GWD-R-P
DRMAA-WG / OCCI-WG
drmaa-wg@ogf.org / occi-wg@ogf.org

Peter Tröger, Hasso-Plattner-Institute
(Corresponding Author)

March 2012

OCCI-DRMAA

Status of This Document

Group Working Draft - Proposed Recommendation (GWD-R-P)

(See footnote)[1]

Document Change History

Copyright Notice

Trademark

All company, product or service names referenced in this document are used for identification purposes only and may be trademarks of their respective owners.

Abstract

This document is an extension specification in the Open Cloud Computing Interface (OCCI) document series. It describes an extension of the OCCI Core Model [3] to allow the remote access to a distributed resource management (DRM) system which supports the notion of jobs. The access wire protocol is provided by the OCCI rendering specifications, such as the OCCI HTTP rendering [2]. Since all interface semantics are derived from the Distributed Resource Management Application API Version 2 [4], this document serves also as 'language binding' in the DRMAA document series.

The intended audience for this specification are OCCI and DRMAA implementors. Based on this specification, OCCI implementors can extend their backend support to DRM systems. DRMAA implementors can realize a 'remote' version of their API implementation.

In plain English: This is an API specification for job submission over RESTful protocols such as HTTP.

[1] This is the non-normative annotated version of the specification with line numbers. It includes information concerning the content and why features were included or discarded by the working group. It also emphasizes the consequences of some aspects that may not be immediately apparent. This document in only intended for internal working group discussions.

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 [1].

Contents

1 Introduction

The *Distributed Resource Management Application API Version 2 (DRMAA)* specification defines an interface for tightly coupled, but still portable access to Distributed Resource Management (DRM) systems. The scope is limited to job submission, job control, reservation management, and retrieval of job and machine monitoring information. The DRMAA root specification describes the abstract API concepts and the behavioral rules of a DRMAA-compliant implementation. The programming language representation of the API is defined by a separate *language binding specification.*

The Open Cloud Computing Interface (OCCI) is a RESTful Protocol and API for all kinds of management tasks. The OCCI Core specification consists of a single document defining the OCCI Core Model [3]. The OCCI Core Model can be interacted with *renderings* (including associated behaviors) and expanded through *extensions.* For particular domain, extensions specify additional resource types, their attributes and the actions that can be taken on each resource type. OCCI makes an ideal interoperable boundary interface between the web and the internal resource management system of infrastructure providers.

This document acts as OCCI extension for the domain of DRM systems that are covered by the DRMAA specification. It therefore acts both as OCCI extension and as DRMAA 'language' binding for remote access purposes. The OCCI DRMAA extension details how an OCCI implementation can model and implement the control and monitoring functions of a DRM system as a Service API offering. This API allows for the creation and management of typical resources associated with a DRM system, as defined by the DRMAA specification.

Due to the nature of this specification, no functionality or protocol specifics are provided. All behavioral semantics of an implementation MUST be conformant to the DRMAA specification [4]. All syntactical aspects of the access protocol MUST be conformant to a chosen OCCI binding, such as the HTTP binding [2].

There are other relevant OGF standards for remote APIs in the area of job submission and monitoring. They typically focus on meta-scheduling, while this specification provides a tightly coupled DRM system access with a remote API. OGSA-BES [**?**] defines a service interface for similar job submission purposes, but with strong focus on the abstract notion of resources and WSRF standards [**?**]. OCCI-DRMAA interfaces may serve as backend for a OGSA-BES implementation. JSDL [**?**] is a specification for XML-based job description. OCCI-DRMAA utilizes the more restrictive DRMAA job description scheme in order to implement mandatory job attribute support.

2 Basic concepts

DRMAA interfaces represent activities on instantiatable entities. They are mostly modeled as OCCI resources:

- A `jobsession` resource acts as container for `job` resources and `jobarray` resources.
- A `reservationsession` resource acts as container for `reservation` resources.
- A `monitoringsession` resource acts as representation of server-side information about the DRM system.
- A `job` resource represents one job in the underlying DRM system.

- A `jobarray` resource represents a cluster of jobs that can be controlled and monitored as one in the underlying DRM system.
- A `reservation` resource represents a successfully created advance reservation in the DRM system.

Discuss singleton entry point with query interface (see example). Trick is to return only one location in the drmaa2 collection.

The DRMAA *SessionManager* interface is not explicitely modelled as OCCI resource. Instead, the session management methods are represented with according OCCI resource management actions. All attributes of the DRMAA *SessionManager* interface are mapped to OCCI attributes on the scheme root URI (*http://schemas.ogf.org/occi/drmaa*).

Add DrmaaReflective mapping.

(See footnote)[2]

Are OCCI attributes on the root URI OK ?

DRMAA interface methods that trigger state changes in the DRM system map to OCCI actions on OCCI resources. DRMAA functionalities that lead to the creation of instances represented by OCCI resources are available in two ways, as explicit `jobsession` / `reservationsession` actions and as OCCI resource creation activities (see also [3], Section 4.4.4 and [2], Section 3.4.4). DRMAA interface methods that return named instances (i.e. JobSession::getJobArray) are not translated to OCCI actions, since this kind of retrieval is possible by a direct URI path formulation (i.e. `GET /drmaa/jobarray/[id]`).

