

UNICORE

Version 6 – Architecture

Bernd Schuller, FZ Jülich

UNICORE Migration Workshop
29 October 2008, Langen

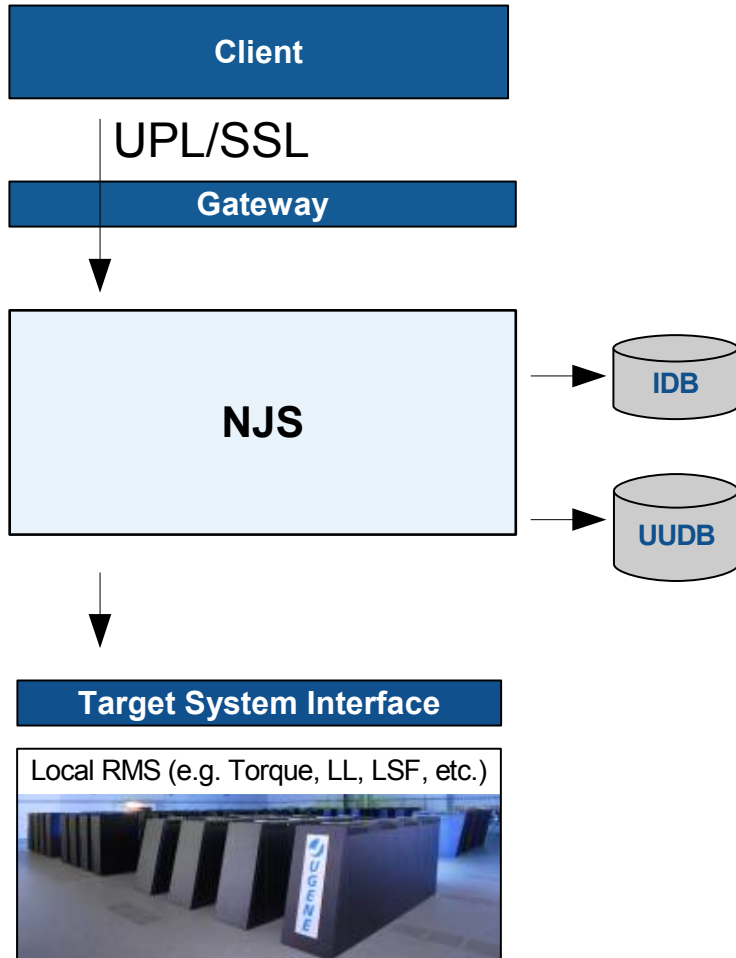
Outline

- From UNICORE 5 to UNICORE 6
- Architecture of UNICORE 6
- Security
- Status and future developments

UNICORE key ideas and „design principles“

- Integrated, complete Grid middleware stack
- Easy to install, configure, administrate and monitor
- Excellent application support
- Workflow support
- Graphical clients
- Support common operating and resource management systems
 - OS: UNIXes, MacOS X, Windows, ...
 - RMS: no-batch, Torque, LoadLeveller, LSF, SGE, ...
- Respect autonomy of resource providers

What is/was wrong with UNICORE 5 ?



- „UNICORE protocol layer“ (UPL)
 - Serialised Java objects over SSL
- Abstract job object (AJO)
 - Java class hierarchy
 - But: unified model
- Central server component (NJS)
 - Monolithic application
 - Few plugin points
 - Hardcoded behaviour (e.g. security)
 - Proprietary solutions for common tasks (e.g. persistence)
 - Scalability limits

UNICORE 6: Standardised inter-component communication

- Communication using Web Services
 - SOAP over HTTPS
- Off-the-shelf, well-tested components
 - Jetty web server, XFire web services toolkit
- Platform and language independence
 - Java, .NET

```
<soap:Envelope>
  <soap:Header>[...]</soap:Header>
  <soap:Body>
    <tss:Submit>
      <jSDL:JobDefinition>...
    </jSDL:JobDefinition>
    </tss:Submit>
  </soap:Body>
</soap:Envelope>
```

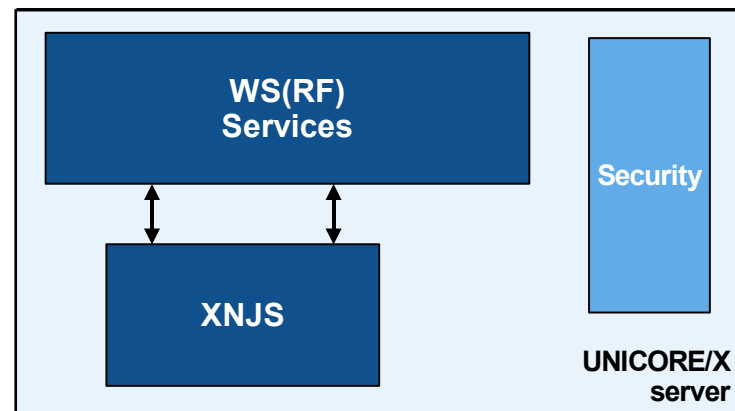
UNICORE 6: Standardised models

- Use XML based, standards-based models where available
- WSRF (Web service resource framework)
 - Stateful extension for Web Services
 - XML document associated with each resource (job, storage, ...)
 - OASIS standard
- JSDL (Job submission description language)
 - Describes single abstract computational job including data stage in/out
 - OGF standard

