

GWD-R  
UR-WG  
ur-wg@ogf.org

A. Cristofori (editor), IGI Bologna  
J. K. Nilsen (editor), Univ. of Oslo  
J. Gordon, STFC RAL London  
M. Jones, Univ. of Manchester  
J. A. Kennedy, RZG Munich  
R. Müller-Pfefferkorn, TU Dresden  
January 2013

## Usage Record – Format Recommendation

### Status of This Document

Group Working Draft (GWD)

### Obsoletes

This document obsoletes GFD-R-P.098[1].

### Copyright Notice

Copyright © Open Grid Forum (2006-2013). Some Rights Reserved. Distribution is unlimited.

### Abstract

For resources to be shared, sites must be able to exchange basic accounting and usage data in a common format. This document describes a common format with which to exchange basic accounting and usage data on different resources. This record format is intended to facilitate the sharing of usage information, particularly in the area of the accounting of jobs, computing, memory, storage and cloud usage but with a structure that allows an easy extension to other resources. This document describes the Usage Record components both in natural language form and annotated XML.

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 communication mechanisms employed to exchange the records, i.e. transport layer, framing, authentication, integrity, etc.

## Contents

Abstract . . . . .	1
Contents . . . . .	2
1 Introduction . . . . .	5
1.1 Context . . . . .	5
1.1.1 History . . . . .	5
1.1.2 What This Document Is Not . . . . .	5
1.2 Format of the Record Specification . . . . .	6
2 Conventions Used in this Document . . . . .	7
2.1 Notational Conventions . . . . .	7
2.2 Meta Properties . . . . .	7
2.2.1 Description . . . . .	7
2.2.2 Metric . . . . .	7
2.2.3 Time Stamps . . . . .	7
2.3 Conventions . . . . .	7
2.4 Supported Data Types . . . . .	8
3 RecordIdentityBlock . . . . .	9
3.1 RecordId . . . . .	9
3.2 CreateTime . . . . .	9
3.3 Site . . . . .	9
3.4 Infrastructure . . . . .	10
4 SubjectIdentityBlock . . . . .	11
4.1 LocalUserId . . . . .	11
4.2 LocalGroupId . . . . .	11
4.3 GlobalUserId . . . . .	11
4.4 GlobalGroupId . . . . .	12
4.5 GlobalGroupAttribute . . . . .	12
5 ComputeUsageBlock . . . . .	13
5.1 CpuDuration . . . . .	13
5.2 WallDuration . . . . .	13
5.3 StartTime . . . . .	13
5.4 EndTime . . . . .	14
5.5 ExecutionHost . . . . .	14
5.5.1 Hostname . . . . .	14
5.5.2 ProcessId . . . . .	15
5.5.3 Benchmark . . . . .	15
5.6 HostType . . . . .	16

5.7	Processors	16
5.8	NodeCount	16
5.9	ExitStatus	17
5.10	Charge	17
6	JobUsageBlock	18
6.1	GlobalJobId	18
6.2	LocalJobId	18
6.3	JobName	18
6.4	MachineName	19
6.5	SubmitHost	19
6.6	Middleware	19
6.7	Queue	20
6.8	TimeInstant	20
6.9	ServiceLevel	20
6.10	Status	21
7	MemoryUsageBlock	22
7.1	MemoryClass	22
7.2	MemoryResourceCapacityUsed	22
7.3	MemoryLogicalCapacityUsed	22
7.4	MemoryResourceCapacityAllocated	23
7.5	StartTime	23
7.6	EndTime	23
7.7	Host	23
7.8	HostType	23
7.9	Charge	24
8	StorageUsageBlock	25
8.1	StorageShare	25
8.2	StorageMedia	25
8.3	StorageClass	25
8.4	DirectoryPath	26
8.5	FileCount	26
8.6	StorageResourceCapacityUsed	26
8.7	StorageLogicalCapacityUsed	27
8.8	StorageResourceCapacityAllocated	27
8.9	StartTime	28
8.10	EndTime	28
8.11	Host	28
8.12	HostType	28
8.13	Charge	28

9	CloudUsageBlock . . . . .	29
9.1	LocalVirtualMachinel . . . . .	29
9.2	GlobalVirtualMachinel . . . . .	29
9.3	Status . . . . .	29
9.4	SuspendDuration . . . . .	30
9.5	Imagel . . . . .	30
9.6	MachineName . . . . .	30
9.7	SubmitHost . . . . .	30
9.8	TimeInstant . . . . .	30
9.9	ServiceLevel . . . . .	31
10	Attributes Matrix . . . . .	32
11	Field Summaries . . . . .	36
11.1	RecordIdentityBlock . . . . .	36
11.2	SubjectIdentityBlock . . . . .	36
11.3	ComputeUsageBlock . . . . .	36
11.4	JobUsageBlock . . . . .	37
11.5	MemoryUsageBlock . . . . .	37
11.6	StorageUsageBlock . . . . .	38
11.7	CloudUsageBlock . . . . .	39
12	Examples . . . . .	40
12.1	Full example . . . . .	40
12.2	Grid example . . . . .	42
12.3	Cloud example . . . . .	43
12.4	Local example . . . . .	43
12.5	Minimal examples . . . . .	44
12.5.1	Job record . . . . .	44
12.5.2	Storage record . . . . .	45
13	XSD Schema . . . . .	46
14	Security Considerations . . . . .	53
15	Contributors . . . . .	54
16	Acknowledgments . . . . .	55
17	Intellectual Property Statement . . . . .	56
18	Disclaimer . . . . .	56
19	Full Copyright Notice . . . . .	56
20	References . . . . .	57

# 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 accounting of different use-cases involves recording:

- General properties related to the record itself
- Properties related to the consumer of the resources
- Usage of one or more resources.

Hence, the usage record schema is made up of a set of blocks for general properties, consumer and distinct resources.

### 1.1.1 History

Before the definition of UR-2.0 different usage record definitions building on the job accounting definition of UR-1.0 [1], started to surface to describe various resource usages, such as the EMI Compute Accounting Record CAR [2], the EMI Storage Accounting Record StAR [3], the EGI Cloud Usage Record CUR <sup>1</sup> and the Storage Accounting Implementation SAI [4]. The definition of UR-2.0 came from the experiences of these record definitions and is built as an easily extensible superset of these new usage records.

### 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

---

<sup>1</sup><https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:Scenario4>

will meet the needs of all projects and resource providers. This document does not address summary records, “grid job” records, consolidated records, or anything other than an atomic resource consumption instantiation. Sufficient resource 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 (e.g., what grids do with the atomic data) are beyond the scope of this document. This definition of UR-2.0 do not enter into detail of how the Usage Record should be used or the way records are transported from the information producers to its consumers. Neither does it enter into implementation details of the accounting sensors.

## 1.2 Format of the Record Specification

UR-2.0 defines the building blocks necessary for the accounting of different resources. This is achieved by combining the blocks in different ways. In this document all the fields and blocks that are part of UR-2.0 will be described. Additionally, example records for accounting of storage, grid, cloud are given. This record specification is aiming at being as general as possible. Different communities should then create their own profiles where they specify the combination of blocks and fields required for their implementation.

## 2 Conventions Used in this Document

### 2.1 Notational Conventions

The key words “MUST” “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” are to be interpreted as described in RFC 2119 [5], except that the words do not appear in uppercase.

### 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 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.

#### 2.2.3 Time Stamps

Time stamps should follow the ISO 8601[6] standard as well. This includes enumerating the time zone, as specified in the standard.

### 2.3 Conventions

Unless otherwise stated, all fields are optional. A required field is only required if the corresponding block is present.

## 2.4 Supported Data Types

1. String: Data of this type has no required restrictions on the length or available characters.
2. Integer
3. Positive Integer: Data of this type must have a value of zero or greater.
4. Non-zero Integer: Data of this type must have a value of one or greater.
5. Float
6. Timestamp: Data of this type must comply with the UTC time zone format specified in ISO 8601.
7. DomainName: Data of this type must comply with RFC 1034[7] 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 RecordIdentityBlock

This block contains the properties related to the identity of the usage record itself.

### Example

```
<ur:RecordIdentityBlock>  
<!! Identity properties go in here -->  
</ur:RecordIdentityBlock>
```

### 3.1 RecordId

A record identity uniquely defines a record in the set of all usage records of the accounting context.

- *RecordId* MUST be present in the *RecordIdentityBlock*.
- *RecordId* MUST have the type string.

### Example

```
<ur:RecordId>"host.example.org/ur/87912469269276"</ur:RecordId>
```

### 3.2 CreateTime

The time when this particular Usage Record was created.

- *CreateTime* MUST be present in the *RecordIdentityBlock*.
- *CreateTime* MUST be an ISO 8601:2004 timestamp.