DRMAA IDL interface attributes map to OCCI attributes. The `readonly` modifier for DRMAA attributes translates to the immutability property. The concept of optional or possibly *UNSET* attributes in DRMAA is mapped to a OCCI attribute multiplicity of 0..X. Id-based or name-based referencing of instances (e.g. of a DRMAA session) is replaced by URI-based referencing.

Put some descriptions somewhere about what happened to the SessionManager::open... methods. They are replaced by knowledge about the URI.

Most DRMAA structures are just used as complex data type for return values or other attributes. They are mapped to OCCI as JSON-serialized structures (see Table 2):

- *ReservationInfo* struct
- *SlotInfo* struct
- *QueueInfo* struct
- *MachineInfo* struct
- *Version* struct

New version: All DRMAA struct instances become resources. When a function returns a struct instance, OCCI-DRMAA will create a new struct resource. The creation may be modeled by a POST

The *JobInfo* structure of DRMAA represents a set of information items for a job. It is modeled as OCCI mixin. This allows to use it both as reporting and filter configuration data structure. The reporting functionality comes from the combination of a `job` resource with the according `jobinfo` mixin. Filtering is supported by querying a `jobsession` resource that is connected with the `jobinfo` mixin for jobs under the given `jobinfo` filter constraints (see [3], Section 4.4.5):

- A `jobinfo` mixin is a set of additional information attributes about `job` resources.

[2] The root URI is not named DRMAA2 as suggested in the DRMAA IDL spec, since there was never a DRMAA OCCI before that might clash with this one.

DRMAA templates are data structures that express complex information entities 'as a whole'. They might be modified by a DRM system after their creation, which makes them additional OCCI resources without actions:

- A `jobtemplate` resource represents a particular job configuration.
- A `reservationtemplate` resource representing an advance reservation configuration.

(See footnote)[3]

The DRMAA session concept models the relationship of *Job* instances to *JobSession* instances, and of *Reservation* instances to *ReservationSession* instances. In OCCI-DRMAA, this is represented by OCCI links between the according resource entities:

- A `joblink` represents the connection of a job to it's job session.
- A `reservationlink` represents the connection of an advance reservation to it's reservation session.

(See footnote)[4]

Enumerations from the DRMAA specification are mapped directly to OCCI-DRMAA:

- JobState
- OperatingSystem
- CpuArchitecture
- ResourceLimitType
- JobTemplatePlaceholder (special mapping rule, see Table 2)
- DrmaaEvent
- DrmaaCapability

Table 1 describes the *Kind* instances for each of the OCCI-DRMAA entities:

In adherence to the DRMAA specification, *reservation*, *reservationsession*, *reservationtemplate* and *reservationlink* only MAY be supported by the OCCI-DRMAA implementation. In case, they MUST be supported as a whole. The support MUST be discoverable through the OCCI Query Interface. All other *Kind* instances MUST be supported.

Mapping of DrmaaCallback: registerEventNotification takes notification URI; if protocol is http, then the POST request sent to the client is standardized

3 Exception Mappings

DeniedByDrmsException: 401 (no auth available) / 403 (auth available, op not allowed) DrmCommunicationException: 500 TryLaterException: 503 with retry header TimeoutException: 410 InternalException:

[3] There was a long debate at OGF 34 if we should model DRMAA templates as serialized structs too. This brings potential rendering problems with long OCCI attribute values. Also, the server may want to report differing job templates, which would bloat the `job` resource return attributes on every request.

[4] We model the jobsession ¡-¿ job relation as OCCI link, especially for using the lifetime model for links. At least Thijs says there is one. Currently, this demands the server to return ALL such links when a resource is retrieved without filtering (see GFD.185, Section 3.4.5). Upcoming OCCI specs will have pagination support for this.

500 InvalidArgumentException: 400 InvalidSessionException: 404 InvalidStateException: 409 OutOfResourceException: 503 without retry header UnsupportedAttributeException: 400 UnsupportedOperationException: 405 ImplementationSpecificException: 500

4 Type system

The OCCI core model supports the notion of 'action attributes' as representation for parameters of an invocable operation (see [3], Section 4.5.4). OCCI attributes are always represented as String (see [3], Figure 2). On the other hand, DRMAA has a central definition of utilized parameter types. For this reason, Table 2 defines a mapping from DRMAA-IDL data types to OCCI string representations in JSON [?]. This allows the proper translation of attribute value encoding. Accordingly, constant values from the DRMAA definitions are mapped to OCCI-DRMAA as shown in Table 3.

Define ISO8601 as date format, to make the OCCI clients happy.

Instead of declaring the maximum number for long values, define the error if the value is too long.

DRMAAs notion of `UNSET` values is mapped to multiplicity of OCCI-DRMAA attributes. All attributes that may hold the value `UNSET` MUST be have a multiplicity ≥ 0.

(See footnote)[5]

5 Root resource

Requests for the root scheme URI MUST return the attributes described in Table 4.

close functions do not map - stateless clients. Destroy maps to DELETE

6 Mapping of DrmaaCapability

ADVANCE RESERVATION: Ask query interface if the according category is supported RESERVE SLOTS: Boolean attribute on the drmaa resource CALLBACK: Ask query interface if registerCallback action is supported BULK JOBS MAXPARALLEL: Boolean attribute on the drmaa resource JT EMAIL / JT STAGING / JT DEADLINE / JT MAXSLOTS / JT ACCOUNTINGID: Ask query interface if the according job template mixins are supported. ALl these mixins inherit from an abstract optional_jt_attribute mixin.

