

UNICORE

Version 6.0

Bernd Schuller, FZ Jülich

Outline

- Architecture overview
- Some feature highlights
- Jobs and storages
- Service registry
- Status and Roadmap

UNICORE 6 „design principles“

- Integrated, complete Grid middleware stack
- Excellent application support
- Workflows
- Graphical clients
- Support many operating and batch systems
 - Unixes, MacOS, Windows
 - TSIs: embedded, LL, Torque, SGE, ...
 - Existing UNICORE 5 TSI installations can be re-used
- Easy installation and configuration
- Easy to administer and monitor

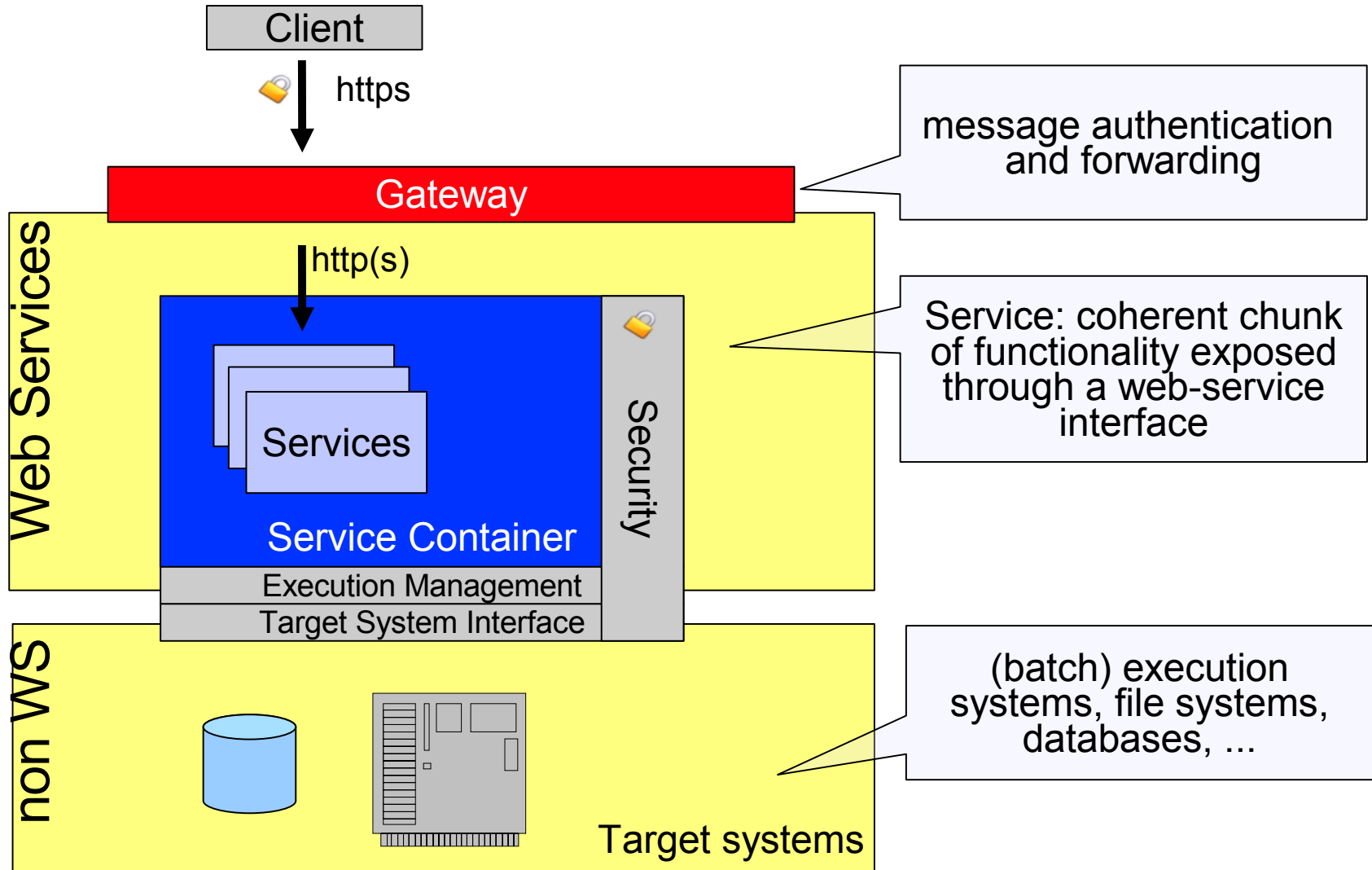
UNICORE 5 plus ...