### Example

```
<ur:CreateTime>2013-05-09T09:06:52Z</ur:CreateTime>
```

### 3.3 Site

The site at which the resource is located. This property should contain a descriptive name of the group of resources which are accounted for in the record. The *Site* value should be constructed in such a way that it is unique within the context where it is used.

- *Site* MUST be a string.

### Example

```
<ur:Site>"ACME-University"</ur:Site>
```

### 3.4 Infrastructure

The infrastructure where the resource was used (e.g., EGI or OSG).

- *Infrastructure* MUST be a string.
- The attribute *description* SHOULD be used to give additional information on the used infrastructure.

#### Example

```
<ur:Infrastructure>"OSG"</ur:Infrastructure>
```

## 4 SubjectIdentityBlock

This block contains the properties related to the identity of the subject accounted for. As all fields in this block are optional, this block SHOULD be associated with a profile that defines at least one field that must be present in the specific implementation.

### Example

```
<ur:SubjectIdentityBlock>  
<!! Identity properties go in here -->  
</ur:SubjectIdentityBlock>
```

### 4.1 LocalUserId

The local identity of the user accountable for the resource consumption (e.g., the Unix user).

- *LocalUserId* MUST be a string.

### Example

```
<ur:LocalUserId>"johndoe"</ur:LocalUserId>
```

### 4.2 LocalGroupId

The local group accountable for the resource consumption (e.g., Unix group).

- *LocalGroupId* MUST be a string.

### Example

```
<ur:LocalGroupId>"projectA"</ur:LocalGroupId>
```

### 4.3 GlobalUserId

The global identity of the user accountable for the resource consumption. The property should identify the user globally, such that clashes do not happen accidentally, e.g. it could be an X500 identity.

- *GlobalUserId* MUST be a string.

### Example

```
<ur:GlobalUserId>"/O=Grid/OU=example.org/CN=John Doe"</ur:GlobalUserId>
```

## 4.4 GlobalGroupId

The global group accountable for the resource consumption. The property should identify the group globally, such that clashes do not happen accidentally, e.g. using a Fully Qualified Domain Name (FQDN) to construct it. In a Grid context, this would typically be the name of the Virtual Organization (VO).

- *GlobalGroupId* MUST be a string.

### Example

```
<ur:GlobalGroupId>"binarydatapproject.example.org"</ur:GlobalGroupId>
```

## 4.5 GlobalGroupAttribute

Supplemental traits of the group property, e.g., a sub-group, role or authority. This makes it possible to account for segments of a group, while still being able to account for the group as a whole. The attribute *type* denotes the type of the group property.

- *GlobalGroupAttribute* MAY be present multiple times.
- *GlobalGroupAttribute* MUST be a string.
- If *GlobalGroupAttribute* is specified, *GlobalGroupId* MUST be present.
- If *GlobalGroupAttribute* is specified, the attribute *type* MUST exist.
- The attribute *type* MUST be a string.

### Example

```
<ur:GlobalGroupAttribute ur:type="subgroup">ukusers</ur:GlobalGroupAttribute>
```

## 5 ComputeUsageBlock

This block contains the properties related to compute usage.

### Example

```
<ur:ComputeUsageBlock>  
<!--|Compute Record properties go in here -->  
</ur:ComputeUsageBlock>
```

### 5.1 CpuDuration

The CPU time consumed. If the task ran on many cores/processors/nodes, all separate consumptions shall be aggregated in this value. This has an impact for example on MPI usage, where the consumption of all the “nodes” get aggregated into this CPU consumption.

- *CpuDuration* MUST be present in the *ComputeUsageBlock*.
- *CpuDuration* MUST contain a time duration as defined in ISO 8601:2004[6].

### Example

```
<ur:CpuDuration>PT3600S</ur:CpuDuration>
```

### 5.2 WallDuration

WallClock time elapsed during the process execution. In the case of parallel applications (like MPI) *WallDuration* might be lower than *CpuDuration*.

- *WallDuration* MUST be present in the *ComputeUsageBlock*.
- *WallDuration* MUST contain a time duration as defined in ISO 8601:2004[6].

### Example

```
<ur:WallDuration>PT3600S</ur:WallDuration>
```

### 5.3 StartTime

A timestamp indicating the time at which the measured resource consumption started. Together with *EndTime* this defines a period over which the resource has been consumed.

- *StartTime* MUST be present in the *ComputeUsageBlock*.
- *StartTime* MUST be an ISO 8601:2004 timestamp.

- The time zone may be specified as Z (UTC) or (+—)hh:mm. Time zones that are not specified are considered undetermined.

### Example

```
<ur:StartTime>2013-05-31T11:00:00</ur:StartTime>
```

## 5.4 EndTime

A timestamp indicating the time at which the measured resource consumption ended. Together with *StartTime* this defines a period over which the resource has been consumed.

- *EndTime* MUST be present in the *ComputeUsageBlock*.
- *EndTime* MUST be an ISO 8601:2004 timestamp.
- The time zone may be specified as Z (UTC) or (+—)hh:mm. Time zones that are not specified are considered undetermined.

### Example

```
<ur:EndTime>2013-05-31T12:00:00</ur:EndTime>
```

## 5.5 ExecutionHost

This property is a container for various information about the host where the application was executed. As an example, in case of MPI processes, more than one *ExecutionHost* property can be specified.

- *ExecutionHost* MAY be present multiple times.
- *ExecutionHost* MUST NOT have a value.
- *ExecutionHost* MUST contain at least one value.

### Example

```
<ur:ExecutionHost>
  <!-- Various host properties go in here -->
</ur:ExecutionHost>
```

### 5.5.1 Hostname

The name of the *ExecutionHost*.

- *Hostname* MUST be a string.

- *Hostname* MUST be under *ExecutionHost*.
- *Hostname* MUST be present if *ExecutionHost* is present.
- The attribute *primary* MAY be present in this element.
- The attribute *primary* MUST be a boolean.

### Example

```
<ur:Hostname primary=false>"compute-0-1.abel.uio.no"</ur:Hostname>
```

### 5.5.2 ProcessId

The process ID of the process running at the host. For example this could be used in case of MPI processes that use multiple hosts.

- *ProcessId* MUST be under *Host*.
- *ProcessId* MAY be present multiple times in *Host*.
- *ProcessId* MUST be a Non-zero Integer.

### Example

```
<ur:ProcessId>1042</ur:ProcessId>
```

### 5.5.3 Benchmark

This element is used to insert computing benchmarks associated to the host.

- *Benchmark* MUST be under *Host*.
- *Benchmark* MAY be present multiple times in *Host*.
- *Benchmark* MUST be a float.
- The *type* attribute MUST be present in the element.
- The *type* attribute type MUST be a string.
- At least the following types should be supported:
  - *Si2k* – SpecInt2000
  - *Sf2k* – SpecFloat2000
  - *HEPSPEC* – HEPSpec

### Example

```
<ur:Benchmark type="Si2k">3.14</ur:Benchmark>
```

## 5.6 HostType

The type of service according to a namespace-based classification. The namespace MAY be related to a middleware name, an organization or other concepts. `org.ogf.glue.*` is reserved for types defined by the OGF GLUE Working Group<sup>2</sup>.

- The *HostType* field type MUST be a string.

### Example

```
<ur:HostType>"org.nordugrid.ares"</ur:HostType>
```

## 5.7 Processors

The number of processors used or requested. A processor definition may be dependent on the machine architecture. Typically, *Processors* is equivalent to the number of physical CPUs used. For example, if a process uses two cluster “nodes”, each node having 16 CPUs each, the total number of processors would be 32. In addition, a processor can consist of several cores which may be used independently. In the example above the use of dual-core processors would thus increase the number to 64.

- *Processors* MUST be of type Non-zero Integer.

### Example

```
<ur:Processors>1</ur:Processors>
```

## 5.8 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 one physical node (machine) with 16 processors.

- *NodeCount* MUST be of type Non-zero Integer.

### Example

```
<ur:NodeCount>1</ur:NodeCount>
```

---

<sup>2</sup><https://forge.ogf.org/sf/go/projects.glue-wg/wiki>



## 5.9 ExitStatus

This element allows to specify the numeric exit status value for the application.

- The *ExitStatus* property SHOULD be present in the *ComputeUsageBlock*.
- The value of *ExitStatus* MUST be of type Integer

## 5.10 Charge

This property may be used for economic accounting purposes. It is the charge applied to the users resource usage.

The attribute *unit* defines the currency used to report the charge, while the attribute *formula* describes the cost computation formula adopted to compute the applied charge.

- *Charge* MUST be a float.
- The attribute *unit* MUST be present if *Charge* is specified.
- The attribute *unit* MUST be an ISO 4217<sup>3</sup> alphabetic code.
- The attribute *formula* MUST be present if *Charge* is specified.
- The attribute *formula* MUST be a string.