Add optional job template attributes as mixins, all mixin attributes must be optional. Leave open if a POST on job template resource adds automatically all mixins, or if the client is doing this.

RT DURATION / RT MACHINEOS / RT MACHINEARCH: see above. All these mixins inherit from an abstract optional_rt_attribute mixin.

RT STARTNOW: Boolean attribute on the drmaa resource

7 jobsession resource

Table 5 describes the actions available for a `jobsession` resource. Required attributes on actions are always mutable. Table 6 describes the attributes delivered on retrieval of a `jobsession` resource.

Map job categories to empty mixins, which all inherit

[5] This sentence is a safeguard for ourselves, in case we forgot something in the tables.

The original `JobSession::getJobArray` method is not represented as attribute or action. Instead, the server MUST support the retrieval of a specifically named `jobarray` resource based on the manual creation of an according URI beginning with *http://schemas.ogf.org/occi/drmaa#jobarray*.

The original `JobSession:getJobs` method is also not represented as attribute or action. Instead, the `jobsession` resource MUST return the references to all attached `job` resource instances as `joblink` OCCI links. This retrieval operation MUST also support a mixin-based filtering of this `job` set.

Not clear how the jobsession resource can transform to the jobsession+jobinfo resource for this activity.

On creation of a `jobsession` resource, the `contact` and / or the `sessionName` attribute MAY be provided by the client as part of the request. This maps to original DRMAA *SessionManager::createJobSession* method signature.

8 reservationsession resource

Table 7 describes the actions available for a `reservationsession` resource. Required attributes on actions are always mutable. Table 8 describes the attributes delivered on retrieval of a `reservationsession` resource.

The original `ReservationSession::getReservation` method is not represented as attribute or action. Instead, the server MUST support the retrieval of a specifically named `reservation` resource based on the manual creation of an according URI beginning with *http://schemas.ogf.org/occi/drmaa#reservation*.

The original `ReservationSession:getReservations` method is also not represented as attribute or action. Instead, the `reservationsession` resource MUST return the references all attached `reservation` resource instances as `reservationlink` OCCI links.

On creation of a `reservationsession` resource, the `contact` and / or the `sessionName` attribute MAY be provided by the client as part of the request. This maps to original DRMAA *SessionManager::createReservationSession* method signature.

9 monitoringsession resource

Table 9 describes the attributes delivered on retrieval of a `monitoringsession` resource. This resource has no actions defined.

The original `MonitoringSession:getAllJobs` method is not represented as attribute or action. Instead, the `monitoringsession` resource MUST return the references to all known `job` resource instances as OCCI links. This retrieval operation MUST also support a mixin-based filtering of this `job` set.

The original `MonitoringSession:getAllReservations` method is also not represented as attribute or action. Instead, the `monitoringsession` resource MUST return the references all known `reservation` resource instances as OCCI links.

The original `MonitoringSession:getAllQueues` method is represented with the `queue` attribute. On retrieval of the `monitoringsession` resource, the implementation MUST return a `queue` attribute for each supported queue in the target system. This retrieval operation MUST also support the provisioning of `queue` attributes by the client for filtering of the result. Implementations MUST support the filtering by queue name. Implementations MAY support the filtering by implementation-specific *QueueInfo* attributes.

The original `MonitoringSession:getAllMachines` method is represented with the `machine` attribute. On retrieval of the `monitoringsession` resource, the implementation MUST return a `machine` attribute for each execution host in the target DRM system. This retrieval operation MUST also support the provisioning of `machine` attributes by the client for filtering of the result. Implementations MUST support the filtering by machine name. Implementations MAY support the filtering by other (DRMAA-)mandatory or implementation-specific *MachineInfo* attributes.

On creation of a `monitoringsession` resource, the `contact` attribute MAY be provided by the client as part of the request. This maps to original DRMAA *SessionManager::openMonitoringSession* method signature.

10 job resource

Table 12 describes the actions available for a `job` resource. Table 13 describes the attributes delivered for the `job` resource.

The DRMAA *sessionName* attribute was replaced by the `session` resource link, which fulfills the same purpose, but fits better to the OCCI semantics.

The `jobTemplate` and `session` attributes are mutable by the client, in order to allow the implicit triggering of job execution by creating a `job` resource. In this case, it MUST be ensured that the client provides both of them with valid values. All other attempts to create `job` resources from the client side MUST fail.

(See footnote)[6]

The `job` resource can be combined with the `jobinfo` mixin.

explain use cases

11 jobarray resource

Table ?? describes the actions available for a `jobarray` resource. Table ?? describes the attributes delivered for the `jobarray` resource.

The DRMAA *sessionName* attribute was replaced by the `session` resource link, which fulfills the same purpose, but fits better to the OCCI semantics.

The original `JobArray:jobs` attribute is not represented as attribute or action. Instead, the `jobarray` resource MUST return the references to all jobs in the array as OCCI `job` links.

12 reservation resource

Table 14 describes the actions available for a `reservation` resource. Table 15 describes the attributes delivered for the `reservation` resource.

