which they are defined.

- This type of usage record MAY contain any of the base properties.
- This type of usage record MUST contain at least one of the following properties:
  - LocalJobId
  - GlobalJobId
- The Host property MAY repeat within the record. Each occurrence of the Host property SHOULD contain a unique value for the job usage being reported.
- Job Level properties SHOULD be contained within a “job” property for each job

## 6. **Aggregate Accounting**

Aggregate accounting reports the accounting data in aggregate (summarized form). The properties listed in this document are not required to appear within the usage record structure in the order in which they are defined.

- Aggregate accounting MAY contain any of base properties listed in this document, which appear exactly once per job.
- Aggregate properties SHOULD be contained within an “aggregate” property.

## 7. **XML Overview**

### 7.1 **Encoding**

7.1.1 **XML Structure:** This specification uses XML Schema documents conforming to the W3C XML Schema Specification [XML\_Schema] and normative text to describe the syntax and semantics of XML encoded usage records.

7.1.2 **Transport:** This specification defines the structure of information that may be represented in a compliant XML document. No requirements are placed on the encoding of this document for a particular transport. Therefore, instance documents may be represented in ASCII or Unicode text. Further, we envision that many of the systems using this data definition will be OGSA compliant systems and therefore preferences to the http/ https protocols may occur. However, a usage record may be communicated via any lower level transport that is acceptable to the using parties.

### 7.2 **XML Conventions**

#### 7.2.1 **Element names**

- Element names SHALL be in UCC convention (example: <UpperCamelCaseElement/>).
- Capitalization of element names in external specifications SHALL remain consistent with the initial specification (example: ds:KeyInfo).
- Acronyms SHOULD be avoided.
- Underscores (\_), periods (.) and dashes(-) MUST NOT be used.
- Element names MUST comply with all XML Schema specific naming rules.

#### 7.2.2 **Attribute names**

- Attribute names SHALL be in LCC convention (example: <UpperCamelCaseElement lowerCamelCaseAttribute="attributevalue"/>).
- Capitalization of attribute names in external specifications SHALL remain consistent with the initial specification.

- Acronyms SHOULD be avoided.
- Underscores (`_`), periods (`.`) and dashes (`-`) MUST NOT be used.
- Attribute names MUST comply with all XML Schema specific naming rules.

### 7.2.3 Enumerated attribute values

- Attributes of type enumeration SHALL have values in LCC convention (example: `<UpperCamelCaseElement enumAttribute="valueOne"/>`).
- Capitalization of enumerated attribute values in external specifications SHALL remain consistent with the initial specification.

### 7.2.4 Empty and white space only values: This specification places the following restrictions on string values:

- All string values that do not require white space characters other than the space character should be defined as the **token** type. This type does not permit leading or trailing spaces, two or more adjacent spaces, or any other white space character. Refer to the XML Schema specification [XML\_SCHEMA] for additional details on this type.
- Identity constraints SHOULD use the schema identity constraint mechanisms rather than ID, IDREF or IDREFS definitions. Thus, any element that may be used as an identifier SHOULD use a data type that strictly controls white space, such as token, NCName or QName.
  - If a key identity constraint is declared for a particular element or attribute, then it MUST be present for each member of the participating nodeset. Additional information on the identity constraint can be found in [XML\_SCHEMA].
- No element declared in the UsageRecord schema specifies the `xsd:nillable` attribute. Therefore, empty element content means a string of zero length. Any schema extension that declares an element as nillable MUST not equate the nil string and a zero-length string. An instance document must use the `xsi:nil` attribute to set the value to nil `<MyNillableElement xsi:nil="true"/>` whereas a zero-length string is represented by an empty element `<MyElement/>`.

### 7.2.5 Time values

#### 7.2.5.1 Discrete time values: All time values that represent a discrete instance in time MUST be declared as the primitive XML Schema type **dateTime**. Values are represented with a character string corresponding to the date and time components specified in ISO 8601 [ISO8601]. According to [XML\_SCHEMA], implementations must handle fractional seconds to six digits, and MUST round additional digits. This specification places several additional semantic constraints on time values contained in conforming instance documents:

- Although the **dateTime** data type permits negative values (to identify times B.C.E), semantically, compliant documents MUST not contain such values.
- Both the date and time components of **dateTime** MUST be present.
- All points in time MUST indicate a specific time zone to permit a total ordering on across all represented time values as well as comparisons between times. The suffix ‘Z’ represents Coordinated Universal Time.”

#### 7.2.5.2 Interval time values: All time values that represent an interval of time MUST be declared as the primitive XML Schema type **duration**. A character string as specified in ISO 8601 represents values. Each numeral representing year, month, day, hour, minute or second is combined with a terminator to create a component of the duration. Each

component may be omitted. This specification places additional semantic constraints on time durations within conforming instance documents:

- Negative durations **MUST** not be present.
- The smallest granularity component of duration **MUST** be used when representing time duration. For example, one month is not comparable to 30 days and thus duration should include the day component if resource usage is typically measured in increments of one day. To avoid these inherent comparison and conversion difficulties, only resources that are typically measured in month and year intervals should use that component of duration.

**7.2.6 Comparing Usage Record Values:** Two UsageRecord elements that carry identical values for the recordId attribute of the RecordIdentity element (see §10.1) **MUST** be considered identical. It is left to the application producing or consuming UsageRecord instances to enforce this constraint. Further, it is also left to each site to ensure that the possibility of producing recordId values identical to the values used as recordId values of a different site is minimized. Although two UsageRecords with different recordId values **MAY** carry identical usage information, they are not considered identical instances. It is left to the application producing or consuming Usage Record instances to specify how this situation will be addressed.

**7.2.7 Encoding within KeyInfo elements:** [XML\_DSIG] identifies several rules for encoding distinguished names within its X509IssuerSerial, X509SubjectName and KeyName child elements. These encoding rules are found at the end of [XML\_DSG] §4.4.4 of the specification.

**7.2.8 Extension Framework:** The xsi:type attribute is used in instance documents to indicate which derived type we are using in an instance document. Therefore, if custom resource definitions are derived from any of the extensibility elements, then the instance documents using that content model must identify the derived type within the appropriate element declaration. See §12 for additional information on the extension points defined within this specification.

## 7.3 Schema

### 7.3.1 Schema Listings

The Schema listing can be found as Appendix D of this document.

---

Listings from the Usage Record Schema appear like this

---

In case of disagreement between the schema file and this specification, the schema file takes precedence.

**7.3.2 Schema Organization and Namespaces:** The usage record structures are defined in a schema [URWG-XSD] associated with the following namespace. When referenced in this document, this namespace is associated with the prefix urf.

---

<http://schema.ogf.org/urf/2003/09/urf>

---

The digital signature components are defined in a schema [XML\_SIG] associated with the following namespace: This schema is imported into the URF schema to directly use its definitions. When referenced in this document, this namespace is associated with the prefix ds.

```
http://www.w3.org/2000/09/xmldsig#
```

All simple data types referenced in this document are built into the W3C XML Schema Datatypes specification. When referenced in this document, this namespace is associated with the prefix xsd.

```
http://www.w3.org/2001/XMLSchema
```

The XMLSchema-instance namespace defines several attributes that are used in element definitions. When referenced in this document, this namespace is associated with the prefix xsi.

```
http://www.w3.org/2001/XMLSchema-instance
```

**7.3.2.1 Schema Header and Namespace Declarations:** The following schema fragment defines the XML namespaces and other header information:

```
<xsd:schema
attributeFormDefault="qualified"
elementFormDefault="qualified"
targetNamespace="http://schema.ogf.org/urf/2003/09/urf"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:urf="http://schema.ogf.org/urf/2003/09/urf"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<xsd:annotation>
<xsd:documentation xml:lang="en">Usage Record Working Group XML
Schema definition</xsd:documentation>
</xsd:annotation>
<xsd:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/xmldsig-core/xmldsig-core-
schema.xsd"/>
...
</schema>
```

## 8. Global Element Definitions

The global definitions are those that may be used compliantly with this specification as root elements for an XML document or which may be used as extension points within or included within other XML Schema definitions. Although the elements that represent individual resources that are consumed are defined at a global level within the xml schema, semantically, a compliant instance document should only be rooted with one of the following elements or an element that derives from one of these elements. To facilitate extensibility, these elements participate in the substitution group mechanism from [XML\_SCHEMA] according to the following abstract element definition :

```
<xsd:element abstract="true" name="Usage"
type="urf:UsageRecordType"/>
```

### 8.1 UsageRecord Element

The UsageRecord element encapsulates a single Usage Record. The UsageRecordType complex type dictates the particular structure of this element. All specific usage record elements should extend or restrict this element. Any structure that contains usage record information should reference the UsageRecord element so that extensions or restrictions of the element are automatically handled. This element should contain all the information that is generic to a usage record and addressed by this specification. Any extensions to or restrictions of the UsageRecord should not redefine this base structure.

```
<xsd:element name="UsageRecord" substitutionGroup="urf:Usage"
type="urf:UsageRecordType"/>
```

### 8.2 JobUsageRecord Element

This element definition establishes the structure of job usage record as it derives from the generic UsageRecordType.

```
<xsd:element name="JobUsageRecord" substitutionGroup="urf:Usage">
<xsd:complexType>
<xsd:complexContent>
<xsd:extension base="urf:UsageRecordType"/>
</xsd:complexContent>
</xsd:complexType>
</xsd:element>
```

### 8.3 UsageRecords Element

This element definition allows a set of UsageRecords to be grouped together with a single common root element.

```
<xsd:element name="UsageRecords">
<xsd:complexType>
<xsd:sequence>
<xsd:element maxOccurs="unbounded" minOccurs="0"
ref="urf:Usage"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
```

## 9. Global Attribute and Attribute Group Definitions

Global attributes, and attribute groups, are attributes that are common to many of the element definitions contained in this specification. A global definition of these maintains a standard semantic meaning within each context that they are used. By defining them globally, we ensure a standard definition for the appropriate information.