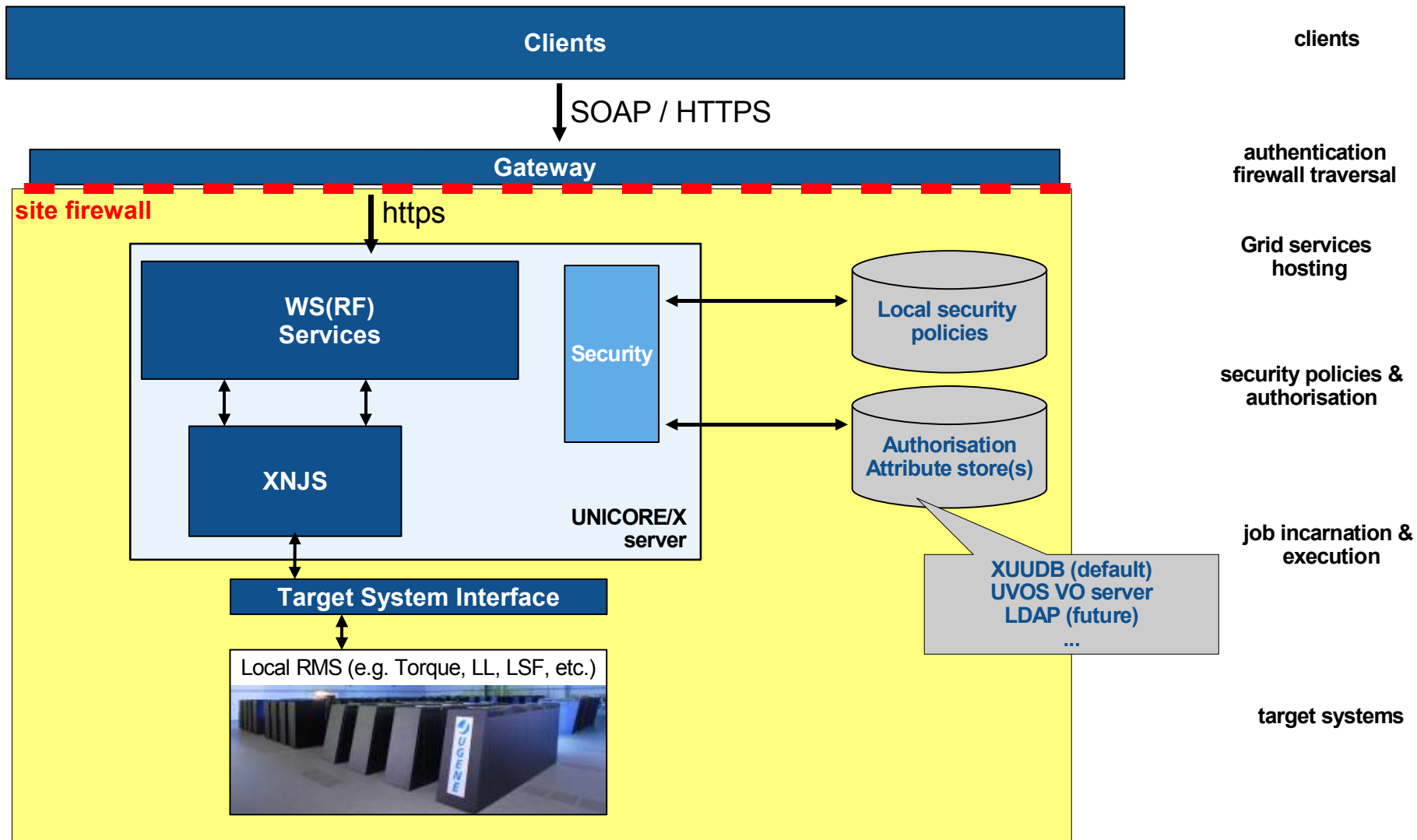


UNICORE 6: extensible core

- Replace monolithic NJS by
 - Web services / WSRF hosting environment WSRF lite
 - New execution manager (XNJS)
 - Flexible security architecture