- Openness, extensibility, interoperability
 - Service-oriented
 - Web services foundation
 - Communications are no longer tied to the Java platform
 - Highly extensible core engine

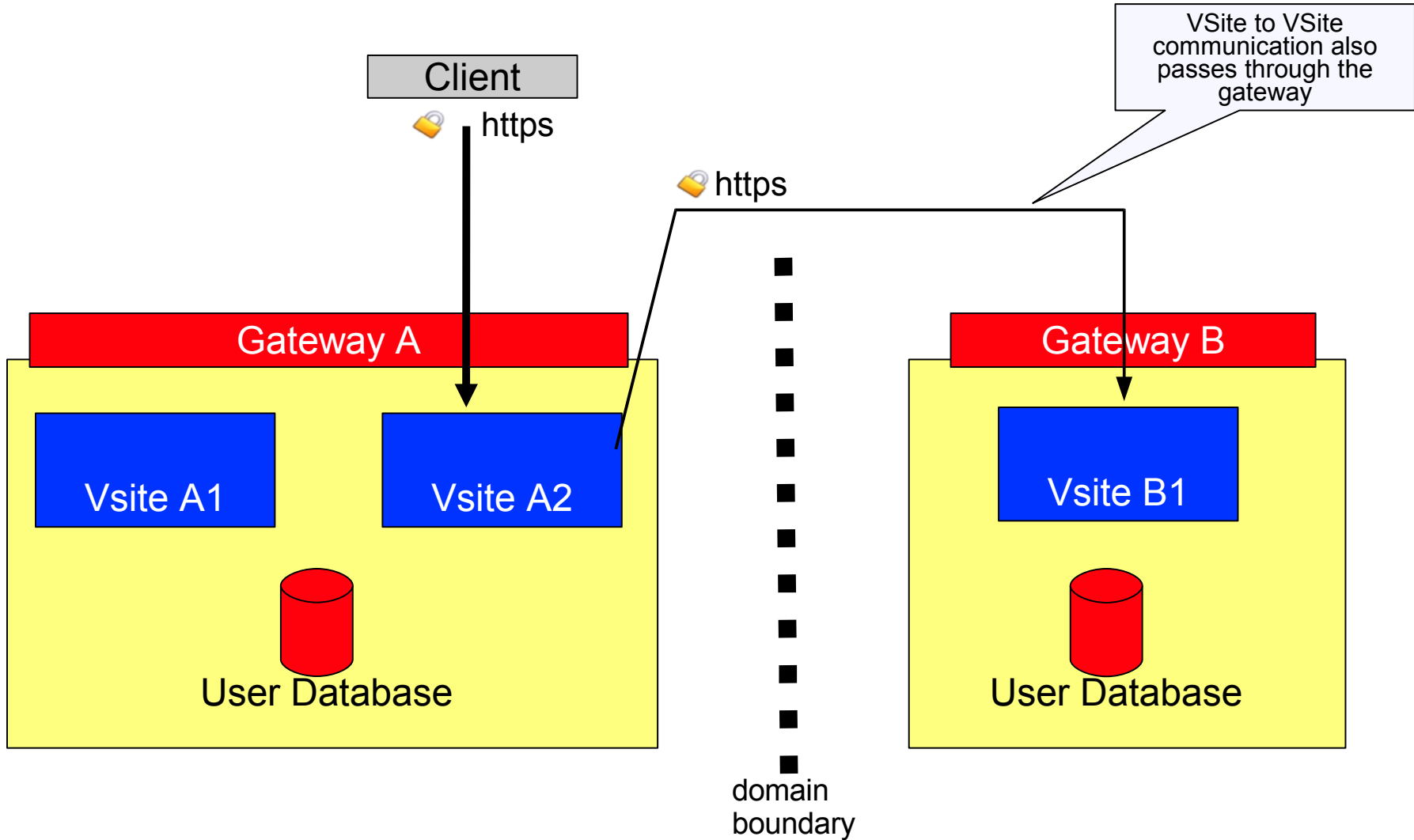
... added value

- Performance and scalability
 - Fast default file transfer
 - Tested and proven scalability of core components
- State of the art software
 - Uses Java 5 features (generics, JMX)
 - Jetty 6 web server
 - XFire SOAP stack
 - HSQLDB embedded database
 - ...