When referencing a global attribute, an element must specify whether the use of that attribute is *required* or *optional* within its context. Each element must also identify the default attribute value to be assumed, if applicable, when the attribute is omitted from the instance document. Note that during XML Canonicalization, any attribute for which a default value is specified within the constraining schema will be inserted into the instance

document with that default value by the canonicalizer if no value is supplied for that attribute.

When referencing a global attribute group, the characteristics of each member attribute are defined directly in the attribute group rather than in the context of a referring element.

### 9.1 Description

This attribute provides a mechanism for additional, optional information to be attached to a Usage Record element. The value of this attribute MAY provide clues to the semantic context to use while interpreting or examining the value of the owning element. As there are no constraints on the contents of this attribute, it is defined as a string rather than a token.

```
<xsd:attribute name="description" type="xsd:string" />
```

### 9.2 storageUnit

This attribute represents the unit of volume measure that should be applied to the value of its owning element. Each element that references this global attribute MUST define its use as required and identify the appropriate default value. The legal values for this attribute are listed below. Each value combines a prefix that identifies a multiplier and a suffix that identifies the base quantity. For example, the suffix -B represents volume in bytes and the prefix K- implies a kilo multiplier. Therefore, KB as the designated unit of measure value MUST be interpreted as kilobytes. Please refer to Appendix C for a listing of all valid values. When this attribute is used without a corresponding phaseUnit, no assumption may be made regarding the relationship between the usage represented by the referring element and the overall duration represented with the UsageRecord instance. Volume SHOULD not be measured in terms of characters, words or blocks as these types of measurement are typically architecture specific and require additional information to be interpreted correctly. Any usage locally represented with such a unit MUST be converted into one of the legal units.

```
<xsd:attribute name="storageUnit">
  <xsd:simpleType>
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="b"/>
      <xsd:enumeration value="B"/>
      <xsd:enumeration value="KB"/>
      <xsd:enumeration value="MB"/>
      <xsd:enumeration value="GB"/>
      <xsd:enumeration value="TB"/>
      <xsd:enumeration value="PB"/>
      <xsd:enumeration value="EB"/>
      <xsd:enumeration value="Kb"/>
      <xsd:enumeration value="Mb"/>
      <xsd:enumeration value="Gb"/>
      <xsd:enumeration value="Tb"/>
      <xsd:enumeration value="Pb"/>
      <xsd:enumeration value="Eb"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
```

### 9.3 **phaseUnit**

This attribute represents the a duration of time that is relevant to the usage reported in the referring element. See §2.2.2.2 for additional information on the use of duration.

```
<xsd:attribute name="phaseUnit" type="xsd:duration"/>
```

### 9.4 **intervallicVolume** Attribute Group:

This attribute group identifies the measurements that must accompany usage measured in two distinct dimensions. These dimension are the volume of resource consumed and the interval of time for which the resource was used. All attributes within this group are required. Therefore, any element that references this attribute group **MUST** provide values for each attribute.

```
<xsd:attributeGroup name="intervallicVolume">
<xsd:attribute ref="urf:storageUnit" use="required"/>
<xsd:attribute ref="urf:phaseUnit" use="required"/>
</xsd:attributeGroup>
```

### 9.5 **metric**

This attribute identifies the method of measurement used for quantifying the associated resource consumption if there are multiple methods by which to measure resource usage. This attribute is intended to provide a standard mechanism for including information about the how the measurement is performed, regardless of the resource that is measured. Therefore, the definition of metric within the schema document does not predefine the actual value that may appear for this attribute, either globally for all elements or uniquely to a specific element containing the attribute. However, particular to each element definition in this specification document that may carry the metric attribute, a minimal set of values are defined that must be semantically supported as values for that specific resource type. As an example, §11.2.2 indicates that the following values must be recognized for the metric associated with disk usage: average, total, min, max. This minimal set does not disqualify other values, such as "average scratch", as prohibited. It only implies that required entities sharing usage information in the standard usage record format must independently establish an understanding of those additional values. For the standard definitions unique to each metric value for a particular resource element, please refer directly to the resource specifications in §10 and §11 of this document. The following rules **MUST** be followed when creating metric value definitions for any resource.

- Any use of this attribute **MUST** not attempt to differentiate between requested and utilized quantities of resource usage within a single record, even if this differentiation is pertinent to the final assessed charge. Requested resources may be represented by a second usage record, containing an appropriate value for the Status attribute, as agreed upon by the implementation context.
- Each element that differentiates usage according to the metric **MUST** define attribute usage as required and identify the appropriate default value.
- The use of *average* as a metric value **MUST** be defined by the arithmetic mean.
- The use of *average-95* as a metric value **MUST** be defined by the 95<sup>th</sup> percentile calculation for average.
- Each metric value definition must differentiate between per-node and across-node measurements, when both measurements are feasible.

```
<xsd:attribute name="metric" type="xsd:token" />
```

## 9.6 type

This attribute identifies the type of the resource being measure when quantifying the associated resource consumption. This attribute is intended to provide a standard mechanism for including information about the type of measurement used, regardless of the resource that is measured. Therefore, the definition of type within the schema document does not predefine the actual value that may appear for this attribute, either globally for all elements or unique to a specific element containing the attribute. However, particular to each element definition in this specification document that may carry the type attribute, a minimal set of values are defined that must be semantically supported as values for that specific resource type. As an example, §11.2.3 indicates that the following values must be recognized for the type associated with disk usage: scratch, temp. This minimal set does not disqualify other values, such as "high speed disk". It only implies that required entities sharing usage information in the standard usage record format must independently establish an understanding of those additional values. For the standard definitions unique to each type value for a particular resource element, please refer directly to the resource specifications in §10 and §11 of this document. Any use of this attribute **MUST** not attempt to differentiate between requested and utilized quantities of resource usage within a single record, even if this differentiation is pertinent to the final assessed charge.

```
<xsd:attribute name="type" type="xsd:token" />
```

## 9.7 unit

This attribute may be used to express a unit of measure that is not quantifiable in terms of volume, time or a combination of both. For example, this attribute may represent any of the following units of measure "number of requests", "operations", "requests", "signals", "faults", "context switches", "swaps", "page reclaims" or "messages." However, there are no specific values for this attribute identified within this specification that **MUST** be supported.

```
<xsd:attribute name="unit" type="xsd:token" />
```

## 10. Global Type Definitions

### 10.1 UsageRecordType Complex Type

This complex type definition establishes the structure of the generic usage record. The properties section §3, above, form the basis for the definition of the UsageRecordType components.

There are two mandatory elements of the UsageRecordType. These elements are RecordIdentity and Status. This implies that each element of this type that appears in an instance document must contain these child elements. The descriptions for those specific elements (see §10.1 and §10.6) define the semantics for their unique content models.

**10.1.1 Content Model:** The elements contained in the UsageRecordType content model may appear in various orderings, so long as they adhere to the following rules:

- All identity, job name, charge and status elements **MUST** appear before any specific resource elements.
- All resources that are differentiated by metric **MUST** appear before other resources
- If multiple CpuDuration elements appear, they **MUST** appear together in single group.
- If multiple ProjectName elements appear, they **MUST** appear together in a single group.
- If multiple Host elements appear, they **MUST** appear together in a single group.
- If any elements represent resource usage via the extensibility framework appear, they **MUST** appear after the specifically defined resource elements.

```

<xsd:complexType name="UsageRecordType">
  <xsd:sequence>
    <xsd:element maxOccurs="1" minOccurs="1"
      ref="urf:RecordIdentity"/>
    <xsd:element maxOccurs="1" minOccurs="0" ref="urf:JobIdentity"/>
    <xsd:element maxOccurs="unbounded" minOccurs="0"
      ref="urf:UserIdentity"/>
    <xsd:element maxOccurs="1" minOccurs="0" ref="urf:JobName"/>
    <xsd:element maxOccurs="1" minOccurs="0" ref="urf:Charge"/>
    <xsd:element maxOccurs="1" minOccurs="1" ref="urf:Status"/>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>The elements grouped together in this choice
          may be
          represented within a usage record multiple times. Each of these
          appearances
          must be differentiated by the metric and/or type associated with
          the element. </xsd:documentation>
      </xsd:annotation>
      <xsd:element ref="urf:Disk"/>
      <xsd:element ref="urf:Memory"/>
      <xsd:element ref="urf:Swap"/>
      <xsd:element ref="urf:Network"/>
      <xsd:element ref="urf:TimeDuration"/>
      <xsd:element ref="urf:TimeInstant"/>
      <xsd:element ref="urf:ServiceLevel"/>
    </xsd:choice>
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element maxOccurs="1" minOccurs="0" ref="urf:WallDuration"/>
      <xsd:sequence maxOccurs="1" minOccurs="0">
        <xsd:element maxOccurs="2" minOccurs="0" ref="urf:CpuDuration"/>
      </xsd:sequence>
      <xsd:element maxOccurs="1" minOccurs="0" ref="urf:NodeCount"/>
      <xsd:element maxOccurs="1" minOccurs="0" ref="urf:Processors"/>
      <xsd:element maxOccurs="1" minOccurs="0" ref="urf:EndTime"/>
      <xsd:element maxOccurs="1" minOccurs="0" ref="urf:StartTime"/>
      <xsd:element maxOccurs="1" minOccurs="0" ref="urf:MachineName"/>
      <xsd:element maxOccurs="1" minOccurs="0" ref="urf:SubmitHost"/>
      <xsd:element maxOccurs="1" minOccurs="0" ref="urf:Queue"/>
      <xsd:sequence maxOccurs="1" minOccurs="0">
        <xsd:element maxOccurs="unbounded" minOccurs="0"
          ref="urf:ProjectName"/>

```

```

</xsd:sequence>
<xsd:sequence maxOccurs="1" minOccurs="0">
<xsd:element maxOccurs="unbounded" minOccurs="0" ref="urf:Host"/>
</xsd:sequence>
<xsd:sequence maxOccurs="1" minOccurs="0">
<xsd:choice maxOccurs="unbounded" minOccurs="0">
<xsd:element ref="urf:PhaseResource"/>
<xsd:element ref="urf:VolumeResource"/>
<xsd:element ref="urf:Resource" />
<xsd:element ref="urf:ConsumableResource" />
</xsd:choice>
</xsd:sequence>
</xsd:choice>
</xsd:sequence>
</xsd:complexType>

```

## 10.2 **domainNameType** Simple Type

This simple type definition establishes the structure of a Fully Qualified Domain Name in string format. Any valid value provided follows the format of a fully qualified domain name (FQDN) as defined by RFC 1034 [RFC1034]:

*FQDNs can be up to 255 characters long and can contain alphabetic and numeric characters and the “-“ and the “.” characters.*

```

<xsd:simpleType name="domainNameType">
<xsd:restriction base="xsd:string">
<xsd:pattern value="([a-zA-Z0-9][a-zA-Z0-9'\-']*[a-zA-Z0-9]\.)*([a-zA-Z0-9][a-zA-Z0-9'\-']*[a-zA-Z0-9])?"/>
<xsd:maxLength value="255"/>
</xsd:restriction>
</xsd:simpleType>

```

## 11. **Usage Record Common Properties**

The complete specifications for each element that MAY appear in the generic usage record element are presented below. A quick reference table summarizes these elements in Appendix B. Each of these elements MUST appear at most once within a UsageRecord. Thus, the usage details represented by these elements cannot be differentiated based on distinct metrics within each appearance of an element.