### Example

```
<ur:Charge ur:formula="42 units/CPU/day" ur:unit="USD">1.75</ur:Charge>
```

---

<sup>3</sup>[http://www.iso.org/iso/currency\\_codes](http://www.iso.org/iso/currency_codes)

## 6 JobUsageBlock

This block contains the properties related to job usage. The block is intended to be used for jobs submitted to a batch system, a Grid or something similar.

### Example

```
<ur:JobUsageBlock>  
<!!Job Record properties go in here -->  
</ur:JobUsageBlock>
```

### 6.1 GlobalJobId

The global identity of the job. The property should identify the job globally, such that clashes do not happen accidentally. This could for example be a combination of a time stamp, a local job id and a host name.

- The *GlobalJobId* field type MUST be a string.

### Example

```
<ur:GlobalJobId>"host.example.org/ab1234/2013-05-09T09:06:52Z"</ur:GlobalJobId>
```

### 6.2 LocalJobId

The local identity of the job. For example, this may be the ID the job got assigned by the local resource management system (LRMS).

- *LocalJobId* MUST be a string.

### Example

```
<ur:LocalJobId>"ab1234"</ur:LocalJobId>
```

### 6.3 JobName

A descriptive name of the job. It has to be stressed that user defined job names are often difficult to retrieve from an accounting perspective and are not suitable for reliable accounting purposes.

- *JobName* MUST be a string.

### Example

```
<ur:JobName>"HiggsGammaGamma42"</ur:JobName>
```

## 6.4 MachineName

A descriptive name of the machine on which the job ran. This may be a system hostname, the LRMS server hostname or a sites name for a cluster of machines. The identification of the machine by name may assume the context of the site or Grid in which the machine participates, i.e. machine names may be unique within a specific site or Grid, but do not need to be unique globally.

- *MachineName* SHOULD be present in the *JobUsageBlock*.
- *MachineName* MUST be a string.

### Example

```
<ur:MachineName>"ce.example.org"</ur:MachineName>
```

## 6.5 SubmitHost

The host from which the jobs was submitted.

- In a Grid environment *SubmitHost* MUST report the Computing Element Unique ID.
- The *SubmitHost* field type MUST be a string.

### Example

```
<ur:SubmitHost>  
  "nordugrid-cluster-name=ce.example.org,Mds-Vo-name=local,o=grid"  
</ur:SubmitHost>
```

## 6.6 Middleware

The purpose of this element is to mark whether the job was submitted locally or through a (Grid) middleware. At least the values “local” and “grid” MUST be supported. The attribute *description* SHOULD be used to give additional information on the used middleware.

- *Middleware* MUST be a string.
- The values “local” and “grid” MUST be supported.
- The attribute *description* SHOULD be used.
- The attribute *description* MUST be a string.

### Example

```
<ur:Middleware ur:description="ARC CE">"grid"</ur:Middleware>
```

## 6.7 Queue

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

- *Queue* MUST be a string.
- The attribute *description* MAY be specified.
- The attribute *description* MUST be a string.

### Example

```
<ur:Queue ur:description="execution">"Bigmem"</ur:Queue>
```

## 6.8 TimeInstant

Time instant related to the user payload. Three optional values for the attribute *type* are defined as they are of common usage by batch systems. The semantic is derived from Torque.

- *TimeInstant* SHOULD be present in the *JobUsageBlock*.
- *TimeInstant* MAY be present multiple times.
- *TimeInstant* MUST be an ISO 8601:2004 timestamp.
- The attribute *type* MUST be a string.
- The following three values of the attribute *type* are defined and SHOULD be reported:
  - *Ctime* - Time job was created
  - *Qtime* - Time job was queued
  - *Etime* - Time job became eligible to run

### Example

```
<ur:TimeInstant ur:type="Etime">2013-05-31T10:59:42</ur:TimeInstant>
```

## 6.9 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.

- *ServiceLevel* MUST be a String.

### Example

```
<ur:ServiceLevel>BigMem</ur:ServiceLevel>
```

## 6.10 Status

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 systems accounting record. The semantic meaning of status is site dependent.

- *Status* MUST be of type String.
- *Status* MUST exist in the record.
- *Status* 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.
- The *Status* property MAY support other values, as agreed upon within the implementation context.

### Example

```
<ur:Status>"aborted"</ur:Status>
```

## 7 MemoryUsageBlock

This block contains the properties related to memory usage. The block may be present several times to account for different types of memory (e.g., RAM and swap).

### Example

```
<ur:MemoryUsageBlock>  
<!! Memory Usage properties go in here -->  
</ur:MemoryUsageBlock>
```

### 7.1 MemoryClass

The class of memory used. RAM and swap must be supported but others might be specified. This is a descriptive value, which allows the memory system to provide details about the memory used.

- *MemoryClass* MUST be present in the *MemoryUsageBlock*.
- The *MemoryClass* field type MUST be a string.
- The values “RAM” and “swap” MUST be supported.

### Example

```
<ur:MemoryClass>"RAM"</ur:MemoryClass>
```

### 7.2 MemoryResourceCapacityUsed

The number of physical bytes used on the memory system (e.g., the amount of memory resources used for this process). This is the main metric for measuring memory consumption.

- *MemoryResourceCapacityUsed* MUST be present in the *MemoryUsageBlock*.
- The *MemoryResourceCapacityUsed* attribute field type MUST be a positive Integer.

### Example

```
<ur:MemoryResourceCapacityUsed>14728</ur:MemoryResourceCapacityUsed>
```

### 7.3 MemoryLogicalCapacityUsed

The number of bytes allocated for this process on the memory system (e.g., the amount of memory resources made available for this process). This value may be higher than the

*MemoryResourceCapacityUsed* because it may also include bytes that are not really used by the process.

- The *MemoryLogicalCapacityUsed* attribute field type MUST be a positive Integer.

#### Example

```
<ur:MemoryLogicalCapacityUsed>56437</ur:MemoryLogicalCapacityUsed>
```

### 7.4 MemoryResourceCapacityAllocated

The number of bytes required by the process (e.g., the memory requested in a job description).

- The *MemoryResourceCapacityAllocated* attribute field type MUST be a positive Integer.

#### Example

```
<ur:MemoryResourceCapacityAllocated>42000</ur:MemoryResourceCapacityUsed>
```

### 7.5 StartTime

Same as *StartTime* in *ComputeUsageBlock* (see chapter 5.3).

### 7.6 EndTime

Same as *EndTime* in *ComputeUsageBlock* (see chapter 5.4).

### 7.7 Host

The system on which the resources have been consumed. This value should be chosen in such a way that it globally identifies the system, on which resources are being consumed (e.g. the Fully Qualified Domain Name of the system could be used).

- *Host* MUST be a string.

#### Example

```
<ur:Host>host.example.org</ur:Host>
```

### 7.8 HostType

Same as *HostType* in *ComputeUsageBlock* (see chapter 5.6).

## 7.9 Charge

Same as *Charge* in *ComputeUsageBlock* (see chapter 5.10).



## 8 StorageUsageBlock

This block contains the properties related to storage usage.

### Example

```
<ur:StorageUsageBlock>  
<!--Storage Record properties go in here -->  
</ur:StorageUsageBlock>
```

### 8.1 StorageShare

The part of the storage system which is accounted for in the record. For a storage system, which is split into several logical parts, this can be used to account for consumption on each of these parts. The value should be able to identify the share of the storage system, given the storage system property.

- *StorageShare* MUST be a string.

### Example

```
<ur:StorageShare>pool-003</ur:StorageShare>
```

### 8.2 StorageMedia

The media type of storage that is accounted for in the record (e.g. “disk” or “tape”.) This allows for accounting of different backend storage types.

- *StorageMedia* MUST be a string.

### Example

```
<ur:StorageMedia>disk</ur:StorageMedia>
```

### 8.3 StorageClass

The class of the stored data, e.g. “pinned”, “replicated” or “precious”. It is a descriptive value, which allows to provide details about the stored data.

- *StorageClass* MUST be a string.
- The values “pinned”, “replicated” and “precious” MUST be supported.

### Example

```
<ur:StorageClass>replicated</ur:StorageClass>
```

## 8.4 DirectoryPath

The directory path being accounted for. If the property is included in the record, the record should account for all usage in the directory and only that directory.

- *DirectoryPath* MUST be a string.

### Example

```
<ur:DirectoryPath>/projectA</ur:DirectoryPath>
```

## 8.5 FileCount

The number of files which are accounted for in the record.

- *FileCount* MUST be a positive non-zero integer.

### Example

```
<ur:FileCount>42</ur:FileCount>
```

## 8.6 StorageResourceCapacityUsed

The number of bytes used on the storage system or storage share where appropriate. This is the main metric for measuring storage resource consumption. It should include all resources for which the identity of the record is accountable for.