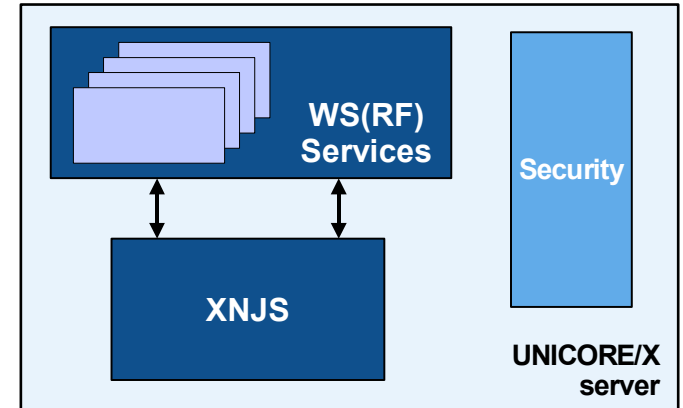


UNICORE 6: single Grid site

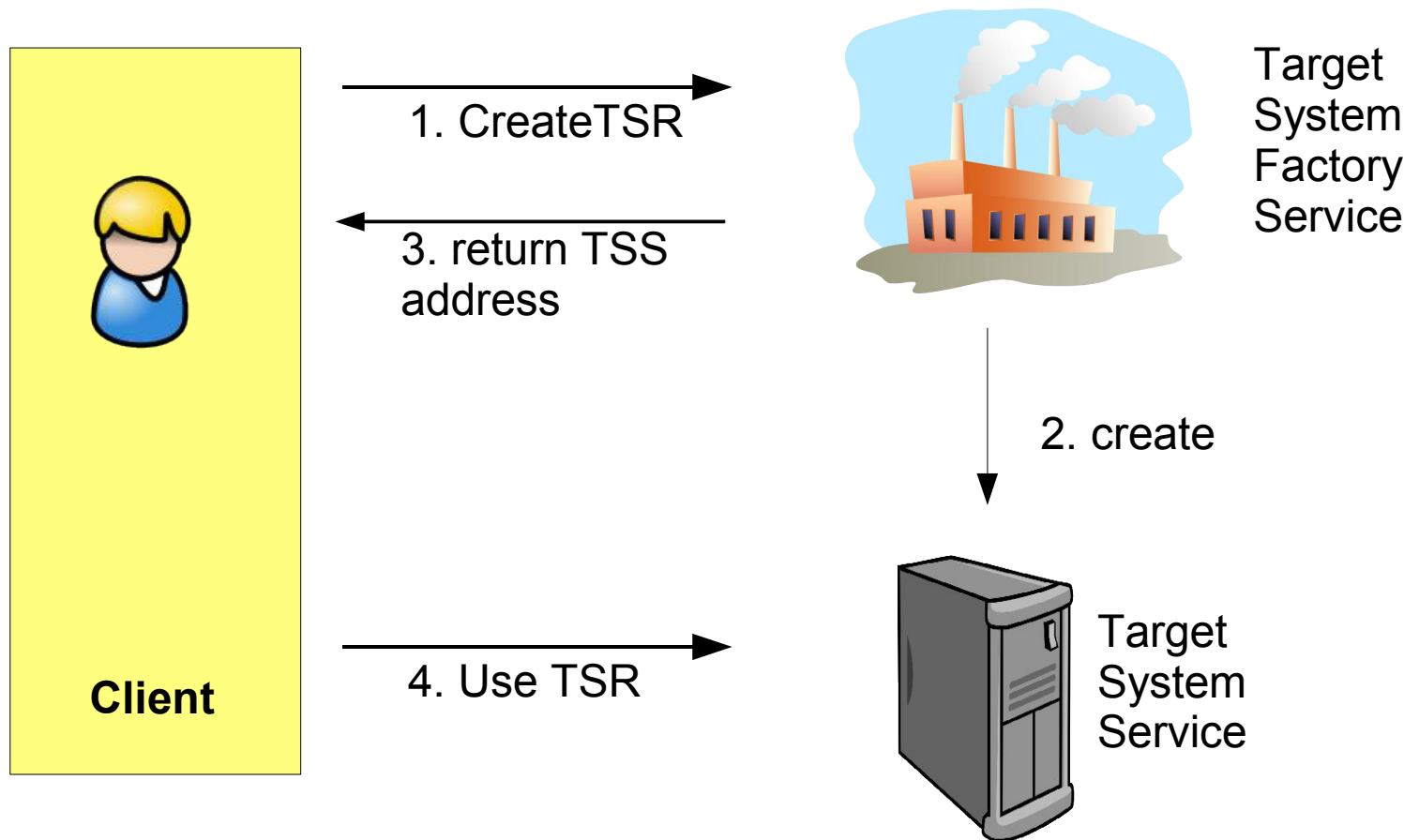


Base Services

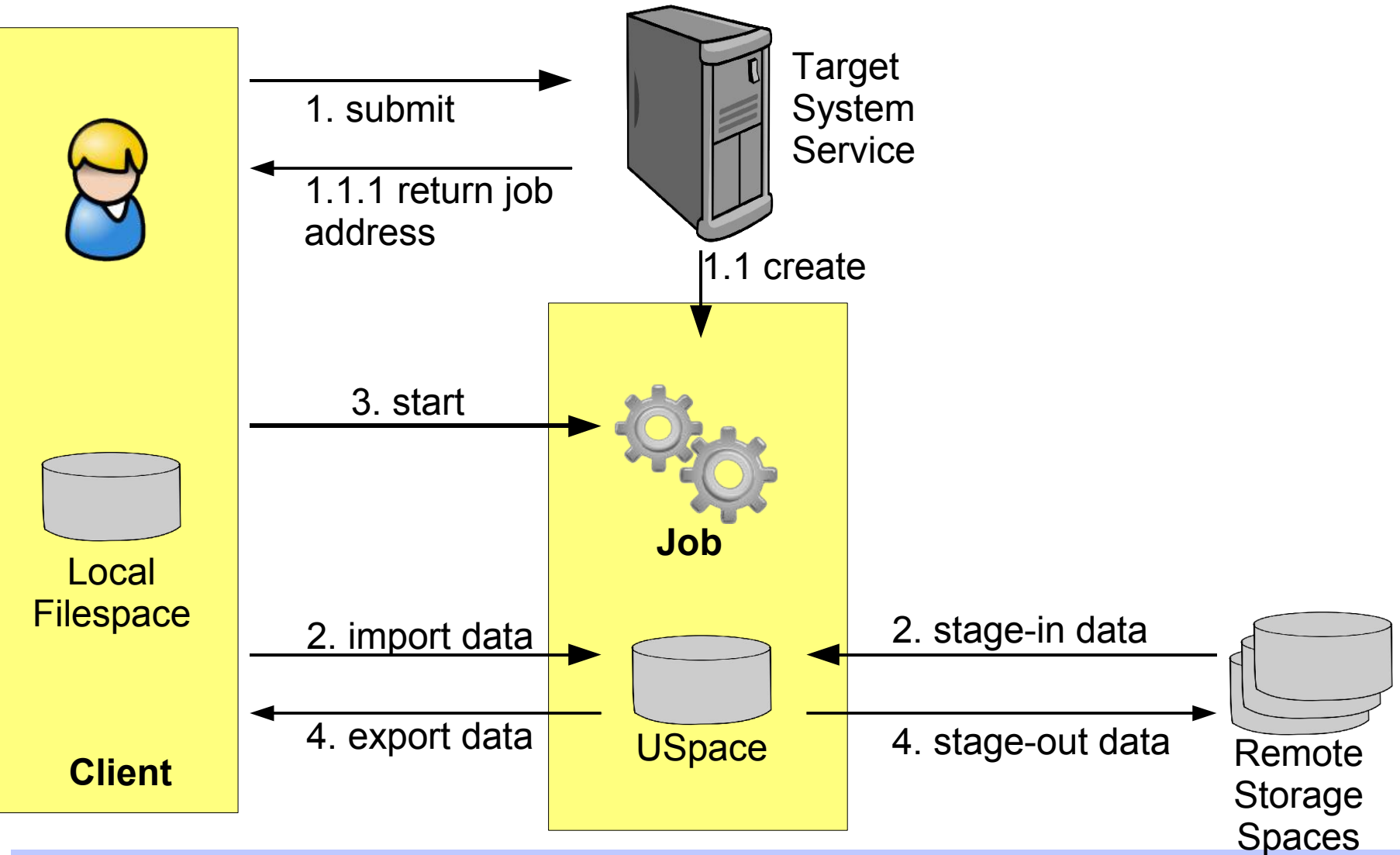
- UNICORE Atomic Services
 - Target system creation
 - Job submission, job management
 - File system access
 - File import/export control
- Registry
 - Publish services (address, service description)
 - Shareable between sites
 - Single point of entry for clients
- Emerging standards (OGSA-*)
 - OGSA-BES (basic execution services)



