UNICORE 7 – Middleware Services for Distributed and Federated Computing

Krzysztof Benedyczak¹, Bernd Schuller², Maria Petrova El-Sayed², Richard Grunzke³

¹ Interdisciplinary Center for Math. and Comp. Modelling, Warsaw, Poland
² JSC, Forschungszentrum Jülich GmbH, Germany
³ ZIH, Technische Universität Dresden, Germany

HPCS 2016, Innsbruck, July 21, 2016
Outline

- UNICORE – UNiform Interface to COmputing and data REsources
- Services for Distributed and Federated computing
  - UNICORE : overview
  - Unity : user authentication and identity management
  - UNICORE Web Portal
  - RESTful APIs
- Summary and outlook
Local batch system LoadLeveller

- Login/Password
- qsub, qstat, mpirun, ...
- /usr/local/apps/myapp/bin/myapp, ...
- ~/mydata/2011/job123/ergebnisse.txt, ...
How can I ...

- ... use multiple, heterogeneous systems seamlessly,
- ... manage my job input data and results?
- ... across multiple systems? Workflows?
- ... integrate HPC and big data into custom applications and portals?
UNICORE

Clients

Web
Command line
GUI
API

Services

Workflows
Jobs
Data Management
Discovery

Resources

Compute
Storage

Security

Federations

Policies

Users
Typical UNICORE installation at a HPC center
Workflow enactment
- Task execution
  - TargetSystemFactory
  - TargetSystem
  - JobManagement
  - Reservations

StorageFactory
- StorageManagement
- FileTransfer
- Metadata

Registry
- Resource Broker

Services

- Workflows
- Jobs
- Data Management
- Discovery
Default setup

Access to resource manager and file system via TargetSystemInterface (TSI) daemon installed on the cluster login node(s)
Factory services: virtualisation support

- Set up a virtual image during initialisation phase
- Aim at OpenStack VMs, Amazon EC2, ...

1. createTSS()
   - provide parameters

2. return TSS address

3. setup

4. use when ready, to submit jobs etc
Storage Management Service

- File systems
- Apache HDFS
- S3
- CDMI (prototype)
- ...

Client

<table>
<thead>
<tr>
<th>mkdir, ls, rm, stat, ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>upload download</td>
</tr>
<tr>
<td>server-to-server copy</td>
</tr>
</tbody>
</table>

SMS
Storage Management Service: more than a file system

- Initiate file transfers
  - Multi-protocol support
  - Scheduled server-to-server copy

- Metadata management
  - Schema-free, key-value
  - Indexed via Lucene, searchable

- Rule-based data processing
  - New files automatically trigger actions
  - e.g. metadata extraction, compression, etc
Factory services: virtualisation support

- Different types of storage backends can be supported
- User can select and provide required parameters
- Batch systems (Torque, Slurm, LoadLeveler, GridEngine, ...)
- Apache Hadoop (YARN)
- Direct execution (e.g. on Windows)
- ... (extensible)

- File systems
- Apache HDFS
- S3
- ... (extensible)
Federated access: security is key
UNICORE – Basic security flow

- **Service invocation:** a web service call is made to a UNICORE service
- **Authentication:** who is the user?
  - Results in the user's X.500 DN ("CN=..., O=..., OU=…, C=...")
- **Assign attributes** to the user
  - Standard attributes: role, Unix ID, groups, etc.
  - Custom attributes: (e.g. S3 access and secret keys)
- **Authorisation**
  - Add context: e.g. who owns the resource?
  - Check resource policies (ACLs)
  - Check server policies (XACML)
- → **Allow or deny** the request
Delegation

- Allow Service to work on behalf of the user

- UNICORE solution based on SAML
  - Use chain of signed assertions
  - Trust always delegated to particular server
  - Can be validated and audited

1. User submits job
2. Server A uploads results
End-user authentication in UNICORE

- Pre-UNICORE 7: X.509 client certificates REQUIRED for end-users
  - Users tend to hate them
    - All sorts of usage issues
  - Lack of understanding leads to lack of security (sending keys via email etc)
  - Users understand passwords
    - and it is relatively easy to teach basic security measures
Certificate-less end-user authentication

- **Goal:** no end-user certificates (not even short-lived)

- **Approach**
  - Use *signed SAML assertions* for authentication
  - Issued and signed by a trusted service
  - Flexible solution is required: e.g. want support for existing SAML Identity providers, federations like DFN AAI, OAuth, etc

- **Implications**
  - Client – server TLS is not client-authenticated
  - End-user cannot sign anything (no „non-repudiation“ guarantee)
Introducing Unity

- Complete **Authentication and Identity Management** solution
- Manage users and user attributes, group membership
- Separate, standalone product: [www.unity-idm.eu](http://www.unity-idm.eu)
- Increasing take-up: e.g. HBP, EuDAT

![Diagram](image)
Unity architecture
Managing Unity via web application

**UNITY administration interface**

Logged as: Default Administrator [entity id: 1]

- Contents management
- Registrations management
- Schema management
- Server management

Groups

- Root (/)
  - A
  - D
  - portal

Group /portal members

<table>
<thead>
<tr>
<th>ENTITY</th>
<th>IDENTITY TYPE</th>
<th>IDENTITY</th>
<th>x500Name</th>
<th>CN=Demo User,O=UNICORE,C=EU</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3]</td>
<td>userName</td>
<td>demo</td>
<td></td>
<td>ENA</td>
</tr>
<tr>
<td>[3]</td>
<td>persistent</td>
<td>5c1e8334-e268-4ddd-a7c7-3097bc320813</td>
<td></td>
<td>ENA</td>
</tr>
</tbody>
</table>