### 11.1 **RecordIdentity** Element

This attribute group combines all the attribute information that may identify a particular a Usage Record or Usage Records element. However, no such identity constraint is defined directly within this schema. Each UsageRecord element MUST contain a child RecordIdentity element.

11.1.1 **recordId** Attribute: This attribute contains a token that may be referenced using the identity constraint mechanisms defined in XML Schema. It is used to identify a unique instance of a Usage Record. Attribute values must satisfy the following properties:

- Any party that assigns a recordId MUST ensure that there is negligible probability that it, or any other party, will assign the same value to a different UsageRecord.
- Each Usage Record may declare at most one identifier.

- 11.1.2 **createTime** Attribute: This attribute contains a `dateTime` that defines when this particular `UsageRecord` was created. This attribute **MUST** reflect the creation of the actual record and not when the consumption it reports occurred. See §6.2.5 for additional information about representing an instance in time.
- 11.1.3 **ds:KeyInfo** Element: `KeyInfo` is an optional element within the XML Digital Signature Specification [XML\_DSIG] that “enables the recipients(s) to obtain the key needed to validate a signature.” As each key ultimately correlates to a particular identified entity, this specification uses the `KeyInfo` fields to identify the entity that created a particular Usage Record. The following rules should be followed when populating the various sub-elements of this `KeyInfo` with identifying values:
- The `RetrievalMethod` element **MUST** be present. The *Type* attribute of the element **MUST** contain the appropriate URI (see §4.4 of XML-DSIG for a complete list) to identify the type of identity in use. No other components of the `RetrievalMethod` element are required by this specification. The `RetrievalMethod` identified **MUST** correspond to the additional information represented within `KeyInfo`. Supported type URIs must include:
    - <http://www.w3.org/2000/09/xmldsig#DSAKeyValue>
    - <http://www.w3.org/2000/09/xmldsig#RSAKeyValue>
    - <http://www.w3.org/2000/09/xmldsig#X509Data>
    - <http://www.w3.org/2000/09/xmldsig#rawX509Certificate>
  - The following rules apply to the specific types `KeyInfo` data, according to the selected *RetrievalMethod Type*. Note that XML-Dsig allows multiple identifier types within a single `KeyInfo` element.
    - *KeyName* contains one of a string identifier related to the key-pair, the distinguished name or the email address identifying the creator. If *KeyName* identifies a named key-pair, `RetrievalMethod` type **MUST** identify the type of keys contained within the key-pair.
    - *KeyValue* contains the public key associated with the creator. If *KeyValue* is used, `RetrievalMethod` type **MUST** identify the type of the public key.
    - *X509Data* contains information from the X.509 certificate associated with the creator. `RetrievalMethod` type **MUST** reference <http://www.w3.org/2000/09/xmldsig#X509Data>. The relevant rules from [XML-Dsig], summarized below, must be followed when representing X509 information.
      - At least one of *X509IssuerSerial*, *X509SubjectName*, *X509SKI* or *X509Certificate* must be present.
      - All elements from §2.c.i referring to a single certificate must be grouped within a single *X509Data* element.
      - All certificates appearing in an *X509Data* element **MUST** relate to the identity of the creator or be part of a certification chain terminating in a certificate that identifies the creator.
        - If *RetrievalMethod Type* is <http://www.w3.org/2000/09/xmldsig#rawX509Certificate>, the contents of `KeyInfo` **MUST** be a binary *X509Certificate*. No particular encoding is required by this specification.
  - No specific ordering must be imposed on the elements contained within `KeyInfo`.

See section §6.2.7 for additional restrictions on value representation within a `KeyInfo` structure.

```

<xsd:element name="RecordIdentity">
  <xsd:complexType >
    <xsd:sequence maxOccurs="1" minOccurs="0">
      <xsd:element ref="ds:KeyInfo" />
    </xsd:sequence>
    <xsd:attribute name="recordId" type="xsd:token" use="required"/>
    <xsd:attribute name="createDate" type="xsd:dateTime"
      use="optional"/>
  </xsd:complexType>
</xsd:element>

```

## 11.2 JobIdentity

This element uniquely identifies the job associated with this usage. The scope representation of identity may be either local or global. An identifier of at least one scope MUST appear in the UsageRecord. Identifiers of both scopes MAY appear in a single UsageRecord. If both appear, there must be a semantic correlation between the presented local and global job identifiers. There is no upper bound on the number of job identifiers that MAY be associated with a single UsageRecord.

- 11.2.1 **GlobalJobId**: The value of this element represents a global identity for the job. No restrictions on the format of this value are required by this specification.
- 11.2.2 **LocalJobId**: The value of this element represents a local identity for the job. No restrictions on the format of this value are required by this specification. The value of this element MAY require additional information to be correctly interpreted.
- 11.2.3 **ProcessId**: If all processes associated with a job are tracked as part of usage, each individual process identifier MAY be reported with this element. Each individual process MUST be represented in a unique ProcessId element.

```

<xsd:element name="JobIdentity">
  <xsd:complexType>
    <xsd:sequence minOccurs="1">
      <xsd:element name="GlobalJobId" type="xsd:string" minOccurs="0"/>
      <xsd:element name="LocalJobId" type="xsd:string" minOccurs="0"/>
    <xsd:sequence >
      <xsd:element name="ProcessId" minOccurs="0" maxOccurs="unbounded"
        type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

## 11.3 UserIdentity

This element uniquely identifies the user associated with this usage. The scope representation of identity may be either local or global. Identifiers of both scopes MAY appear in a single UsageRecord. If both appear, there must be a semantic correlation between the presented local and global user identifiers. There is no upper bound on the number of user identifiers that MAY be associated with a single UsageRecord.

- 11.3.1 **LocalUserId:** The value of this element represents a local identity for the user. No restrictions on the format of this value are required by this specification. For example, this may report the local uid under which the job executed.
- 11.3.2 **GlobalUsername:** If the grid supports global usernames, this element should contain that value. There are no restrictions on the format of this value.
- 11.3.3 **ds:KeyInfo:** The contents of this element represent the global identity for the user associated with this usage. Similarly to the RecordIdentity element, rules for binding data elements to this structure assist the interpretation of data contained. The following rules should be followed when populating the various sub-elements of this KeyInfo with identifying values:
- The RetrievalMethod element MUST be present. The *Type* attribute of the element MUST contain the appropriate URI (see section § 4.4 of XML-DSIG for a complete list) to identify the type of identity in use. No other components of the RetrievalMethod element are required by this specification. The RetrievalMethod identified MUST correspond to the additional information represented within KeyInfo. Supported type URIs must include:
    - <http://www.w3.org/2000/09/xmldsig#DSAKeyValue>
    - <http://www.w3.org/2000/09/xmldsig#RSAKeyValue>
    - <http://www.w3.org/2000/09/xmldsig#X509Data>
    - <http://www.w3.org/2000/09/xmldsig#rawX509Certificate>
  - The following rules apply to the specific types KeyInfo data, according to the selected *RetrievalMethod Type*. Note that XML-Dsig allows multiple identifier types within a single *KeyInfo* element.
    - *KeyName* contains one of a string identifier related to the key-pair, the distinguished name or the email address identifying the creator. If *KeyName* identifies a named key-pair, *RetrievalMethod* type MUST identify the type of keys contained within the key-pair.
    - *KeyValue* contains the public key associated with the creator. If *KeyValue* is used, *RetrievalMethod* type MUST identify the type of the public key.
    - *X509Data* contains information from the X.509 certificate associated with the creator. *RetrievalMethod* type MUST reference <http://www.w3.org/2000/09/xmldsig#X509Data>. The relevant rules from [XML-Dsig], summarized below, must be followed when representing X509 information.
      - At least one of *X509IssuerSerial*, *X509SubjectName*, *X509SKI* or *X509Certificate* must be present.
      - All elements from §2.c.i referring to a single certificate must be grouped within a single *X509Data* element.
      - All certificates appearing in an *X509Data* element MUST relate to the identity of the creator or be part of a certification chain terminating in a certificate that identifies the creator.
    - If *RetrievalMethod Type* is <http://www.w3.org/2000/09/xmldsig#rawX509Certificate>, the contents of *KeyInfo* MUST be a binary *X509Certificate*. No particular encoding is required by this specification.
  - No specific ordering must be imposed on the elements contained within *KeyInfo*.

See section §6.2.7 for additional restrictions on value representation within a *KeyInfo* structure.