Target system factory



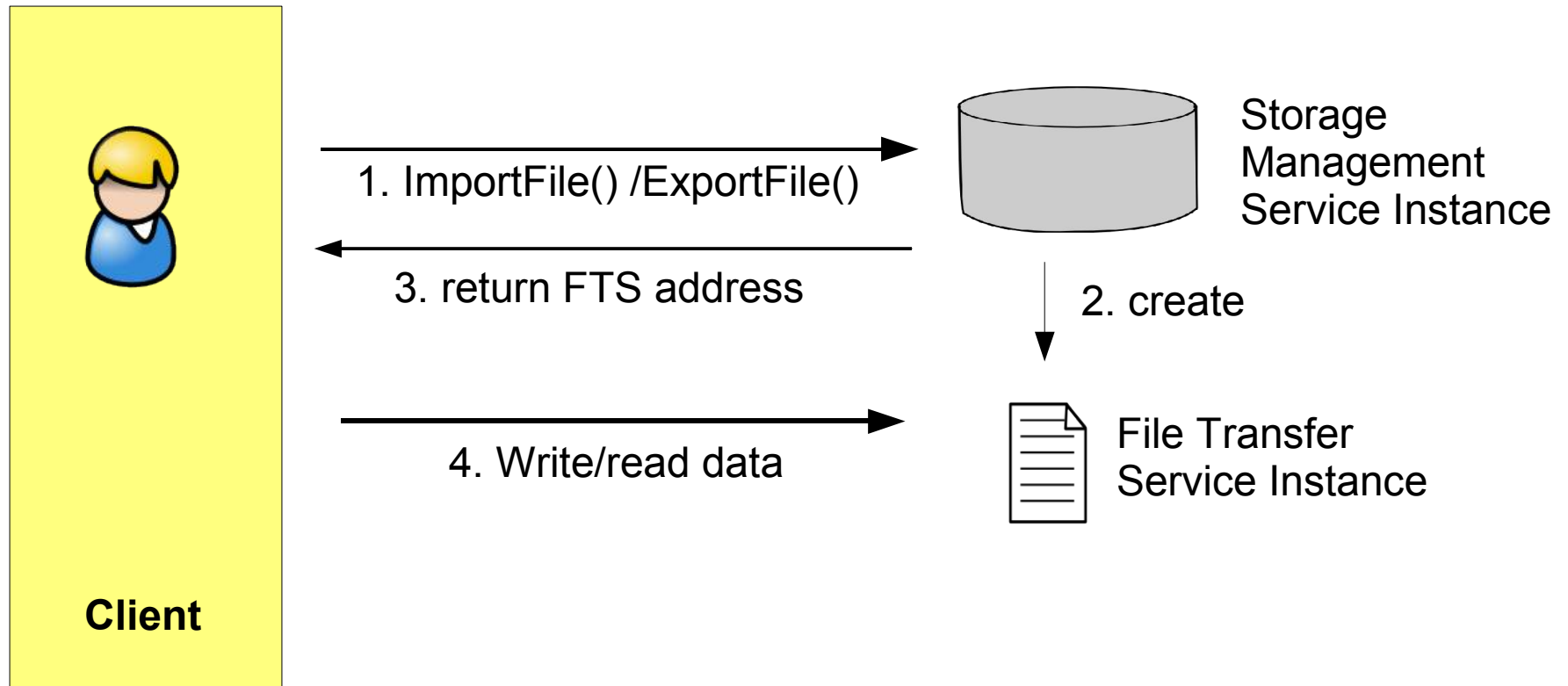
Jobs and storages



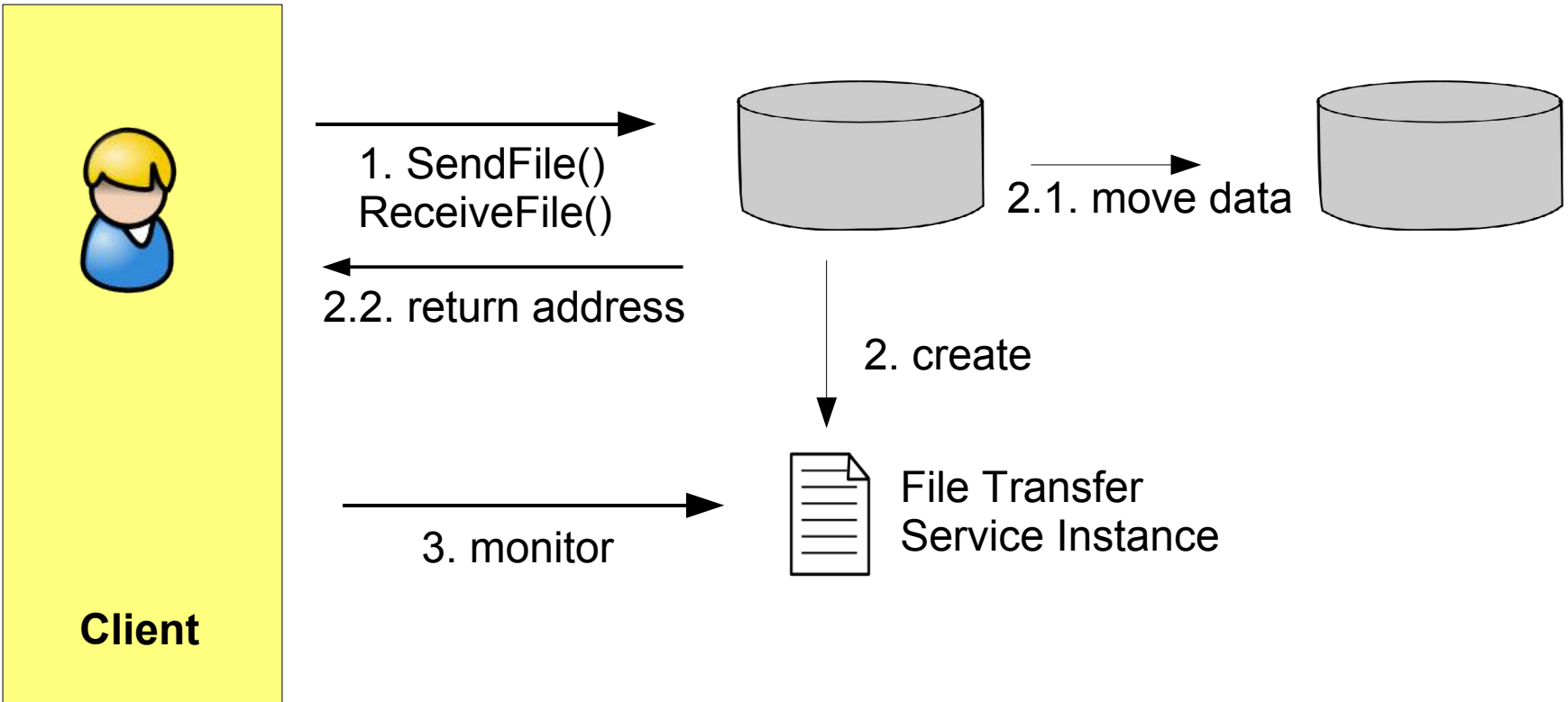
Job description

- JSDL 1.0 (OGF standard)
- Application / Executable to execute
 - Application name / version
(mapped to executable by UNICORE)
 - or: Executable path
 - Arguments and environment variables
- Data stage-in/stage-out specification
- Resources requested

