Security Model and Harmonization Options/Plans in context of the European Middleware Initiative

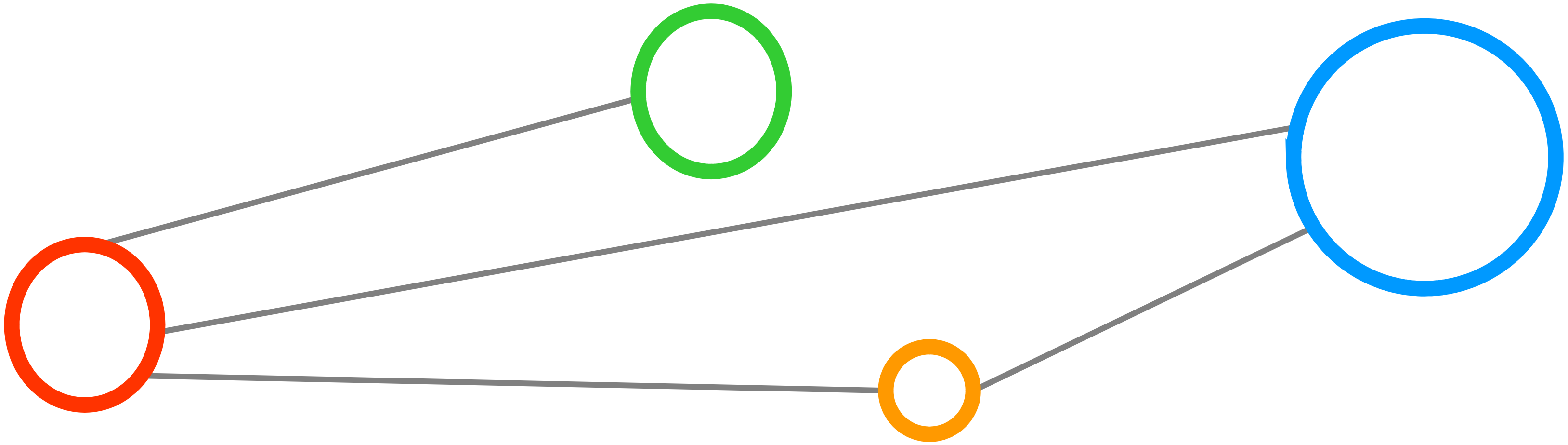
http://www.unicore.eu

Morris Riedel (m.riedel@fz-juelich.de)
Jülich Supercomputing Centre (JSC)

Outline

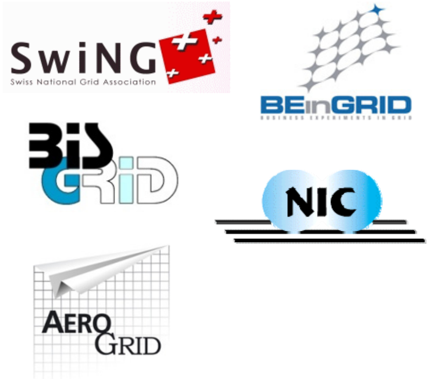
- UNICORE 101
- Security Remarks & Constraints
- Foundational Security Elements
- Security Setups
- One Example
- Future Topics
- Discussions

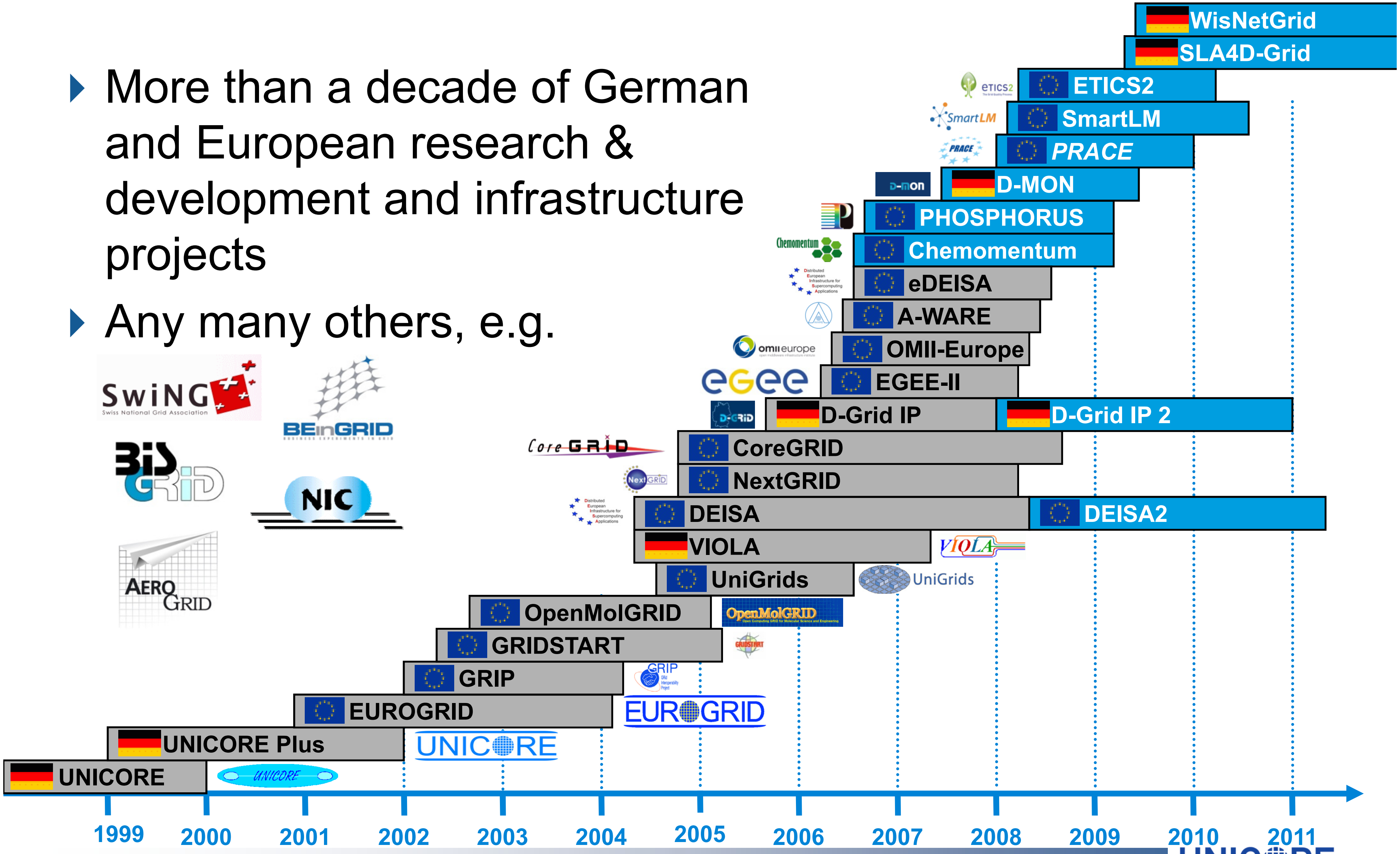
UNICORE 101



UNICORE Projects

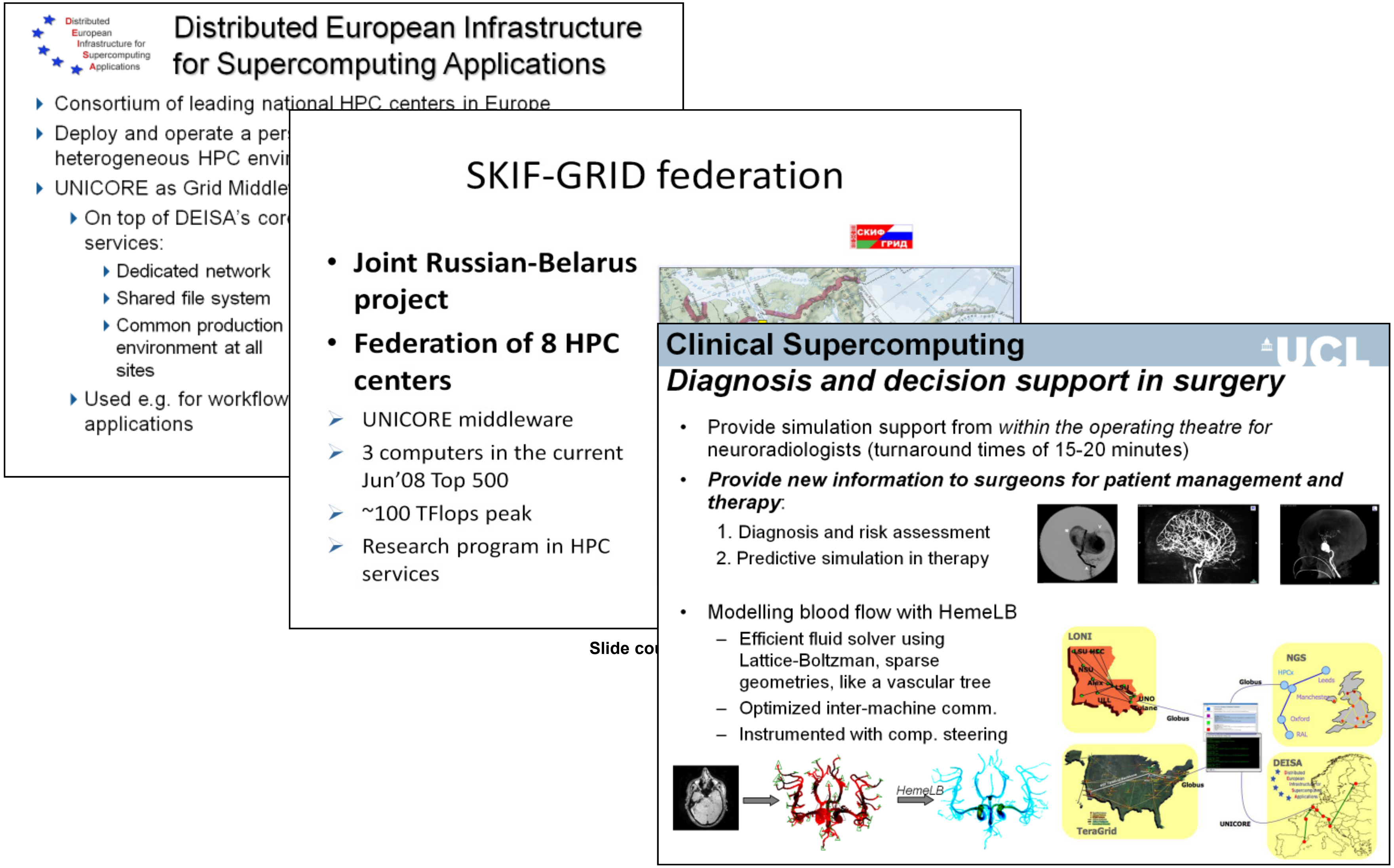
- More than a decade of German and European research & development and infrastructure projects
- Any many others, e.g.





UNICORE COMMUNITY

UNICORE in Supercomputing



Distributed European Infrastructure for Supercomputing Applications

Distributed European Infrastructure for Supercomputing Applications

- Consortium of leading national HPC centers in Europe
- Deploy and operate a per
  heterogeneous HPC envi
- UNICORE as Grid Middle
  - On top of DEISA's cor
    services:
    - Dedicated network
    - Shared file system
    - Common production environment at all sites
  - Used e.g. for workflow applications

SKIF-GRID federation

- **Joint Russian-Belarus project**
- **Federation of 8 HPC centers**
  - UNICORE middleware
  - 3 computers in the current Jun'08 Top 500
  - ~100 TFlops peak
  - Research program in HPC services