*StorageResourceCapacityUsed* can include reserved space, file metadata, space used for redundancy in RAID setups, tape holes, or similar. The decision about including such “additional” space is left to the resource owner but should be made known to the user e.g. via the usage policy or a service level agreement. In contrary the *StorageLogicalCapacityUsed* denotes the pure file size (see chapter 8.7). If available, reserved space can be recorded explicitly with *StorageResourceCapacityAllocated* (see chapter 8.8).

- *StorageResourceCapacityUsed* MUST be present in the *StorageUsageBlock*.
- *StorageResourceCapacityUsed* MUST be a non-negative integer.
- *StorageResourceCapacityUsed* SHOULD include all resources that are used to store the files.
- *StorageResourceCapacityUsed* MAY also include resources that are no longer in use but are unavailable for reuse (e.g., if a file is removed from tape, the tape may not be immediately available for reuse), as documented in the appropriate service level agreement or usage policy documents.

### Example

```
<ur:StorageResourceCapacityUsed>14728</ur:StorageResourceCapacityUsed>
```

#### Implementation Note:

Using bytes saves us from the argument of discussing if 1000 or 1024 should be used as a base. However, this also means that the number reported can be very large. Therefore any implementation should use at least a 128-bit integer to hold this variable (a signed 64-bit integer will overflow at 8 Exabytes).

## 8.7 StorageLogicalCapacityUsed

The number of “logical” bytes used on the storage system . By ”logical” is meant the sum of bytes of the files stored, i.e. excluding reservation, any underlying replicas of files, RAID overhead etc.

- *StorageLogicalCapacityUsed* MUST be a non-negative integer.

### Example

```
<ur:StorageLogicalCapacityUsed>13617</ur:StorageLogicalCapacityUsed>
```

#### Implementation Note:

Same as for *StorageResourceCapacityUsed* property (see chapter 8.6).

## 8.8 StorageResourceCapacityAllocated

The number of bytes allocated on the storage system or storage share. Depending on the implementation this property may be equal to *StorageResourceCapacityUsed*. However, it should only take into account space allocated to the entity described in the record, not resources used for redundancy in RAID setups, tape holes, or similar.

- *StorageLogicalCapacityUsed* MUST be a non-negative integer.

### Example

```
<ur:StorageResourceCapacityAllocated>14624</sr:StorageResourceCapacityAllocated>
```

#### Implementation Note:

Same as for *StorageResourceCapacityUsed* property (see chapter 8.6).

## 8.9 StartTime

Same as *StartTime* in *ComputeUsageBlock* (see chapter 5.3).

## 8.10 EndTime

Same as *EndTime* in *ComputeUsageBlock* (see chapter 5.4).

## 8.11 Host

Same as *Host* in *MemoryUsageBlock* (see chapter 7.7).

## 8.12 HostType

Same as *HostType* in *ComputeUsageBlock* (see chapter 5.6).

## 8.13 Charge

Same as *Charge* in *ComputeUsageBlock* (see chapter 5.10).

## 9 CloudUsageBlock

This block contains the properties related to cloud usage.

Example

**Example**

```
<ur:CloudUsageBlock>
<!--Cloud Record properties go in here -->
</ur:CloudUsageBlock>
```

### 9.1 LocalVirtualMachineId

The local identity of the Virtual Machine. For example, this may be the ID assigned to by the Cloud management system.

- *LocalVirtualMachineId* MUST be a string.

**Example**

```
<ur:LocalVirtualMachineId>"ab1234"</ur:LocalVirtualMachineId>
```

### 9.2 GlobalVirtualMachineId

The global identity of the Virtual Machine. The property should identify the Virtual Machine globally, such that clashes do not happen accidentally. This could for example be a combination of time stamp, local Virtual Machine ID and host name.

- *GlobalVirtualMachineId* MUST be a string.

**Example**

```
<ur:GlobalVirtualMachineId>
  host.example.org/ab1234/2013-05-09T09:06:52Z
</ur:GlobalVirtualMachineId>
```

### 9.3 Status

The status of the Virtual Machine.

- *Status* MUST be present in the *CloudUsageBlock*.
- *Status* MUST be string.

- *Status* MUST support the following values:
  - completed – The execution is completed.
  - started – The execution started at the time this usage record was generated.
  - suspended – The execution was suspended at the time this usage record was generated.
- *Status* MAY support other values, as agreed upon within the implementation context.

### Example

```
<ur:Status>"started"</ur:Status>
```

## 9.4 SuspendDuration

The amount of time in which the Virtual Machine status was “suspended”.

- *SuspendDuration* MUST be present if the property *Status* (see chapter 9.3) of the Virtual Machine is “suspended”.
- *SuspendDuration* MUST be a time duration as defined in ISO 8601:2004.

## 9.5 ImageId

The ID of the image used to instantiate the Virtual Machine.

- *ImageId* MUST be a string.

### Example

```
<ur:ImageId>"UbuntuImage2013"</ur:ImageId>
```

## 9.6 MachineName

Same as *MachineName* in *JobUsageBlock* (see chapter 6.4).

## 9.7 SubmitHost

Same as *SubmitHost* in *JobUsageBlock* (see chapter 6.5).

## 9.8 TimeInstant

Same as *TimeInstant* in *JobUsageBlock* (see chapter 6.8).

## 9.9 ServiceLevel

Same as *ServiceLevel* in *JobUsageBlock* (see chapter 6.9).

## 10 Attributes Matrix

UR 2.0	UR 1.0	StAR 1.0	SAI 1.0	CAR 1.0	FedCloud 1.0	GLUE 2.0
<b>Record-Identity-Block</b>		<b>Record-Identity</b>		<b>Record-Identity</b>		
recordId	Record-Identity	recordId	Record-Identity	recordId	Record-Id/Storage-RecordId	
createTime		createTime	TimeInstant			
Site		Site		Site	Site	Admin-Domain.-Name
					ZoneName	
					TimeZone	
Infrastructure				Infrastructure		
<b>Subject-Identity-Block</b>		<b>Subject-Identity</b>		<b>User-Identity</b>		
LocalUserId	LocalUserId	LocalUser	LocalUserId	LocalUserId	LocalUserId	
LocalGroupId		LocalGroup		LocalGroup	LocalGroup-Id	
GlobalUserId		UserIdentity	GlobalUser-Name	GlobalUser-Name	GlobalUser-Name	
Global-GroupId	Global-Username	Group	Project-Name	Group	FQAN	
Global-Group-Attribute	Project-Name	Group-Attribute	Global-Group	Group-Attribute		
<b>Compute-Usage-Block</b>				<b>JobIdentity</b>		
CpuDuration	CpuDuration			CpuDuration	CpuDuration	
WallDuration	WallDuration			WallDuration	WallDuration	
StartTime	StartTime			StartTime	StartTime	
EndTime	EndTime			EndTime	EndTime	
HostName	Host			Host		
ProcessId	ProcessId			ProcessId		
Benchmark				ServiceLevel		
HostType					CloudType	ServiceType.t
Processors	Processors			Processors	CpuCount	
ExitStatus				ExitStatus		
NodeCount	NodeCount			NodeCount		
Charge	Charge			Charge		



<b>JobUsage-Block</b>				<b>Job-Identity</b>		
GlobalJobId	GlobalJobId			GlobalJobId		
LocalJobId	LocalJobId			LocalJobId		
JobName	JobName			JobName		
Machine-Name	Machine-Name			Machine-Name		
SubmitHost	SubmitHost			SubmitHost		
Middleware						
Queue	Queue			Queue		
	Time-Duration			Time-Duration		
TimeInstant	TimeInstant			TimeInstant		
ServiceLevel	ServiceLevel					
	Extensions			Extensions		
Status	Status			Status		
<b>Memory-Usage-Block</b>						
Memory-Class	Memory/Swap			Memory/Swap	Memory	
Memory-Resource-Capacity-Used						
Memory-Logical-Capacity-Used	Memory/Swap			Memory/Swap	Memory	
Memory-Resource-Capacity-Allocated						
StartTime						
EndTime						
Host						
HostType					CloudType	ServiceType.t
Charge						
<b>Storage-Usage-Block</b>						
StorageShare		StorageShare				
StorageMedia	(Disk)	StorageMedia				
StorageClass		StorageClass	ServiceLevel			