Storage and FileTransfer



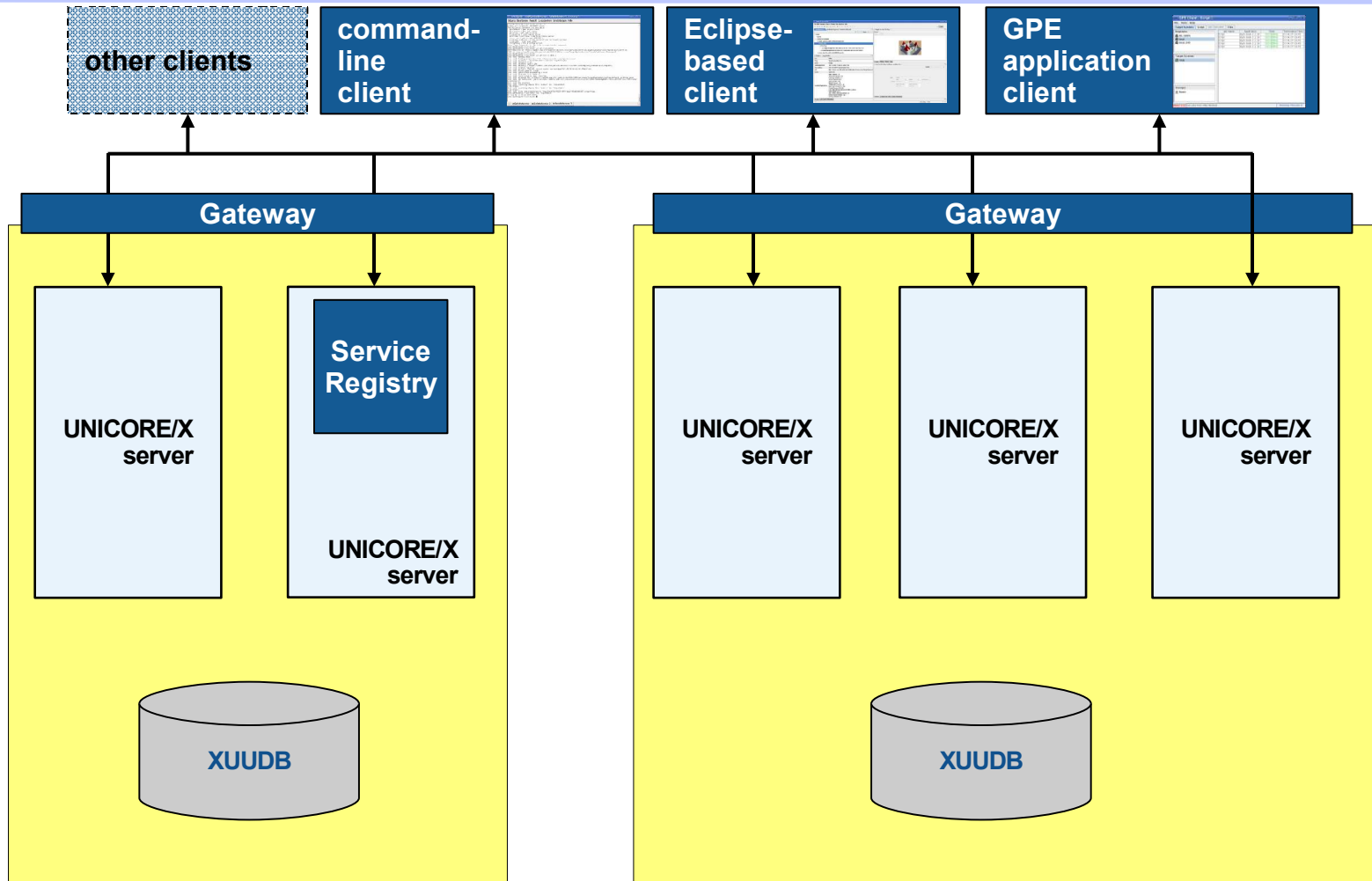
Server-to-server file transfer



File transfer options

- Builtin: OGSA ByteIO
 - Sends data embedded into SOAP messages
 - Single port, full stack (Gateway, SSL connections)
 - Performance ~400kB/sec
 - Rich interface (POSIX-like, block read/write etc)
- Builtin: BFT transfer (based on HTTPs)
 - Single port, full stack: up to 9Mb/sec
 - With Gateway bypass: up to 18Mb/sec
 - Simple interface (bulk write, read supports byte ranges)
- Plus: alternative mechanisms can be plugged in
 - UDT, GridFTP, parallel HTTP...