```

<xsd:element name="UserIdentity">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="1" minOccurs="0" name="LocalUserId"
        type="xsd:string"/>
      <xsd:element maxOccurs="1" minOccurs="0" name="GlobalUserName"
        type="xsd:string"/>
      <xsd:element maxOccurs="1" minOccurs="0" ref="ds:KeyInfo"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

#### 11.4 JobName

This element specifies the name of the job or application that generated this usage. There is no requirement that this element contain a unique identifier for a job. For unique identification of a job, the JobIdentity (see §10.2) MUST be used.

11.4.1 **description:** This optional attribute provides additional information about job name.

```

<xsd:element name="JobName">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute ref="urf:description" use="optional"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>

```

#### 11.5 Charge

This element specifies the total charge, in the generating system's allocation unit, associated with the usage represented by this structure. The charge MAY be reported without any additional information regarding the usage that generated this charge. However, the value reported MAY include premiums or discounts assessed on the actual usage represented within this record. Therefore, it might not be possible to reconstruct the reported charge from the specific usage reported.

11.5.1 **description:** This optional attribute may provide information about the meaning associated with the reported charge.

11.5.2 **unit:** This attribute specifies the unit of measurement in which the charge for usage is reported. There are no values that must be supported by implementations.

11.5.3 **formula:** This attribute provides information about the charge calculation applied to this particular usage. Given that a particular charge formula may be complex, this attribute MAY contain only a description of the formula applied. Each implementation MAY specify semantic value that can be interpreted from the value contained in this attribute.

```

<xsd:element name="Charge">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:float">
        <xsd:attribute ref="urf:description" use="optional"/>
        <xsd:attribute ref="urf:unit" use="optional"/>
        <xsd:attribute name="formula" type="xsd:string" use="optional"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>

```

```

</xsd:simpleContent>
</xsd:complexType>
</xsd:element>

```

## 11.6 Status

This required element specifies a completion status associated with the usage as a string. If no value is reported for this element, no inference regarding the current job status may be made. The value for this element should report any completion status for a job. For example, this may represent the exit status of an interactive running process or the exit status from the batch queuing system's accounting record. Any resource usage reported in this record should be assumed as consumed regardless of the value, or lack thereof, specified for status. There is no defined semantic meaning attached to any particular value that may appear, other than the following pre-defined values. Compliant implementations must support the following values for status, and other values at their discretion.

- aborted – A policy or human intervention caused the job to cease execution
- completed – The execution completed
- failed – Execution halted without external intervention
- held – Execution is held at the time this usage record was generated
- queued – Execution was queued at the time this usage record was generated
- started – Execution started at the time this usage record was generated
- suspended – Execution was suspended at the time this usage record was generated

**11.6.1 description:** This optional attribute may provide information about the meaning associated with the reported status.

```

<xsd:element name="Status">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:token">
        <xsd:attribute ref="urf:description" use="optional"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>

```

## 11.7 WallDuration

This element specifies the wall clock time that elapsed.

**11.7.1 description:** This optional attribute provides additional information about the wall clock time reported

```

<xsd:element name="WallDuration">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:duration">
        <xsd:attribute ref="urf:description" use="optional"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>

```

## 11.8 CpuDuration

This element specifies the CPU time used, summed over all processes associated with this usage.

11.8.1 **description:** This optional attribute provides additional information about the CPU time reported

11.8.2 **usageType:** This optional attribute specifies whether the CPU time reported is user or system CPU time. The recognized values for this attribute are:

- user
- system

This element appears at most once with a particular value of usageType

```
<xsd:element name="CpuDuration">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:duration">
<xsd:attribute ref="urf:description" use="optional"/>
<xsd:attribute name="usageType">
<xsd:simpleType>
<xsd:restriction base="xsd:token">
<xsd:enumeration value="user"/>
<xsd:enumeration value="system"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
```

## 11.9 EndTime

This element specifies the time at which usage consumption ended. The end time may depend upon the queue system, file staging etc. If no EndTime value is reported, no assumption about the completion status of the job may be inferred. Only the Status (see §11.6) may provide specific information regarding completion status.

11.9.1 **description:** This optional attribute provides additional information about end time reported.

```
<xsd:element name="EndTime">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:dateTime">
<xsd:attribute ref="urf:description" use="optional"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
```

## 11.10 StartTime

This element specifies the time at which usage consumption started. The start time may depend upon the queue system, file staging etc.

**11.10.1description:** This optional attribute provides additional information about start time reported.

```
<xsd:element name="StartTime">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:dateTime">
        <xsd:attribute ref="urf:description" use="optional"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

### 11.11 MachineName

This element specifies the name of the machine on which a job ran.

**11.11.1description:** This optional attribute provides additional information about machinename.

```
<xsd:element name="MachineName">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="urf:domainNameType">
        <xsd:attribute ref="urf:description" use="optional"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

### 11.12 Host

This element specifies the name of the host on which a job ran.

**11.12.1description:** This optional attribute provides additional information about host.

**11.12.2primary:** This attribute indicates whether this host acted as the primary host for the execution that incurred this usage. This attribute contains a Boolean value which **MUST** be true if this host was the primary host. The default value for this attribute is false.

```
<xsd:element name="Host">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="urf:domainNameType">
        <xsd:attribute ref="urf:description" use="optional"/>
        <xsd:attribute default="false" name="primary"
          type="xsd:boolean"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

### 11.13 SubmitHost

This element specifies the name of the host from which a job was submitted.

**11.13.1description:** This optional attribute provides additional information about submithost.

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

```

<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="urf:domainNameType">
<xsd:attribute ref="urf:description" use="optional"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>

```

### 11.14 Queue

This element specifies the name of the queue from which the job executed or was submitted.

**11.14.1 description:** This optional attribute provides additional information about queue.

```

<xsd:element name="Queue">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:string">
<xsd:attribute ref="urf:description" use="optional"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>

```

### 11.15 ProjectName

This element specifies the name/identifier of the project or charge group associated with this usage.

**11.15.1 description:** This optional attribute provides additional information about project, for example a human readable project name.

```

<xsd:element name="ProjectName">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:string">
<xsd:attribute ref="urf:description" use="optional"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>

```

## 12. Differentiated Properties

These element definitions represent usage that may appear multiple times within a single UsageRecord if usage is measured with several distinct metrics or several types of the usage are included. If there are multiple appearances of a particular element, they **MUST** carry a unique combination of values for the metric and type attributes. The minimal set of supported values for metric and type is indicated within each element definition. Each element that can be differentiated by a metric is defined within a single choice structure. Please refer to the XML Schema documentation for details on the characteristics of the choice structure.

## 12.1 Network

This element specifies the network usage rate of transfer

12.1.1 **description:** This optional attribute may provide information about the type of network usage reported.

12.1.2 **metric:** This attribute specifies the metric for the reported network usage. The values that MUST be supported for this attribute are:

- average – the average flow rate over the entire usage window.
- total – volume of data transferred in the specified unit.
- min – minimum flow rate in the specified units.
- max – maximum flow rate in the specified units.

The default value for this attribute is total.

```
<xsd:element name="Network">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:positiveInteger">
        <xsd:attribute ref="urf:description" use="optional"/>
        <xsd:attributeGroup ref="urf:intervallicVolume"/>
        <xsd:attribute default="total" ref="urf:metric" use="optional"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

## 12.2 Disk:

This element specifies the disk storage for this usage.

12.2.1 **description:** This optional attribute may provide information about the type of disk usage reported. For example, it may be scratch space, network storage, etc. Adding the `xsd:anyAttribute` would allow a further description of the elements to include something like optical, tape or whatever implementation attributes are required.

12.2.2 **metric:** This attribute specifies the metric for the reported disk usage. The values that MUST be supported for this attribute are:

- average
- total
- min
- max

The default value for this attribute is total.

12.2.3 **type:** This attribute specifies the type of the reported disk usage. The values that MUST be supported for this attribute are:

- scratch
- temp

```
<xsd:element name="Disk">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:positiveInteger">
        <xsd:attribute ref="urf:description" use="optional"/>
        <xsd:attributeGroup ref="urf:intervallicVolume"/>
        <xsd:attribute default="total" ref="urf:metric" use="optional"/>
        <xsd:attribute ref="urf:type" use="optional"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

```

</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>

```

### 12.3 Memory

This element specifies the memory usage.

12.3.1 **description:** This optional attribute may provide information about the type of memory usage reported. For example, it may be virtual memory, paged memory, shared memory, dedicated memory, etc

12.3.2 **metric:** This attribute specifies the metric for the reported memory usage. The values that MUST be supported for this attribute are:

- average
- total
- max
- min

The default value for this attribute is total.

12.3.3 **type:** This attribute specifies the type of the reported memory usage. The values that MUST be supported for this attribute are:

- shared
- physical
- dedicated

```

<xsd:element name="Memory">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:positiveInteger">
        <xsd:attribute ref="urf:description" use="optional"/>
        <xsd:attributeGroup ref="urf:intervallicVolume"/>
        <xsd:attribute default="total" ref="urf:metric" use="optional"/>
        <xsd:attribute ref="urf:type" use="optional"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>

```

### 12.4 Swap

This element specifies the swap usage.

12.4.1 **description:** This optional attribute may provide information about the type of swap usage reported.

12.4.2 **metric:** This attribute specifies the metric for the reported swap usage. The values that MUST be supported for this attribute are:

- average
- total
- max
- min

The default value for this attribute is total.

12.4.3 **type:** This attribute specifies the type of the reported swap usage. There are no values that MUST be supported for this attribute.

```

<xsd:element name="Swap">

```

```

<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:positiveInteger">
<xsd:attribute ref="urf:description" use="optional"/>
<xsd:attributeGroup ref="urf:intervallicVolume"/>
<xsd:attribute default="total" ref="urf:metric" use="optional"/>
<xsd:attribute ref="urf:type" use="optional"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>

```

## 12.5 NodeCount

This element specifies the number of nodes used. The definition of a node may depend upon architecture.

- 12.5.1 **description:** This optional attribute provides additional information about the node count reported, for example it may report architecture (cluster vs. smp) associated with this usage record.