Slide co

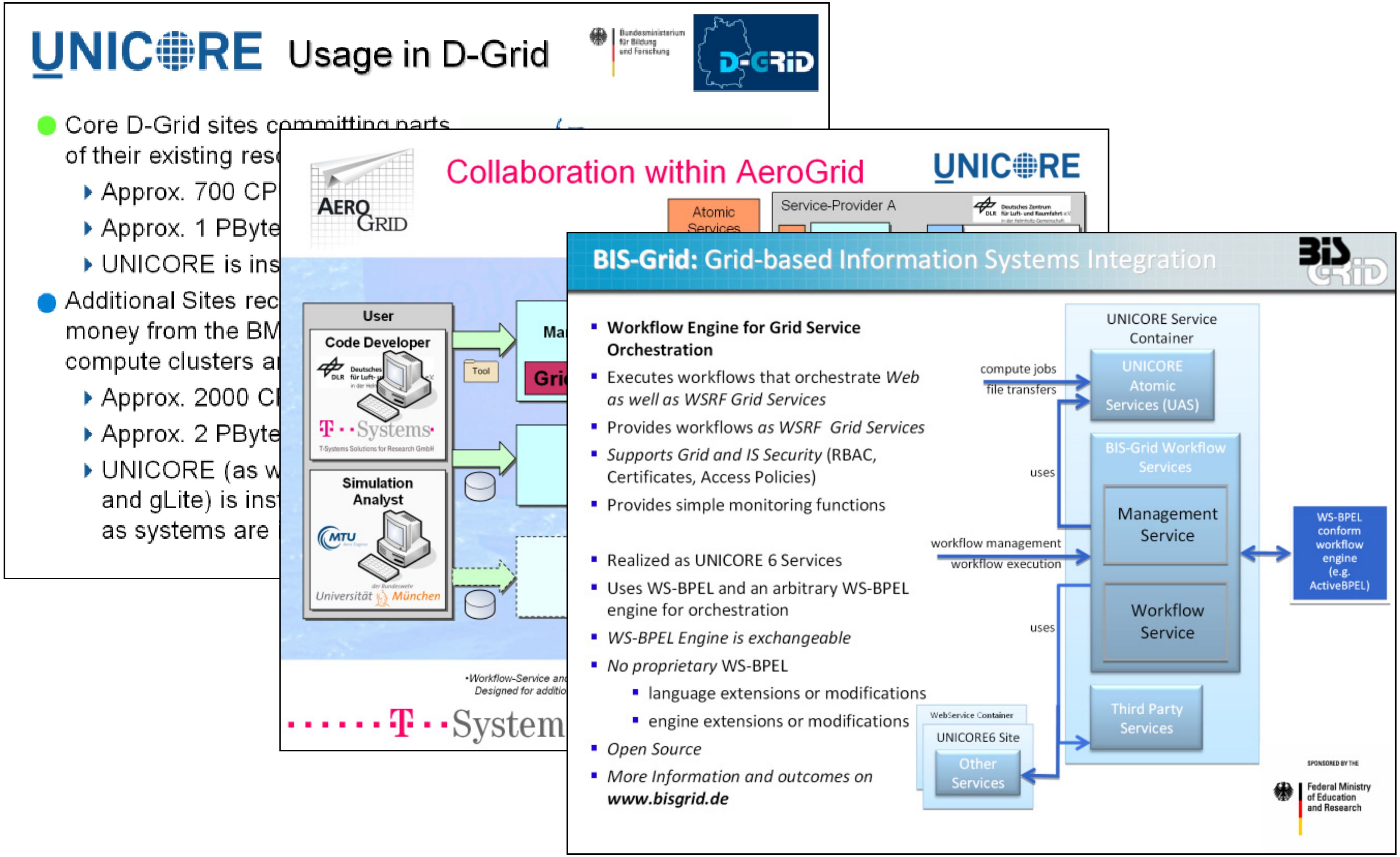
Clinical Supercomputing

Diagnosis and decision support in surgery

- Provide simulation support from *within the operating theatre for* neuroradiologists (turnaround times of 15-20 minutes)
- ***Provide new information to surgeons for patient management and therapy***:
  1. Diagnosis and risk assessment
  2. Predictive simulation in therapy
- Modelling blood flow with HemeLB
  - Efficient fluid solver using Lattice-Boltzman, sparse geometries, like a vascular tree
  - Optimized inter-machine comm.
  - Instrumented with comp. steering

Slide courtesy of Peter Coveney (UCL)

UNICORE in National Grids



UNICORE Usage in D-Grid

Bundesministerium für Bildung und Forschung

D-GRID

- Core D-Grid sites committing parts of their existing res
  - Approx. 700 CP
  - Approx. 1 PByte
  - UNICORE is ins
- Additional Sites rec money from the BM compute clusters a
  - Approx. 2000 C
  - Approx. 2 PByte
  - UNICORE (as w and gLite) is inst as systems are

Collaboration within AeroGrid

UNICORE

AERO GRID

Atomic Services

Service-Provider A

DLR Deutsches Zentrum für Luft- und Raumfahrt e.V. in der Helmholtz-Gemeinschaft

User

Code Developer

DLR Deutsches für Luft- in der Helm

T··Systems·

T-Systems Solutions for Research GmbH

Simulation Analyst

MTU Aero Engines

Universität der Bundeswehr München

Tool

Ma

Gri

•Workflow-Service an

Designed for additio

T··System

BIS-Grid: Grid-based Information Systems Integration

BIS GRID

- **Workflow Engine for Grid Service Orchestration**
- Executes workflows that orchestrate *Web as well as WSRF Grid Services*
- Provides workflows *as WSRF Grid Services*
- *Supports Grid and IS Security* (RBAC, Certificates, Access Policies)
- Provides simple monitoring functions

- Realized as UNICORE 6 Services
- Uses WS-BPEL and an arbitrary WS-BPEL engine for orchestration
- *WS-BPEL Engine is exchangeable*
- *No proprietary* WS-BPEL
  - language extensions or modifications
  - engine extensions or modifications
- *Open Source*
- *More Information and outcomes on* ***www.bisgrid.de***

UNICORE Service Container

compute jobs

file transfers

UNICORE Atomic Services (UAS)

uses

BIS-Grid Workflow Services

Management Service

workflow management

workflow execution

Workflow Service

WS-BPEL conform workflow engine (e.g. ActiveBPEL)

uses

Third Party Services

WebService Container

UNICORE6 Site

Other Services

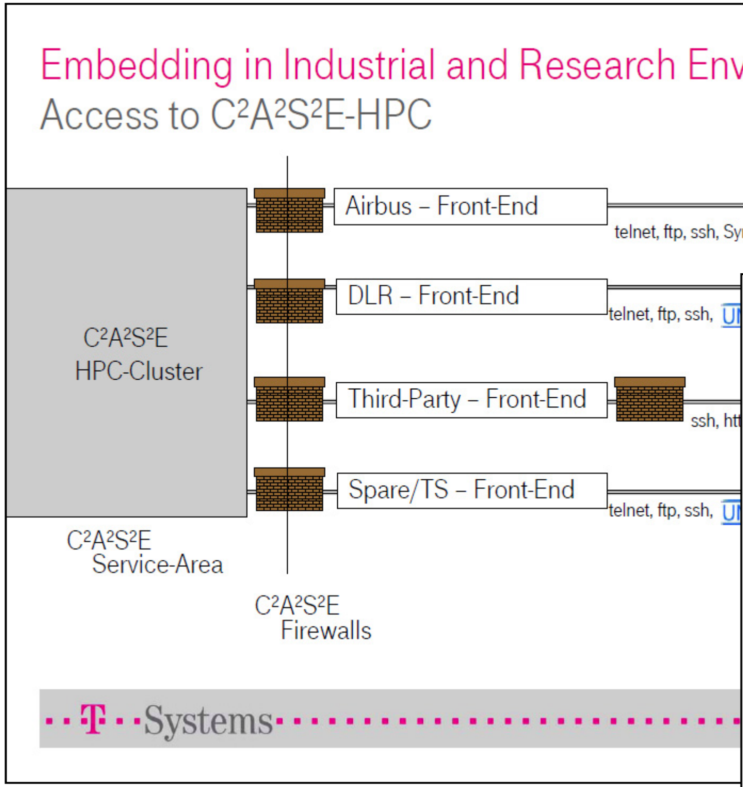
SPONSORED BY THE

Federal Ministry of Education and Research

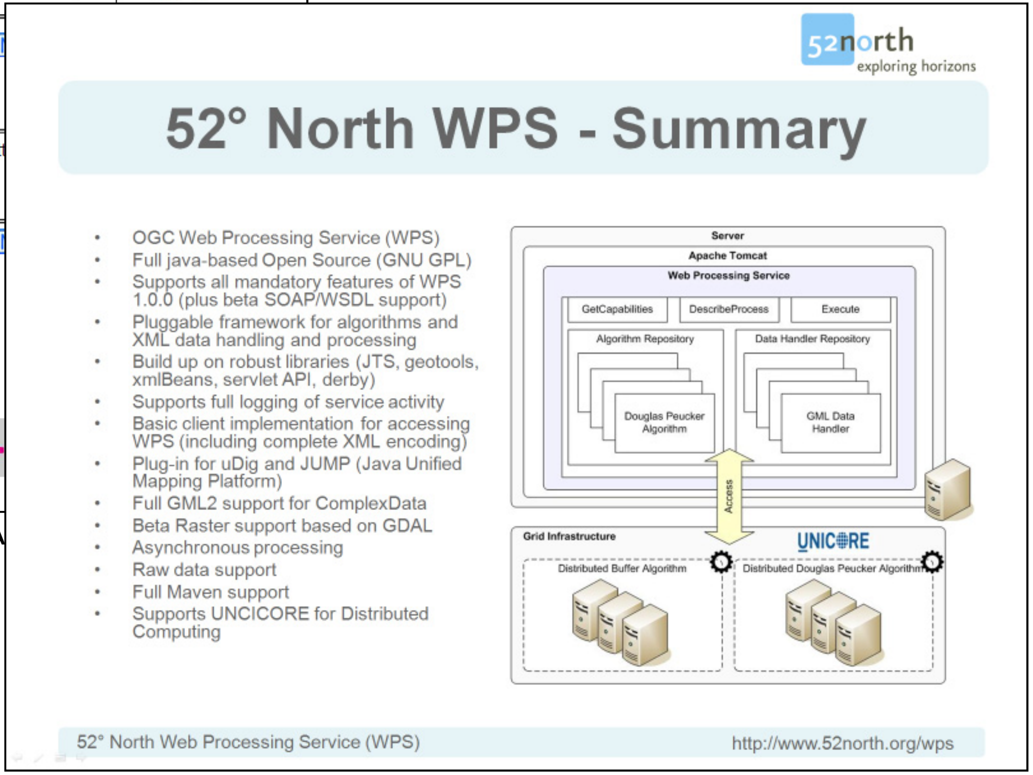
Slide courtesy of André Höing (TU Berlin)

UNICORE Commercial Environments

Embedding in Industrial and Research Environments: Access to C²A²S²E-HPC



Slide courtesy of A


52north
exploring horizons

52° North WPS - Summary

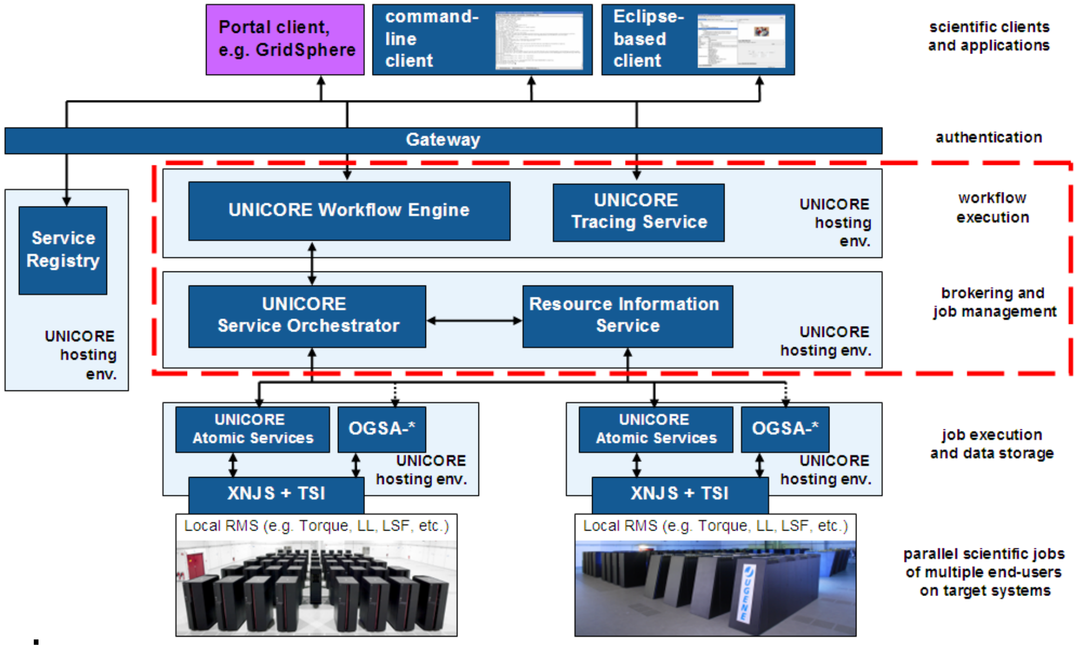
- OGC Web Processing Service (WPS)
- Full java-based Open Source (GNU GPL)
- Supports all mandatory features of WPS 1.0.0 (plus beta SOAP/WSDL support)
- Pluggable framework for algorithms and XML data handling and processing
- Build up on robust libraries (JTS, geotools, xmlBeans, servlet API, derby)
- Supports full logging of service activity
- Basic client implementation for accessing WPS (including complete XML encoding)
- Plug-in for uDig and JUMP (Java Unified Mapping Platform)
- Full GML2 support for ComplexData
- Beta Raster support based on GDAL
- Asynchronous processing
- Raw data support
- Full Maven support
- Supports UNCICORE for Distributed Computing