Group /portal details

- GROUPS ATTRIBUTES CLASSES
  - UNICORE portal attributes

Attributes of entity [3] in group /portal

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>email</td>
<td><a href="mailto:test@example.com">test@example.com</a></td>
</tr>
<tr>
<td>cn</td>
<td></td>
</tr>
</tbody>
</table>

Information

- Directly defined
- Created at 11/16/14 10:19 AM updated at 11/16/14 10:19 AM

- Value
  - test@example.com
Example: Unity authentication assertion

<urn:Assertion>...
<dsig:Signature>...</dsig:Signature>
<urn:Subject>
<urn:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">CN=Demo User,O=UNICORE,C=EU</urn:NameID>
<urn:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:sender-vouches">
<urn:SubjectConfirmationData NotOnOrAfter="2014-11-16T10:30:23.334Z"/>
</urn:SubjectConfirmation>
</urn:Subject>
<urn:AttributeStatement>
<urn:Attribute Name="cn">
<urn:AttributeValue>Demo User</urn:AttributeValue>
</urn:Attribute>
<urn:Attribute Name="email">
<urn:AttributeValue>test@example.com</urn:AttributeValue>
</urn:Attribute>
<urn:Attribute Name="memberOf">
<urn:AttributeValue>/portal</urn:AttributeValue>
</urn:Attribute>
</urn:AttributeStatement>
</urn:Assertion>

1. authenticate
1.1. return attributes
Name: ...
Attributes: ...

Unity

User

X.509

Unity authentication assertion
Portal
Third-party science gateways
UCC
UFTP
Eclipse-base Rich Client
Custom clients
RESTful API
Java APIs
Based on the Eclipse framework

Building, submitting and monitoring jobs and workflows

Integrated data and storage management

X.509 and Unity for AuthN

*Mostly targeted at expert users*
UNICORE Portal

- Aim for a simple, easy-to-use web application

- Simple use cases
  - Job and (limited) workflow management
  - Data management

- Less details exposed to user

- Implementation choices
  - Java-based, VAADIN web framework
  - Use UNICORE SOAP/WS APIs
UNICORE Portal – various

- Several „list“ views, e.g. jobs, sites

![Jobs Browser](image)

- Simple workflow creation
- JavaScript
UNICORE Portal: Data manager

Select **remote** storage

Initiate **data movement** (direct, not via portal)
RESTful APIs to UNICORE services
Use case: the Human Brain Project's HPAC platform

https://collab.humanbrainproject.eu

1. authenticate
   returns OIDC token

2. pass OIDC Bearer token

3. OIDC Bearer token
   returns signed SAML

3.1 validate OIDC

https://collab.humanbrainproject.eu

- BSC: HPC site
- CINECA: HPC site
- CSCS: HPC site
- JSC: HPC site
- KIT: S3 storage

Use case: the Human Brain Project's HPAC platform
SOAP and WS(RF) – Defining UNICORE since 2004/2005

Pros

- Strongly typed – XML schema based
- SOAP header/envelope mechanism
- WS-Security, SAML, etc are well established

Cons

- CPU intensive (XML processing, XML signatures)
- Complex interface (look at a typical WSDL!)
- Only Java and C# can be realistically used on the client side
RESTful APIs to UNICORE Services

- REST
  - Document / Resource oriented approach
  - HTTP semantics (GET, PUT, POST, DELETE, error codes, caching, …)
  - Multiple message formats and resource representations can be used
    - JSON, XML, HTML, …
  - Several authentication options (HTTP basic, OAuth, …)
  - Clients in all languages (even `curl` or `wget`)

- Keep SOAP/WS (for backwards compatibility), fully in sync with RESTful APIs
Example: job submission

```json
job.u:
{
    "Executable": "/bin/echo",
    "Arguments": ["Hello World"],
}
```

```
$> curl -X POST -H "Content-Type: application/json"
   --data-binary @job.u
   https://localhost:8080/DEMO SITE/rest/core/jobs

HTTP/1.1 201 Created
Content-Type: application/json; charset=ISO8859-1
Location: https://localhost:8080/DEMO SITE/rest/core/jobs/74198236e970429db55ca7d59c831f14
```
Summary

- UNICORE – a complete solution for building federations

- Main progress in UNICORE 7
  - Simplify user experience
    - Make X.509 user certs obsolete
    - Web portal targetted at non-expert users
  - Simplify integration options
    - Complete set of RESTful APIs for computing and data
      → bringing HPC to the Web!
  - Widen integration options
    - Unity as a universal solution for federated identify management solution
Outlook

- Consolidate and simplify
  - Installation and configuration
  - Packaging and automation of deployment

- Add/extend support for
  - Cloud resources (OpenStack, EC2, …)
  - Virtualised applications (Docker)

- [http://www.unicore.eu](http://www.unicore.eu)
Team / Thank you

- Björn Hagemeier, Valentina Huber, André Giesler, Boris Orth, Mariya Petrova, Jedrzej Rybicki, Rajveer Saini, Bernd Schuller and many others at JSC
- Krzysztof Benedyczak, Marcelina Borcz, Rafał Kluszczynski, Piotr Bała and others at ICM / Warsaw University
- Richard Grunzke and others at Technical University Dresden
- Students: Burak Bengi, Maciej Golik, Konstantine Muradov
- … many others who reported bugs, suggested features, contributed code and provided patches

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 604102 (Human Brain Project)
A federation software suite

- Secure and seamless access to compute and data resources
- Focus on scientific applications and workflows
- Complies with typical HPC centre policies
- Complete solutions: APIs, clients, services, ...
- Java/Python based, supports UNIX, MacOS, Windows and many resource management systems (Torque, Slurm, SGE, …)
- Long development history (since 1997)
- Open source, BSD licensed, visit http://www.unicore.eu