```

<xsd:element name="NodeCount">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:positiveInteger">
<xsd:attribute ref="urf:description" use="optional"/>
<xsd:attribute default="total" ref="urf:metric" use="optional"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>

```

- 12.5.2 **metric:** This attribute specifies the metric for the reported node usage. The values that MUST be supported for this attribute are:

- average
- total
- min
- max

## 12.6 Processors

This element specifies the number of processors used. A processor definition may depend on machine architecture.

- 12.6.1 **description:** This optional attribute provides additional information about the number of processors reported, for example it may report machine architecture associated with this usage record.

```

<xsd:element name="Processors">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:positiveInteger">
<xsd:attribute ref="urf:description" use="optional"/>
<xsd:attribute ref="urf:metric" use="optional"/>
<xsd:attribute name="consumptionRate" type="xsd:float"
use="optional"/>
</xsd:extension>

```

```

</xsd:simpleContent>
</xsd:complexType>
</xsd:element>

```

12.6.2 **metric**: This attribute specifies the metric for the reported processor usage. The values that MUST be supported for this attribute are:

- average
- total
- min
- max

12.6.3 **consumptionRate**: This attribute specifies the consumption rate for the reported processor usage. The consumption rate is a scaling factor that indicates the average percentage of utilization. It intends to facilitate the application of fair charging when accounting for multiplexed jobs.

For example, we might have

```
<Processors consumptionrate="0.67">2</Processors>
```

This says that of the two processors, they were only about 2/3rds utilized on average across the duration of the job. Presumably, the user will only be charged ( $2 * \text{per\_processor\_charge\_rate} * 0.67 * \text{wall\_clock\_duration\_of\_job}$ ).

## 12.7 TimeDuration

This element specifies the any additional measure of time duration that is relevant to the reported usage.

12.7.1 **description**: This optional attribute may provide information about the type of duration being reported. For example, it may be connection time within a multi-tasking queue.

12.7.2 **type**: This attribute specifies the metric for the reported time duration associated with this usage. The values that MUST be supported for this attribute are:

- submit
- connect
- dedicated

There is no default value for this attribute. The metric attribute MUST be present within the element.

```

<xsd:element name="TimeDuration">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:duration">
        <xsd:attribute ref="urf:type" use="required"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>

```

## 12.8 TimeInstant

This element specifies the any additionally identified discrete time that is relevant to the usage reported.

12.8.1 **description:** This optional attribute may provide information about the type of time being reported. For example, it may report the submit time, rather than start time, of the job.

12.8.2 **type:** This attribute specifies the type for the reported time instant associated with the usage. The values that **MUST** be supported for this attribute are:

- submit

There is no default value for this attribute. The type attribute **MUST** be present within the element.

```
<xsd:element name="TimeInstant">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:dateTime">
<xsd:attribute ref="urf:type" use="required"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
```

## 12.9 ServiceLevel

This element identifies the service level, or quality of service, associated with this resource consumption. Each installation may support several service levels. The values reported are installation specific and may affect charging. For example, service level may represent a combination of priority, expansion and roles associated with the reported usage.

```
<xsd:element name="ServiceLevel">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:token">
<xsd:attribute ref="urf:type" use="required"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
```

12.9.1 **type:** This attribute specifies the type for the reported service level associated with the usage. The values that **MUST** be supported for this attribute are:

- defined policy

There is no default value for this attribute.

## 13. Extension Framework

The XML formats for representing Usage Records have been designed with consideration given to extensibility for implementation specific requirements. However, the use of extensions may reduce interoperability and therefore the introduction of extensions **SHOULD** be carefully considered.

These element definitions represent a mechanism for sites to exchange data that does not correspond to one of the common properties (for example, perhaps grid telescope power). The specific types of extension elements that are defined for use – or further extension – are detailed in the following sub-sections. A site may choose to extend these basic

resource element definitions with additional attributes or child elements. However, the site must define the new semantics of additionally imposed structure.

### 13.1 Type Definitions

Standard type definitions allow sites to define elements that are explicitly typed by these definitions or to extend or restrict the generic type definitions as appropriate to the data needed for their resource usage records.

**13.1.1 ResourceType:** This type definition provides a mechanism to represent the consumption of an additional resource within the usage record. For example, the quote identifier, the executable or application name may be reported with this element.

**13.1.1.1 description:** This optional attribute provides additional information about the resource reported. Although its usage is optional, the value of this attribute can be used to distinguish between multiple resources that are included in a single usage record. For example, it may indicate which resource element contains the quote identifier and another contains the name of the executable used when both are reported within a single usage record. Therefore, if multiple resources elements are included in a single usage record, this attribute SHOULD be present.

```
<xsd:complexType name="ResourceType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute ref="urf:description" use="optional"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

**13.1.2 ConsumableResourceType:** This type definition provides a mechanism to represent the consumption of an additional resource within the usage record that is measured. For example, this resource element may report a standard support charge allocated to usage or the number of priority processes created to execute a job.

**13.1.2.1 description:** This optional attribute provides additional information about the resource reported. Although its usage is optional, the value of this attribute can be used to distinguish between multiple resources that are included in a single usage record. Therefore, when multiple resources elements are included in a single usage record, this attribute SHOULD be present.

**13.1.2.2 unit:** This attribute specifies the unit of measurement in which the consumed resource for this usage is reported. There are no values that must be supported by implementations.

```
<xsd:complexType name="ConsumableResourceType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:float">
      <xsd:attribute name="units" type="xsd:string" use="optional"/>
      <xsd:attribute ref="urf:description" use="optional"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

### 13.2 Element Definitions

Global element definitions allow sites to use these global element definitions directly rather than defining particular typed elements.

- 13.2.1 **Resource**: This optional element may provide information about a resource associated with the usage.

```
<xsd:element name="Resource" type="urf:ResourceType"/>
```

Example:

```
<Resource urf:description="quoteId">1435</Resource>
```

- 13.2.2 **ConsumableResource**: This optional element may provide information about a measured resource associated with the usage.

```
<xsd:element name="ConsumableResource"
type="urf:ConsumableResourceType"/>
```

Example:

```
<urf:ConsumableResource urf:description="mt process count"
urf:units="processes">1919</urf:ConsumableResource>
```

- 13.2.3 **PhaseResource**: This optional element extends the ConsumableResource element to report extended resource usage that measures resource usage with the phaseUnit attribute.

```
<xsd:element name="PhaseResource">
<xsd:complexType>
<xsd:complexContent>
<xsd:extension base="urf:ConsumableResourceType">
<xsd:attribute ref="urf:phaseUnit" use="optional"/>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:element>
```

- 13.2.4 **VolumeResource**: This optional element extends the ConsumableResource element to report extended resource usage that measures resource usage with the storageUnit attribute.

```
<xsd:element name="VolumeResource">
<xsd:complexType>
<xsd:complexContent>
<xsd:extension base="urf:ConsumableResourceType">
<xsd:attribute ref="urf:storageUnit" use="optional"/>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:element>
```

## 14. Example Usage Records

### 14.1 Sample Usage Record 1

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<JobUsageRecord xmlns="http://schema.ogf.org/urf/2003/09/urf"
  xmlns:urf="http://schema.ogf.org/urf/2003/09/urf"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schema.ogf.org/urf/2003/09/urf
file:/Users/bekah/Documents/GGF/URWG/urwg-schema.09.xsd">
  <RecordIdentity
    urf:recordId="http://www.emsl.pnl.gov/mscf/colony/PBS.1234.
0" urf:createTime="2003-08-13T18:56:56Z" />
  <JobIdentity>
    <LocalJobId>PBS.1234.0</LocalJobId>
  </JobIdentity>
  <UserIdentity>
    <LocalUserId>scottmo</LocalUserId>
  </UserIdentity>
  <Charge>2870</Charge>
  <Status>completed</Status>
  <Memory urf:storageUnit="MB">1234</Memory>
  <ServiceLevel urf:type="QOS">BottomFeeder</ServiceLevel>
  <Processors>4</Processors>
  <ProjectName>mscfops</ProjectName>
  <MachineName>Colony</MachineName>
  <WallDuration>PT1S</WallDuration>
  <StartTime>2003-08-13T17:34:50Z</StartTime>
  <EndTime>2003-08-13T18:37:38Z</EndTime>
  <NodeCount>2</NodeCount>
  <Queue>batch</Queue>
  <Resource urf:description="quoteId">1435</Resource>
  <Resource urf:description="application">NWChem</Resource>
  <Resource
    urf:description="executable">nwchem_linux</Resource>
</JobUsageRecord>

```

## 14.2 Sample Usage Record 2

```

<?xml version="1.0" encoding="UTF-8"?>
<UsageRecord xmlns="http://schema.ogf.org/urf/2003/09/urf"
  xmlns:urf="http://schema.ogf.org/urf/2003/09/urf"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schema.ogf.org/urf/2003/09/urf
file:/Users/bekah/Documents/GGF/URWG/urwg-
schema.09.02.xsd">
  <RecordIdentity urf:createTime="2003-08-15T14:25:56Z"
    urf:recordId="urn:nasa:arc:usage:82125.lomax.nas.nasa.gov:0
"/>
  <urf:JobIdentity>
    <urf:LocalJobId>82125.lomax.nas.nasa.gov</urf:LocalJobId>
  </urf:JobIdentity>
  <urf:UserIdentity>
    <urf:LocalUserId>foobar</urf:LocalUserId>
  </urf:UserIdentity>
  <Status urf:description="pbs exit status">0</Status>
  <urf:Memory urf:metric="max" urf:storageUnit="KB"
    urf:type="virtual">1060991</urf:Memory>
  <urf:Processors urf:metric="total">32</urf:Processors>
  <urf:EndTime>2003-06-16T08:24:32Z</urf:EndTime>

```

```
<urf:ProjectName urf:description="local charge
group">gl3563</urf:ProjectName>
<urf:Host urf:primary="true">lomax.nas.nasa.gov</urf:Host>
<urf:Queue>lomax</urf:Queue>
<urf:WallDuration>PT45M48S</urf:WallDuration>
<urf:CpuDuration>PT15S</urf:CpuDuration>
<urf:Resource urf:description="pbs-jobname">m0.20a-
7.0b0.0v</urf:Resource>
</UsageRecord>
```

## 15. Security Considerations

There may be security concerns that should be addressed with respect to usage data. Possible security issues might include:

- Non-repudiation
- Confidentiality of certain elements
- Integrity
- Secure Transport

Recommendation of required solutions for these security concerns are out of scope for this layer. Another layer should address the necessary security requirements.