52° North Web Processing Service (WPS) http://www.52north.org/wps

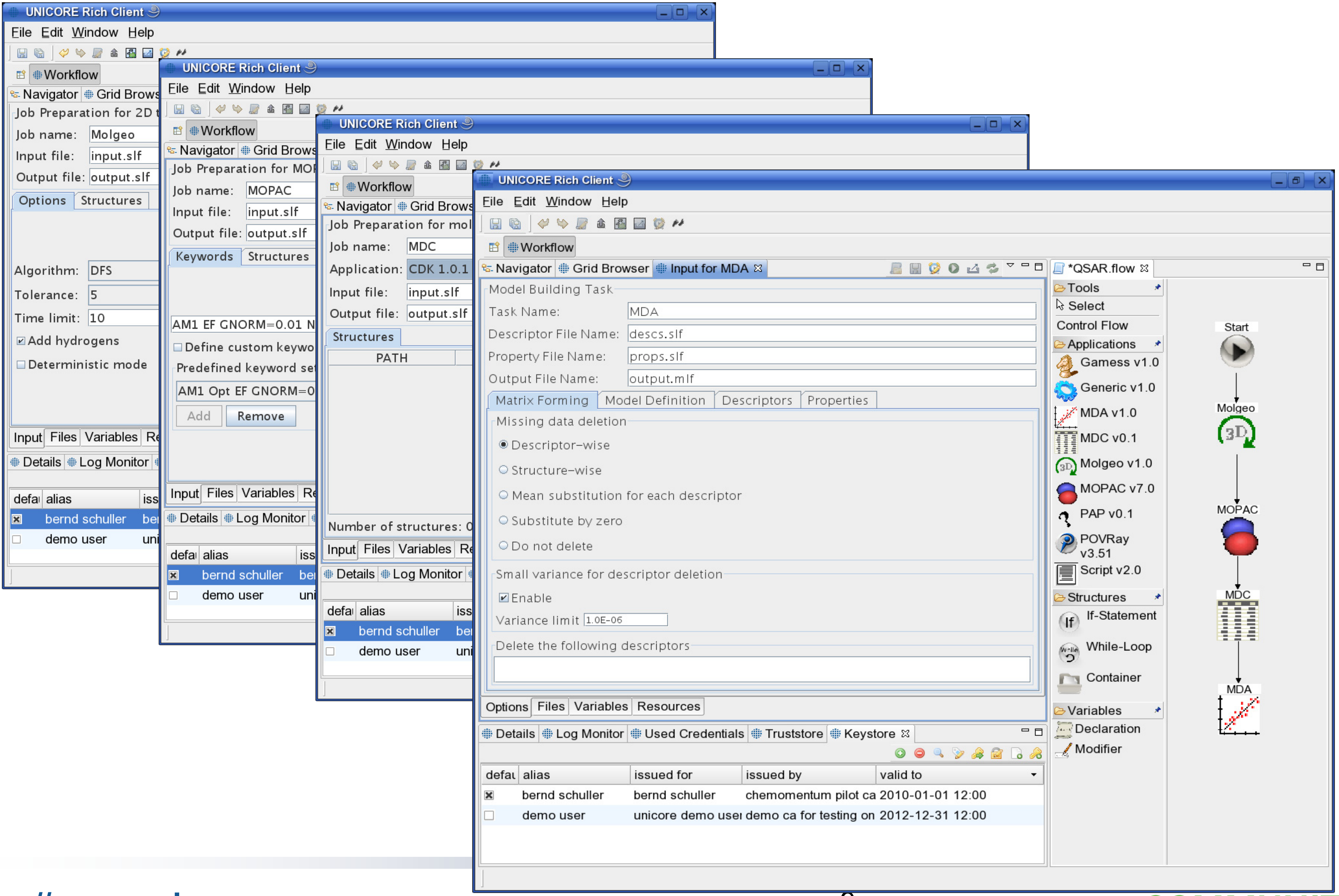
Slide courtesy of Bastian Baranski (52° North & University Münster)

Workflows in UNICORE 6

- Two layer architecture for scalability
- Workflow engine
  - Based on Shark open-source **XPDL** engine
  - Pluggable, domain-specific workflow languages
- Service orchestrator
  - Job execution and monitoring
  - Callback to workflow engine
  - Brokering based on pluggable strategies
- Clients
  - GUI client based on **Eclipse**
  - Commandline submission of workflows is also possible


Portal client, e.g. GridSphere
command-line client
Eclipse-based client
scientific clients and applications
Gateway
authentication
Service Registry
UNICORE hosting env.
UNICORE Workflow Engine
UNICORE Tracing Service
UNICORE hosting env.
workflow execution
UNICORE Service Orchestrator
Resource Information Service
UNICORE hosting env.
brokering and job management
UNICORE Atomic Services
OGSA-*
UNICORE hosting env.
XNJS + TSI
UNICORE Atomic Services
OGSA-*
UNICORE hosting env.
XNJS + TSI
job execution and data storage
Local RMS (e.g. Torque, LL, LSF, etc.)
Local RMS (e.g. Torque, LL, LSF, etc.)
parallel scientific jobs of multiple end-users on target systems

Example: Life Science Workflow



UNICORE Rich Client

File Edit Window Help

Workflow

Navigator | Grid Brows

Job Preparation for 2D t

Job name: Molgeo

Input file: input.slf

Output file: output.slf

Options | Structures

Algorithm: DFS

Tolerance: 5

Time limit: 10

☑ Add hydrogens

□ Deterministic mode

Input | Files | Variables | Re

Details | Log Monitor

defa	alias	iss
☒	bernd schuller	be
□	demo user	uni

UNICORE Rich Client

File Edit Window Help

Workflow

Navigator | Grid Brows

Job Preparation for MOI

Job name: MOPAC

Input file: input.slf

Output file: output.slf

Keywords | Structures

AM1 EF GNORM=0.01 N

□ Define custom keywo

Predefined keyword se

AM1 Opt EF GNORM=0

Add | Remove

Input | Files | Variables | Re

Details | Log Monitor

defa	alias	iss
☒	bernd schuller	be
□	demo user	uni

UNICORE Rich Client

File Edit Window Help

Workflow

Navigator | Grid Brows

Job Preparation for mol

Job name: MDC

Application: CDK 1.0.1

Input file: input.slf

Output file: output.slf

Structures

PATH

Number of structures: 0

Input | Files | Variables | Re

Details | Log Monitor

defa	alias	iss
☒	bernd schuller	be
□	demo user	uni

UNICORE Rich Client

File Edit Window Help

Workflow

Navigator | Grid Browser | Input for MDA

Model Building Task

Task Name: MDA

Descriptor File Name: descs.slf

Property File Name: props.slf

Output File Name: output.mlf

Matrix Forming | Model Definition | Descriptors | Properties

Missing data deletion

● Descriptor–wise

○ Structure–wise

○ Mean substitution for each descriptor

○ Substitute by zero

○ Do not delete

Small variance for descriptor deletion

☑ Enable

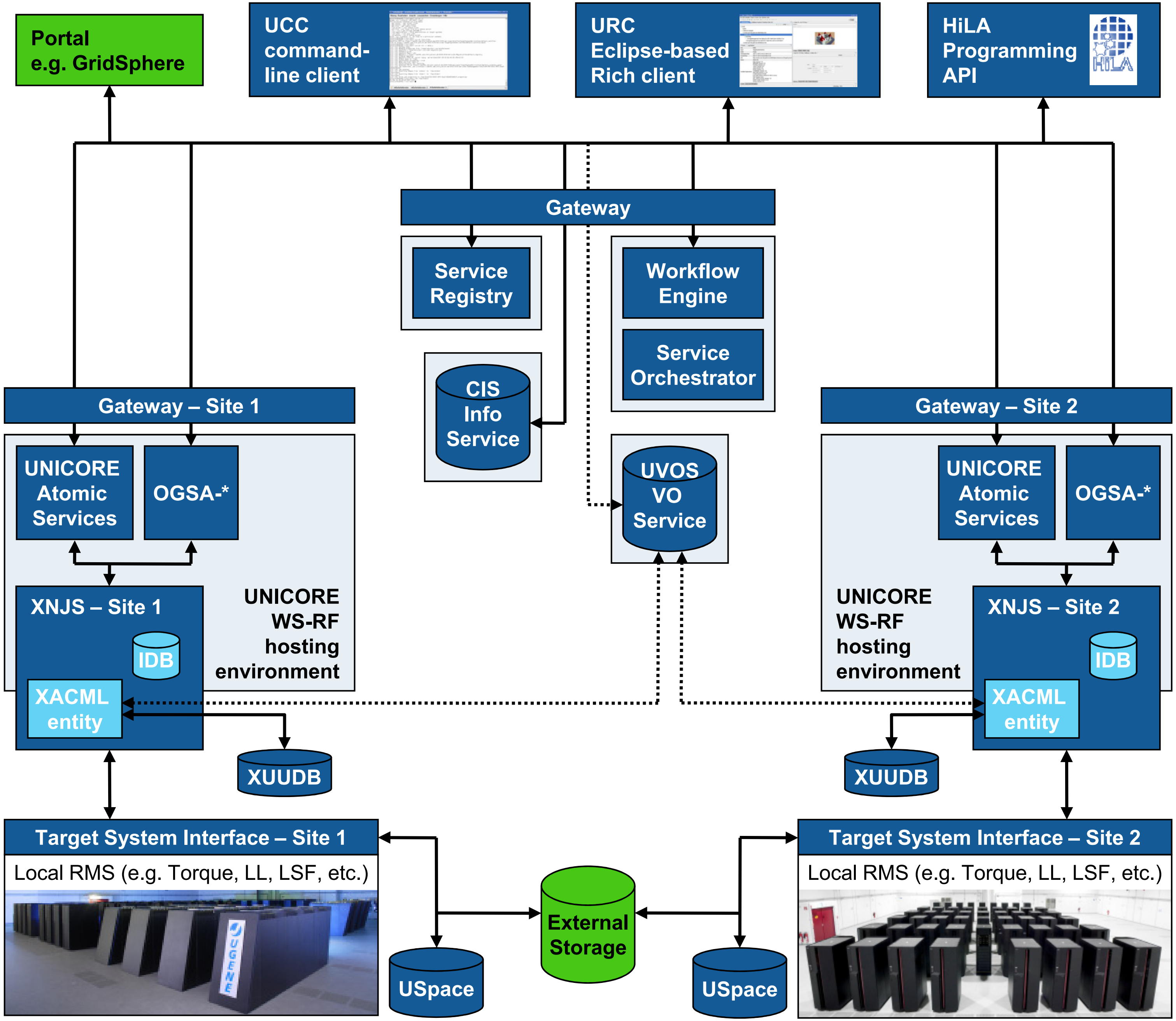
Variance limit 1.0E–06

Delete the following descriptors

Options | Files | Variables | Resources

Details | Log Monitor | Used Credentials | Truststore | Keystore

defau	alias	issued for	issued by	valid to
☒	bernd schuller	bernd schuller	chemomentum pilot ca	2010-01-01 12:00
□	demo user	unicore demo usei	demo ca for testing on	2012-12-31 12:00



Portal e.g. GridSphere

UCC command-line client

URC Eclipse-based Rich client

HiLA Programming API

Gateway

Service Registry

Workflow Engine

Service Orchestrator

CIS Info Service

UVOS VO Service

Gateway – Site 1

UNICORE Atomic Services

OGSA-*

XNJS – Site 1

IDB

XACML entity

UNICORE WS-RF hosting environment

XUUDB

Target System Interface – Site 1

Local RMS (e.g. Torque, LL, LSF, etc.)