Directory-Path		Directory-Path	Project-Partition			
FileCount		FileCount				
Storage-Resource-Capacity-Used	Disk	Resource-Capacity-Used	Disk		Disk	
Storage-Logical-Capacity-Used	Disk	Logical-Capacity-Used				
Storage-Resource-Capacity-Allocated		Resource-Capacity-Allocated				
StartTime		StartTime	Time-Instant/ Time-Duration			
EndTime		EndTime	Time-Instant/ Time-Duration			
Host		Storage-System	Host		CloudType	
HostType			StorageType			ServiceType.t
Charge			Charge			
			LocalFileId			
			GlobalFileId			
			Status			
			SubmitHost			
			Operation-Type			
<b>Cloud-Usage-Block</b>						
Local-Virtual-MachineId					Machine-Name	
Global-Virtual-MachineId					VMUUID	
Status					Status	
Suspend-Duration					Suspend-Duration	

ImageId					ImageId	
TimeInstant						
ServiceLevel						
SubmitHost						
Machine- Name						
<b>Network- Usage- Block</b>						
					Network- Type	
			Network	Network	Network- Inbound	
			Network	Network	Network- Outbound	
			Protocol- Type			

## 11 Field Summaries

### 11.1 RecordIdentityBlock

Element	Attribute	Short Description	Field Type	Requirement
<i>RecordId</i>		Identity of the record	String	REQUIRED
<i>CreateTime</i>		Time of creation of the record	ISO8601	REQUIRED
<i>Site</i>		The site where resource resides	String	OPTIONAL
<i>Infrastructure</i>		The infrastructure where the resource was used	String	OPTIONAL
	description	Additional information on the used infrastructure	String	RECOMMENDED if <i>Infrastructure</i> exists

### 11.2 SubjectIdentityBlock

Element	Attribute	Short Description	Field Type	Requirement
<i>LocalUserId</i>		Identity of the local user	String	OPTIONAL
<i>LocalGroupId</i>		Identity of the local group	String	OPTIONAL
<i>GlobalUserId</i>		Global identity of the user	String	OPTIONAL
<i>GlobalGroupId</i>		Global identity of the group	String	REQUIRED if <i>GlobalGroupAttribute</i> exists
<i>GlobalGroupAttribute</i>	type	Global group attribute	String	OPTIONAL
		Type of attribute	String	REQUIRED if <i>GlobalGroupAttribute</i> exists

### 11.3 ComputeUsageBlock

Element	Attribute	Short Description	Field Type	Requirement
<i>CPUDuration</i>		CPU duration	ISO 8601:2004	REQUIRED
<i>WallDuration</i>		Wall duration	ISO 8601:2004	REQUIRED
<i>StartTime</i>		Start of consumption	ISO 8601:2004	REQUIRED
<i>EndTime</i>		End of consumption	ISO 8601:2004	REQUIRED
<i>ExecutionHost</i>		Host where application is executed		OPTIONAL
<i>Hostname</i>		Name of the execution host	String	REQUIRED if <i>Host</i> is present

<i>ProcessId</i>		UNIX pid of the process	Non-zero Integer	OPTIONAL
<i>Benchmark</i>	type	Benchmark associated with execution host Benchmark type	String String	OPTIONAL OPTIONAL
<i>HostType</i>		Type of service	String	OPTIONAL
<i>Processors</i>		Number of processors requested/used	Non-zero Integer	OPTIONAL
<i>NodeCount</i>		Number of nodes requested/used	Non-zero Integer	OPTIONAL
<i>ExitStatus</i>		Exit status of the process	Integer	REQUIRED
<i>Charge</i>	unit	The charge to the user for the resource used Currency used	Float ISO 4217:2008 Currency	OPTIONAL REQUIRED if <i>Charge</i> is present
	formula	Cost consumption formula	String	REQUIRED if <i>Charge</i> is present

## 11.4 JobUsageBlock

Element	Attribute	Short Description	Field Type	Requirement
<i>GlobalJobId</i>		Global identity of the job	String	OPTIONAL
<i>LocalJobId</i>		Local identity of the job	String	OPTIONAL
<i>JobName</i>		Description of the job	String	OPTIONAL
<i>MachineName</i>		Computer facility	String	RECOMMENDED
<i>SubmitHost</i>		Host submitting the job	String	OPTIONAL
<i>Middleware</i>	description	Type of middleware	String	OPTIONAL
		Description of the middleware used	String	REQUIRED
<i>Queue</i>	description	Queue name	String	OPTIONAL
		Description of the queue name	String	OPTIONAL
<i>TimeInstant</i>	type	Time instant related to the user payload	ISO 8601:2004	OPTIONAL
		Type of time instant	String	OPTIONAL
<i>ServiceLevel</i>		Type of service level	String	OPTIONAL
<i>Status</i>		Status of the job	String	REQUIRED

## 11.5 MemoryUsageBlock

Element	Attribute	Short Description	Field Type	Requirement
<i>MemoryClass</i>		Class of memory	String	REQUIRED
<i>MemoryResource-CapacityUsed</i>		Bytes used	Integer	REQUIRED
<i>MemoryLogical-CapacityUsed</i>		Logical bytes used	Integer	OPTIONAL

<i>MemoryResource-CapacityAllocated</i>		Bytes allocated	Integer	OPTIONAL
<i>StartTime</i>		Start of consumption	ISO 8601:2004	REQUIRED
<i>EndTime</i>		End of consumption	ISO 8601:2004	REQUIRED
<i>Host</i>		Host where resource is consumed	String	OPTIONAL
<i>HostType</i>		Type of service	String	OPTIONAL
<i>Charge</i>	unit	The charge to the user for the resource used Currency used	Float ISO 4217:2008 Currency	OPTIONAL REQUIRED if <i>Charge</i> is present
	formula	Cost consumption formula	String	REQUIRED if <i>Charge</i> is present

## 11.6 StorageUsageBlock

Element	Attribute	Short Description	Field Type	Requirement
<i>StorageShare</i>		Part of the storage used	String	OPTIONAL
<i>StorageMedia</i>		Details on the stored data	String	OPTIONAL
<i>StorageClass</i>		Class of the stored data	String	OPTIONAL
<i>DirectoryPath</i>		Directory path of the data accounted	String	OPTIONAL
<i>FileCount</i>		Number of files accounted	Non-zero Integer	OPTIONAL
<i>StorageResource-CapacityUsed</i>		Bytes used	Integer	REQUIRED
<i>StorageLogical-CapacityUsed</i>		Logical bytes used	Integer	OPTIONAL
<i>StorageResource-CapacityAllocated</i>		Bytes allocated	Integer	OPTIONAL
<i>StartTime</i>		Start of consumption	ISO 8601:2004	REQUIRED
<i>EndTime</i>		End of consumption	ISO 8601:2004	REQUIRED
<i>Host</i>		Host in which the storage is consumed	String	OPTIONAL
<i>HostType</i>		Type of service	String	OPTIONAL
<i>Charge</i>	unit	The charge to the user for the resource used Currency used	Float ISO 4217:2008 Currency	OPTIONAL REQUIRED if <i>Charge</i> is present
	formula	Cost consumption formula	String	REQUIRED if <i>Charge</i> is present

## 11.7 CloudUsageBlock

Element	Attribute	Short Description	Field Type	Requirement
<i>LocalVirtualMachine-Id</i>		Local Id of the Virtual Machine	String	OPTIONAL
<i>GlobalVirtual-MachineId</i>		Global Id of the Virtual Machine	String	OPTIONAL
<i>Status</i>		Status of the Virtual Machine	String	REQUIRED
<i>SuspendDuration</i>		Amount of time in suspension	ISO 8601:2004	OPTIONAL
<i>ImageId</i>		Virtual Machine Image Id	String	OPTIONAL
<i>TimeInstant</i>		Time instant related to the user payload	ISO 8601:2004	OPTIONAL
	type	Type of time instant	String	OPTIONAL
<i>ServiceLevel</i>		Type of service level	String	OPTIONAL
<i>SubmitHost</i>		Host submitting the request for the Virtual Machine	String	OPTIONAL
<i>MachineName</i>		Computer facility	String	RECOMMENDED

## 12 Examples

### 12.1 Full example

Full example including all defined elements. Note that this does not necessarily make a useful working example as some resource blocks do not necessarily make sense when used together in a single record.

#### Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ur:UsageRecord xmlns="http://schema.ogf.org/urf/2013/04/urf"
  xmlns:ur="http://schema.ogf.org/urf/2013/04/urf"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schema.ogf.org/urf/2013/04/urf">
  <ur:RecordIdentityBlock>
    <ur:RecordId>"host.example.org/ur/87912469269276"</ur:RecordId>
    <ur:CreateTime>2013-05-09T09:06:52Z</ur:CreateTime>
    <ur:Site>"ACME-University"</ur:Site>
    <ur:Infrastructure>"OSG"</ur:Infrastructure>
  </ur:RecordIdentityBlock>
  <ur:SubjectIdentityBlock>
    <ur:LocalUserId>"johndoe"</ur:LocalUserId>
    <ur:LocalGroupId>"projectA"</ur:LocalGroupId>
    <ur:GlobalUserId>"/O=Grid/OU=example.org/CN=John Doe"</ur:GlobalUserId>
    <ur:GlobalGroupId>"binarydatapoint.example.org"</ur:GlobalGroupId>
    <ur:GlobalGroupAttribute ur:type="subgroup">ukusers</ur:GlobalGroupAttribute>
  </ur:SubjectIdentityBlock>
  <ur:ComputeUsageBlock>
    <ur:CpuDuration>PT3600S</ur:CpuDuration>
    <ur:WallDuration>PT3600S</ur:WallDuration>
    <ur:StartTime>2013-05-31T11:00:00</ur:StartTime>
    <ur:EndTime>2013-05-31T12:00:00</ur:EndTime>
    <ur:ExecutionHost>
      <ur:Hostname>"compute-0-1.abel.uio.no"</ur:Hostname>
      <ur:ProcessId>1042</ur:ProcessId>
      <ur:Benchmark ur:type="si2k">3.14</ur:Benchmark>
      <ur:Benchmark ur:type="hepspec">42</ur:Benchmark>
    </ur:ExecutionHost>
    <ur:HostType>"org.nordugrid.arex"</ur:HostType>
    <ur:Processors>1</ur:Processors>
    <ur:NodeCount>1</ur:NodeCount>
```



```

    <ur:ExitStatus>0</ur:ExitStatus>
    <ur:Charge ur:formula="42 units/CPU/day" ur:unit="USD">1.75</ur:Charge>
</ur:ComputeUsageBlock>
<ur:JobUsageBlock>
  <ur:GlobalJobId>"host.example.org/ab1234/2013-05-09T09:06:52Z"</ur:GlobalJobId>
  <ur:LocalJobId>"ab1234"</ur:LocalJobId>
  <ur:JobName>"HiggsGammaGamma42"</ur:JobName>
  <ur:MachineName>"ce.example.org"</ur:MachineName>
  <ur:SubmitHost>
    "nordugrid-cluster-name=ce.example.org,Mds-Vo-name=local,o=grid"
  </ur:SubmitHost>
  <ur:Middleware ur:description="ARC CE">"grid"</ur:Middleware>
  <ur:Queue ur:description="execution">"Bigmem"</ur:Queue>
  <ur:TimeInstant ur:type="Ctime">2013-05-31T10:30:00</ur:TimeInstant>
  <ur:TimeInstant ur:type="Qtime">2013-05-31T10:31:00</ur:TimeInstant>
  <ur:TimeInstant ur:type="Etime">2013-05-31T10:59:42</ur:TimeInstant>
  <ur:ServiceLevel>"Bigmem"</ur:ServiceLevel>
  <ur>Status>"aborted"</ur>Status>
</ur:JobUsageBlock>
<ur:MemoryUsageBlock>
  <ur:MemoryClass>"RAM"</ur:MemoryClass>
  <ur:MemoryResourceCapacityUsed>14728</ur:MemoryResourceCapacityUsed>
  <ur:MemoryLogicalCapacityUsed>56437</ur:MemoryLogicalCapacityUsed>
  <ur:MemoryResourceCapacityAllocated>42000</ur:MemoryResourceCapacityAllocated>
  <ur:StartTime>2013-05-31T11:00:00</ur:StartTime>
  <ur:EndTime>2013-05-31T12:00:00</ur:EndTime>
  <ur:Host>"compute-0-1.abel.uio.no"</ur:Host>
  <ur:HostType>"org.nordugrid.arex"</ur:HostType>
  <ur:Charge ur:formula="42 units/MB/day" ur:unit="USD">0.0735</ur:Charge>
</ur:MemoryUsageBlock>
<ur:StorageUsageBlock>
  <ur:StorageShare>pool-003</ur:StorageShare>
  <ur:StorageMedia>disk</ur:StorageMedia>
  <ur:StorageClass>replicated</ur:StorageClass>
  <ur:DirectoryPath>/projectA</ur:DirectoryPath>
  <ur:FileCount>42</ur:FileCount>
  <ur:StorageResourceCapacityUsed>14728</ur:StorageResourceCapacityUsed>
  <ur:StorageLogicalCapacityUsed>13617</ur:StorageLogicalCapacityUsed>
  <ur:StorageResourceCapacityAllocated>14624</ur:StorageResourceCapacityAllocated>
  <ur:StartTime>2013-05-07T09:31:40Z</ur:StartTime>
  <ur:EndTime>2013-05-08T09:29:42Z</ur:EndTime>
  <ur:Host>host.example.org</ur:Host>

```

```

        <ur:HostType>"org.dcache.storage"</ur:HostType>
        <ur:Charge ur:formula="42 units/MB" ur:unit="USD">0.01473</ur:Charge>
    </ur:StorageUsageBlock>
    <ur:CloudUsageBlock>
        <!--Cloud Record properties go in here -->
        <ur:Status>"running"</ur:Status>
    </ur:CloudUsageBlock>
</ur:UsageRecord>

```

## 12.2 Grid example

Example of how a grid usage record may look like.

### Example

```

<?xml version="1.0" encoding="UTF-8"?>
<ur:UsageRecord xmlns="http://schema.ogf.org/urf/2013/04/urf"
  xmlns:ur="http://schema.ogf.org/urf/2013/04/urf"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schema.ogf.org/urf/2013/04/urf">
  <ur:RecordIdentityBlock>
    <ur:RecordId>"host.example.org/ur/87912469269276"</ur:RecordId>
    <ur:CreateTime>2013-05-09T09:06:52Z</ur:CreateTime>
    <ur:Infrastructure>"EGI"</ur:Infrastructure>
  </ur:RecordIdentityBlock>
  <ur:SubjectIdentityBlock>
    <ur:GlobalGroupId>"binarydataproject.example.org"</ur:GlobalGroupId>
    <ur:GlobalGroupAttribute ur:type="subgroup">ukusers</ur:GlobalGroupAttribute>
  </ur:SubjectIdentityBlock>
  <ur:ComputeUsageBlock>
    <ur:CpuDuration>PT3600S</ur:CpuDuration>
    <ur:WallDuration>PT3600S</ur:WallDuration>
    <ur:StartTime>2013-05-31T11:00:00</ur:StartTime>
    <ur:EndTime>2013-05-31T12:00:00</ur:EndTime>
    <ur:ExecutionHost>
      <ur:Hostname>"compute-0-1.abel.uio.no"</ur:Hostname>
      <ur:ProcessId>1042</ur:ProcessId>
      <ur:Benchmark ur:type="si2k">3.14</ur:Benchmark>
      <ur:Benchmark ur:type="hepspec">42</ur:Benchmark>
    </ur:ExecutionHost>
    <ur:HostType>"org.nordugrid.arex"</ur:HostType>
    <ur:ExitStatus>0</ur:ExitStatus>
  </ur:ComputeUsageBlock>

```

```

<ur:JobUsageBlock>
  <ur:GlobalJobId>"host.example.org/ab1234/2013-05-09T09:06:52Z"</ur:GlobalJobId>
  <ur:MachineName>"ce.example.org"</ur:MachineName>
  <ur:SubmitHost>
    "nordugrid-cluster-name=ce.example.org,Mds-Vo-name=local,o=grid"
  </ur:SubmitHost>
  <ur:Middleware ur:description="ARC CE">"grid"</ur:Middleware>
  <ur:Queue ur:description="execution">"Bigmem"</ur:Queue>
  <ur:TimeInstant ur:type="Ctime">2013-05-31T10:30:00</ur:TimeInstant>
  <ur:TimeInstant ur:type="Qtime">2013-05-31T10:31:00</ur:TimeInstant>
  <ur:TimeInstant ur:type="Etime">2013-05-31T10:59:42</ur:TimeInstant>
  <ur:ServiceLevel>"Bigmem"</ur:ServiceLevel>
  <ur>Status>"aborted"</ur>Status>
</ur:JobUsageBlock>
</ur:UsageRecord>

```

## 12.3 Cloud example

## 12.4 Local example

Example of how a storage record accounting for a local user could look like.

### Example

```

<?xml version="1.0" encoding="UTF-8"?>
<ur:UsageRecord xmlns="http://schema.ogf.org/urf/2013/04/urf"
  xmlns:ur="http://schema.ogf.org/urf/2013/04/urf"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schema.ogf.org/urf/2013/04/urf">
  <ur:RecordIdentityBlock>
    <ur:RecordId>"host.example.org/ur/87912469269276"</ur:RecordId>
    <ur:CreateTime>2013-05-09T09:06:52Z</ur:CreateTime>
    <ur:Infrastructure>"OSG"</ur:Infrastructure>
  </ur:RecordIdentityBlock>
  <ur:SubjectIdentityBlock>
    <ur:LocalUserId>"johndoe"</ur:LocalUserId>
  </ur:SubjectIdentityBlock>
  <ur:StorageUsageBlock>
    <ur:StorageResourceCapacityUsed>13617</ur:StorageResourceCapacityUsed>
    <ur:StartTime>2013-05-07T09:31:40Z</ur:StartTime>
    <ur:EndTime>2013-05-08T09:29:42Z</ur:EndTime>
    <ur:Host>"compute-0-1.abel.uio.no"</ur:Host>
  </ur:StorageUsageBlock>
</ur:UsageRecord>

```