**16. Document Credits****16.1 Author Information**

Rodney Mach  
University of Michigan  
rmach@umich.edu

Rebekah Lepro  
NASA Ames Research Center  
rlepro@arc.nasa.gov  
(650)604-4359

Scott Jackson  
Pacific Northwest National Laboratory  
scott.Jackson@pnl.gov  
(509)376-2205

Laura McGinnis (Editor)  
Pittsburgh Supercomputing Center  
lfm@psc.edu  
412-268-5642

**16.2 Contributor Information**

Donal Fellows  
University of Manchester  
donal.k.fellows@manchester.ac.uk

Christopher Baumbauer  
San Diego Supercomputer Center  
cab@sdsc.edu

Jian Zhang  
xCelera Corp.  
jian@xcelera.com

Mi Young Koo  
NASA Ames Research Center  
mkoo@nas.nasa.gov

Andy Anderson  
San Diego Supercomputer Center  
andya@sdsc.edu

Devin Kowatch  
San Diego Supercomputer Center  
devink@sdsc.edu

John Moores  
Sandial Corp.  
jmoores@sandial.com

## 17. Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

## 18. Full Copyright Notice

Copyright (C) Open Grid Forum (2006-2007). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the OGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the OGF Document process must be followed, or as required to translate it into languages other than English.

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

This document and the information contained herein is provided on an "As Is" basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

## 19. References

- [KEYWORDS ] RFC 2119. *Key words for use in RFCs to Indicate Requirement Levels*. S. Bradner. March 1997.  
<http://www.ietf.org/rfc/rfc2119.txt>
- [FED\_XML] US Federal CIO Council, “*Draft Federal XML Developers Guidelines*”, April 2002. Available online at [http://xml.gov/documents/in\\_progress/developersguide.pdf](http://xml.gov/documents/in_progress/developersguide.pdf).
- [ISO8601] Houston, G., “*ISO 8601:1988 Date/Time Representations*”, 1988. Available online at <http://hydracen.com/dx/iso8601.htm>.
- [NLD] Mach, Rodney et al., “*Accounting Interchange Natural Language Description (Requirements)*”, The Global Grid Forum. November 5, 2003. Available online at <http://forge.gridforum.org/projects/ur-wg>.
- [OGSA] S. Tuecke et al., “*Open Grid Services Infrastructure Version 1.0*”, The Global Grid Forum June 27 2003.
- [RFC1034] Mockapetris, P., “*Domain Names – Concepts and Facilities*”, IETF RFC, January 1999.
- [RFC2119] S. Bradner, “*Key Words for Use in RFCs to Indicate Requirement Levels*”, RFC 2119, March 1997.
- [URWG\_XSD] R. Lepro et al., “*Usage Record XML Format Schema version 1.12*”, The Global Grid Forum September 3 2003. Available online at [http://forge.gridforum.org/tracker/index.php?group\\_id=90&atid=414](http://forge.gridforum.org/tracker/index.php?group_id=90&atid=414)
- [XML] Bray, T., et al, “*Extensible Markup Language (XML) 1.0 (Second Edition)*”, 6 October 2000.
- [XML\_C14N] Siddiqui, Bilal, “*XML Canonicalization*”, Sept. 18, 2002. Available online at <http://webservices.xml.com/pub/a/ws/2002/09/18/c14n.html>.
- [XML\_DSIG] Mark Bartel et al, “*XML Signature Syntax and Processing*”, World Wide Web Consortium Recommendation, February 2002.
- [XML\_SCHEMA] D. Beech, M. Maloney, N. Mendelshohn, “*XML Schema Part 1: Structures Working Draft*”, April 2000.

## Appendix A – Current Practices Survey

### A-1 Survey Background

In order to allow users to execute application on remote computer systems using some mechanism established between sites, sites should be able to provide the resource usage information in commonly defined record format. Typically, each site has its own accounting record and resource attributes to describe the resources that are being tracked at individual sites. Thus, the common terms to describe a resource attributes must be identified for all resources that are being tracked at sites to exchange resource usage information between sites. To accomplish this goal, a survey was conducted to collect all the resources that are being tracked and terms that are used to describe resources at different sites. Based on responses received from sites (NASA, NPACI, PNNL, NCSA, and ANL), resources are grouped together to show the terms used to describe the same resources in following tabularized "Survey Results" section.

Based on survey results, a proposed common set of Usage Record fields are identified with the common terms to describe the resources. This proposed set doesn't limit the extension of the set as the new and/or additional resource attributes are encountered. An individual sites is not required to report all the resource attributes listed in the proposed set but should report resources that are currently being tracked at the individual site. These survey results are intended for use by other groups in the Grid community to reference resources that are currently being tracked and indicate common terms used to describe such resources.

### A-2 Survey Results

Terms used at different sites	Globus Resource Specification Language (RSL)	Data type	Reference/Description
LOGIN_ NAME, user, userName, username, UserName, USERNAME		Text/char	User's login name corresponding to user Id in /etc/ passwd file
uid, userId		Int	User identification number from the /etc/ passwd file
Type		String	This field indicates the type of transaction being recorded, such as doWithdrawal, doDeposit, doTransfer, modifyAllocation, deleteAccount, etc.
AuthName		String	Authorized userid performing the transaction
ACCOUNT, project, AccountName, projectName, GROUPNAME	project	Text/char	Users account name where usage will be charged
projid, projectName		int	The account ID
JOB_ ID, jid, jid_ num, jobId, jobid		Number/int	Job id where job was submitted to the batch queue.
Session_ id		Number	session id from the originating system

Terms used at different sites	Globus Resource Specification Language (RSL)	Data type	Reference/Description
Id		char	Identifier indicating the job_id, session_id, reservation_id, quote_id, allocation_id, etc according to context
Pid		Number/ int	The process identifier assigned by the operating system during the life of the process.
client, hostname		Text/ char	Name of the system job was executed.
Machine		String	Machine name (This could be a list of machines (systems) for a job which spans clusters and each machine could be a composite name composed of the host, partition, cluster, site, and/ or enterprise)
QUEUE_ID		Text	Queueing system identification code. NQS id, LoadLeveler cluster id or LSF id
QUEUE, qname, queue, Queue, queueName	queue	Text/ char	Queue name where job was executed. (LSF - job was submitted)
QWAIT		Number	Queue wait time for batch jobs
QUEUE_DATE, submitTime, QueueTime		Number/ long	The date the job was queued to the batch system. Number of seconds since the Epoch in GMT.
JOB_QUEUE_DATE		Date	This is the date the job was submitted in the date format
START_DATE, start_time, beginTime, StartTime		Number/ long	The date the job was started by the system. Number of seconds since the Epoch in GMT
JOB_START_DATE		Date	This is the date the job started running in date format.
END_DATE, end_time, Event Time, EndTime, finishTime		Number/ long	The date the job was completed by the system. Number of seconds since the Epoch in GMT
JOB_END_DATE		Date	This is the date the job ended in date format.
REQUESTED_PROCS, ncpus, Processors, limitNpe, numProcessors	count	Number/ int	Number of processors requested at job submission time.
nprocs, peakNpe, maxNumProcesors, MAXPROCS		Number/ int	The number of CPU's used.
MINPROCS		Number/ int	

Terms used at different sites	Globus Resource Specification Language (RSL)	Data type	Reference/Description
NODES, Nodes		Number	Cumulative sum of all nodes allocated to the job. (number_of_nodes times cpu_per_node)
num_nodes		int	Number of nodes used. $\max((\text{cputime} + \text{process\_per\_node} - 1) / \text{process\_per\_node}, (\text{memory} + \text{mb\_per\_node} - 1) / \text{mb\_per\_node})$
Nodemask		char	A hexadecimal string representing the bit mask specifying the nodes (a pair of processors) to be associated with this job.
NodeType		String	Type of node might factor into performance and charge rate
MAXPAR		Number	Maximum node partition. Largest number of processors allocated to parallel applications within the job. On all systems except Cray T3E systems this number will be the same as NODES. On Cray T3E systems multiple parallel applications per job can be run, therefore, MAXPAR will describe the largest number of NODES allocated for the entire job.
Cpupercnt		percent	The maximum percentage of a cpu which the job used. A value of 100 means 1 cpu. This cannot be set, it is only reported.
ProconsumptionRate		Number	Percentage of Total CPU used for prorating charge – a decimal number between 0 and 1
CPU_TIME, cput, connect_time, cputime, CPUTime, cpuTime		Number/ long	CPU time used by all processes of job
CONNECT_TIME		Number	Connect time for interactive session
Pcput	max_cpu_time	long	Maximum amount of CPU time used by any single process in the job.
user_cpu, ru_utime		long/ double	The user CPU time in seconds
sys_cpu, ru_stime		long/ double	The system CPU time in seconds
interactive_cpu		double	Interactive cpus used (user_cpu + sys_cpu)

Terms used at different sites	Globus Resource Specification Language (RSL)	Data type	Reference/Description
Batch_cpu		double	Batch cpus used (user_cpu + sys_cpu)
mt_user_cpu		double	The total user CPU time in seconds in multitasking (MT) queue (user_cpu)
mt_sys_cpu		double	total system CPU time in seconds in MT (sys_cpu)
mt_connect		double	total connect time in MT queue
mt_nconnect		double	The sum of (connect_time * nprocs) in each of the CPU's in MT queue.
mt_non- mt		double	number of seconds which are not multitasking in MT queue. (user_cpu - mt_nconnect)
WALLCLOCK, walltime, Wallclock, runTime	max_wall_time	Number/long	Wall clock time which elapsed while the job was in running state. For clusters where a node is exclusively allocated the wallclock is multiplied by the number of processors yielding wallclock processor hours. Therefore, on an IBM SP system this is actually the wallclock node hours or "wallclock * number of cpus"
REQUESTED_TIME, limitRuntime		Number/long	Amount of time requested at queue submission time for resource time, either wallclock time for parallel jobs or cpu time for vector/ DMP systems
MAXMEMORY, high_mem		Number/long	Memory high water mark for entire job
MEMORY		Number	Memory usage in Kcore- hours
REQUESTED_MEM, limitMem		Number	Amount of memory requested at job submission time
Pmem		size	Maximum amount of virtual memory (workingset) used by any single process of the job.
vmem, memory, maxRSwap, mem, Memory, peakMem	max_memory	size	Maximum amount of virtual memory used by all concurrent processes in the job.
workingset		size	Maximum amount of physical memory used by any single process of the job.