USpace

External Storage

USpace

Gateway – Site 2

UNICORE Atomic Services

OGSA-*

UNICORE WS-RF hosting environment

XNJS – Site 2

IDB

XACML entity

XUUDB

Target System Interface – Site 2

Local RMS (e.g. Torque, LL, LSF, etc.)

scientific clients and applications

X.509, Proxies, SOAP, WS-RF, WS-I, JSDL

web service stack

central services running in WS-RF hosting environments

OGSA-RUS, UR, GLUE 2.0

authentication

OGSA-ByteIO, OGSA-BES, JSDL, HPC-P, OGSA-RUS, UR

Grid services hosting

job incarnation

X.509, XACML, SAML, Proxies

authorization

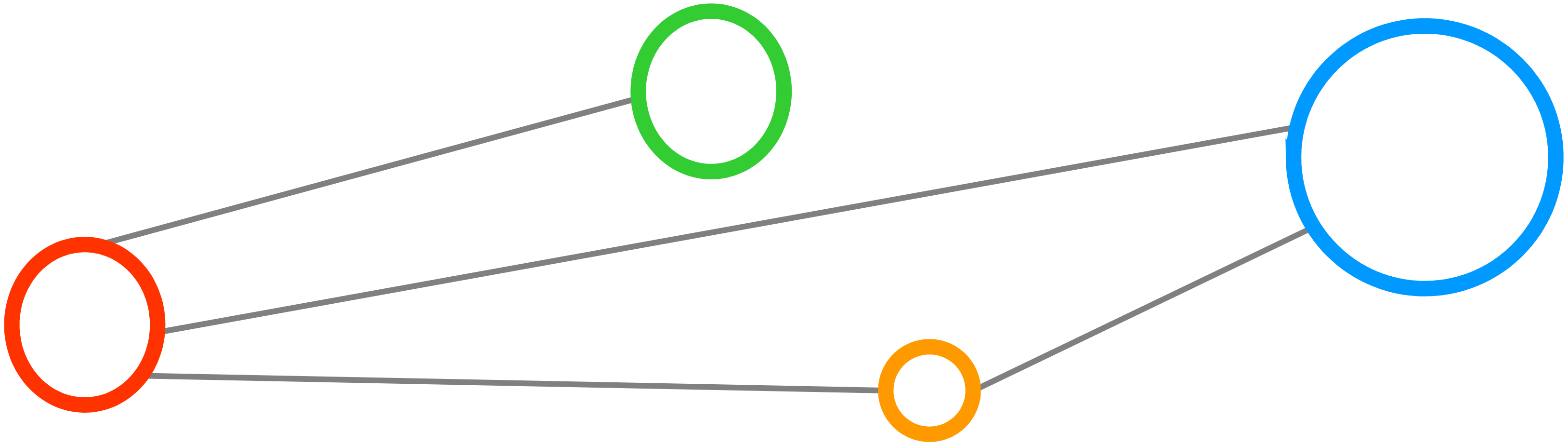
DRMAA

GridFTP, Proxies

data transfer to external storages

http://www.unicore.eu

Security Remarks & Constraints



Security Remarks & Constraints (1)

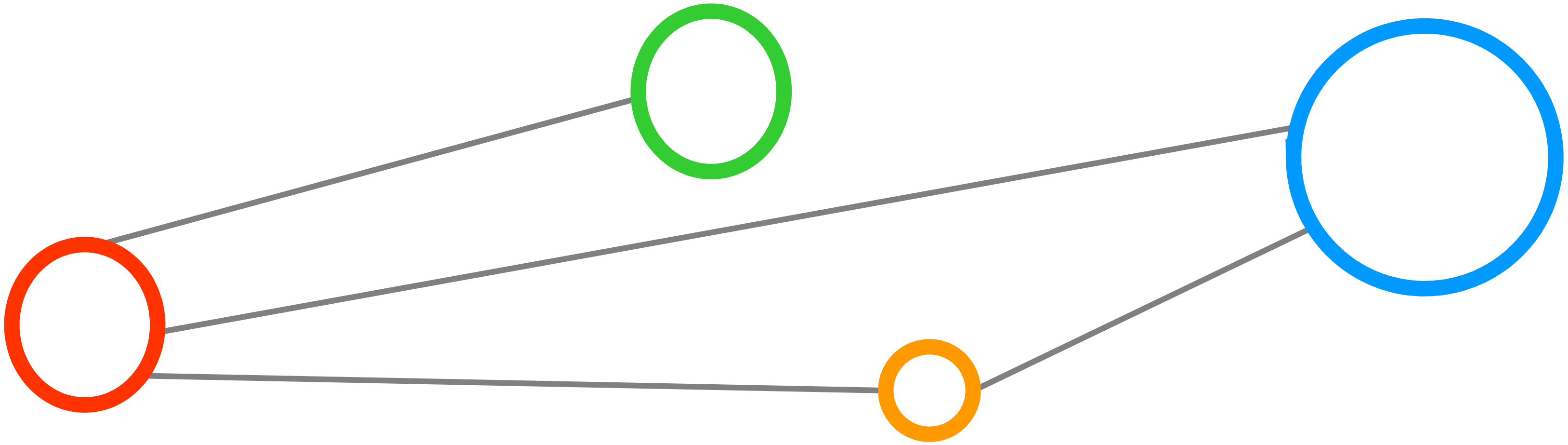
- Important foundation:
  - Major difference what security the middleware supports and what security is deployed on infrastructures
  - Infrastructures should (and have to) decide on their own what security setup is necessary (not what middleware provides)
- UNICORE mostly used in environments driven by High Performance Computing (HPC), e.g. DEISA, SKIF-Grid,
  - Driver of requirements in terms of security
- Boundary condition: no pool accounts
  - Each HPC user typically has to map to one specific UNIX account

Security Remarks & Constraints (2)

- Requirements for full X.509 end-entity certificates
  - Many HPC administrators see ‘plain proxies’ as unsecure (since proxies become part of OpenSSL this might change)
  - Proxies are not bad, but the way they are used in Grids is bad…
  - Issues: no constraints in what users do with them, no revocation mechanism possible even if they may have rather short lifetimes, overheads in creating proxies per hop, no commercial tooling and support available (often either forms of the certificate’s DNs or path validation causes problems), etc.

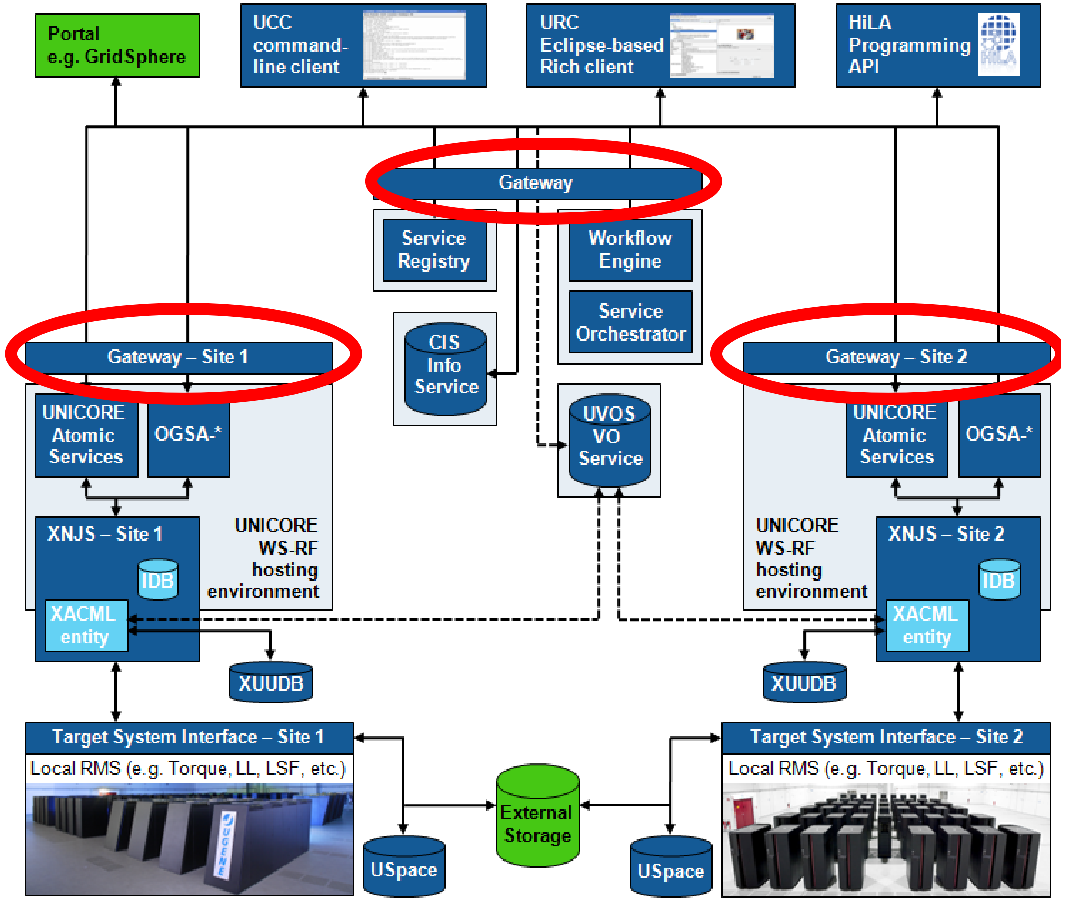
- UNICORE can be also used in non HPC-driven environments
  - Different optional setups can be used to satisfy administrators and decision makers in various Grid infrastructures
- Interoperability becomes more and more a driver of new features…, e.g. using proxy-certificates, bridging to Grids,…

Foundational Security Elements



Foundational Security Elements (1)

- Authentication via the UNICORE Gateway Component
  - Accepts Transport Level Security (TLS) connections established with full X.509 end-entity-certificates
  - Checks whether certificates are still valid (lifetime), not revoked, and signed by a Certificate Authority (CA) that is being trusted (i.e. TrustStore)
  - Uses Certificate Revocation Lists (CRL) for revocation



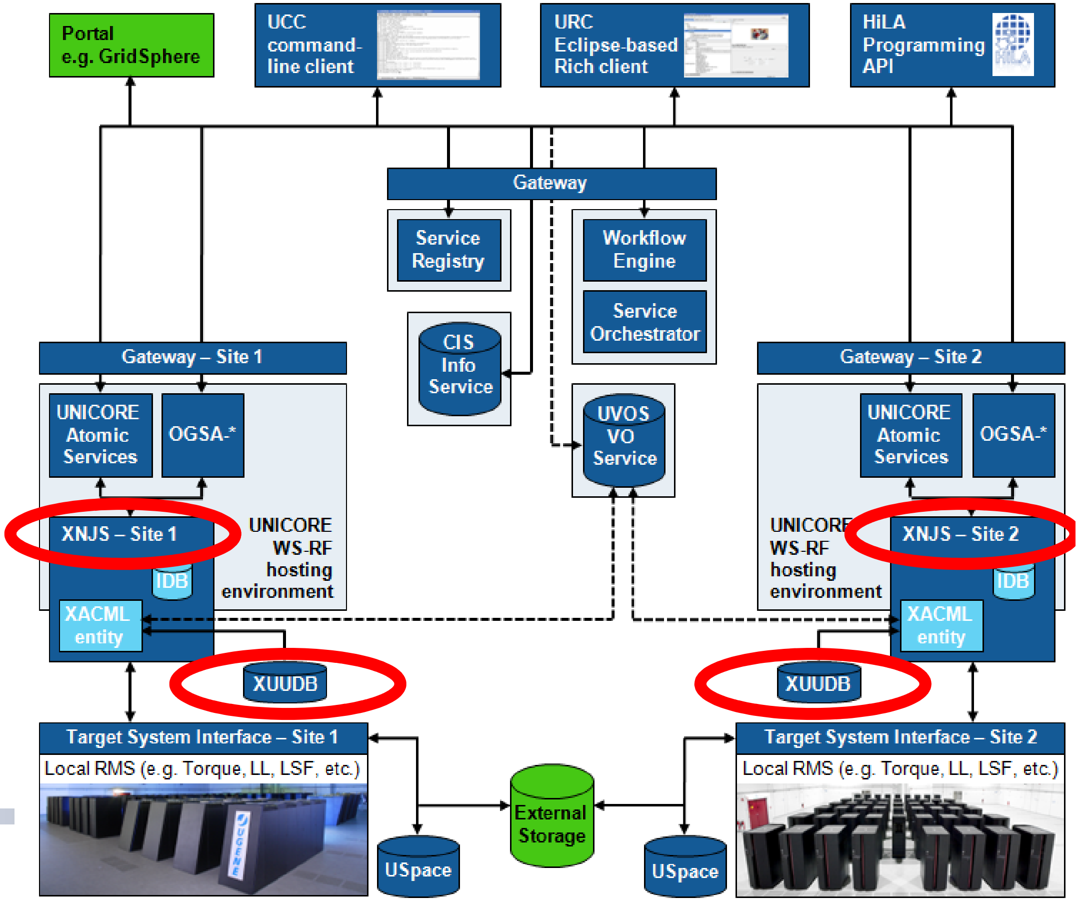
http://www.unicore.eu

Foundational Security Elements (2)

- SAML Assertions
  - User assertion: created and signed by the end user
  - Consignor assertion: created at the gateway, contains information about the certificate used for SSL
  - Trust delegation assertion
    - Initial TD is created and signed by the user. Delegates trust to a particular identity (i.e. a single DN!)
    - The trusted server can extend the TD to another identity
    - Builds a single, verifiable chain of trust
      (vs. proxy chains that only track DNs that are may not distinguishable, e.g. …/CN=Morris → machine-name/hop)
    - Maximum chain length and TD lifetime configurable
- Server just needs to trust the Gateway and the authorization servers (e.g. XUUDB, UNICORE VO - Service, ….)