The DRMAA *sessionName* attribute was replaced by the `session` resource link, which fulfills the same purpose, but fits better to the OCCI semantics.

The `reservationTemplate` and `session` attributes are mutable by the client, in order to allow the implicit triggering of reservation requesting by creating a `reservation` resource. In this case, it MUST be ensured that the client provides both attributes with valid values. All other attempts to create `reservation` resources from the client side MUST fail.

[6] Starting of bulk jobs through this would mess up to much.

13 jobinfo mixin

Table 16 describes the attributes delivered by the usage of the `jobinfo` mixin on a resource. There are no actions defined by this mixin.

Implementations SHOULD consider that some of the `jobinfo` attributes are mandatory on job information reporting (see [?], Section 5.5). All attributes MUST be optional on mixin-based job filtering.

(See footnote)[7]

14 jobtemplate resource

Table 17 describes the attributes delivered for the `jobtemplate` resource. The resource has no actions.

(See footnote)[8]

The implementation MUST render all `jobtemplate` attributes immutable when the template is linked from a `job` resource.

(See footnote)[9]

15 reservationtemplate resource

Table 18 describes the attributes delivered for the `reservationtemplate` resource. The resource has no actions.

(See footnote)[10]

The implementation MUST render all `reservationtemplate` attributes immutable when the template is linked from a `reservation` resource.

(See footnote)[11]

16 Example Interactions with HTTP Rendering

Use ”drmaa2” everywhere.

GFD-P-R.185 [2] describes the rendering of OCCI through a RESTful HTTP interface. The following example shows how typical OCCI-DRMAA interactions would be modeled based on this rendering.

[7] DRMAA makes no statements on mandatory attributes in the filtering case, so we make all mixin attributes optional. On the other hand, this makes job information reporting more flexible than it should be. For example, an implementation could decide to not report allocatedMachines as job information, which is a violation of the DRMAA spec.

[8] DRMAA says that all attributes may have the value UNSET on submission, which maps to mupliplicity zero.

[9] This is the case when the job template resource represents the properties of a running job. The other case is the creation of a job template resource by the client for job submission purposes.

[10] DRMAA says that all attributes may have the value UNSET on submission, which maps to mupliplicity zero.

[11] This is the case when the reservation template resource represents the properties of a valid reservation. The other case is the creation of a reservation template resource by the client for reservation creation purposes.

16.1 Query DRMAA interfaces

```
> GET /-/ HTTP/1.1
> [...]
> Category: drmaa2;scheme="http://schemas.ogf.org/drmaa2"

< HTTP/1.1 200 OK
< [...]
< Category: drmaa2;scheme="http://schemas.ogf.org/drmaa2";class="kind";
rel="http://schemas.ogf.org/occi/core#resource";
            location="/drmaa2/";title="DRMAAv2 Interfaces";
            attributes="occi.drmaa2.drmsName occi.drmaa2.drmsVersion occi.drmaa2.drmaaName occi.drmaa2.

> GET /drmaa2/ HTTP/1.1
> [...]

< HTTP/1.1 200 OK
< Content-type: text/uri-list
< [...]
< http://example.com/drmaa2
```

Spezifizieren, das Collection immer Kardinalitaet 1 hat

16.2 Determine the DRM system information

```
> GET /drmaa2 HTTP/1.1
> [...]

< HTTP/1.1 200 OK
< [...]
< X-OCCI-Attribute: occi.drmaa2.drmsName="Platform LSF"
< X-OCCI-Attribute: occi.drmaa2.drmsVersion="{"major":"42", "minor":"0"}
< X-OCCI-Attribute: occi.drmaa2.drmaaName="Thijs's OCCI-DRMAA backend"
< X-OCCI-Attribute: occi.drmaa2.drmaaVersion="{"major":"2", "minor":"17"}
< [...]
```

16.3 Get all existing job sessions

¿ GET /drmaa2/jobsession/ HTTP/1.1 ¿ [...]

¡ HTTP/1.1 200 OK ¡ Content-type: text/uri-list ¡ [...] ¡ http://example.com/drmaa2/jobsession/17

16.4 Create a job session

```
> POST /drmaa2/jobsession/ HTTP/1.1
> [...]
> X-OCCI-Attribute: occi.drmaa2.contact="headnode.testbed.platform.com"
> X-OCCI-Attribute: occi.drmaa2.sessionName="MyTestSession"
> [...]
```

```
< HTTP/1.1 201 CREATED
< [...]
< Location: http://example.com/drmaa/jobsession/session1
< [...]
```

16.5 Submit a job

Step 1 - Create a `jobtemplate` *resource*:

```
> POST /drmaa/jobtemplate/ HTTP/1.1
> [...]
> X-OCCI-Attribute: occi.drmaa.remoteCommand="/bin/date"
> X-OCCI-Attribute: occi.drmaa.machineOS="LINUX"
> X-OCCI-Attribute: occi.drmaa.email="peter@troeger.eu"
> X-OCCI-Attribute: occi.drmaa.email="tmetsch@platform.com"
> X-OCCI-Attribute: occi.drmaa.emailOnTerminated=true
> [...]

< HTTP/1.1 201 CREATED
< [...]
< Location: http://example.com/drmaa/jobtemplate/template1
< [...]
```