UNICORE 6 architecture



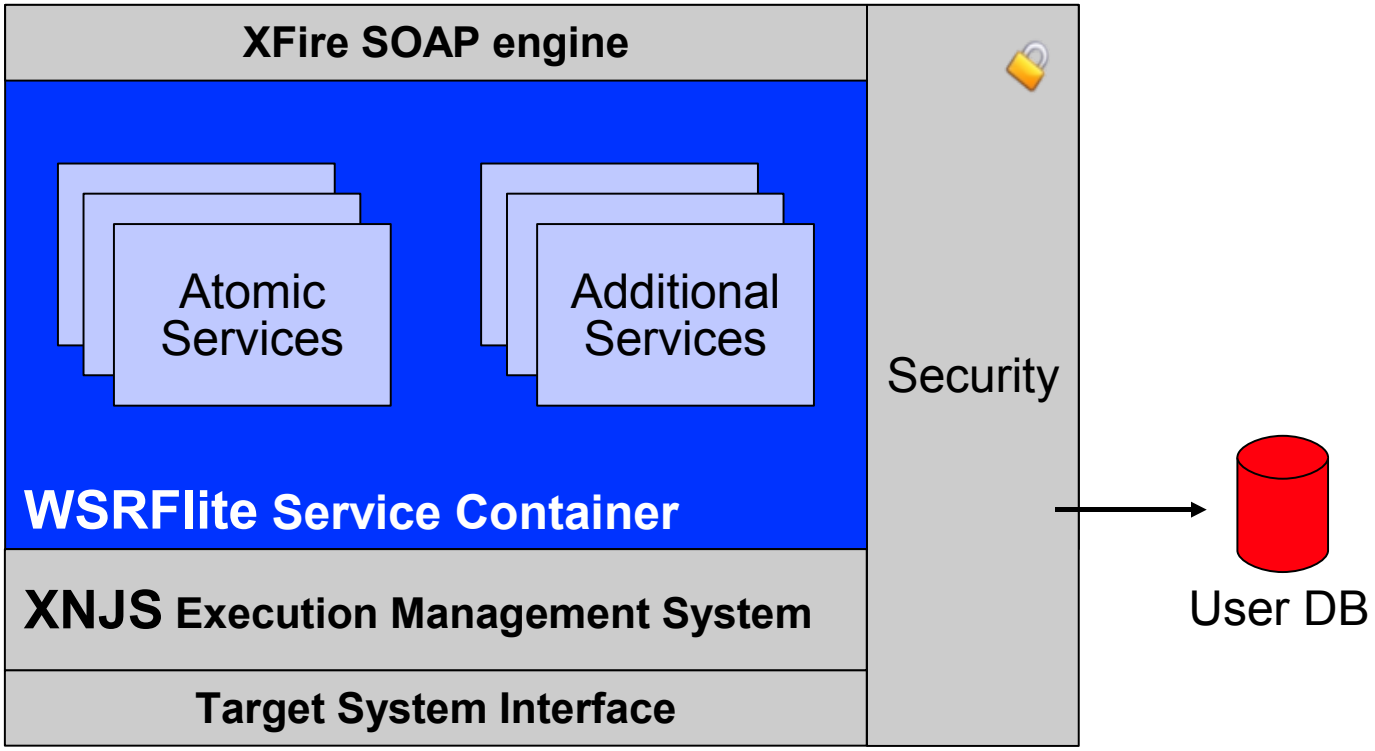
USites, VSites and all that



Feature highlights

- Fast WSRF 1.2 hosting environment
- XNJS: scalable and extensible execution engine
- Full set of basic Grid services
- Plugable file transfer mechanisms
- GPE: flexible client framework
- Simple, lightweight installation
- Java management extensions (JMX) support

Detailed view: VSite



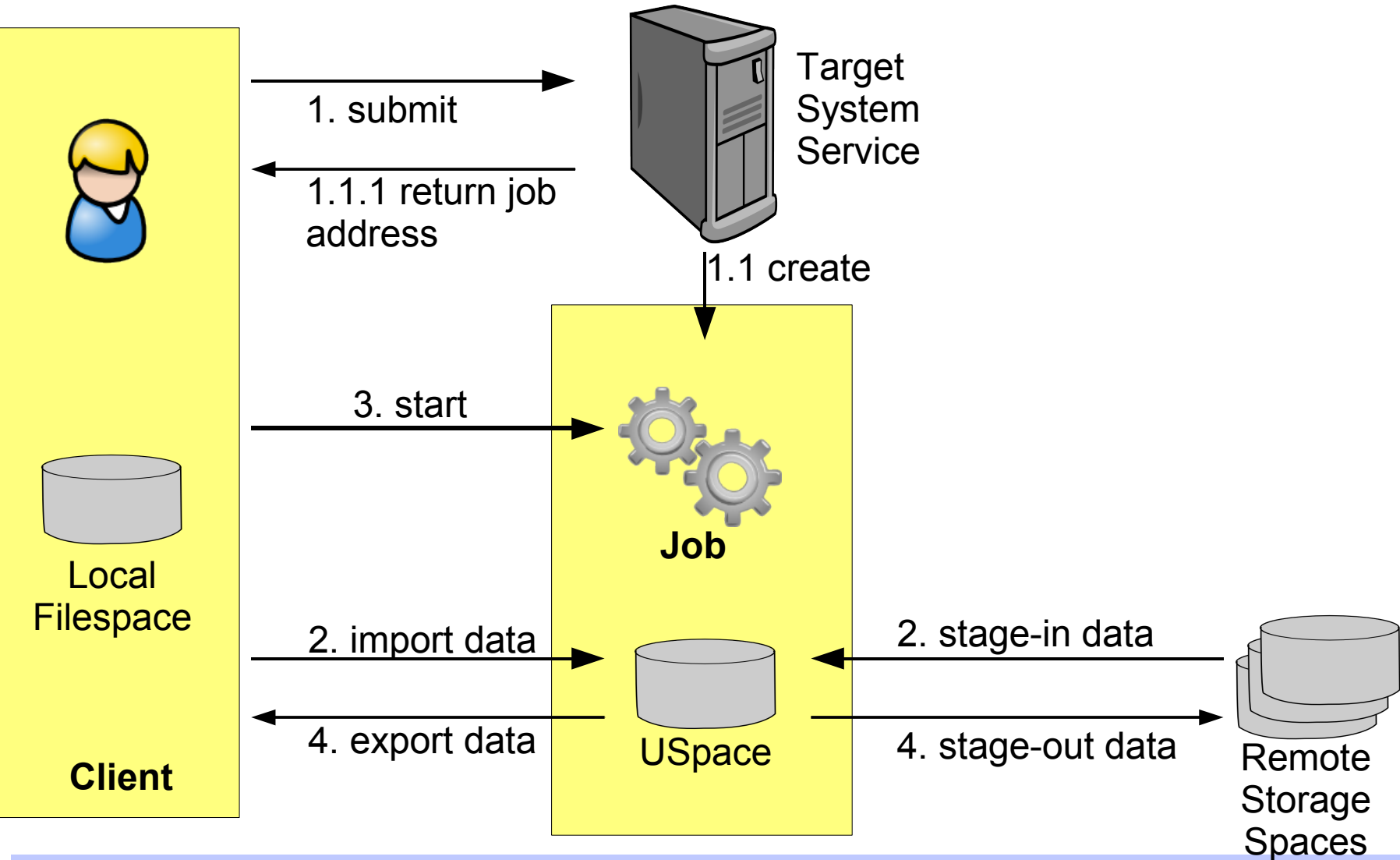
Service container (WSRFlite)