Foundational Security Elements (3)

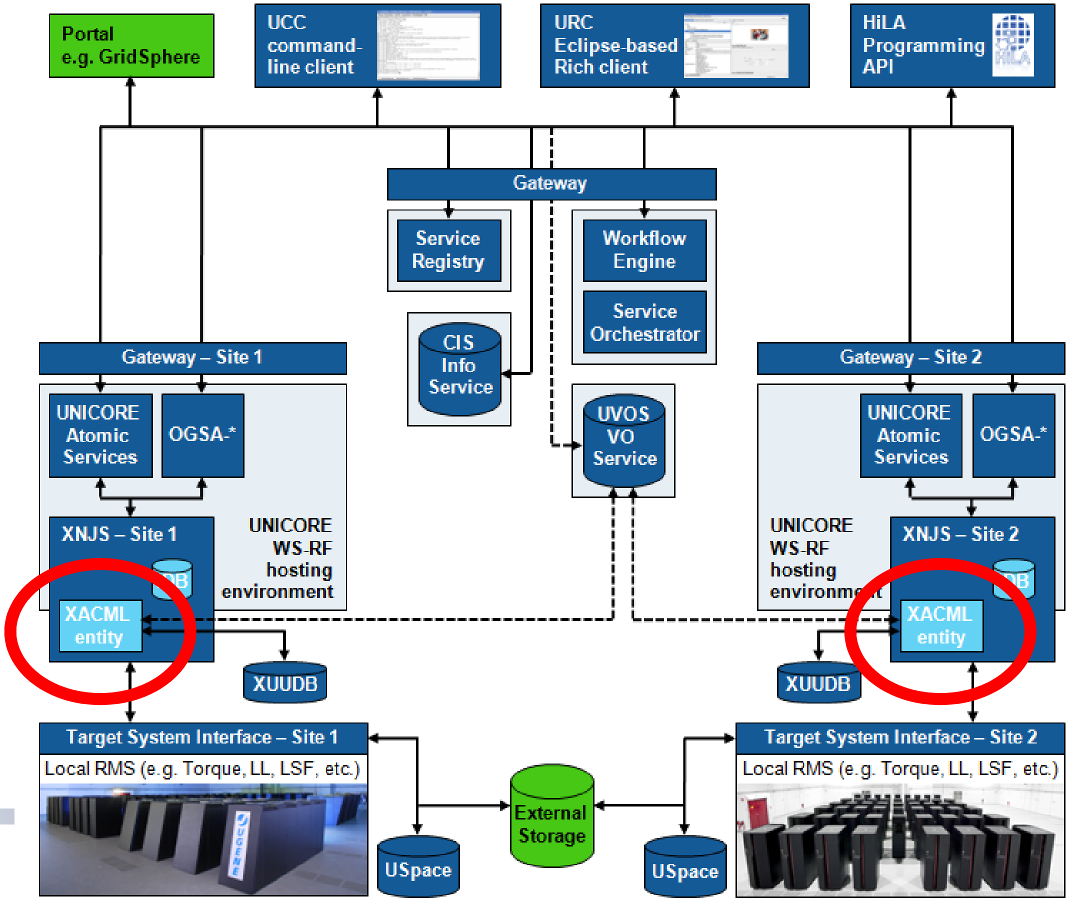
- Numerous Authorization Capabilities through handler design
  - Gateway forwards requests to UNICORE server
  - UNICORE server processes a handler chain before each service invocation happen
  - Different authorization handlers exist
    - SAMLInHandler, SAMLVOMSInHandler, …
  - Each Handler puts security content (i.e. attributes, roles) in ‘Security context’
  - XUUDB callout to map X509 to local account and role (e.g. “user”)



http://www.unicore.eu

Foundational Security Elements (4)

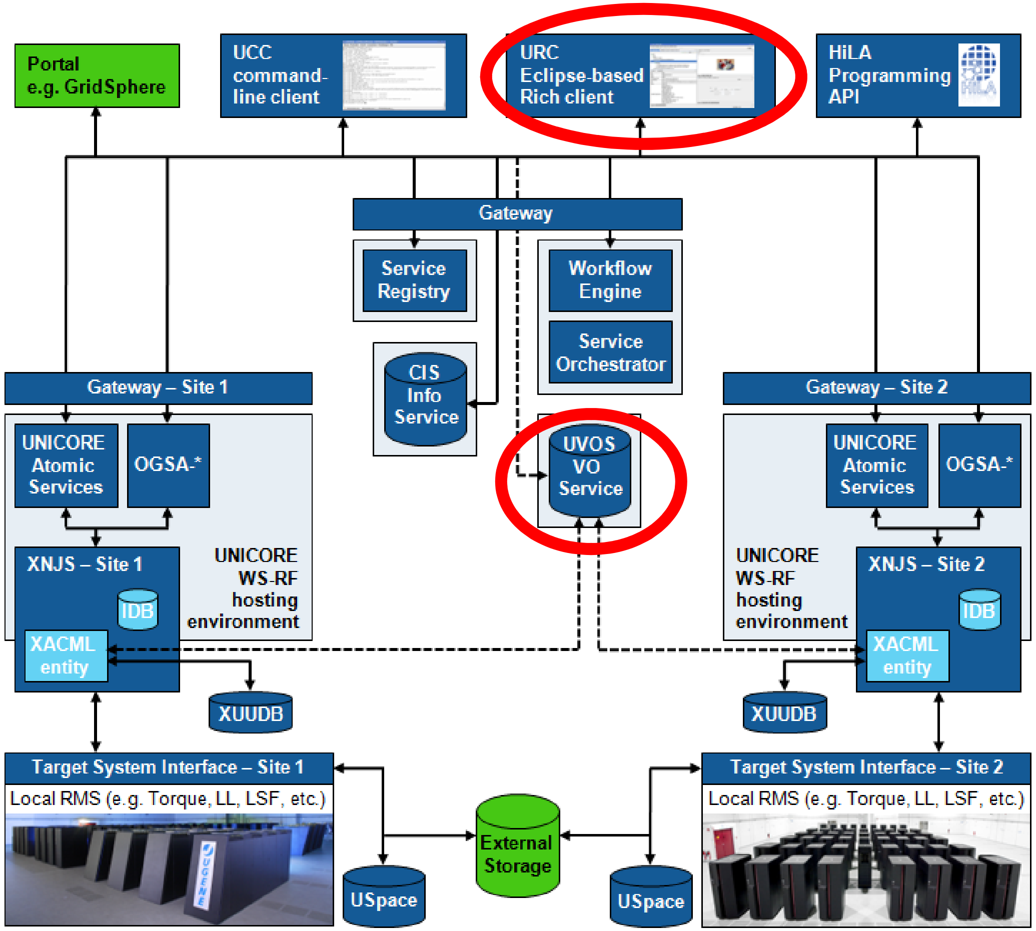
- High flexibility in policy definition and enforcement
  - Each UNICORE server provides an XACML-based policy (XACML = eXtensible Access Control Markup Language)
  - Before each invocation, the security context (filled with information from handlers) is checked against the XACML policy
  - Simple yes/no decision if end-users get access to systems



http://www.unicore.eu

Foundational Security Elements (5)

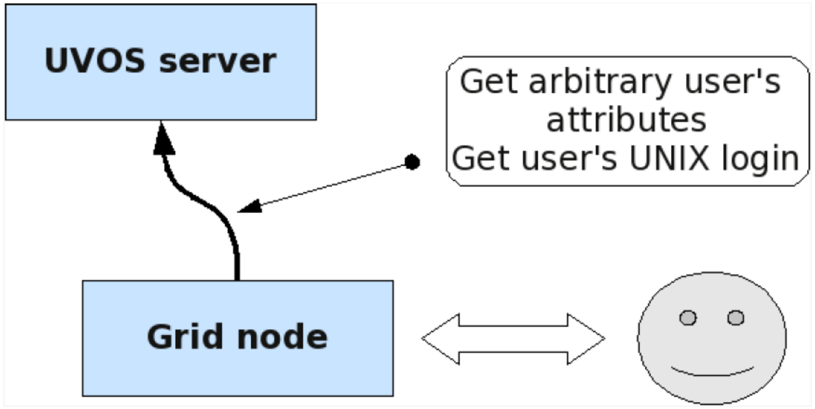
- Support for attributes of Virtual Organizations (VOs)
  - SAML-based UNICORE VO Service (UVOS) developed (in parallel to OMII-Europe SAML-based VOMS service)
  - SAML Requests to service according to SAML standard
  - Membership can be conveniently configured from VO administrators via a UNICORE Rich Client plug-in
  - Xlogin local accounts retrieval according to user identity



http://www.unicore.eu

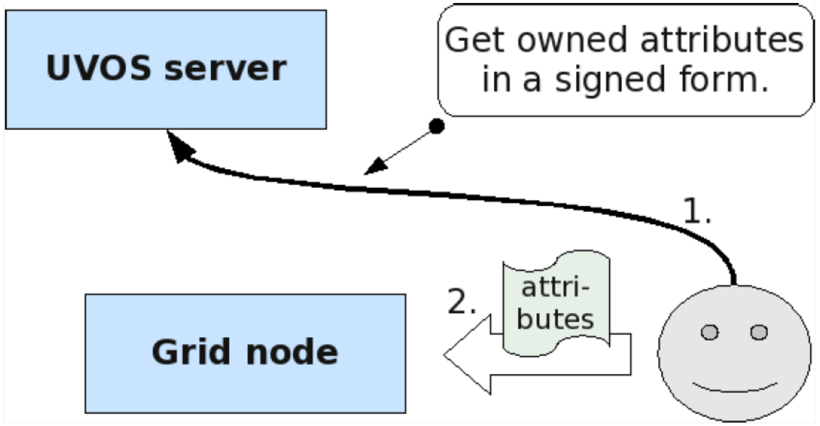
Foundational Security Elements (6)

- UVOS & "Pull Model"
  - A service contacts the VO server to obtain the attributes of a user who tries to use it.
  - The attributes received from the VO server can be used for an authorization later inside UNICORE
- Evaluation from usage
  - Pull mode is transparent for the grid users
  - More difficult for grid administrators to set it up since grid site must be correctly configured to use the corresponding UVOS



Foundational Security Elements (7)

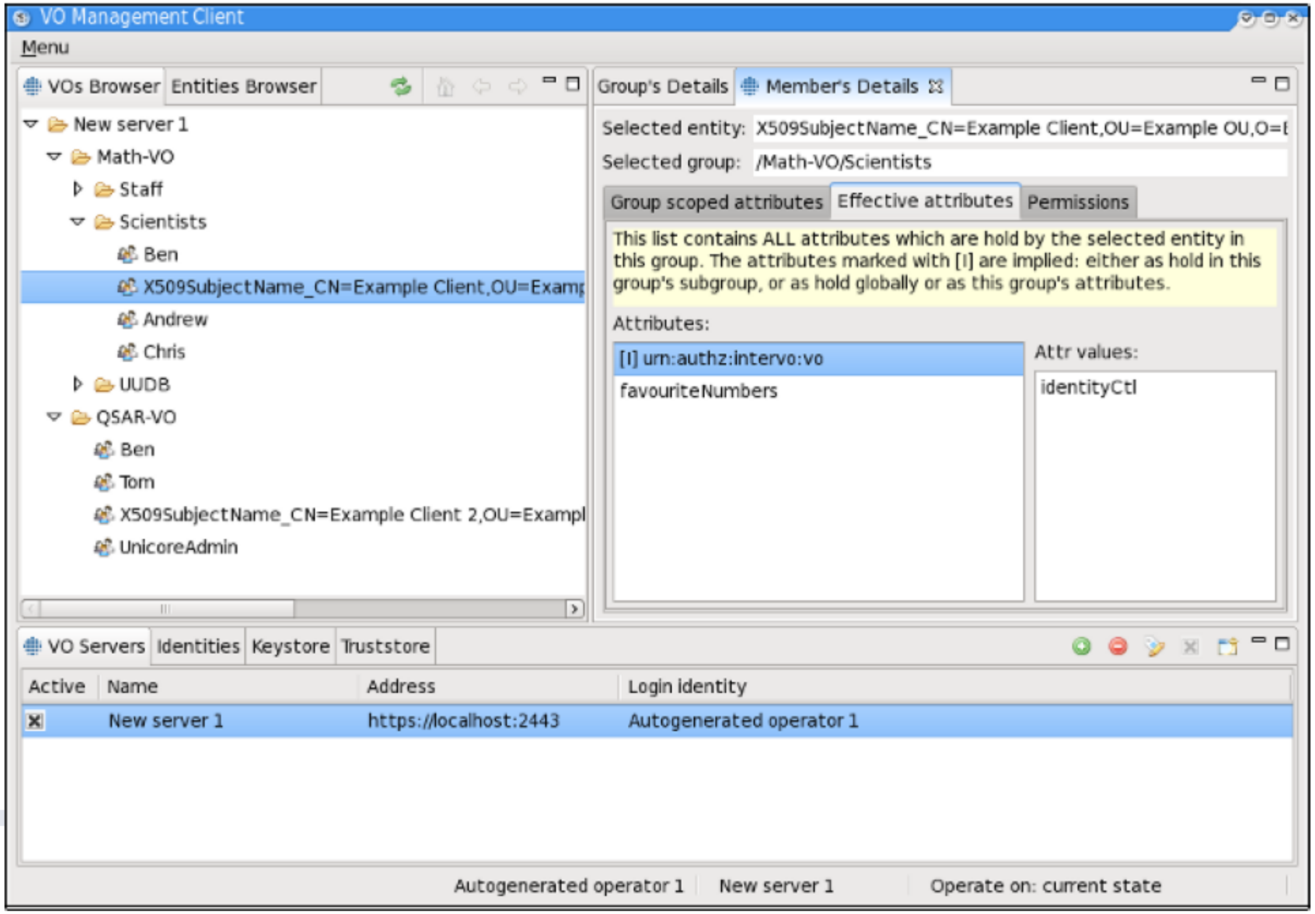
- UVOS & “Push Model”
  - A user has to contact a VO server (via client) and get the list of possessed attributes in a signed assertion
  - This assertion can be attached to the requests which are sent to the grid services (i.e. Grid node)
- Evaluation from usage
  - More scalable in terms of server administration and easier to set up
  - It requires user interaction and a problem with expired assertions arises (cp. With proxy renewal problem)



Foundational Security Elements (8)

- UVOS Interaction Protocols & standards
  - The management clients use a custom WS interface
  - The consumers use the open standard SAML 2.0 as a protocol

- Administrative GUI within the Eclipse-based UNICORE Rich Client



Selected entity: X509SubjectName_CN=Example Client,OU=Example OU,O=E

Selected group: /Math-VO/Scientists

This list contains ALL attributes which are hold by the selected entity in this group. The attributes marked with [I] are implied: either as hold in this group's subgroup, or as hold globally or as this group's attributes.