```

    </ur:StorageUsageBlock>
</ur:UsageRecord>

```

## 12.5 Minimal examples

### 12.5.1 Job record

Minimal useful job record for a job run by the ukusers team of the binarydatapoint.example.org group.

#### Example

```

<?xml version="1.0" encoding="UTF-8"?>
<ur:UsageRecord xmlns="http://schema.ogf.org/urf/2013/04/urf"
  xmlns:ur="http://schema.ogf.org/urf/2013/04/urf"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schema.ogf.org/urf/2013/04/urf">
  <ur:RecordIdentityBlock>
    <ur:RecordId>"host.example.org/ur/87912469269276"</ur:RecordId>
    <ur:CreateTime>2013-05-09T09:06:52Z</ur:CreateTime>
  </ur:RecordIdentityBlock>
  <ur:SubjectIdentityBlock>
    <ur:GlobalGroupId>"binarydatapoint.example.org"</ur:GlobalGroupId>
    <ur:GlobalGroupAttribute ur:type="subgroup">ukusers</ur:GlobalGroupAttribute>
  </ur:SubjectIdentityBlock>
  <ur:ComputeUsageBlock>
    <ur:CpuDuration>PT3600S</ur:CpuDuration>
    <ur:WallDuration>PT3600S</ur:WallDuration>
    <ur:StartTime>2013-05-31T11:00:00</ur:StartTime>
    <ur:EndTime>2013-05-31T12:00:00</ur:EndTime>
    <ur:ExecutionHost>
      <ur:Hostname>"compute-0-1.abel.uio.no"</ur:Hostname>
    </ur:ExecutionHost>
    <ur:ExitStatus>0</ur:ExitStatus>
  </ur:ComputeUsageBlock>
  <ur:JobUsageBlock>
    <ur:GlobalJobId>"host.example.org/ab1234/2013-05-09T09:06:52Z"</ur:GlobalJobId>
    <ur:MachineName>"ce.example.org"</ur:MachineName>
    <ur:ServiceLevel>"Bigmem"</ur:ServiceLevel>
    <ur>Status>"aborted"</ur>Status>
  </ur:JobUsageBlock>
</ur:UsageRecord>

```

### 12.5.2 Storage record

Minimal storage record that is actually useful. There is no SubjectIdentity block, which should be interpreted as the record accounts for all usage on the storage system.

#### Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ur:UsageRecord xmlns="http://schema.ogf.org/urf/2013/04/urf"
  xmlns:ur="http://schema.ogf.org/urf/2013/04/urf"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schema.ogf.org/urf/2013/04/urf">
  <ur:RecordIdentityBlock>
    <ur:RecordId>"host.example.org/ur/87912469269276"</ur:RecordId>
    <ur:CreateTime>2013-05-09T09:06:52Z</ur:CreateTime>
  </ur:RecordIdentityBlock>
  <ur:StorageUsageBlock>
    <ur:StorageResourceCapacityUsed>13617</ur:StorageResourceCapacityUsed>
    <ur:StartTime>2013-05-07T09:31:40Z</ur:StartTime>
    <ur:EndTime>2013-05-08T09:29:42Z</ur:EndTime>
    <ur:Host>host.example.org</ur:Host>
  </ur:StorageUsageBlock>
</ur:UsageRecord>
```

## 13 XSD Schema

```

<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema attributeFormDefault="qualified"
  elementFormDefault="qualified" targetNamespace="http://schema.ogf.org/urf/2013/04/urf"
  xmlns:ur="http://schema.ogf.org/urf/2013/04/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 2.0 XML Schema definition
    </xsd:documentation>
  </xsd:annotation>

  <xsd:complexType name="UsageRecordType">
    <xsd:sequence>
      <xsd:element maxOccurs="1" minOccurs="1" ref="ur:RecordIdentityBlock" />
      <xsd:element maxOccurs="1" minOccurs="0" ref="ur:SubjectIdentityBlock" />
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="ur:ComputeUsageBlock" />
      <xsd:element maxOccurs="1" minOccurs="0" ref="ur:JobUsageBlock" />
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="ur:MemoryUsageBlock" />
      <xsd:element maxOccurs="unbounded" minOccurs="0" ref="ur:StorageUsageBlock" />
      <xsd:element maxOccurs="1" minOccurs="0" ref="ur:CloudUsageBlock" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element abstract="true" name="Usage" type="ur:UsageRecordType" />
  <xsd:element name="UsageRecord" substitutionGroup="ur:Usage"
    type="ur:UsageRecordType" />
  <xsd:element name="UsageRecords">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element maxOccurs="unbounded" minOccurs="0"
          ref="ur:Usage" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="RecordIdentityBlock" type="ur:RecordIdentityBlockType" />
  <xsd:element name="SubjectIdentityBlock" type="ur:SubjectIdentityBlockType" />
  <xsd:element name="ComputeUsageBlock" type="ur:ComputeUsageBlockType" />
  <xsd:element name="JobUsageBlock" type="ur:JobUsageBlockType" />
  <xsd:element name="MemoryUsageBlock" type="ur:MemoryUsageBlockType" />

```

```
<xsd:element name="StorageUsageBlock" type="ur:StorageUsageBlockType" />
<xsd:element name="CloudUsageBlock" type="ur:CloudUsageBlockType" />

<!-- RecordIdentityBlock definition -->
<xsd:complexType name="RecordIdentityBlockType">
  <xsd:sequence>
    <xsd:element maxOccurs="1" minOccurs="1" ref="ur:RecordId" />
    <xsd:element maxOccurs="1" minOccurs="1" ref="ur:CreateTime" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:Site" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:Infrastructure" />
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="RecordId" type="xsd:string" />
<xsd:element name="CreateTime" type="xsd:dateTime" />
<xsd:element name="Site" type="xsd:string" />
<xsd:element name="Infrastructure" type="ur:InfrastructureType" />

<xsd:complexType name="InfrastructureType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="description" type="xsd:string" use="optional" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<!-- SubjectIdentityBlock definition -->
<xsd:complexType name="SubjectIdentityBlockType">
  <xsd:sequence>
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:LocalUserId" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:LocalGroupId" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:GlobalUserId" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:GlobalGroupId" />
    <xsd:element maxOccurs="unbounded" minOccurs="0" ref="ur:GlobalGroupAttribute" />
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="LocalUserId" type="xsd:string" />
<xsd:element name="LocalGroupId" type="xsd:string" />
<xsd:element name="GlobalUserId" type="xsd:string" />
<xsd:element name="GlobalGroupId" type="xsd:string" />
<xsd:element name="GlobalGroupAttribute" type="ur:GlobalGroupAttributeType" />
```

```

<xsd:complexType name="GlobalGroupAttributeType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="type" type="xsd:string" use="required" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<!-- ComputeUsageBlock definition -->
<xsd:complexType name="ComputeUsageBlockType">
  <xsd:sequence>
    <xsd:element maxOccurs="1" minOccurs="1" ref="ur:CpuDuration" />
    <xsd:element maxOccurs="1" minOccurs="1" ref="ur:WallDuration" />
    <xsd:element maxOccurs="1" minOccurs="1" ref="ur:StartTime" />
    <xsd:element maxOccurs="1" minOccurs="1" ref="ur:EndTime" />
    <xsd:element maxOccurs="unbounded" minOccurs="0" ref="ur:ExecutionHost" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:HostType" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:Processors" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:NodeCount" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:ExitStatus" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:Charge" />
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="CpuDuration" type="xsd:duration" />
<xsd:element name="WallDuration" type="xsd:duration" />
<xsd:element name="StartTime" type="xsd:dateTime" />
<xsd:element name="EndTime" type="xsd:dateTime" />
<xsd:element name="ExecutionHost" type="ur:ExecutionHostType" />
<xsd:element name="HostType" type="xsd:string" />
<xsd:element name="Processors" type="xsd:positiveInteger" />
<xsd:element name="NodeCount" type="xsd:positiveInteger" />
<xsd:element name="ExitStatus" type="xsd:integer" />
<xsd:element name="Charge" type="ur:ChargeType" />

<xsd:complexType name="ExecutionHostType">
  <xsd:sequence>
    <xsd:element maxOccurs="1" minOccurs="1" ref="ur:Hostname" />
    <xsd:element maxOccurs="unbounded" minOccurs="0" ref="ur:ProcessId" />
    <xsd:element maxOccurs="unbounded" minOccurs="0" ref="ur:Benchmark" />
  </xsd:sequence>