- State of the art web services tools:
 - XFire, XmlBeans, Jetty
- Specifications
 - WSRF (version 1.2 final)
 - WS ServiceGroup
 - WS BaseNotification
 - WS-I „plain“ web services
- Characteristics:
 - High performance
 - Ease of configuration
 - Developer friendly, embeddable, flexible

Base Services

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

Jobs and storages



XNJS execution engine

- Execution management system
 - Submit and manage jobs
 - Control file transfers
- JSDL 1.0 support
 - extensions: POSIX, „HPC“, soon: SPMD
- UNICORE concepts
 - TSI, USpace, abstraction, incarnation
- Scalable and fast
- Extensible
 - Logging, tracing, monitoring, notifications
 - Third-party system integration

File transfer

- Plugable mechanisms
 - Both for client-server and server-server transfers
- Default mechanism: OGSA ByteIO
 - Sends data as SOAP messages
 - Single port, full stack (Gateway, SSL connections)
 - Reasonable performance of ~400kB/sec (for comparison: UPL ~ 200kB/sec)
- Room for improvements
 - Use MTOM
 - Use non-SOAP protocols
 - HTTP, TCP/IP, UDP hole punching, ...

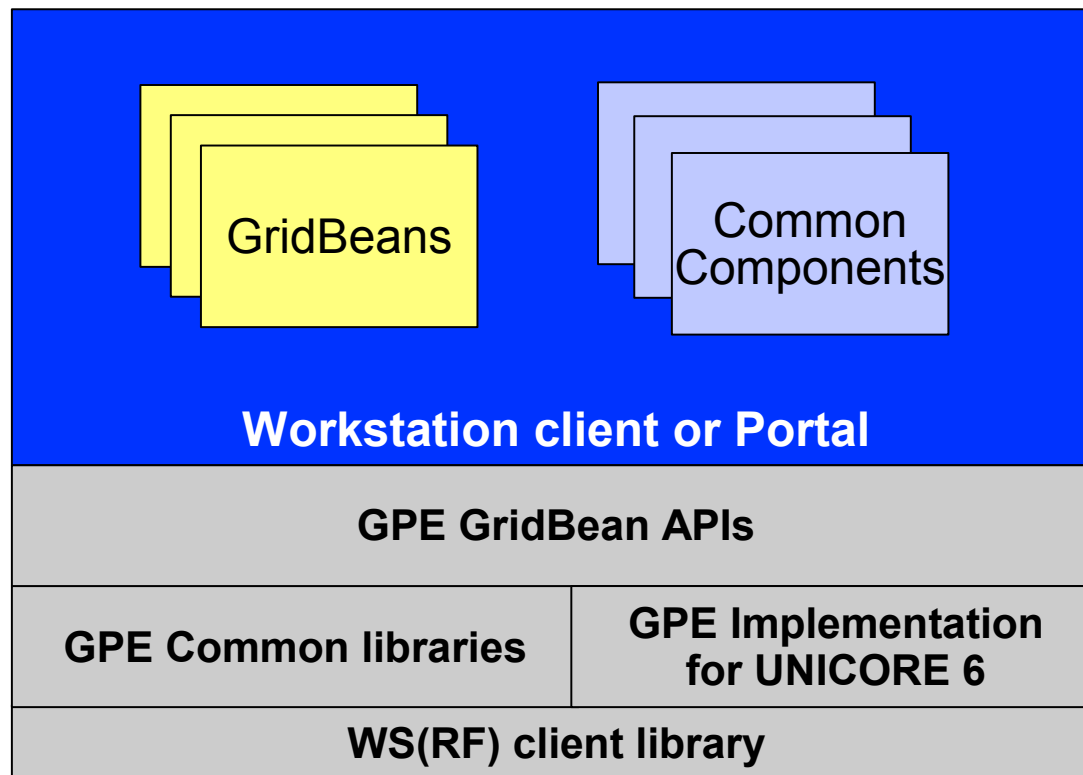
Clients: GPE for UNICORE

- Application client
 - Single application, easy to use
- Expert client
 - Workflows
 - Replacement for UNICORE 5 client
- Web portal client