Replace multiple attributes with the same name by list data type as argument (e.g. email)

Step 2, Option 1 - Perform a `jobsession` *action*:

```
> POST /drmaa/jobsession/session1?action=runjob HTTP/1.1
> [...]
> X-OCCI-Attribute: jobTemplate="/drmaa/jobtemplate/template1"
> [...]

< HTTP/1.1 201 CREATED
< [...]
< Location: http://example.com/drmaa/job/job42
< [...]
```

Step 2, Option 2 - Create a `job` *resource*:

```
> POST /drmaa/job/ HTTP/1.1
> [...]
> X-OCCI-Attribute: occi.drmaa.session="/drmaa/jobsession/session1"
> X-OCCI-Attribute: occi.drmaa.jobTemplate="/drmaa/jobtemplate/template1"
> [...]

< HTTP/1.1 201 CREATED
< [...]
< Location: http://example.com/drmaa/job/job43
< [...]
```

16.6 Fetch filtered list of jobs

```
> GET /drmaa/job/ HTTP/1.1
> [...]
> X-OCCI-Attribute: occi.drmaa2.queueName="foo"
> [...]

< HTTP/1.1 200 OK
< Content-type: text/uri-list
< [...]
< http://example.com/drmaa2/job/job43
< http://example.com/drmaa2/job/job44
```

When it works with an HTTP verb, do it; otherwise use an OCCI action

```



\subsection{Wait for job start}

\begin{verbatim}
> GET /drmaa/job/job43?action=waitstarted HTTP/1.1
> [...]
> X-OCCI-Attribute: occi.drmaa2.timeout="..."
> [...]

< HTTP/1.1 202 ACCEPTED
< [...]
< Location: /drmaa2/job/job43/waithandle1
< [...]

> GET /drmaa/job/job43/waithandle1 HTTP/1.1
> [...]
< HTTP/1.1 404 NOT FOUND

> GET /drmaa/job/job43/waithandle1 HTTP/1.1
> [...]
< HTTP/1.1 410 GONE

> GET /drmaa/job/job43/waithandle1 HTTP/1.1
> [...]
< HTTP/1.1 301 MOVED PERMANENTLY
< [...]
< Location: /drmaa2/job/job43
< [...]
```

404: Still waiting, 410: timeout happened, 301: Wait successful

16.7 Wait for some job to start

```
> GET /drmaa/jobsession/js44?action=waitanystarted HTTP/1.1
> [...]
> X-OCCI-Attribute: occi.drmaa2.timeout="..."
> X-OCCI-Attribute: occi.drmaa2.jobs="http://example.com/drmaa2/job/job44"
> X-OCCI-Attribute: occi.drmaa2.jobs="http://example.com/drmaa2/job/job42"
> [...]

< HTTP/1.1 202 ACCEPTED
< [...]
< Location: /drmaa2/jobsession/js44/waithandle1
< [...]

> GET /drmaa2/jobsession/js44/waithandle1 HTTP/1.1
> [...]
< HTTP/1.1 404 NOT FOUND

> GET /drmaa2/jobsession/js44/waithandle1 HTTP/1.1
> [...]
< HTTP/1.1 410 GONE

> GET /drmaa2/jobsession/js44/waithandle1 HTTP/1.1
> [...]
< HTTP/1.1 301 MOVED PERMANENTLY
< [...]
< Location: /drmaa2/job/job42
< [...]
```

16.8 Query if advanced reservation is supported, negative answer

```
> GET /-/ HTTP/1.1
> [...]
> Category: reservationsession;scheme="http://schemas.ogf.org/drmaa2"

< HTTP/1.1 204 NOCONTENT
< [...]

```

Check OCCI-compliant negative query answer

16.9 Query if advanced reservation is supported, positive answer

```
> GET /-/ HTTP/1.1
> [...]
> Category: reservationsession;scheme="http://schemas.ogf.org/drmaa2"

< HTTP/1.1 200 OK
< [...]
< Category: reservationsession;scheme="http://schemas.ogf.org/drmaa2";class="kind";
rel="http://schemas.ogf.org/occi/core#resource";
            location="/drmaa2/reservationsession/";title="DRMAAv2 Advance Reservation Sessions";
            attributes="occi.drmaa2.reservationsession.contact occi.drmaa2.reservationsession.sessionNa
```

16.10 Query all existing reservation sessions

¿ GET /drmaa2/reservationsession/ HTTP/1.1 ¿ [...]