Attributes:

- [I] urn:authz:intervo:vo
- favouriteNumbers

Attr values:

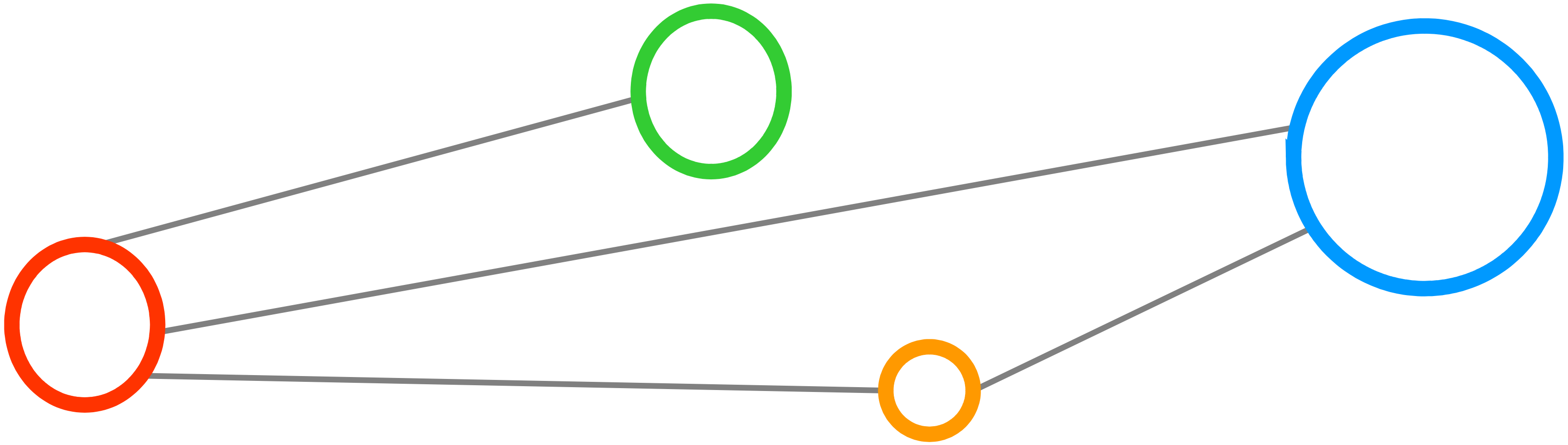
- identityCtl

Active	Name	Address	Login identity
☒	New server 1	https://localhost:2443	Autogenerated operator 1

Autogenerated operator 1 | New server 1 | Operate on: current state

http://www.unicore.eu

Security Setups



Security Setups: Default Deployments

- Authentication
  - Gateway: authentication of end-users based on certificates obtained from TLS connection
- Authorization
  - Services typically use the XUUDB
  - XUUDB maps the user certificate to UNIX accounts
    - Limited set of attributes: e.g. role = user
    - In “DN mode” only the DN of certificates is relevant
- Delegation
  - SAML-based Trust Delegation using SAML Assertions and full end-entity certificates only

Security Setup: Grid Interoperability

- Authentication
  - Gateway: authentication of end-users based on certificates obtained from TLS connection
  - Optional proxy-validation chain algorithm (often used by portals) (allows for OpenSSL-based TLS with proxies but no GSI connections)
- Attribute-based Authorization
  - DN-XUUDB maps DN of certificates to UNIX accounts
  - Attributes obtained from SAML assertions provided by SAML-based UVOS (or SAML-based VOMS)
- Delegation
  - Also SAML-based TD so far
  - (Maybe adopting proxy-based delegation as optional setup)

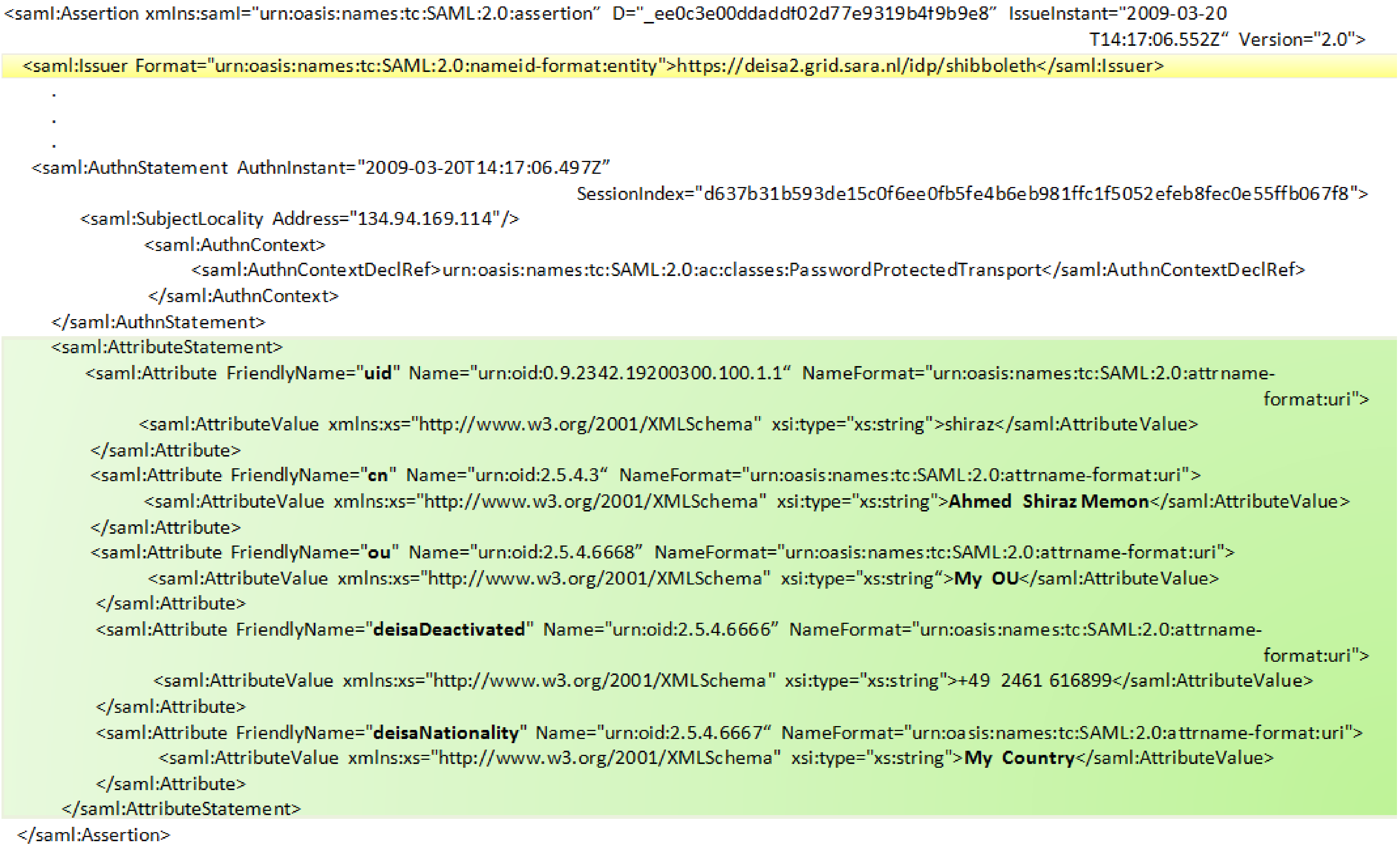
Security Setup: Shibboleth Federations (1)

- Demonstrated at last DEISA review: UNICORE and Shibboleth
- Shibboleth View:
- GridShib-CA is used as a ServiceProvider from the Shibboleth perspective whereas Online CA from user's point of view.
  - This allows client to fetch his/her credentials in X.509 from Shibboleth
  - Move towards new Shibboleth (avoiding GridShib) in progress
- User authenticates with his institutions IDP (Shibboleth based Identity Provider)
  - Get access to the GridShib-CA (the SP and Online CA)
  - Enable users to download "manually" the SLC (Short Lived Credentials) with embedded SAML assertions via Web browser

Security Setup: Shibboleth Federations (2)

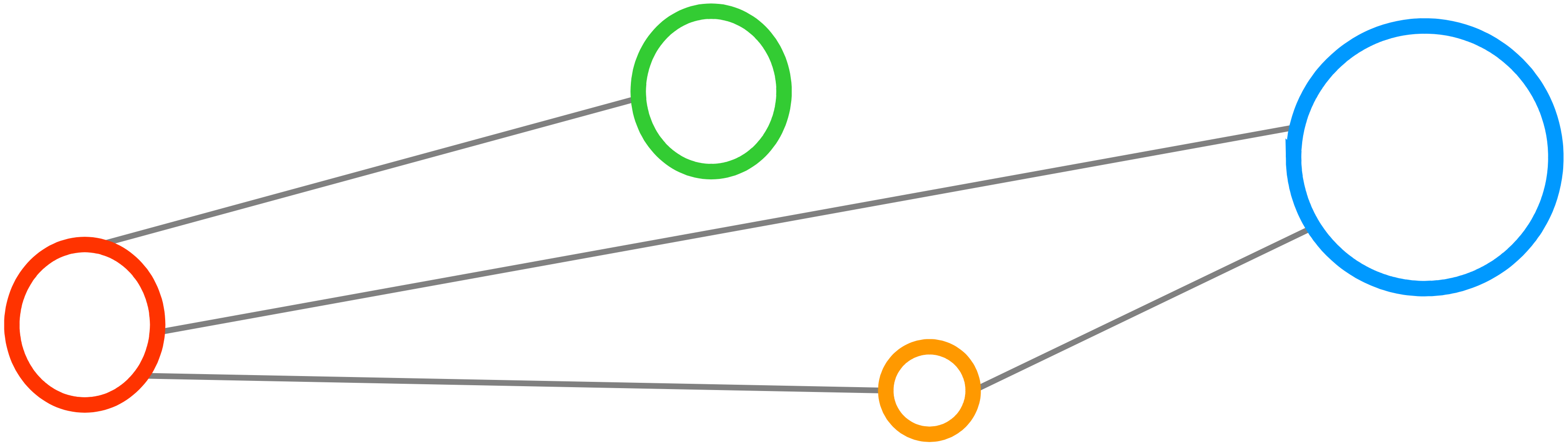
- UNICORE View - Client Side:
  - User imports the downloaded SLC into his/her keystore.
  - By executing "slc" command using ucc to extend the existing keystore otherwise creating the new one
  - User connects to the target system while authenticating himself at the SSL/TLS level
    - Along with that the SAML assertions are being extracted and appended into the (service) requesting SOAP header
- UNICORE View - Server side:
  - The incoming SAML assertion is being parsed at the server side
  - SAML assertion containing user attributes are being matched against XUUDB and XACML policy store to make final decision is being made, contemplates whether the client is allowed to execute that request on particular service or not

Security Setup: Shibboleth Federations (3)