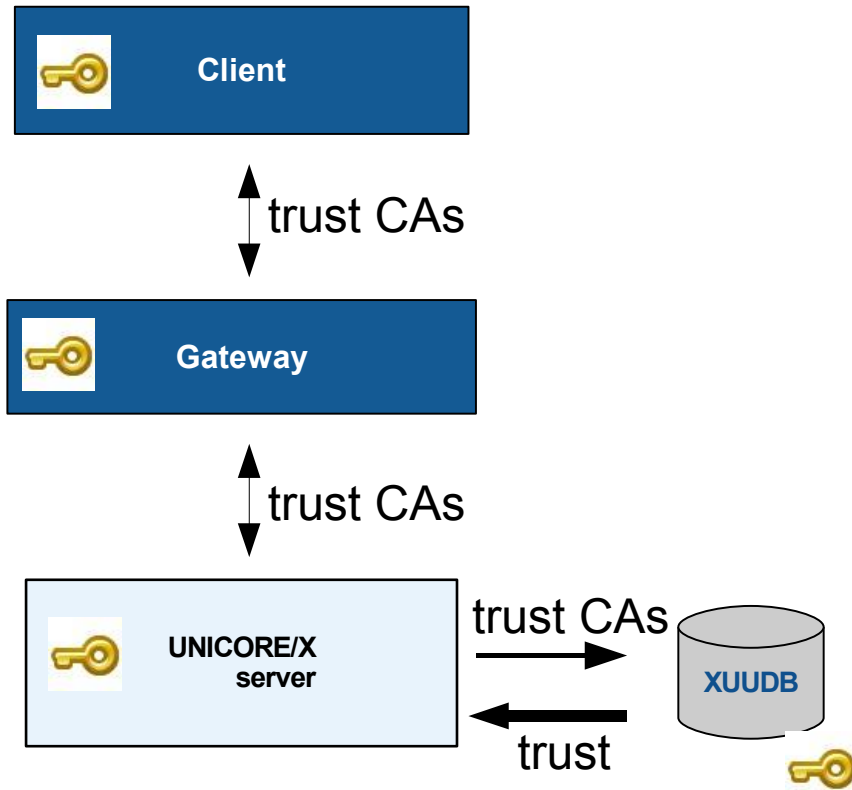
UNICORE 6: multiple Grid sites



Security overview

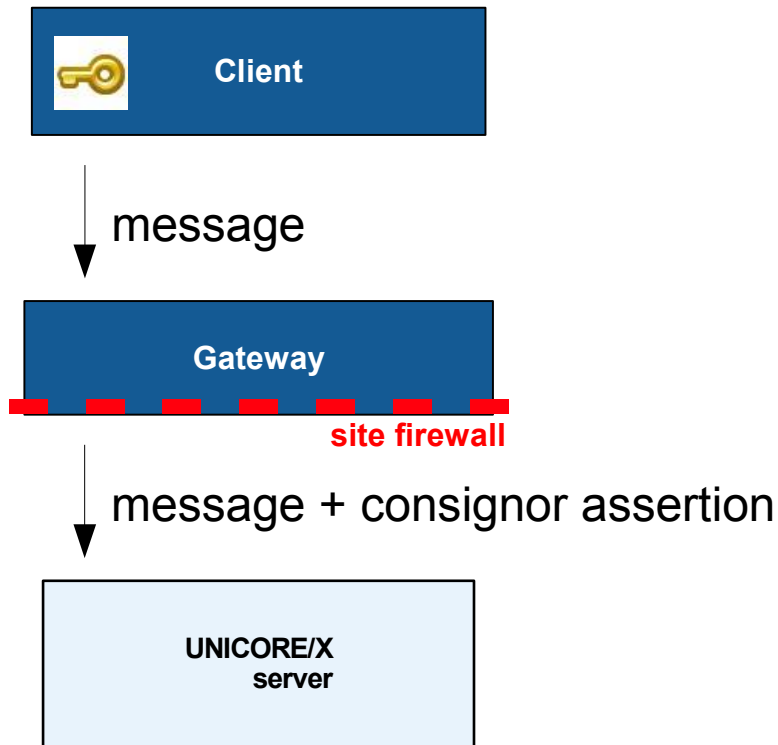
- Users and server components are identified by **X.509** certificates
- Communication paths secured by client authenticated **SSL**
- Messages contain additional security information in the SOAP header (**SAML** assertions)
- Important messages are signed (**XML security**)
- Attributes pulled from XUADB or SAML-based VO server
- Local access control governed by **XACML** policies

X.509 / SSL



- Users and server components need **X.509** certificates 🗝️
- Issued by certification authorities (CAs)
- Client / Gateway mutually trust CAs
- Gateway / server mutually trust CAs
- XUADB must trust the peer directly

Security I: Role of the gateway



- Client to Gateway : SSL
- Client identifies to gateway using his identity 🗝️
- Gateway inserts a SAML assertion into the message („consignor“)
- Gateway forwards the message to the target site
- Gateway receives reply and sends it to the client

Security II – web service layer

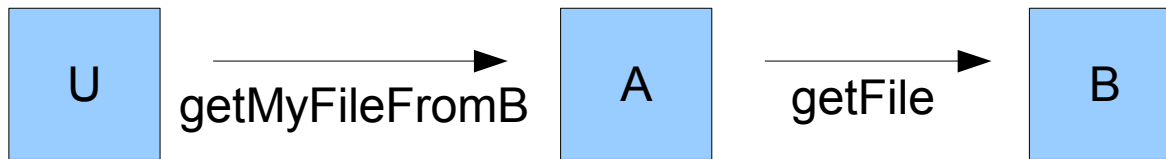
- Extract security info from message
 - Consignor: who sent the message
 - Trust delegation? Who is the original user?
 - Check digital signature
- Lookup security attributes in XUADB
 - Input: user's identity (certificate or DN)
 - Output: Unix login, role, projects, ...
- Check compliance to standard policies
 - Important operations require a digital signature
 - Job submit, storage access, WSRF destroy()

Security III – configurable policies

- Authorisation process
 - Each service call is intercepted and must pass authorisation (transparent to service developer)
 - Policy decision governed by
 - Who tries to access (DN of user)
 - Which service, which method, who owns the service
 - Rules are described by XML policy (**XACML 1.0**)
- If authorisation succeeds, backend activities can use the user's attributes
 - e.g.: XNJS uses Unix login to perform work

Security IV – trust delegation

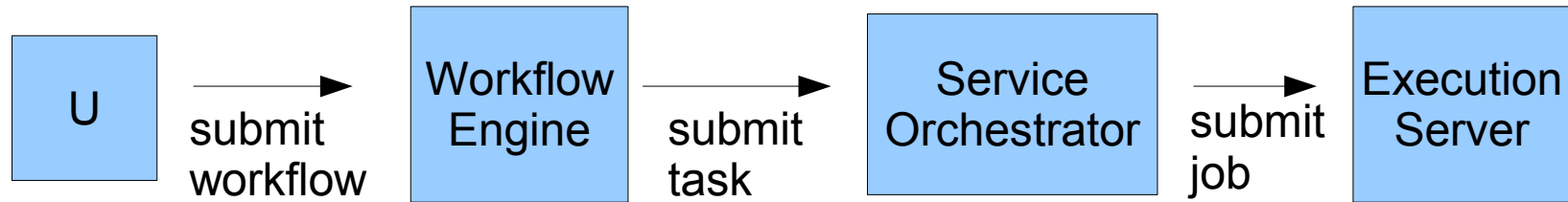
- Common problem: user needs to delegate rights to a service
 - e.g. file from Server B needed to execute job on Server A



- Clients can send a SAML trust delegation token
 - „User U trusts Server A“
 - Digitally signed by U
- Server A adds this token to his messages to Server B
- Server B checks validity and treats request „as if“ sent by User U.

Security V – extended trust delegation

- Trust delegation chain
 - e.g. workflow execution



- Trust delegation chain can be extended
 - „User U trusts Server A trusts Server B“
 - Digitally signed by U and A
- User can **limit** chain length and time of validity

(in-)Security VI: Proxy certificates

- Certificates derived from other certificates
 - signed by user, not CA
- Widely used in other Grid systems
- Third party software (gssssh, gridftp, Kerberos, ...)
- UNICORE offers optional support
 - Gateway module for using proxy certs for SSL
 - UNICORE/X can lookup proxy certs in XUADB
 - future: implement solution used in UNICORE 5
 - generate proxy cert on the client and send it to the server
 - needed for e.g. gridftp