¡ HTTP/1.1 200 OK ¡ Content-type: text/uri-list ¡ [...] ¡ http://example.com/drmaa2/reservationsession/rsess5 ¡ http://example.com/drmaa2/reservationsession/rsess4711 ¡ http://example.com/drmaa2/reservationsession/rsess42

16.11 Request an advance reservation

Step 1 - Create a `reservationtemplate` *resource*:

```
> POST /drmaa/reservationtemplate/ HTTP/1.1
> [...]
> X-OCCI-Attribute: occi.drmaa.startTime="2012-11-11T11:11:11"
> X-OCCI-Attribute: occi.drmaa.endTime="2012-11-12T00:00:00"
> X-OCCI-Attribute: occi.drmaa.minSlots=2
> X-OCCI-Attribute: occi.drmaa.maxSlots=5000
> [...]

< HTTP/1.1 201 CREATED
< [...]
< Location: http://example.com/drmaa/reservationtemplate/rtpl4711
< [...]
```

Step 2, Option 1 - Perform a `reservationsession` *action*:

```
> POST /drmaa2/reservationsession/rsess5?action=requestreservation HTTP/1.1
> [...]
> X-OCCI-Attribute: reservationTemplate="/drmaa/reservationtemplate/rtpl4711"
> [...]

< HTTP/1.1 201 CREATED
< [...]
< Location: http://example.com/drmaa/reservation/rs99xy
< [...]
```

Step 2, Option 2 - Create a `reservation` *resource*:

```
> POST /drmaa/reservation/ HTTP/1.1
> [...]
> X-OCCI-Attribute: occi.drmaa.session="/drmaa/reservationsession/rsess5"
> X-OCCI-Attribute: reservationTemplate="/drmaa/reservationtemplate/rtpl4711"
> [...]

< HTTP/1.1 201 CREATED
< [...]
< Location: http://example.com/drmaa/reservation/rs99xy
< [...]
```

16.12 Control a job

```
> POST /drmaa/job/job77?action=suspend HTTP/1.1
> [...]

< HTTP/1.1 200 OK
< [...]
```

16.13 Get the list of machines

Step 1 - Create a `monitoringsession` *resource*:

```
> POST /drmaa/monitoringsession/ HTTP/1.1
> [...]
> X-OCCI-Attribute: occi.drmaa.contact="headnode.testbed.platform.com"
> [...]

< HTTP/1.1 201 CREATED
< [...]
< Location: http://example.com/drmaa/monitoringsession/monitor8
< [...]
```

Step 2 - Fetch the `monitoringsession` *resource to get the machine list*:

```
> GET /drmaa/monitoringsession/monitor8 HTTP/1.1
> [...]

< HTTP/1.1 200 OK
< [...]
< X-OCCI-Attribute: occi.drmaa.machine="{"name":"exec1.testbed.platform.com","available":true,"sockets"
< X-OCCI-Attribute: occi.drmaa.machine="{"name":"exec2.testbed.platform.com","available":false,"sockets
< [...]
```

17 Security Considerations

Security considerations from a DRM system point of view are clarified by the DRMAA root specification. An implementation MUST consider the regulations and security advices given there.

The DRMAA API does not specifically assume the existence of a particular security infrastructure in the DRM system. The scheduling scenario described herein presumes that security is handled at the point of interaction with the DRM system. It is assumed that credentials owned by the application using the API are in effect for the DRMAA implementation too, so that it acts as stakeholder for the application. This relays the responsibility of authentication to the OCCI rendering specification that is used to realize the wire protocol of an implementation.

DRMAA implementers SHOULD guard their product against buffer overflows that can be exploited through DRMAA enabled interactive applications or portals. Implementations of the DRMAA API will most likely require a network to coordinate subordinate DRM system requests. However, the API makes no assumptions about the security posture provided by the networking environment. Therefore, application developers SHOULD also consider the security implications of “on-the-wire” communications in this case.

18 Contributors

19 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.

20 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.

21 Full Copyright Notice

22 References

[1] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels, March 1997.

[2] Thijs Metsch and Andy Edmonds. Open Cloud Computing Interface - RESTful HTTP Rendering. http://www.ogf.org/documents/GFD.185.pdf, June 2011.

[3] Ralf Nyren, Andy Edmonds, Alexander Papaspyrou, and Thijs Metsch. Open Cloud Computing Interface - Core. http://www.ogf.org/documents/GFD.183.pdf, April 2011.

[4] Peter Tröger, Roger Brobst, Daniel Gruber, Mariusz Mamonski, and Daniel Templeton. Distributed Resource Management Application API Version 2 (DRMAA). http://www.ogf.org/documents/GFD.194.pdf, January 2012.

List of Tables

Term	Scheme	Title	Related Kind
jobsession	`<schema>/drmaa#`	Job Session resource	`<schema>/core#resource`
reservationsession	`<schema>/drmaa#`	Reservation Session resource	`<schema>/core#resource`
monitoringsession	`<schema>/drmaa#`	Monitoring Session resource	`<schema>/core#resource`
job	`<schema>/drmaa#`	Job resource	`<schema>/core#resource`
jobarray	`<schema>/drmaa#`	Job Array resource	`<schema>/core#resource`
reservation	`<schema>/drmaa#`	Reservation resource	`<schema>/core#resource`
jobinfo	`<schema>/drmaa/job#`	Job Information Mixin	-
jobtemplate	`<schema>/drmaa#`	Job Template resource	`<schema>/core#resource`
reservationtemplate	`<schema>/drmaa#`	Reservation Template resource	`<schema>/core#resource`
joblink	`<schema>/drmaa#`	Relation from `jobsession` to `job` resource	`<schema>/core#link`
reservationlink	`<schema>/drmaa#`	Relation from `reservationsession` to `reservation` resource	`<schema>/core#link`

Table 1: The *Kind* instances defined for OCCI-DRMAA. The base URL *http://schemas.ogf.org/occi* has been replaced with `<schema>` in this table for a better readability experience.