Terms used at different sites	Globus Resource Specification Language (RSL)	Data type	Reference/Description
maxRMem		size	Maximum amount of resident memory used by all processes in the job.
NUMMPPJOBS		Number	Number of parallel applications run in this job. On all systems except Cray T3E systems this number will be one. On Cray T3E systems multiple parallel applications per job can be run, therefore, NUMMPPJOBS will describe the number of parallel applications run in this job.
kword_ sec, kword- minutes		double	memory integral in seconds and in minutes( kword_ sec/ 60)
I_ O, Mbytes I/ O, io_ kbytes, IO		Number/ double	IO usage in megabytes or kbytes transferred
IOread		number/ double	total number of bytes read by the job
IOwrite		Number/ double	total number of bytes written by the job
Iobread		Number/ double	total number of bytes read by the job to block devices
Iobwrite		Number/ double	total number of bytes written by the job to block devices.
io_ physreq( Physical I/ O)		double	The number of physical I/ O requests
DISK, Disk		Number	Disk storage used or Disk Charge in units defined by CPU: disk blocks or other.
Network		int	Network used (withdrawals) or requested (reservations) by job [could be AVG, TOT, or MAX]
EXPF		Number	Expansion factor. (QWAIT+WALLCLOCK)/WALLCLOCK This gives whether queue times are proportional to job size
File		Size	The largest size of any single file that may be created by the job.
Fsblkused		long	The number of file system blocks consumed during the job.
Nice		int	The nice value under which the job is to be run.
PRIORITY		Number	Priority weight value.

Terms used at different sites	Globus Resource Specification Language (RSL)	Data type	Reference/Description
JOB_COMP_STATUS, Status, jStatus		Number	Number representing completion status of the job.
ExitStatus		int	UNIX exit status of the job.
KillReason		Text/ char	if killed, reason the job was killed(npe, mem, cputime, runtime)
command, Executable		char	The name of the executable or system command
APP_NAME, jobName, JobName	executable	Text/ char	Job or Application name.
Class		String	Class of job (batch, interactive, etc.)
JobType		String	Here you could distinguish between RMS job types, NQS, PBS, LSF, LL, etc.
QOS		String	Quality of Service
Total_charge, Charge		double, float	The total charge of the job in system's billing unit.. Amount debited or credited to account or allocation/ reservation/ quotation
SU		Number	Total charge for this job in System Billing Units. (seconds)

### A-3 Datatypes

The resource data type corresponds to the following units.

Date: Date in human readable format

Number: specifies the maximum amount in terms or time period, integer, long integer, or double.

Text: Specifies the character representation of string

time: specifies a maximum time period the resource can be used. Time is expressed in seconds as an integer, or in the form [[ hours:] minutes:] seconds[. milliseconds]

size: specifies the maximum amount in terms of bytes or words. It is expressed in the form integer[ suffix]. The suffix is a multiplier defined in the following table, The size of a word is the word size on the execution host.

- b or w bytes or words.
- kb or kw Kilo (1024) bytes or words.
- mb or mw Mega (1,048,576) bytes or words.
- gb or gw Giga (1,073,741,824) bytes or words.

unitary: The maximum amount of a resource which is expressed as a simple integer.

int: specifies the numeric representation in integer

long: specifies the numeric representation in long integer.

double: specifies the numeric representation in float.

char: specifies the character representation.

String: specifies the character representation.

percent: specifies the numeric representation as a percentage.

**A-4 Common Usage Record Fields**

<b>Resource Name</b>	<b>Descriptions</b>
Username	User's login name corresponding to user Id in /etc/ passwd file
ProjectName	Users account name where usage will be charged
JobId	Job id where job was submitted to the batch queue for batch jobs. Process Id for the interactive jobs
Queue	Queue name where job was executed or submitted depending on the batch system
GridId	User's global unique Id that identifies the user. Distinguish Name in the user's X509 certificate
FromHost	System name where job was submitted from
execHost	System name where job ran on.
StartTime	The date job started running in date time format (UTC timezone)
EndTime	The date job completed in date time format (UTC timezone)
Processors	Number of processors either used or requested that each center uses for billing purpose..
NumNodes	Number of nodes used.
Cputime	CPU time used, summed over all processes in the job.
Walltime	Wall clock time which elapsed while the job was in the running state.
Memory	Maximum amount of virtual memory used by all concurrent processes in the job.
Disk	Disk storage used or Disk Charge in units defined by CPU: disk blocks or other.
Network	Network used (withdrawals) or requested (reservations) by job [could be AVG, TOT, or MAX]
jobName	Job or Application name
Status	Number representing completion status of the job.
Charge	The total charge of the job in system's allocation unit

## Appendix B – Common Usage Record Properties

### Table Column Interpretations

In the property tables, the columns are interpreted to have the following meanings:

- Element Name: Name of the XML element (xsd:element)
- Type: Data type defined by Sections 1 & 2 and the W3C XML Schema specification
- Description: Brief description of the meaning of the property

Simple (unstructured) usage record properties are enumerated in the table below.

Element Name	Type	Description
Charge	xsd:float	Total charge for this usage
CPUDuration	xsd:duration	CPU Time used in seconds, summed over all processes in the job
Disk	xsd:positiveInteger	Disk storage for this usage
EndTime	xsd:dateTime	Date and time when usage ended
Host	xsd:token	Host on which the job ran
MachineName	xsd:token	Name of the machine or cluster on where usage was incurred
Memory	xsd:positiveInteger	Memory for this usage
Network	xsd:positiveInteger	Network usage rate of transfer
NodeCount	xsd:positiveInteger	Number of nodes used
Processors	xsd:positiveInteger	Number of processors used
ProjectName	xsd:token	Project or account name
Queue	xsd:token	Queue name
StartTime	xsd:dateTime	Date and time when usage started
Status	xsd:token	Completion Status
SubmitHost	xsd:token	The hostname from which request was submitted
Swap	xsd:positiveInteger	Swap for this usage
TimeDuration	xsd:duration	A specific interval relevant to this usage. Metric and type attributes provide additional information
TimeInstant	xsd:dateTime	A specific instant relevant to this usage. Metric and type attributes provide additional details.
WallDuration	xsd:duration	Wallclock duration of usage in seconds (sometimes referred to as WallClockTime)

Note that this material informs the XML specification but does not constrain it. The common properties listed here are derived from the material presented in Appendix A: Current Practices Survey. The actual XML schema, Appendix D, is derived from this material. Inconsistencies may arise because of the evolutionary nature of this document. The final authority on the Usage Record XML format is the XML schema itself. The body of this document contains the definitive supporting text for the XML specification. Appendices A and B are included for historical reference.

**Appendix C – Units of Measure for Volume**

<b>Abbreviation</b>	<b>Definition</b>	<b>Quantity</b>
b	bit	1 bit
B	byte	1 byte
KB	kilobyte	1024 bytes
MB	megabyte	1024 KB
GB	gigabyte	1024 MB
TB	terabyte	1024 GB
PB	petabyte	1024 TB
EB	exabyte	1024 PB
Kb	kilobit	1024 bits
Mb	megabit	1024 Kb
Gb	gigabit	1024 Mb
Tb	terabit	1024 Gb
Pb	petabit	1024 Tb
Eb	exabit	1024 Pb

**Appendix D - Usage Record XML Schema**

```

<?xml version="1.0" encoding="UTF-8" ?>
- <xsd:schema attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://schema.ogf.org/urf/2003/09/urf"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:urf="http://schema.ogf.org/urf/2003/09/urf"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">Usage Record Working Group XML Schema
definition</xsd:documentation>
  </xsd:annotation>
  <xsd:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/xmldsig-core/xmldsig-core-schema.xsd" />
  <xsd:complexType name="UsageRecordType">
    <xsd:sequence>
      <xsd:element maxOccurs="1" minOccurs="1" ref="urf:RecordIdentity" />
      <xsd:element maxOccurs="1" minOccurs="0" ref="urf:JobIdentity" />
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="urf:UserIdentity" />
      <xsd:element maxOccurs="1" minOccurs="0" ref="urf:JobName" />
      <xsd:element maxOccurs="1" minOccurs="0" ref="urf:Charge" />
      <xsd:element maxOccurs="1" minOccurs="1" ref="urf:Status" />
      <xsd:choice maxOccurs="unbounded" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>The elements grouped together in this choice may be
represented within a usage record multiple times. Each of these appearances
must be differentiated by the metric and/or type associated with the
element.</xsd:documentation>
        </xsd:annotation>
        <xsd:element ref="urf:Disk" />
        <xsd:element ref="urf:Memory" />
        <xsd:element ref="urf:Swap" />
        <xsd:element ref="urf:Network" />
        <xsd:element ref="urf:TimeDuration" />
        <xsd:element ref="urf:TimeInstant" />
        <xsd:element ref="urf:ServiceLevel" />
      </xsd:choice>
      <xsd:choice maxOccurs="unbounded" minOccurs="0">
        <xsd:element maxOccurs="1" minOccurs="0" ref="urf:WallDuration" />
        <xsd:sequence maxOccurs="1" minOccurs="0">
          <xsd:element maxOccurs="2" minOccurs="0" ref="urf:CpuDuration" />
        </xsd:sequence>
        <xsd:element maxOccurs="1" minOccurs="0" ref="urf:NodeCount" />
        <xsd:element maxOccurs="1" minOccurs="0" ref="urf:Processors" />
        <xsd:element maxOccurs="1" minOccurs="0" ref="urf:EndTime" />
        <xsd:element maxOccurs="1" minOccurs="0" ref="urf:StartTime" />
        <xsd:element maxOccurs="1" minOccurs="0" ref="urf:MachineName" />
        <xsd:element maxOccurs="1" minOccurs="0" ref="urf:SubmitHost" />
        <xsd:element maxOccurs="1" minOccurs="0" ref="urf:Queue" />
        <xsd:sequence maxOccurs="1" minOccurs="0">
          <xsd:element maxOccurs="unbounded" minOccurs="0" ref="urf:ProjectName" />
        </xsd:sequence>
        <xsd:sequence maxOccurs="1" minOccurs="0">
          <xsd:element maxOccurs="unbounded" minOccurs="0" ref="urf:Host" />
        </xsd:sequence>
        <xsd:sequence maxOccurs="1" minOccurs="0">
          <xsd:choice maxOccurs="unbounded" minOccurs="0">
            <xsd:element ref="urf:PhaseResource" />
            <xsd:element ref="urf:VolumeResource" />
            <xsd:element ref="urf:Resource" />
            <xsd:element ref="urf:ConsumableResource" />
          </xsd:choice>
        </xsd:sequence>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>