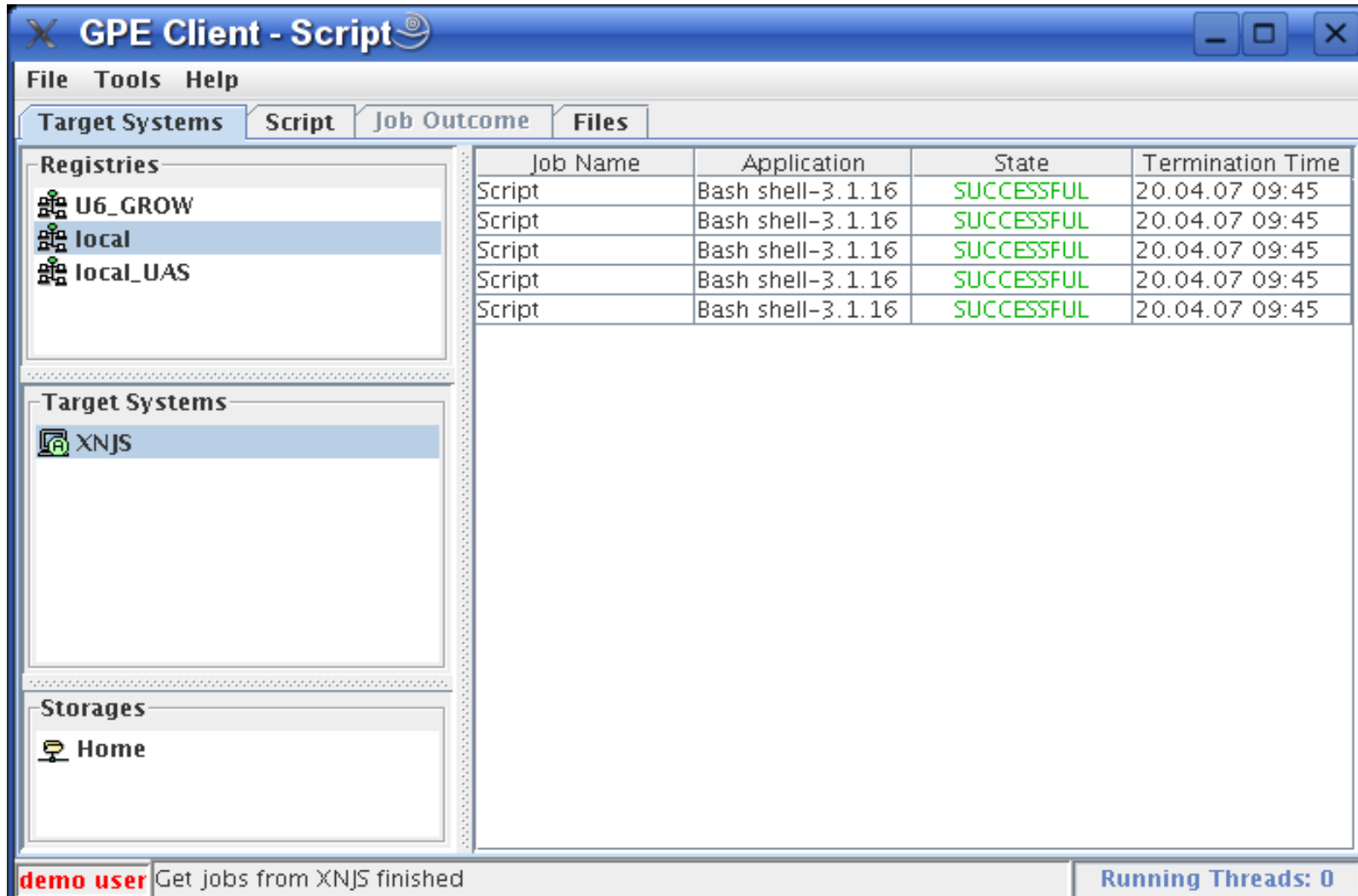
- All are extensible through „GridBeans“
 - GridBeans provide application-specific interfaces

Detailed view: Clients

- Based on Intel's Grid Programming Environment (GPE)



Application client



The screenshot shows the 'GPE Client - Script' application window. The window has a menu bar with 'File', 'Tools', and 'Help'. Below the menu bar are four tabs: 'Target Systems', 'Script', 'Job Outcome', and 'Files'. The 'Job Outcome' tab is active, displaying a table with the following data:

Job Name	Application	State	Termination Time
Script	Bash shell-3.1.16	SUCCESSFUL	20.04.07 09:45
Script	Bash shell-3.1.16	SUCCESSFUL	20.04.07 09:45
Script	Bash shell-3.1.16	SUCCESSFUL	20.04.07 09:45
Script	Bash shell-3.1.16	SUCCESSFUL	20.04.07 09:45
Script	Bash shell-3.1.16	SUCCESSFUL	20.04.07 09:45

On the left side of the window, there are three panels: 'Registries' containing 'U6_GROW', 'local', and 'local_UAS'; 'Target Systems' containing 'XNJS'; and 'Storages' containing 'Home'. At the bottom of the window, a status bar shows 'demo user' on the left, 'Get jobs from XNJS finished' in the middle, and 'Running Threads: 0' on the right.