DRMAA type	OCCI-DRMAA representation
string	JSON string Example: `"/bin/date"`
long / long long	JSON number Example: `42`
double	JSON float Example: `7.02`
boolean	JSON boolean, defaults to `false` if attribute is not set Example: `true`, `false`
struct	JSON dictionary with member names as keys Example: `{"machineName":"node1.drmaa.org", "slots":42}`
Dictionary	JSON dictionary Example: `{"PATH": "/usr/bin", "OMP_NUM_THREADS": "64"}`
enum value	JSON string Example: `"RUNNING"`
JobTemplatePlaceholder	JSON string surrounded by ”$” Example: `"$HOME_DIRECTORY$"`
OrderedStringList	JSON array of strings Example: `["node1.drmaa.org", "node2.drmaa.org"]`
OrderedSlotInfoList	JSON array of JSON dictionaries Example: `[{"machineName": "node1.drmaa.org", "slots": 42}]`
AbsoluteTime	JSON string in ISO8601 format Example: `"2003-04-01T13:01:02"`
TimeAmount	JSON number representing seconds Example: `3600`

Table 2: The data types used for attributes in OCCI-DRMAA.

(See footnote)[12]

DRMAA constant	OCCI-DRMAA representation
ZERO_TIME	JSON number `0`
INFINITE_TIME	JSON number `-1`
NOW	JSON string `"now"`

Table 3: The constants used in OCCI-DRMAA.

Attribute	Type	Multiplicity	Mutability
`occi.drmaa.drmsName`	string	1	Immutable
`occi.drmaa.drmsVersion`	DRMAA Version	1	Immutable
`occi.drmaa.drmaaName`	string	1	Immutable
`occi.drmaa.drmaaVersion`	DRMAA Version	1	Immutable

Table 4: Attributes of the root scheme URI

Action term	Attributes	Attribute type	M.	DRMAA equivalent
`runjob`	`jobTemplate`	`jobtemplate` URI	1	JobSession::runJob()
`runbulkjobs`	`jobTemplate`	`jobtemplate` URI	1	JobSession::runBulkJobs()
	`beginIndex`	long	1	
	`endIndex`	long	1	
	`step`	long	1	
	`maxParallel`	long	1	
`waitanystarted`	`job`	`job` URI	1..*	JobSession::waitAnyStarted()
	`timeout`	DRMAA TimeAmount	1	
`waitanyterminated`	`job`	`job` URI	1..*	JobSession::waitAnyTerminated()
	`timeout`	DRMAA TimeAmount	1	

Table 5: Actions available for a `jobsession` resource

Attribute	Type	Multiplicity	Mutability
`occi.drmaa.contact`	string	1	Mutable
`occi.drmaa.sessionName`	string	1	Mutable
`occi.drmaa.jobCategory`	string	0..*	Immutable

Table 6: Attributes of the `jobsession` resource

Action term	Required attributes	Attribute type	M.	DRMAA equivalent
`requestreservation`	`reservationTemplate`	`reservationtemplate` URI	1	ReservationSession::requestReservati

Table 7: Actions available for a `reservationsession` resource

Attribute	type	Multiplicity	Mutability
`occi.drmaa.contact`	string	1	Mutable
`occi.drmaa.sessionName`	string	1	Mutable

Table 8: Attributes of the `reservationsession` resource

Attribute	Type	Multiplicity	Mutability
`occi.drmaa.contact`	string	1	Mutable
`occi.drmaa.queue`	DRMAA QueueInfo	0..*	Mutable
`occi.drmaa.machine`	DRMAA MachineInfo	0..*	Mutable

Table 9: Attributes of the `monitoringsession` resource

Action term	Required attributes	Attribute type	Multiplicity	DRMAA equivalent
suspend	-	-	-	Job::suspend()
resume	-	-	-	Job::resume()
hold	-	-	-	Job::hold()
release	-	-	-	Job::release()
terminate	-	-	-	Job::terminate()
waitstarted	timeout	DRMAA TimeAmount	1	Job::waitStarted()
waitterminated	timeout	DRMAA TimeAmount	1	Job::waitTerminated()

Table 10: Actions available for a job resource

Attribute	Type	Multiplicity	Mutability
occi.drmaa.jobId	string	1	Immutable
occi.drmaa.session	jobsession URI	0..1	Mutable
occi.drmaa.jobTemplate	jobtemplate URI	1	Mutable
occi.drmaa.state	Enum (DRMAA JobState)	1	Immutable
occi.drmaa.substate	string	0..1	Immutable

Table 11: Attributes of the job resource

Action term	Required attributes	Attribute type	Multiplicity	DRMAA equivalent
suspend	-	-	-	JobArray::suspend()
resume	-	-	-	JobArray::resume()
hold	-	-	-	JobArray::hold()
release	-	-	-	JobArray::release()
terminate	-	-	-	JobArray::terminate()

Table 12: Actions available for a jobarray resource

Attribute	Type	Multiplicity	Mutability
occi.drmaa.jobArrayId	string	1	Immutable
occi.drmaa.session	jobsession URI	0..1	Immutable
occi.drmaa.jobTemplate	jobtemplate URI	1	Immutable

Table 13: Attributes of the jobarray resource

Action term	Required attributes	Attribute type	Multiplicity	DRMAA equivalent
terminate	-	-	-	Reservation::terminate()