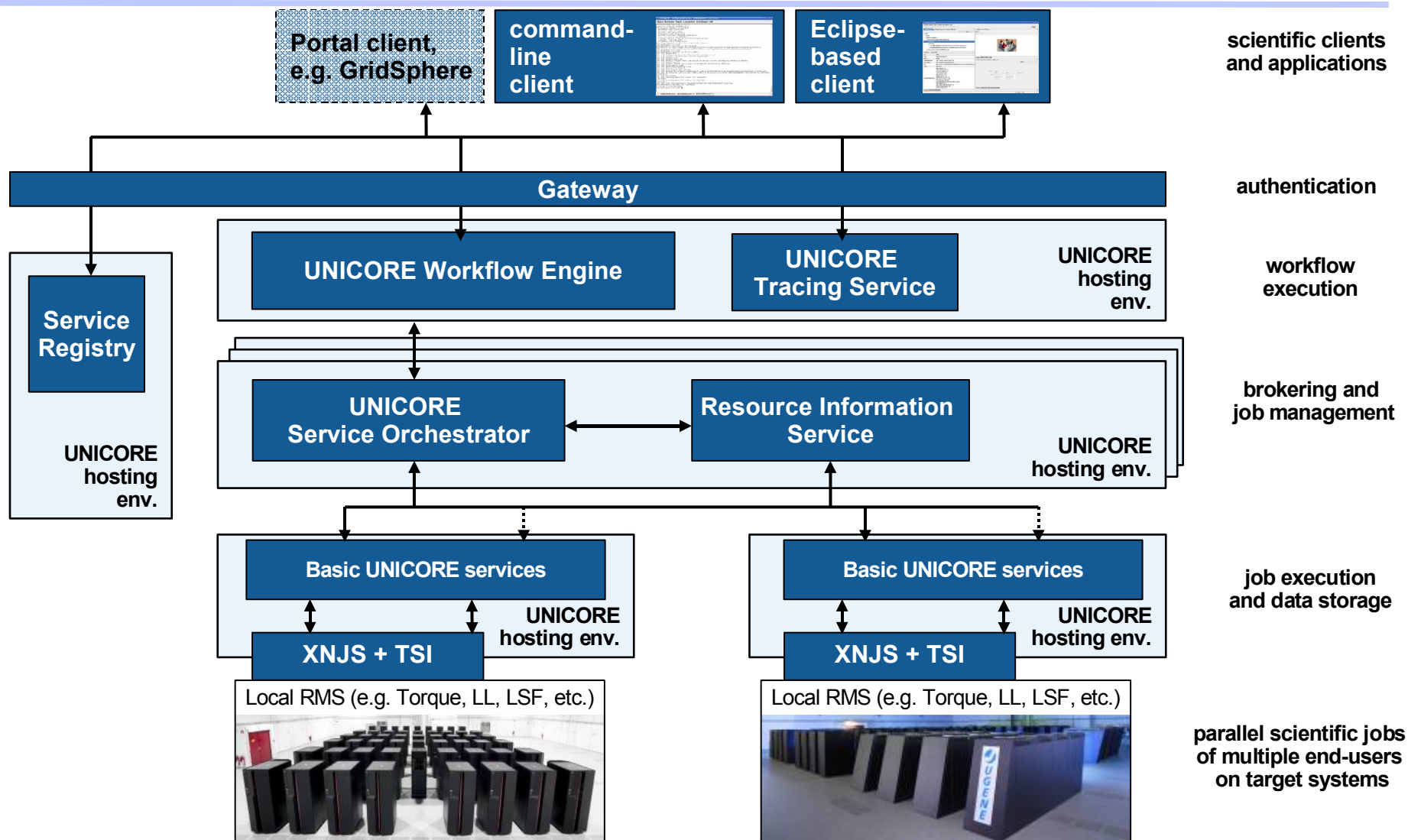
Security VII: VO Management

- Virtual Organisation: community of users sharing resources to achieve common goals
- Goals of VO management
 - simple management of users, groups, and permissions
 - centralised administration
 - keep autonomy of resource providers
- VO Management has been (so far) not common in UNICORE

Option: UVOS server

- Can be used as replacement for the XUADB
- Support for different identity formats
 - email+password, X.509 certificate, X.509 subject DN
- **SAML** support for interoperability.
- Sophisticated features, e.g.
 - VO history
 - Eclipse-based admin client
 - Many authorisation and authentication possibilities

UNICORE 6 Workflow system

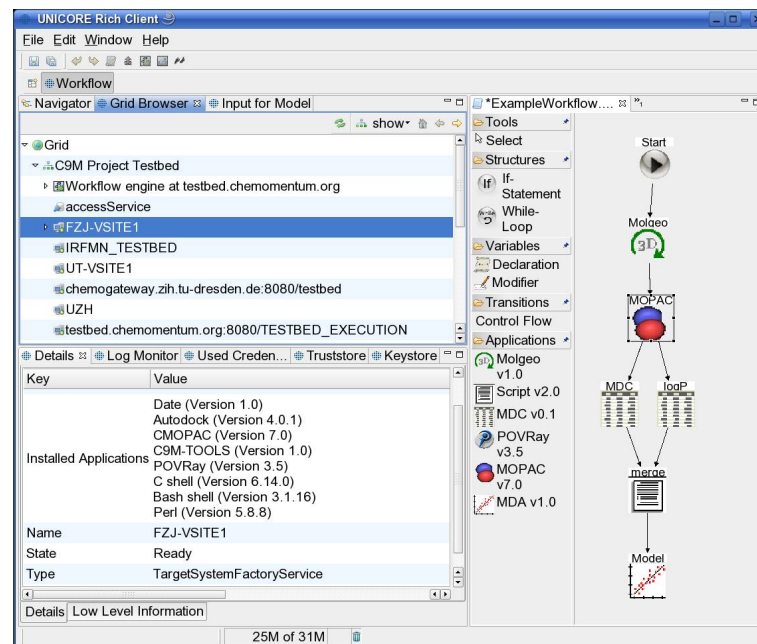


- Two layer architecture for scalability
- Workflow engine
 - Based on Shark open-source XPDL engine
 - Pluggable, domain-specific workflow languages
- Service orchestrator
 - Brokering based on pluggable strategies
 - Job execution and monitoring
 - Callback to workflow engine
 - Multiple instances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

UNICORE 6 Workflow features



- Simple graphs (DAGs)
- Split, merge, synchronize
- Loops, conditions (e.g. exit code checks)
- Workflow variables
- Performance features
 - Loop unrolling
 - (semi-)Automated dataset splitting
- Usage scenarios
 - Scientific workflows
 - Parameter studies



UNICORE 6 status

- Core server v6.1.3 (October 28, 2008)
 - Gateway, Registry, UNICORE/X, XUADB, TSI
 - Basic services (incl. OGSA-BES)
 - Security infrastructure (trust delegation, message signing, SAML-VOMS support)
 - OGSA BytelO and http based file transfer
- UCC 6.1.3 (October 28, 2008)
- Eclipse client 6.1.3, Workflow 6.1.3: soon