Expert client

The screenshot displays the GPE Expert Client 1.4.2_RC2 interface. On the left, a tree view shows the grid structure: Grid, localhost, DEMO-SITE, Workflow_DEMO-SITE, and workflow Example Flow RUNI. The main area shows a workflow diagram with three 'Script' nodes connected by arrows. The top node is highlighted with a green dashed box. To the right, the 'Workflow Input Parameters' tab is active, showing a table of input parameters and an 'Edit' button. Below it, the 'Output Parameters' section shows a table with one output parameter and another 'Edit' button.

Workflow Input Parameters

Name	Type	Value
Source	URL	/home/schuller/foo.txt
InputFileset	FILE_SET	
TargetSystem	XML	localhost!expert_workfl...

Output Parameters

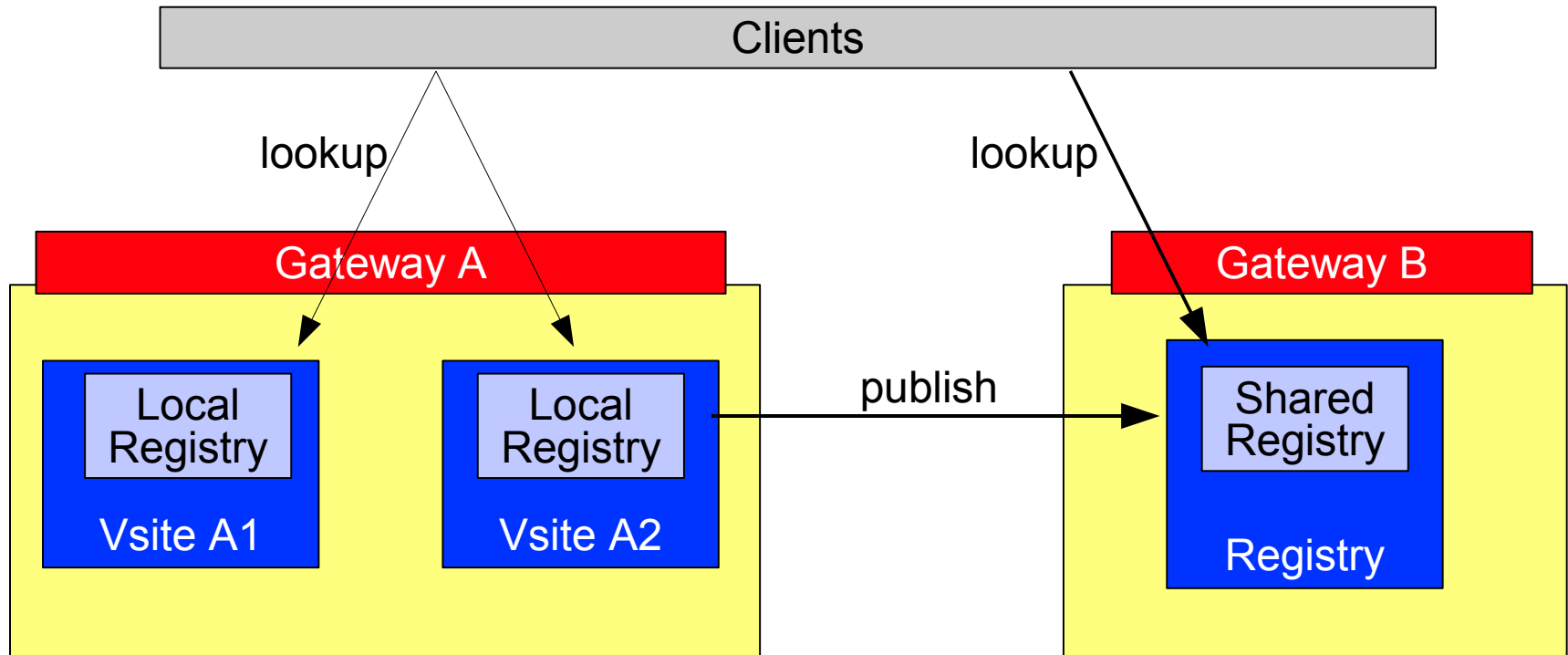
Name	Type	Value
EndPointReference	XML	

demo user | expert_explorer_Workflow_Refreshing finished | Running Threads: 0

Commandline client: UCC

- Run jobs, get status, get output
 - single job and batch mode available
- Transfer files
 - local to remote
 - server-to-server
- Administrative use
 - list jobs, clean up, etc.
- Scripting UNICORE resources using Groovy scripts
- Extensible
 - e.g.: access the Chemomentum workflow system

Service Discovery: local and shared registries



- Shared registry is „single point of entry“ for clients
- Local registries can also be used by clients

Security – basics

- Users and servers are identified by X.509 certificates
- Certificates issued by trusted CAs
- Communication paths secured by client authenticated SSL/TLS
 - Clients – Gateway
 - Gateway – Server
 - Server – XUADB (optional)
 - Server – Gateway
- Messages contain additional security information in the SOAP header
- Messages are not encrypted

Security II – web service layer

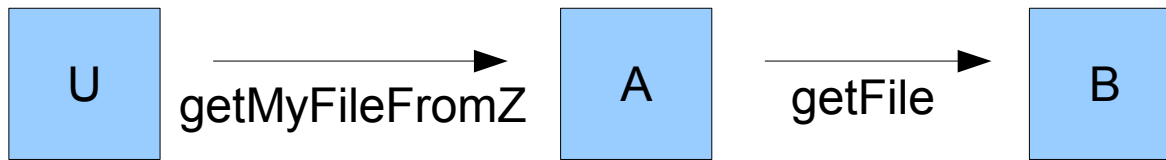
- Extract security info from message
 - User: who is the originator of the message
 - Consignor: who sent the message
 - Trust delegation tokens, digital signature, ...
- Lookup security attributes in XUADB
 - Input: user's identity (certificate or DN)
 - Output: Unix login, role, projects, ...
- Check 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 authZ 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
 - example: file transfer



- 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.