Table 14: Actions available for a reservation resource

Attribute	Type	Multiplicity	Mutability
occi.drmaa.reservationId	string	1	Immutable
occi.drmaa.session	reservationsession URI	0..1	Mutable
reservationTemplate	reservationtemplate URI	1	Mutable

Table 15: Attributes of the reservation resource

Attribute	Type	Multiplicity	Mutability
`occi.drmaa.jobId`	string	0..1	Mutable
`occi.drmaa.exitStatus`	long	0..1	Mutable
`occi.drmaa.terminatingSignal`	string	0..1	Mutable
`occi.drmaa.annotation`	string	0..1	Mutable
`occi.drmaa.jobState`	Enum (DRMAA JobState)	0..1	Mutable
`occi.drmaa.jobSubState`	string	0..1	Mutable
`occi.drmaa.allocatedMachines`	DRMAA OrderedSlotInfoList	0..1	Mutable
`occi.drmaa.submissionMachine`	string	0..1	Mutable
`occi.drmaa.jobOwner`	string	0..1	Mutable
`occi.drmaa.slots`	long	0..1	Mutable
`occi.drmaa.queueName`	string	0..1	Mutable
`occi.drmaa.wallclockTime`	DRMAA TimeAmount	0..1	Mutable
`occi.drmaa.cpuTime`	long	0..1	Mutable
`occi.drmaa.submissionTime`	DRMAA AbsoluteTime	0..1	Mutable
`occi.drmaa.dispatchTime`	DRMAA AbsoluteTime	0..1	Mutable
`occi.drmaa.finishTime`	DRMAA AbsoluteTime	0..1	Mutable

Table 16: Attributes of the `jobinfo` mixin

Attribute	Type	Multiplicity	Mutability
`occi.drmaa.remoteCommand`	string	0..1	Mutable
`occi.drmaa.args`	DRMAA OrderedStringList	0..1	Mutable
`occi.drmaa.submitAsHold`	boolean	0..1	Mutable
`occi.drmaa.rerunnable`	boolean	0..1	Mutable
`occi.drmaa.jobEnvironment`	DRMAA Dictionary	0..1	Mutable
`occi.drmaa.workingDirectory`	string	0..1	Mutable
`occi.drmaa.jobCategory`	string	0..1	Mutable
`occi.drmaa.email`	string	0..*	Mutable
`occi.drmaa.emailOnStarted`	boolean	0..1	Mutable
`occi.drmaa.emailOnTerminated`	boolean	0..1	Mutable
`occi.drmaa.jobName`	string	0..1	Mutable
`occi.drmaa.inputPath`	string	0..1	Mutable
`occi.drmaa.outputPath`	string	0..1	Mutable
`occi.drmaa.errorPath`	string	0..1	Mutable
`occi.drmaa.joinFiles`	boolean	0..1	Mutable
`occi.drmaa.reservationId`	`reservation` URI	0..1	Mutable
`occi.drmaa.queueName`	string	0..1	Mutable
`occi.drmaa.minSlots`	long	0..1	Mutable
`occi.drmaa.maxSlots`	long	0..1	Mutable
`occi.drmaa.priority`	long	0..1	Mutable
`occi.drmaa.candidateMachines`	DRMAA OrderedStringList	0..1	Mutable
`occi.drmaa.minPhysMemory`	long	0..1	Mutable
`occi.drmaa.machineOS`	Enum (DRMAA OperatingSystem)	0..1	Mutable
`occi.drmaa.machineArch`	Enum (DRMAA CpuArchitecture)	0..1	Mutable
`occi.drmaa.startTime`	DRMAA AbsoluteTime	0..1	Mutable
`occi.drmaa.deadlineTime`	DRMAA AbsoluteTime	0..1	Mutable
`occi.drmaa.stageInFiles`	DRMAA Dictionary	0..1	Mutable
`occi.drmaa.stageOutFiles`	DRMAA Dictionary	0..1	Mutable
`occi.drmaa.resourceLimits`	DRMAA Dictionary	0..1	Mutable
`occi.drmaa.accountingId`	string	0..1	Mutable

Table 17: Attributes of the `jobtemplate` resource

Attribute	Type	Multiplicity	Mutability
`occi.drmaa.reservationName`	string	0..1	Mutable
`occi.drmaa.startTime`	DRMAA AbsoluteTime	0..1	Mutable
`occi.drmaa.endTime`	DRMAA AbsoluteTime	0..1	Mutable
`occi.drmaa.duration`	DRMAA TimeAmount	0..1	Mutable
`occi.drmaa.minSlots`	long	0..1	Mutable
`occi.drmaa.maxSlots`	long	0..1	Mutable
`occi.drmaa.jobCategory`	string	0..1	Mutable
`occi.drmaa.userACL`	string	0..*	Mutable
`occi.drmaa.candidateMachines`	DRMAA OrderedStringList	0..1	Mutable
`occi.drmaa.minPhysMemory`	long	0..1	Mutable
`occi.drmaa.machineOS`	Enum (DRMAA OperatingSystem)	0..1	Mutable
`occi.drmaa.machineArch`	Enum (DRMAA CpuArchitecture)	0..1	Mutable

Table 18: Attributes of the `reservationtemplate` resource