Some statistics

- Downloads (April – September 2008)
 - Server quickstart bundle: 400 downloads/month
 - Workflow bundle: 80 downloads/month
 - Clients
 - Eclipse client: 150 downloads / month
 - Commandline client: 90 downloads / month
 - Application client: 40 downloads/month
 - Overall trend: increasing
- Mailing lists
 - unicore-support@lists.sf.net very active

Outlook to the next release

- UNICORE 6.2, estimated end 2008
- New features, such as
 - Enhanced storage service
 - fine grained access control
 - “find“ operation
 - UDT high performance file transfer
 - Enhanced administrative features (dynamic service deployment, improved logging/monitoring)
- Add-ons
 - CIS Information service
- ... and much more
feature tracker: http://sourceforge.net/tracker2/?group_id=102081&atid=633905

Additional components and other activities (incomplete list)

- Enterprise service bus, JBI, BPEL, ... (A-Ware)
- Monitoring, BPEL workflows, ... (D-Grid)
- Meta-scheduling, advance reservation, ... (Phosphorus)
- SAML based VO management (OMII-Europe, Chemomentum)
- Portal activities and web-based clients (OMII Europe, A-Ware, Chemomentum)
- GLUE 2 compliant information service (FZJ)
- High-performance filetransfers UDT, GridFTP (FZJ)
- ... and much more

Clients, clients, clients

- Eclipse client
 - Workflows
 - Highly flexible and extensible
- Application client
 - Single application
- Command line client (UCC)
 - batch processing, tools, performance tests, ...
- Grid APIs
 - HiLA (available today)
 - GAT/SAGA (future)
- Web-based clients

Thank you!

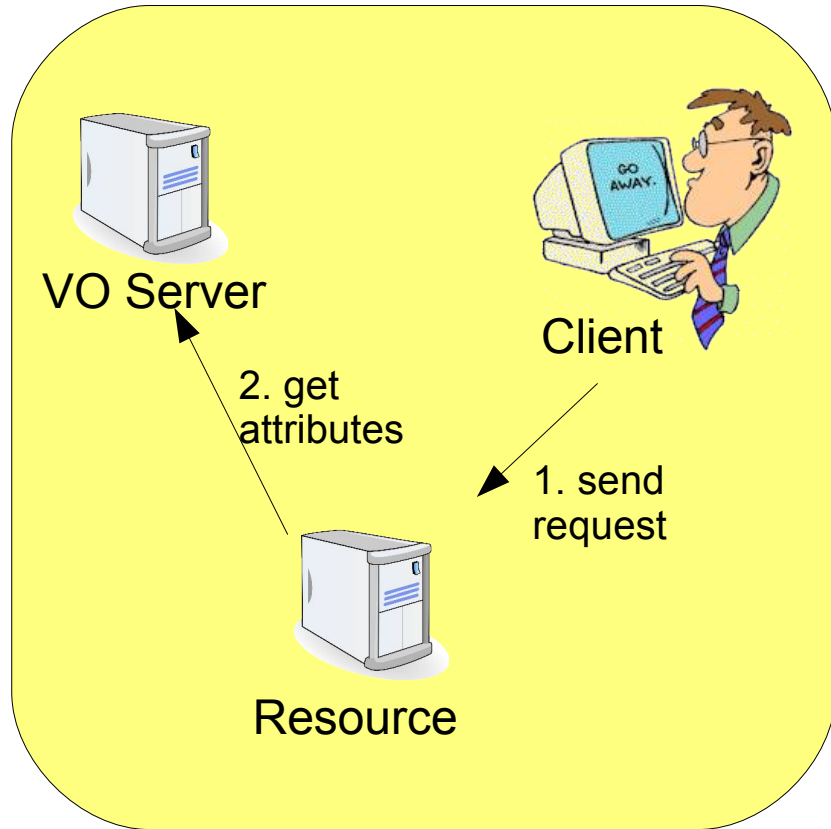


Downloads, documentation, tutorials, mailing lists, community links, and more:

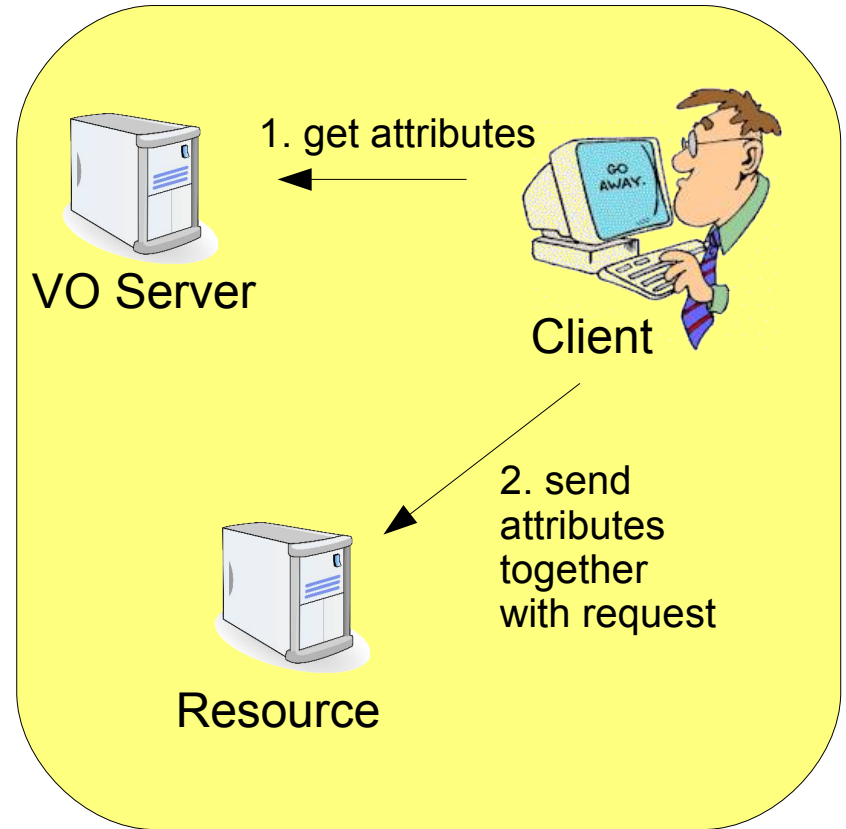
<http://www.unicore.eu>

Extra slides

UVOS: Operational modes



„Pull“ mode



„Push“ mode