Monitoring and managing a VSite

- Java management extensions (JMX) support

The image displays two overlapping screenshots of the J2SE 5.0 Monitoring & Management Console. The left screenshot shows the Summary tab, and the right screenshot shows the MBeans tab.

Summary Tab (Left Screenshot):

- Uptime:** 2 hours 4 minutes
- Total compile time:** 28,791 seconds
- Threads:** Live Threads: 44, Daemon threads: 12
- Memory:** Current heap size: 42,946 kbytes, Maximum heap size: 126,272 kbytes, Objects pending for finalization: 0, Garbage collector: Name = 'PS MarkSweep', Garbage collector: Name = 'PS Scavenge'
- Classes:** Current classes loaded: 4,726, Total classes loaded: 4,766
- Operating System:** Total physical memory: 2,074,580 kbytes, Committed virtual memory: 397,088 kbytes

MBeans Tab (Right Screenshot):

- MBeans Tree:** JMImplementation, WSRFlite (Kernel, Service <FileTransferBFT>, Service <FileTransferRBYTEIO>, Service <FileTransferSBYTEIO>, Service <JobManagement>, Service <Registry>, Service <ServiceGroupEntry>, Service <StorageManagement>, Service <TargetSystemFactoryService>, Service <TargetSystemService>), XNJS (de.fzj.unicore.xnjs.simple.jmx.BasicMBean), java.lang, java.util.logging
- Attributes Table:**

Name	Value
InstanceCount	9
ServerTime	Mittwoch, 18. April 2007 19:35 U...

Refresh button is visible at the bottom right of the MBeans tab.

Installation: server bundle

IzPack - Installation of UNICORE_Quickstart

UNICORE

Select the packs you want to install:
Note: greyed out packs are required.

<input checked="" type="checkbox"/>	Base	56,96 KB
<input checked="" type="checkbox"/>	UNICORE/X	20,05 MB
<input checked="" type="checkbox"/>	Gateway	12,22 MB
<input checked="" type="checkbox"/>	XUADB	8,4 MB
<input checked="" type="checkbox"/>	TSI	793,06 KB
<input checked="" type="checkbox"/>	Doc	13,4 KB

Description
The UNICORE 6 Gateway acts as a web-services firewall for a set of UNICORE/X servers.

Total space Required: 41,51 MB
Available space: 10,84 GB

(Made with IzPack - <http://www.izforge.com/>)

Previous Next Quit

Installation: configuration dialog

IzPack - Installation of UNICORE_Quickstart

Please check the settings for the server host and port values and edit them to fit your wishes.

Please enter the host and port where the gateway will be running.

Gateway Host:

Gateway Port:

Please enter information about the UNICORE/X server.

Server Host:

Server Port:

Vsite Name:

Please enter the host and port where the XUADB server that will be used.

XUADB Host:

XUADB Port:

(Made with IzPack - <http://www.izforge.com/>)

UNICORE 6 status

- Overall status: **release candidate** July 21, 2007
- Functionality
 - Web service /WSRF core
 - Security (message signing, trust delegation)
 - Basic services (registry, jobs, files)
 - Application client
 - BPEL Workflow engine + Expert client
 - Command-line client and scripting tools
- Final release estimated beginning of August

Roadmap

- 6.0 final release: early August, 2007
 - Available functionality as in the release candidate
 - Client-side based on GPE 1.4
- 6.1 release, late 2007
 - Enhanced workflow support
 - Interoperability components (e.g. SAML VOMS)
 - Client-side based on Eclipse and GPE 1.5