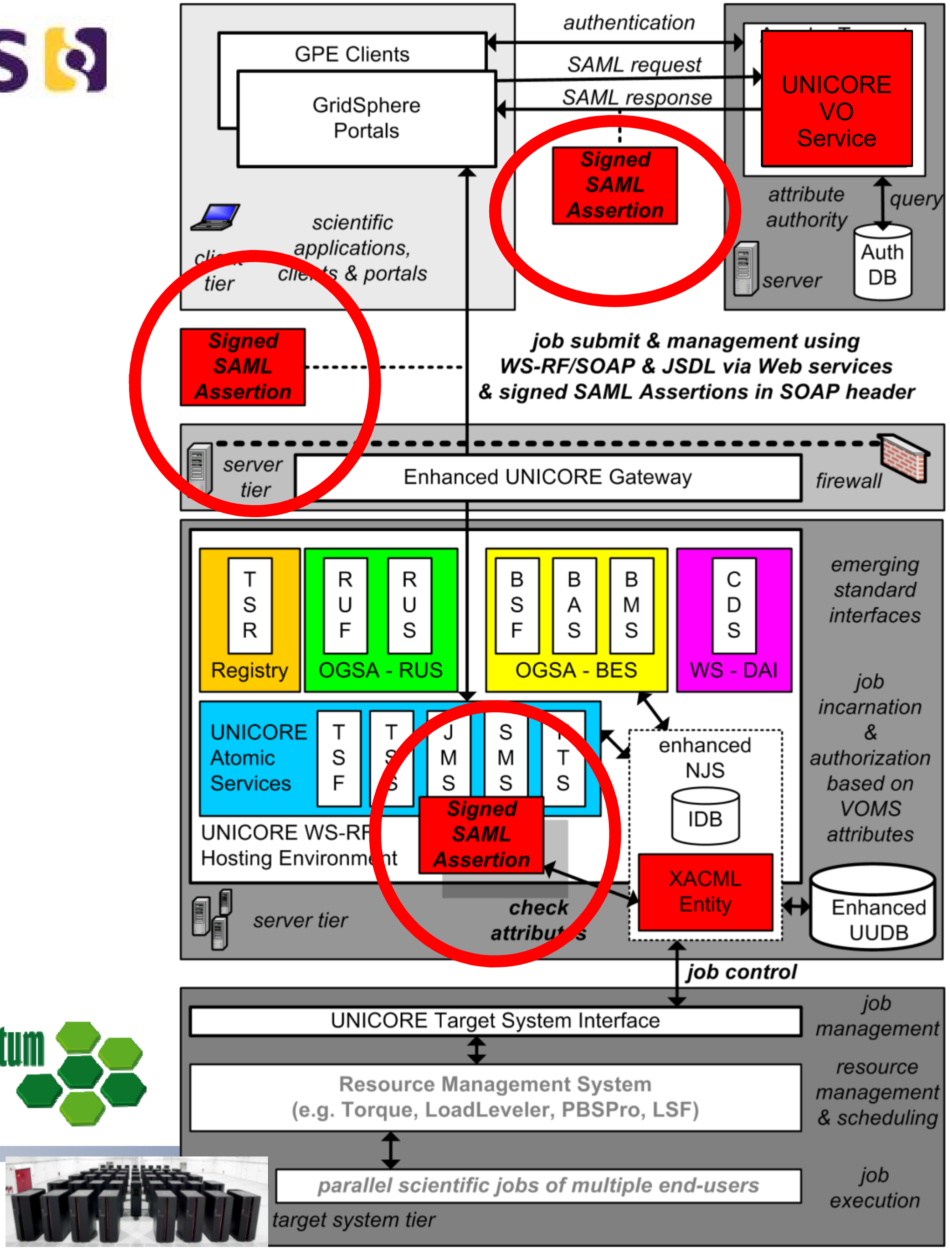
```
<saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"  D="_ee0c3e00ddaddf02d77e9319b4f9b9e8"  IssueInstant="2009-03-20
                                                                                                        T14:17:06.552Z"  Version="2.0">
  <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">https://deisa2.grid.sara.nl/idp/shibboleth</saml:Issuer>
      .
      .
      .
   <saml:AuthnStatement AuthnInstant="2009-03-20T14:17:06.497Z"
                                                            SessionIndex="d637b31b593de15c0f6ee0fb5fe4b6eb981ffc1f5052efeb8fec0e55ffb067f8">
         <saml:SubjectLocality Address="134.94.169.114"/>
                <saml:AuthnContext>
                     <saml:AuthnContextDeclRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</saml:AuthnContextDeclRef>
                </saml:AuthnContext>
     </saml:AuthnStatement>
     <saml:AttributeStatement>
         <saml:Attribute FriendlyName="uid" Name="urn:oid:0.9.2342.19200300.100.1.1"  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
                                                                                                                     format:uri">
               <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"  xsi:type="xs:string">shiraz</saml:AttributeValue>
         </saml:Attribute>
         <saml:Attribute FriendlyName="cn"  Name="urn:oid:2.5.4.3"  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
               <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"  xsi:type="xs:string">Ahmed  Shiraz Memon</saml:AttributeValue>
         </saml:Attribute>
         <saml:Attribute FriendlyName="ou"  Name="urn:oid:2.5.4.6668"  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
               <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"  xsi:type="xs:string">My  OU</saml:AttributeValue>
          </saml:Attribute>
          <saml:Attribute FriendlyName="deisaDeactivated"  Name="urn:oid:2.5.4.6666"  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
                                                                                                                     format:uri">
                <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"  xsi:type="xs:string">+49  2461 616899</saml:AttributeValue>
          </saml:Attribute>
          <saml:Attribute FriendlyName="deisaNationality"  Name="urn:oid:2.5.4.6667"  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
                <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"  xsi:type="xs:string">My  Country</saml:AttributeValue>
          </saml:Attribute>
      </saml:AttributeStatement>
  </saml:Assertion>
```

One Example



Example: Using SAML-based VOMS

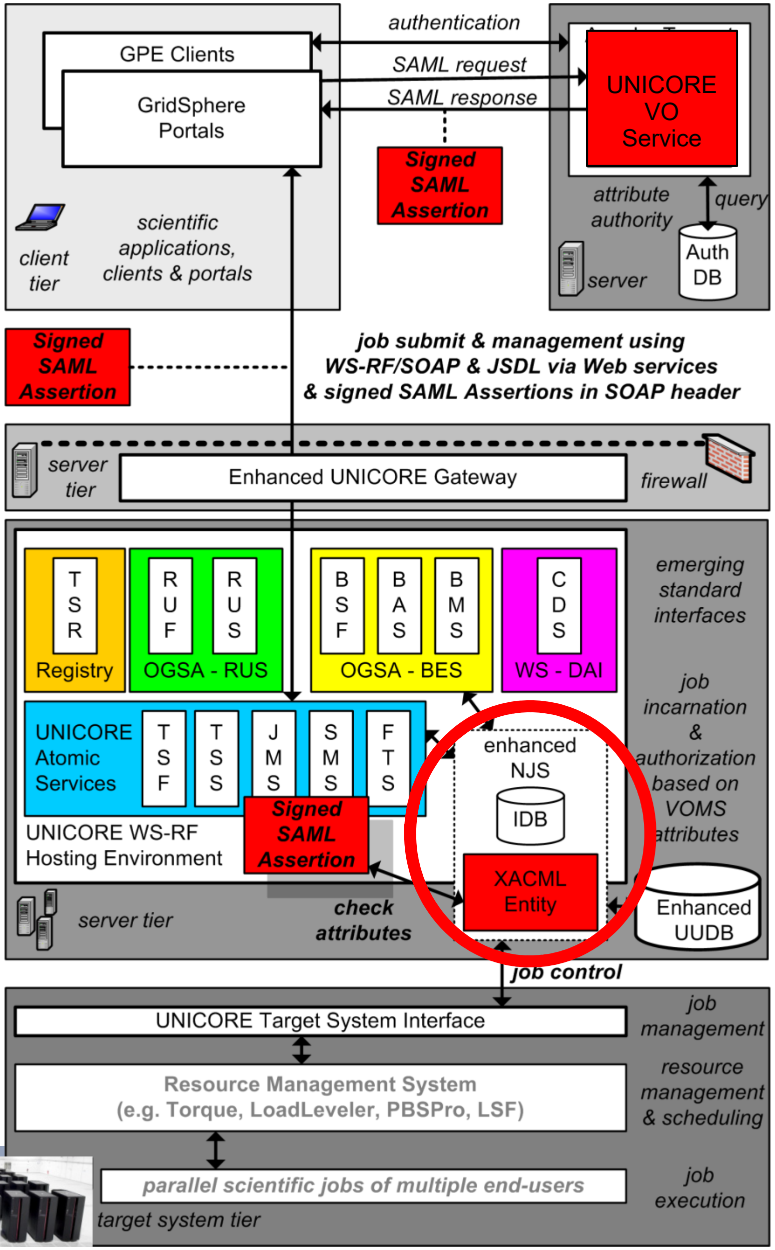
- SAML is an open standard from OASIS
- Used in conjunction with AuthZ Attribute Exchange Profile (OGF AuthZ group) to obtain a SAML assertion from a Policy Information Point (PIP)
- SAML assertions are transported in the SOAP header during WS calls
  - E.g. WS-Security Extensions
- Developed by



http://www.unicore.eu

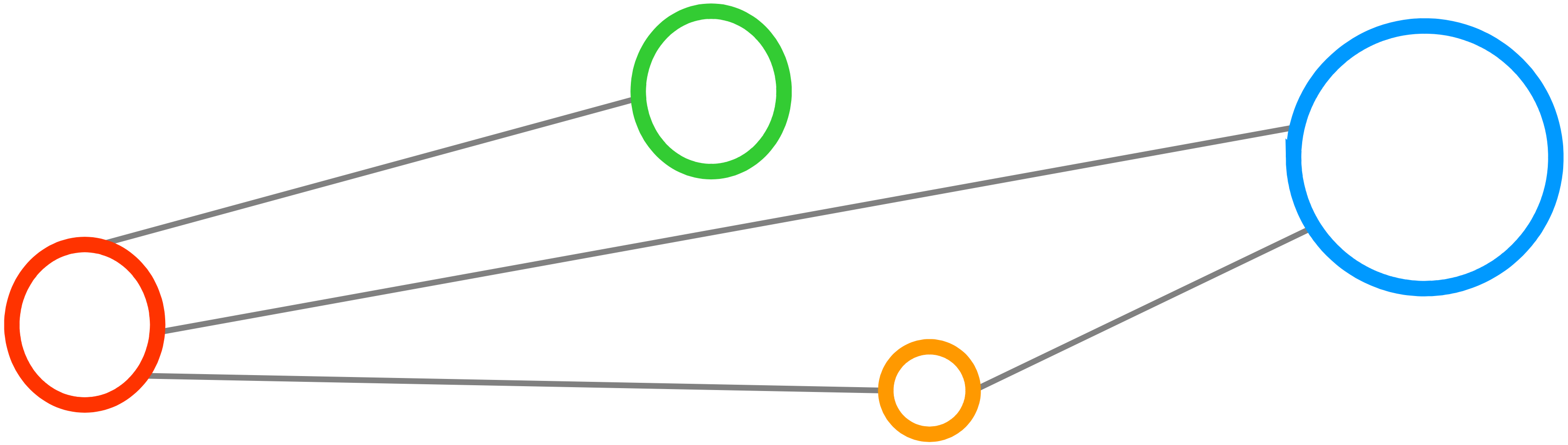
Example: Policy enforcements and decisions

- XACML is an open standard from OASIS
- Allows to define policies that grant/deny access to UNICORE
- Before the XNJS really executes something an XACML callout is performed
- Basically does yes/no decisions
- Heavily used in UNICORE in conjunction with SAML assertions defining roles
- Developed by



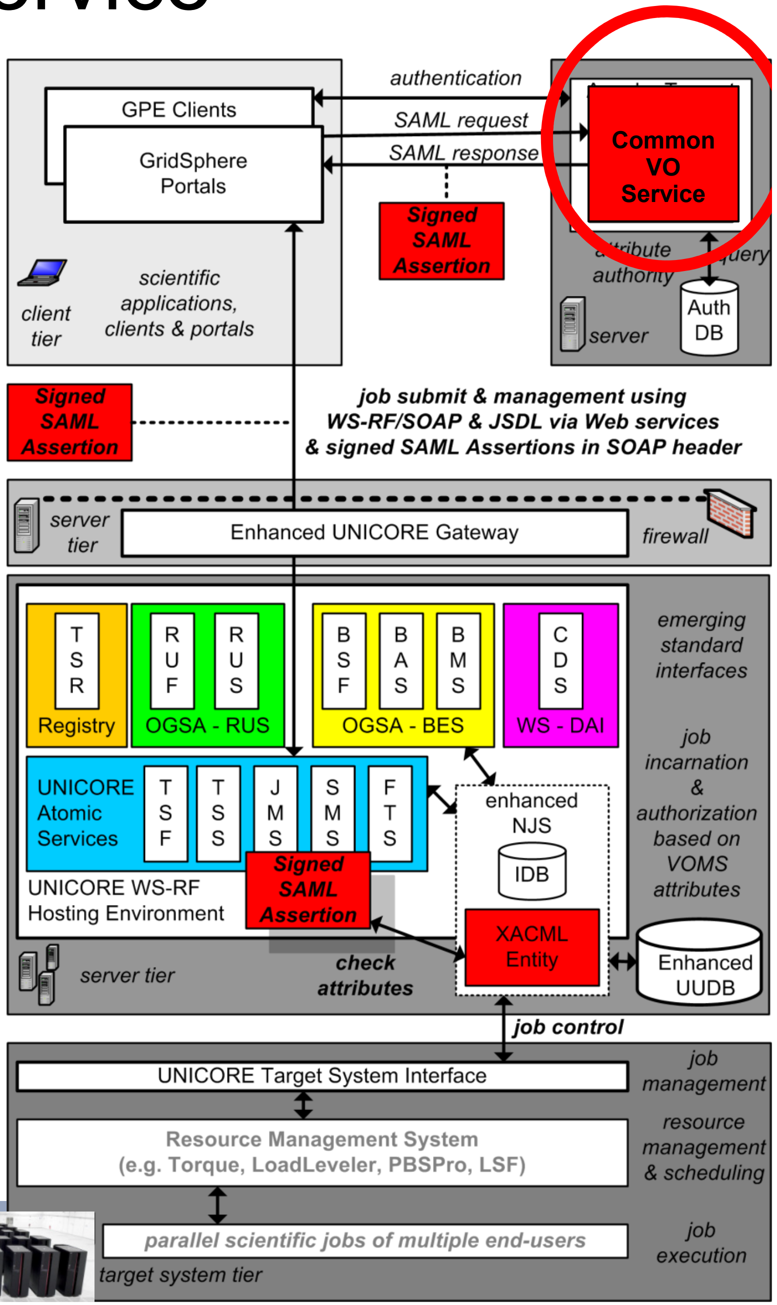
http://www.unicore.eu

Harmonization Options
In context of European Middleware Initiative (EMI)