```



```

</xsd:complexType>

<xsd:element name="Hostname" type="ur:HostnameType" />
<xsd:element name="ProcessId" type="xsd:positiveInteger" />
<xsd:element name="Benchmark" type="ur:BenchmarkType" />

<xsd:complexType name="HostnameType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
<xsd:attribute name="primary" type="xsd:boolean" use="optional" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="BenchmarkType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:float">
<xsd:attribute name="type" type="xsd:string" use="required" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="ChargeType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:float">
<xsd:attribute name="unit" type="xsd:string" use="required" />
<xsd:attribute name="formula" type="xsd:string" use="required" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<!-- JobUsageBlock definition -->
<xsd:complexType name="JobUsageBlockType">
  <xsd:sequence>
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:GlobalJobId" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:LocalJobId" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:JobName" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:MachineName" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:SubmitHost" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:Middleware" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:Queue" />
    <xsd:element maxOccurs="unbounded" minOccurs="0" ref="ur:TimeInstant" />
  </xsd:sequence>
</xsd:complexType>

```

```
<xsd:element maxOccurs="1" minOccurs="0" ref="ur:ServiceLevel" />
<xsd:element maxOccurs="1" minOccurs="1" ref="ur:Status" />
</xsd:sequence>
</xsd:complexType>

<xsd:element name="GlobalJobId" type="xsd:string" />
<xsd:element name="LocalJobId" type="xsd:string" />
<xsd:element name="JobName" type="xsd:string" />
<xsd:element name="MachineName" type="xsd:string" />
<xsd:element name="SubmitHost" type="xsd:string" />
<xsd:element name="Middleware" type="ur:MiddlewareType" />
<xsd:element name="Queue" type="ur:QueueType" />
<xsd:element name="TimeInstant" type="ur:TimeInstantType" />
<xsd:element name="ServiceLevel" type="xsd:string" />
<xsd:element name="Status" type="xsd:string" />

<xsd:complexType name="MiddlewareType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
<xsd:attribute name="description" type="xsd:string" use="optional" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="QueueType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
<xsd:attribute name="description" type="xsd:string" use="optional" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="TimeInstantType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:dateTime">
<xsd:attribute name="type" type="xsd:string" use="optional" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<!-- MemoryUsageBlock definition -->
<xsd:complexType name="MemoryUsageBlockType">
```

```

<xsd:sequence>
  <xsd:element maxOccurs="1" minOccurs="1" ref="ur:MemoryClass" />
  <xsd:element maxOccurs="1" minOccurs="1" ref="ur:MemoryResourceCapacityUsed" />
  <xsd:element maxOccurs="1" minOccurs="0" ref="ur:MemoryLogicalCapacityUsed" />
  <xsd:element maxOccurs="1" minOccurs="0" ref="ur:MemoryResourceCapacityAllocated" />
  <xsd:element maxOccurs="1" minOccurs="1" ref="ur:StartTime" />
  <xsd:element maxOccurs="1" minOccurs="1" ref="ur:EndTime" />
  <xsd:element maxOccurs="1" minOccurs="0" ref="ur:Host" />
  <xsd:element maxOccurs="1" minOccurs="0" ref="ur:HostType" />
  <xsd:element maxOccurs="1" minOccurs="0" ref="ur:Charge" />
</xsd:sequence>
</xsd:complexType>

<xsd:element name="MemoryClass" type="xsd:string" />
<xsd:element name="MemoryResourceCapacityUsed" type="xsd:positiveInteger" />
<xsd:element name="MemoryLogicalCapacityUsed" type="xsd:positiveInteger" />
<xsd:element name="MemoryResourceCapacityAllocated" type="xsd:positiveInteger" />
<xsd:element name="Host" type="xsd:string" />

<!-- StorageUsageBlock definition -->
<xsd:complexType name="StorageUsageBlockType">
  <xsd:sequence>
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:StorageShare" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:StorageMedia" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:StorageClass" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:DirectoryPath" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:FileCount" />
    <xsd:element maxOccurs="1" minOccurs="1" ref="ur:StorageResourceCapacityUsed" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:StorageLogicalCapacityUsed" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:StorageResourceCapacityAllocated" />
    <xsd:element maxOccurs="1" minOccurs="1" ref="ur:StartTime" />
    <xsd:element maxOccurs="1" minOccurs="1" ref="ur:EndTime" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:Host" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:HostType" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:Charge" />
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="StorageShare" type="xsd:string" />
<xsd:element name="StorageMedia" type="xsd:string" />
<xsd:element name="StorageClass" type="xsd:string" />
<xsd:element name="DirectoryPath" type="xsd:string" />

```

```
<xsd:element name="FileCount" type="xsd:positiveInteger" />
<xsd:element name="StorageResourceCapacityUsed" type="xsd:positiveInteger" />
<xsd:element name="StorageLogicalCapacityUsed" type="xsd:positiveInteger" />
<xsd:element name="StorageResourceCapacityAllocated" type="xsd:positiveInteger" />

<!-- CloudUsageBlock definition -->
<xsd:complexType name="CloudUsageBlockType">
  <xsd:sequence>
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:LocalVirtualMachineId" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:GlobalVirtualMachineId" />
    <xsd:element maxOccurs="1" minOccurs="1" ref="ur:Status" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:SuspendDuration" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:ImageId" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:MachineName" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:SubmitHost" />
    <xsd:element maxOccurs="unbounded" minOccurs="0" ref="ur:TimeInstant" />
    <xsd:element maxOccurs="1" minOccurs="0" ref="ur:ServiceLevel" />
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="LocalVirtualMachineId" type="xsd:string" />
<xsd:element name="GlobalVirtualMachineId" type="xsd:string" />
<xsd:element name="SuspendDuration" type="xsd:duration" />
<xsd:element name="ImageId" type="xsd:string" />

</xsd:schema>
```

## 14 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 is out of scope for this layer. Another layer should address the necessary security requirements.

## 15 Contributors

**Andrea Cristofori**

(Corresponding author)

IGI-BOLOGNA

Viale Berti Pichat 6/2, 40127 Bologna

Italy

Email: andrea.cristofori@cnaa.infn.it

**Jon Kerr Nilsen**

(Corresponding author)

University of Oslo

P.O box 1048, Blindern, 0316 Oslo

Norway

Email: j.k.nilsen@fys.uio.no

**John Gordon**

Institution1

Address

Country

Email: jdoe@example.com

**John Alan Kennedy**

Institution1

Address

Country

Email: jdoe@example.com

**Ralph Müller-Pfefferkorn**

Technische Universität Dresden

01062 Dresden

Germany

Email: ralph.mueller-pfefferkorn@tu-dresden.de

**Alison Paker**

Institution1

Address

Country

Email: jdoe@example.com

## 16 Acknowledgments

This document is the work of the Usage Record Working Group of the OGF.

The group would like to thank the authors of other accounting and usage records (namely CAR, StAR, and CUR) who contributed in the discussions as well as the people who commented in the StAR public hearing.

Work on this specification was supported by the European project EMI.

## 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 Disclaimer

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 Full Copyright Notice

Copyright © Open Grid Forum (2006-2013). Some 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 as references to the derived portions on all such copies and derivative works. The published OGF document from which such works are derived, however, 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 new or updated OGF documents in conformance with the procedures defined in the OGF Document Process, or as required to translate it into languages other than English. OGF, with the approval of its board, may remove this restriction for inclusion of OGF document content for the purpose of producing standards in cooperation with other international standards bodies.

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



## 20 References

- [1] RR Mach, R. Lepro-Metz, S. Jackson, and L. McGinnis. Usage record format recommendation gfd-rp. 098. In *Open Grid Forum Recommendation*, 2007.
- [2] A. Guarise. Definition of the compute accounting record (car), 2011. URL <http://cdsweb.cern.ch/record/1449764>.
- [3] JK Nilsen, P. Millar, R. Müller-Pfefferkorn, Z. Molnar, and R. Zappi. Emi star—definition of a storage accounting record. 2011.
- [4] A. Cristofori. *Grid accounting for computing and storage resources towards standardization*. PhD thesis, Università degli Studi di Ferrara, 2011.
- [5] Scott Bradner. Key words for use in RFCs to Indicate Requirement Levels. RFC 2119 (Best Current Practice), March 1997. URL <http://tools.ietf.org/html/rfc2119>.
- [6] M. Wolf and C. Wicksteed. Date and time formats. *W3C NOTE NOTE-datetime-19980827*, August, 1998.
- [7] P. Mockapetris. Rfc 1034: Domain names-concepts and facilities, 1987. URL: <ftp://ftp.isi.edu/in-notes/rfc1034.txt>.