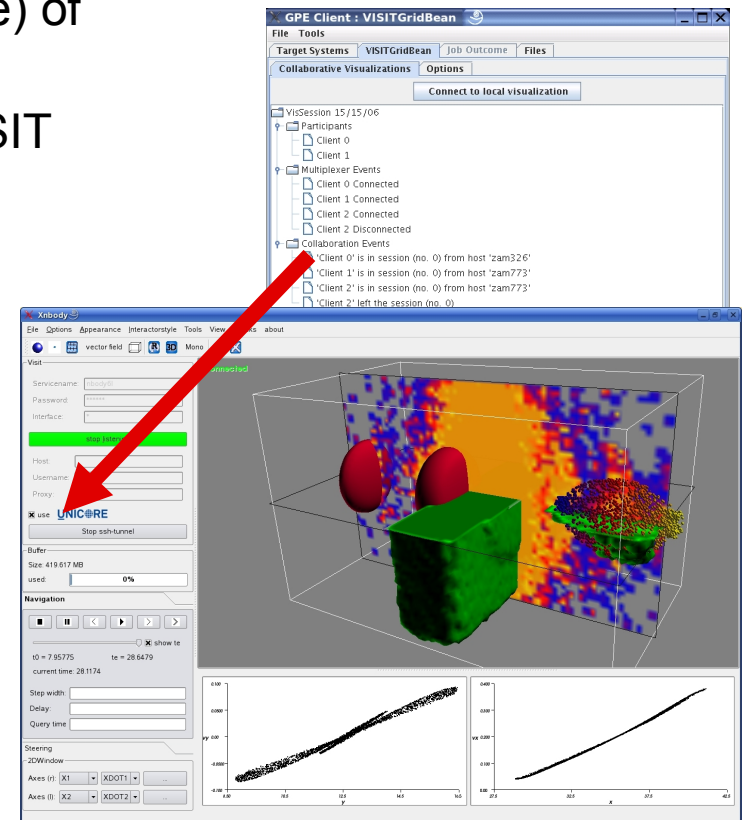
Additional components and other activities (incomplete list)

- Information service
- Workflow options
 - A-Ware (Enterprise service bus, BPEL, ...)
 - Chemomentum (Domain-specific languages, brokering, ...)
- Meta-Scheduling (Phosphorus)
- Security activities
 - VO activities (OMII Europe, Chemomentum, Intel)
- Portal activities (Intel, OMII Europe, A-Ware)
- Monitoring activities (for example in D-Grid)

Use case: visualisation and steering

- Collaborative Online Visualization and Steering

- A framework for scientific simulations and visualizations
 - In addition to usual post-processing (offline) techniques
 - Enables to view the actual status (online) of parallel simulations
 - Based on the communication library VISIT
 - Works with all VISIT-enabled scientific visualizations
- Implemented as an additional UNICORE service
- Collaboration server + multiplexer for geographically dispersed clients
- Usage of UNICORE security infrastructure for single sign-on
- Successfully demonstrated at OGF18, Europar'06, SC'06, ...



... it also works on Windows

The screenshot displays a Windows desktop environment. The desktop background is blue with several icons: Papierkorb, iTunes, QuickTime Player, Eingabeaufforderung, and GPE4Unicore Client. The GPE4Unicore Client application is open, showing a window titled "GPE Client - Application". The window has a menu bar (File, Tools, Help) and several tabs: Target Systems, Application, Job Outcome, and Files. The "Job Outcome" tab is active, displaying a table with the following data:

Job Name	Application	State	Termination Time
NoName	Date-1.0	SUCCESSFUL	20.04.07 11:43

Below the table, there are sections for "Registries" (showing local_XNJS), "Target Systems" (showing XNJS), and "Storages" (showing Home). At the bottom of the application window, a status bar shows "demo user Get jobs from XNJS finished" and "Running Threads: 0".

To the left of the application window, a terminal window titled "Eingabeaufforderung" displays the following log output:

```
20.04.2007 10:43:36 d
INFO: Done
20.04.2007 10:43:41 d
INFO: Reading from co
20.04.2007 10:43:42 d
INFO: Have 3 applicat
20.04.2007 10:43:43 d
INFO: Setting server
20.04.2007 10:43:54 d
INFO: Adding action w
c756e69636f72652d7375
Role: user: role from
Authenticated by: htt
20.04.2007 10:43:54 d
INFO: Creating new ex
client: C=DE,ST=somep
Role: user: role from
Authenticated by: htt
20.04.2007 10:43:54 d
INFO: Xlogin: Bernd S
20.04.2007 10:43:54 d
INFO: Local TS initia
20.04.2007 10:43:56 d
INFO: Local TS initia
20.04.2007 10:43:56 d
INFO: SMS init in <C:
20.04.2007 10:43:56 d
INFO: SMS init in <C:
20.04.2007 10:43:56 d
INFO: Created new Sto
20.04.2007 10:43:57 d
INFO: Setting server
20.04.2007 10:44:00 d
INFO: Setting server
20.04.2007 10:44:02 d
INFO: Job Control: St
20.04.2007 10:44:02 d
INFO: Started 69fc4208
20.04.2007 10:44:02 d
INFO: Submitting...
20.04.2007 10:44:03 de.fzj.unicore.xnjs.simple.LocalTS <init>
INFO: Local TS initialised with base dir <C:\Programme\UNICORE\data\FILESFACE>
20.04.2007 10:44:03 de.fzj.unicore.xnjs.simple.LocalTS exec
INFO: Executing 'C:\WINDOWS\system32\cmd.exe /c date /t' in C:\Programme\UNICORE\data\FILESFACE\69fc4208-0dae-41fa-837d-4a2cd5b934f1
20.04.2007 10:44:04 de.fzj.unicore.xnjs.simple.LocalTS exec
INFO: Done with 'C:\WINDOWS\system32\cmd.exe /c date /t' exit code 0, took 375 ms.
20.04.2007 10:44:07 de.fzj.unicore.uas.impl.tss.rp.ApplicationsResourceProperty update
INFO: Have 3 applications on XNJS
C:\Programme\UNICORE\unicorex\logs>
```