Common VO Service

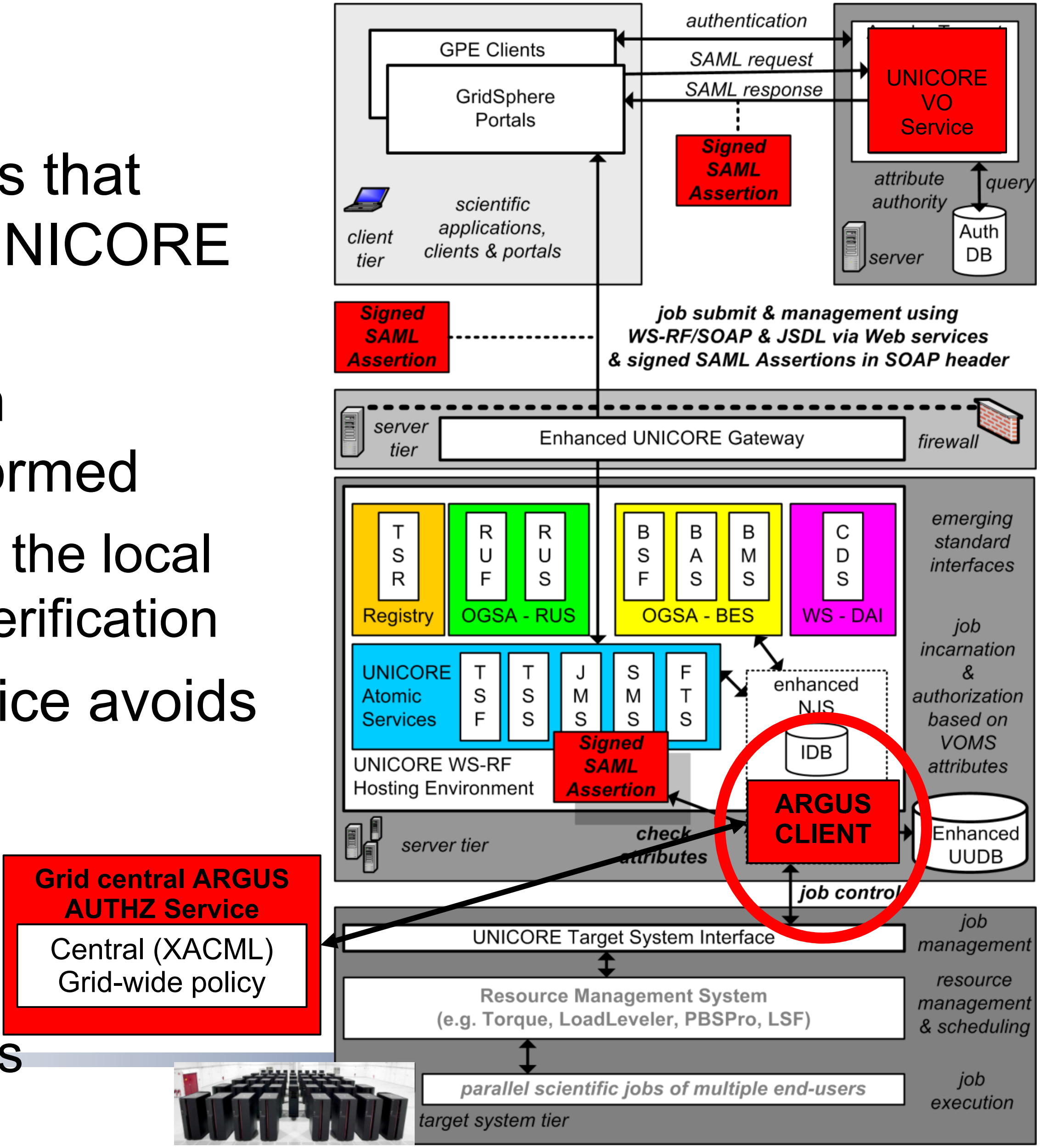
- Two VO services exist with similar scope and functionality
  - SAML-based VOMS as extension of classic VOMS
  - UNICORE VO Service
- Common VO Service
  - Joint development!
- Common VO Attributes
  - E.g. Map VO attributes with DEISA Extreme Computing Initiative (DECI) projects & DEISA Virtual communities



http://www.unicore.eu

Unified EMI Authorization Service

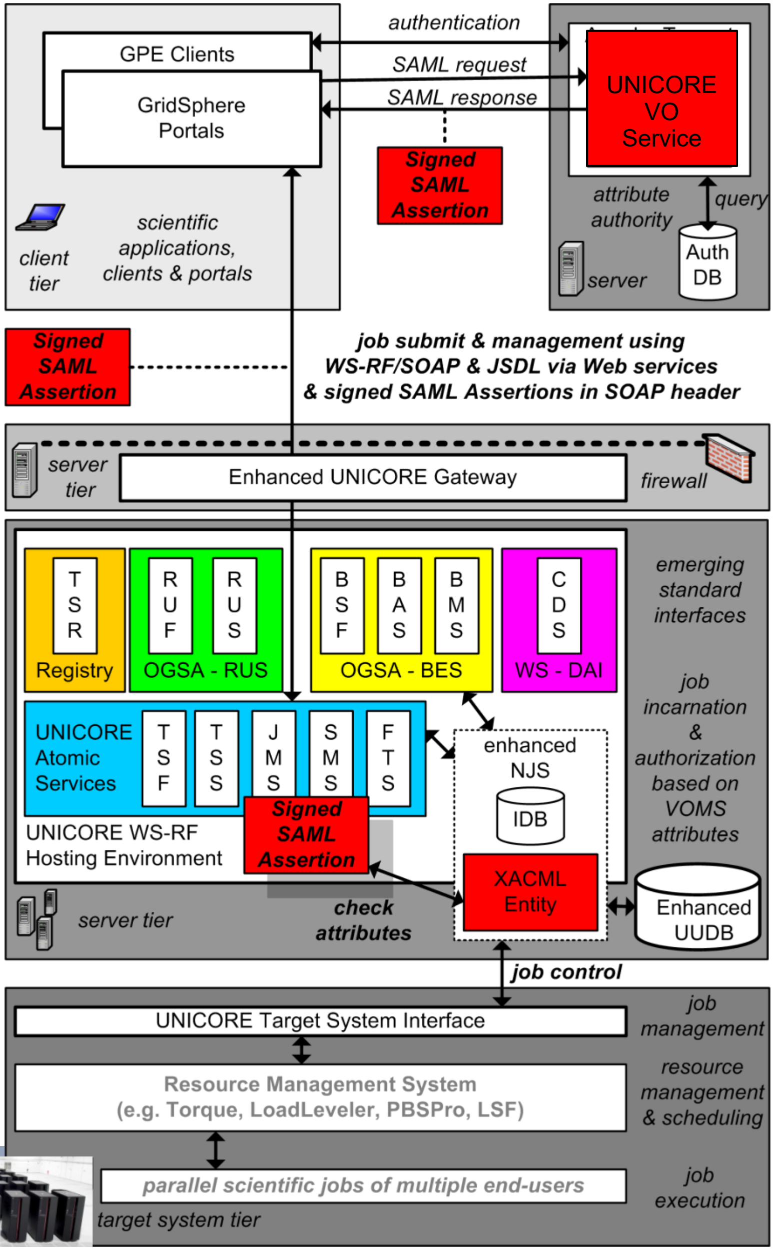
- Allows to define policies that grant/deny access to UNICORE
- Before the XNJS really executes something an ARGUS callout is performed
  - Maybe additionally to the local XACML policy user verification
- Centralized AuthZ service avoids copy&paste of local XACML policy files
  - XACML language is strong enough to express the policies


GPE Clients
GridSphere Portals
client tier
scientific applications, clients & portals
authentication
SAML request
SAML response
Signed SAML Assertion
UNICORE VO Service
attribute authority
query
Auth DB
server
Signed SAML Assertion
job submit & management using WS-RF/SOAP & JSDL via Web services & signed SAML Assertions in SOAP header
server tier
Enhanced UNICORE Gateway
firewall
T S R
Registry
R U F
R U S
OGSA - RUS
B S F
B A S
B M S
OGSA - BES
C D S
WS - DAI
emerging standard interfaces
UNICORE Atomic Services
T S F
T S S
J M S
S M S
F T S
enhanced NJS
IDB
Signed SAML Assertion
UNICORE WS-RF Hosting Environment
job incarnation & authorization based on VOMS attributes
ARGUS CLIENT
Enhanced UUDB
server tier
check attributes
job control
Grid central ARGUS AUTHZ Service
Central (XACML) Grid-wide policy
UNICORE Target System Interface
job management
Resource Management System (e.g. Torque, LoadLeveler, PBSPro, LSF)
resource management & scheduling
parallel scientific jobs of multiple end-users
job execution
target system tier

http://www.unicore.eu

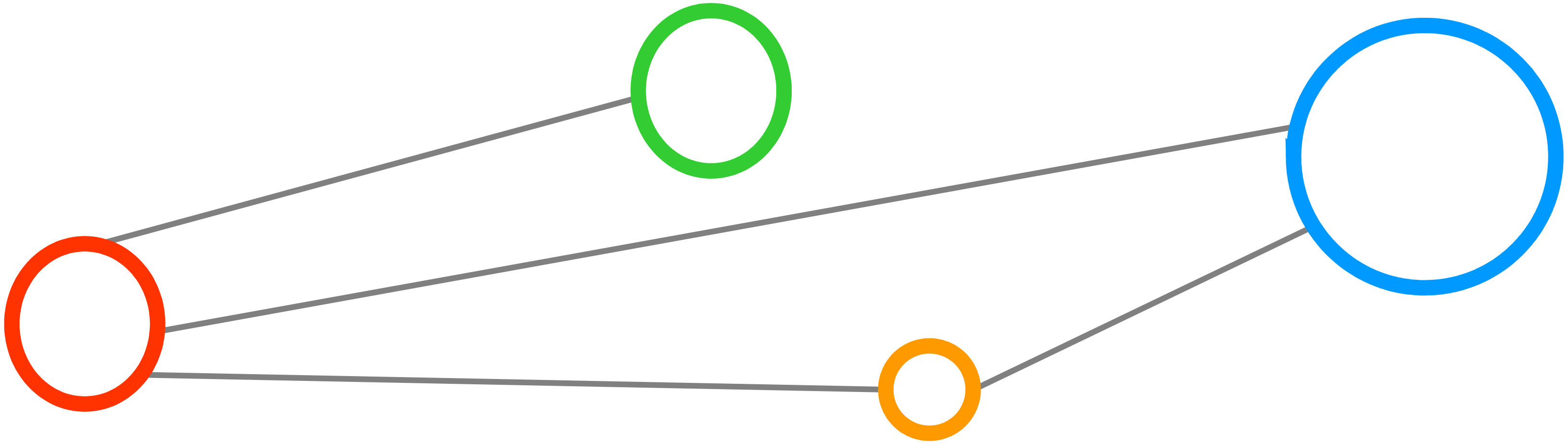
Other Harmonization Options

- Check full interoperability with OpenSSL-based proxies
  - Explore usage of proxies for workflow hops that go beyond UNICORE (i.e. gLite, ARC)
- Joint work on solutions to support Short Lived Credentials (SLCs)
- Move away from the Grid Security Infrastructure (GSI) dependency in some components
  E.g. GridFTP support
- Work towards multi-Grid policies for a production deployment



http://www.unicore.eu

Future Topics



Future Topics

- Kerberos Support
  - Members of UNICORE Community (mostly CEA, France) develop support for Kerberos environments → Work in Progress
- Adopting open standards for interoperability
  - Efforts of Grid Interoperation Now (GIN) and Production Grid Infrastructure (PGI) OGF groups (and others…)
- Using attributes for different xlogins
  - Different xlogins for different Grids (using attributes, e.g. roles)
- Access to Storage systems
  - iRODS and Storage Resource Manager (SRM) adoptions (maybe requires GSI connections on short-term scale)
- UNICORE Rich Client and GLUE capability checks of systems using the UNICORE Common Information Service (CIS)