```

```

    </xsd:sequence>
  </xsd:choice>
</xsd:sequence>
</xsd:complexType>
<xsd:element abstract="true" name="Usage" type="urf:UsageRecordType" />
<xsd:element name="UsageRecord" substitutionGroup="urf:Usage"
type="urf:UsageRecordType" />
<xsd:element name="JobUsageRecord" substitutionGroup="urf:Usage">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:extension base="urf:UsageRecordType" />
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
<xsd:element name="UsageRecords">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="urf:Usage" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!-- Common properties that may be measured with several different metrics
within the same usage record
-->
  <xsd:element name="Network">
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base="xsd:positiveInteger">
          <xsd:attribute ref="urf:description" use="optional" />
          <xsd:attributeGroup ref="urf:intervallicVolume" />
          <xsd:attribute default="total" ref="urf:metric" use="optional" />
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Disk">
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base="xsd:positiveInteger">
          <xsd:attribute ref="urf:description" use="optional" />
          <xsd:attributeGroup ref="urf:intervallicVolume" />
          <xsd:attribute default="total" ref="urf:metric" use="optional" />
          <xsd:attribute ref="urf:type" use="optional" />
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>
- <xsd:element name="Memory">
- <xsd:complexType>
- <xsd:simpleContent>
- <xsd:extension base="xsd:positiveInteger">
  <xsd:attribute ref="urf:description" use="optional" />
  <xsd:attributeGroup ref="urf:intervallicVolume" />
  <xsd:attribute default="total" ref="urf:metric" use="optional" />
  <xsd:attribute ref="urf:type" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
  <xsd:element name="Swap">
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base="xsd:positiveInteger">

```

```

<xsd:attribute ref="urf:description" use="optional" />
<xsd:attributeGroup ref="urf:intervallicVolume" />
<xsd:attribute default="total" ref="urf:metric" use="optional" />
<xsd:attribute ref="urf:type" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="NodeCount">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:positiveInteger">
<xsd:attribute ref="urf:description" use="optional" />
<xsd:attribute default="total" ref="urf:metric" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="Processors">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:positiveInteger">
<xsd:attribute ref="urf:description" use="optional" />
<xsd:attribute ref="urf:metric" use="optional" />
<xsd:attribute name="consumptionRate" type="xsd:float" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="TimeDuration">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:duration">
<xsd:attribute ref="urf:type" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="TimeInstant">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:dateTime">
<xsd:attribute ref="urf:type" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="ServiceLevel">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:token">
<xsd:attribute ref="urf:type" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<!-- This element should appear at most twice within a usage record, with
differing values for usageType for each appearance
-->
<xsd:element name="CpuDuration">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:duration">

```

```

<xsd:attribute ref="urf:description" use="optional" />
<xsd:attribute name="usageType">
<xsd:simpleType>
<xsd:restriction base="xsd:token">
<xsd:enumeration value="user" />
<xsd:enumeration value="system" />
</xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<!-- These common properties should appear at most once within a usage
record, rather than at most once per metric per usage record
-->
<xsd:element name="WallDuration">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:duration">
<xsd:attribute ref="urf:description" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="EndTime">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:dateTime">
<xsd:attribute ref="urf:description" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="StartTime">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:dateTime">
<xsd:attribute ref="urf:description" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="MachineName">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="urf:domainNameType">
<xsd:attribute ref="urf:description" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="SubmitHost">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="urf:domainNameType">
<xsd:attribute ref="urf:description" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="Host">
<xsd:complexType>

```

```

<xsd:simpleContent>
<xsd:extension base="urf:domainNameType">
<xsd:attribute ref="urf:description" use="optional" />
<xsd:attribute default="false" name="primary" type="xsd:boolean" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="Queue">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:string">
<xsd:attribute ref="urf:description" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="JobName">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:string">
<xsd:attribute ref="urf:description" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="ProjectName">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:string">
<xsd:attribute ref="urf:description" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="Status">
<xsd:annotation>
<xsd:documentation>
  Minimum required set =
    {Aborted, Completed, Failed, Held, Queued, Started, Suspended}
</xsd:documentation>
</xsd:annotation>
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:token">
<xsd:attribute ref="urf:description" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="Charge">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:float">
<xsd:attribute ref="urf:description" use="optional" />
<xsd:attribute ref="urf:unit" use="optional" />
<xsd:attribute name="formula" type="xsd:string" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<!-- identity elements
-->

```

```

    <xsd:element name="JobIdentity">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element maxOccurs="1" minOccurs="0" name="GlobalJobId" type="xsd:string"
/>
          <xsd:element maxOccurs="1" minOccurs="0" name="LocalJobId" type="xsd:string"
/>
          <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="0" name="ProcessId"
type="xsd:string" />
          </xsd:sequence>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="UserIdentity">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element maxOccurs="1" minOccurs="0" name="LocalUserId" type="xsd:string"
/>
          <xsd:element maxOccurs="1" minOccurs="0" name="GlobalUserName"
type="xsd:string" />
          <xsd:element maxOccurs="1" minOccurs="0" ref="ds:KeyInfo" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="RecordIdentity">
      <xsd:complexType>
        <xsd:sequence maxOccurs="1" minOccurs="0">
          <xsd:element ref="ds:KeyInfo" />
        </xsd:sequence>
        <xsd:attribute name="recordId" type="xsd:token" use="required" />
        <xsd:attribute name="createTime" type="xsd:dateTime" use="optional" />
      </xsd:complexType>
    </xsd:element>
    <!-- Extensibility Framework
-->
    <xsd:element name="Resource" type="urf:ResourceType" />
    <xsd:element name="ConsumableResource" type="urf:ConsumableResourceType" />
    <xsd:element name="PhaseResource">
      <xsd:complexType>
        <xsd:complexContent>
          <xsd:extension base="urf:ConsumableResourceType">
            <xsd:attribute ref="urf:phaseUnit" use="optional" />
          </xsd:extension>
        </xsd:complexContent>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="VolumeResource">
      <xsd:complexType>
        <xsd:complexContent>
          <xsd:extension base="urf:ConsumableResourceType">
            <xsd:attribute ref="urf:storageUnit" use="optional" />
          </xsd:extension>
        </xsd:complexContent>
      </xsd:complexType>
    </xsd:element>
    <!-- Create a generic consumable resource. Carries the units attribute
-->
    <xsd:complexType name="ConsumableResourceType">
      <xsd:simpleContent>
        <xsd:extension base="xsd:float">
          <xsd:attribute name="units" type="xsd:string" use="optional" />
          <xsd:attribute ref="urf:description" use="optional" />

```

```

</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<!-- Create a generic resource type
-->
<xsd:complexType name="ResourceType">
<xsd:simpleContent>
<xsd:extension base="xsd:string">
<xsd:attribute ref="urf:description" use="optional" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<!-- Global Attribute Definitions
-->
<xsd:attribute name="description" type="xsd:string" />
<!-- Units of measure attribute definitions
-->
<xsd:attribute name="unit" type="xsd:token" />
<xsd:attribute name="storageUnit">
<xsd:simpleType>
<xsd:restriction base="xsd:token">
<xsd:enumeration value="b" />
<xsd:enumeration value="B" />
<xsd:enumeration value="KB" />
<xsd:enumeration value="MB" />
<xsd:enumeration value="GB" />
<xsd:enumeration value="PB" />
<xsd:enumeration value="EB" />
<xsd:enumeration value="Kb" />
<xsd:enumeration value="Mb" />
<xsd:enumeration value="Gb" />
<xsd:enumeration value="Pb" />
<xsd:enumeration value="Eb" />
</xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
<xsd:attribute name="phaseUnit" type="xsd:duration" />
<xsd:attributeGroup name="intervallicVolume">
<xsd:attribute ref="urf:storageUnit" use="optional" />
<xsd:attribute ref="urf:phaseUnit" use="optional" />
</xsd:attributeGroup>
<!-- End units attributes
-->
<xsd:attribute name="metric" type="xsd:token" />
<xsd:attribute name="type" type="xsd:token" />
<!-- Simple type definitions used to constrain values of attributes
-->
<xsd:simpleType name="domainNameType">
<xsd:restriction base="xsd:string">
<xsd:pattern value="([a-zA-Z0-9][a-zA-Z0-9'\-']*[a-zA-Z0-9]\.)*([a-zA-Z0-9][a-zA-Z0-9'\-']*[a-zA-Z0-9])?" />
<xsd:maxLength value="255" />
</xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```