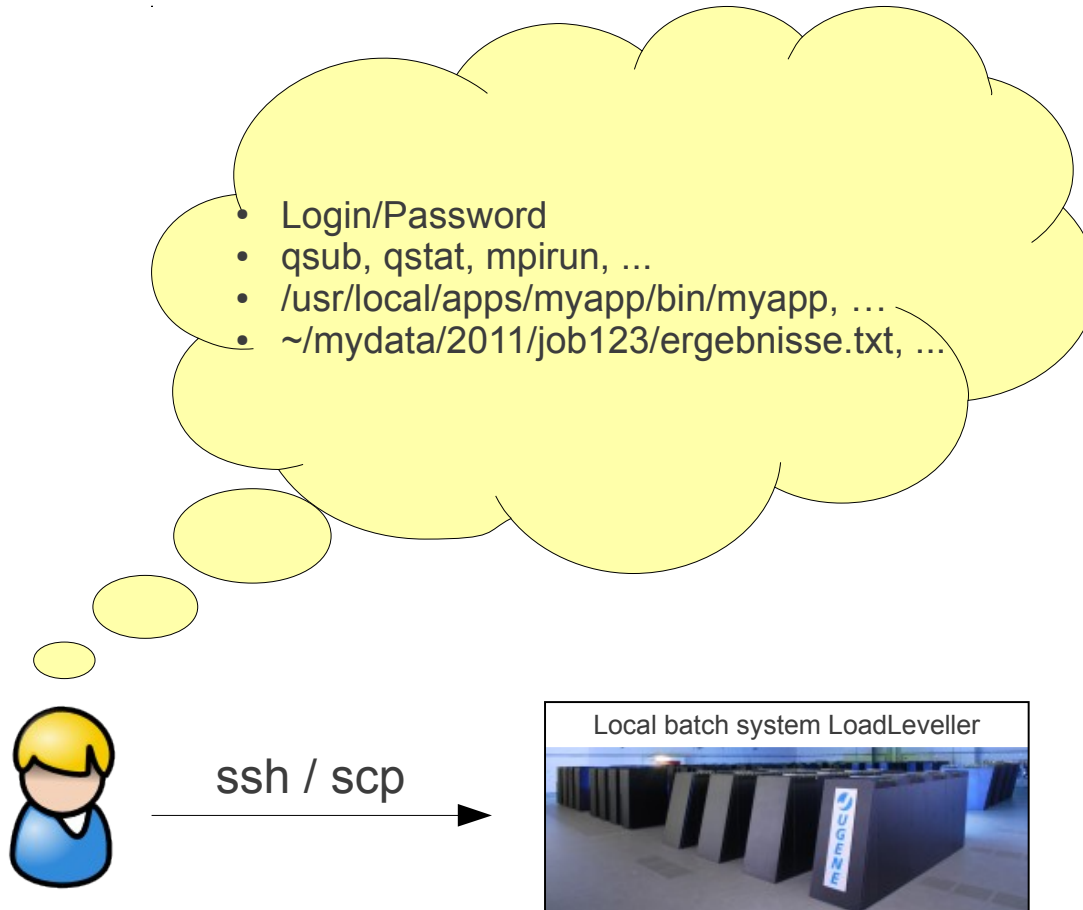


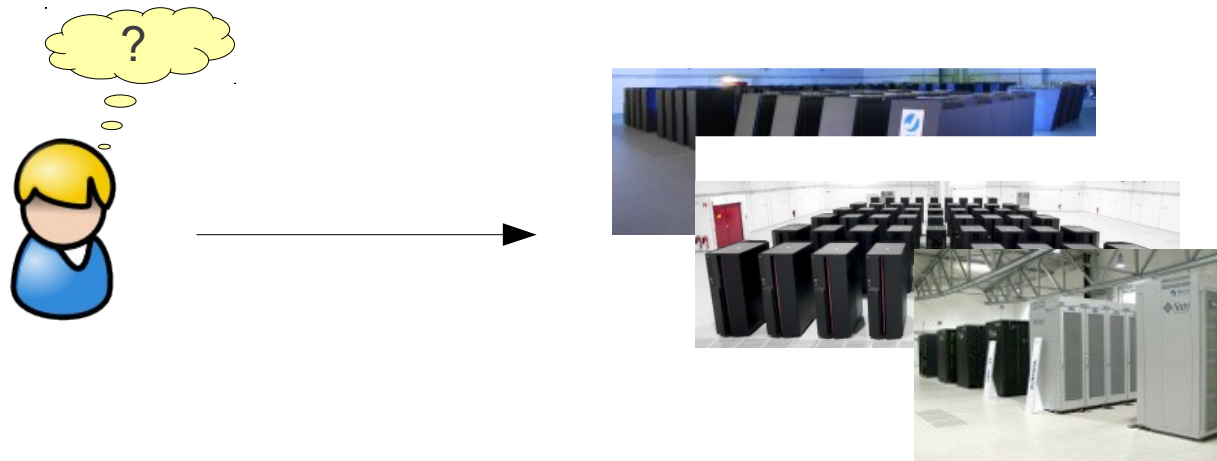
General Introduction to UNICORE

Bernd Schuller and the UNICORE team
Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH
April 1, 2013
UNICORE Tutorial PCT2013, Chelyabinsk

Outline

- UNICORE overview
- Basic services
- Computational jobs and data management
- Workflows
- Clients
- Outlook on future developments



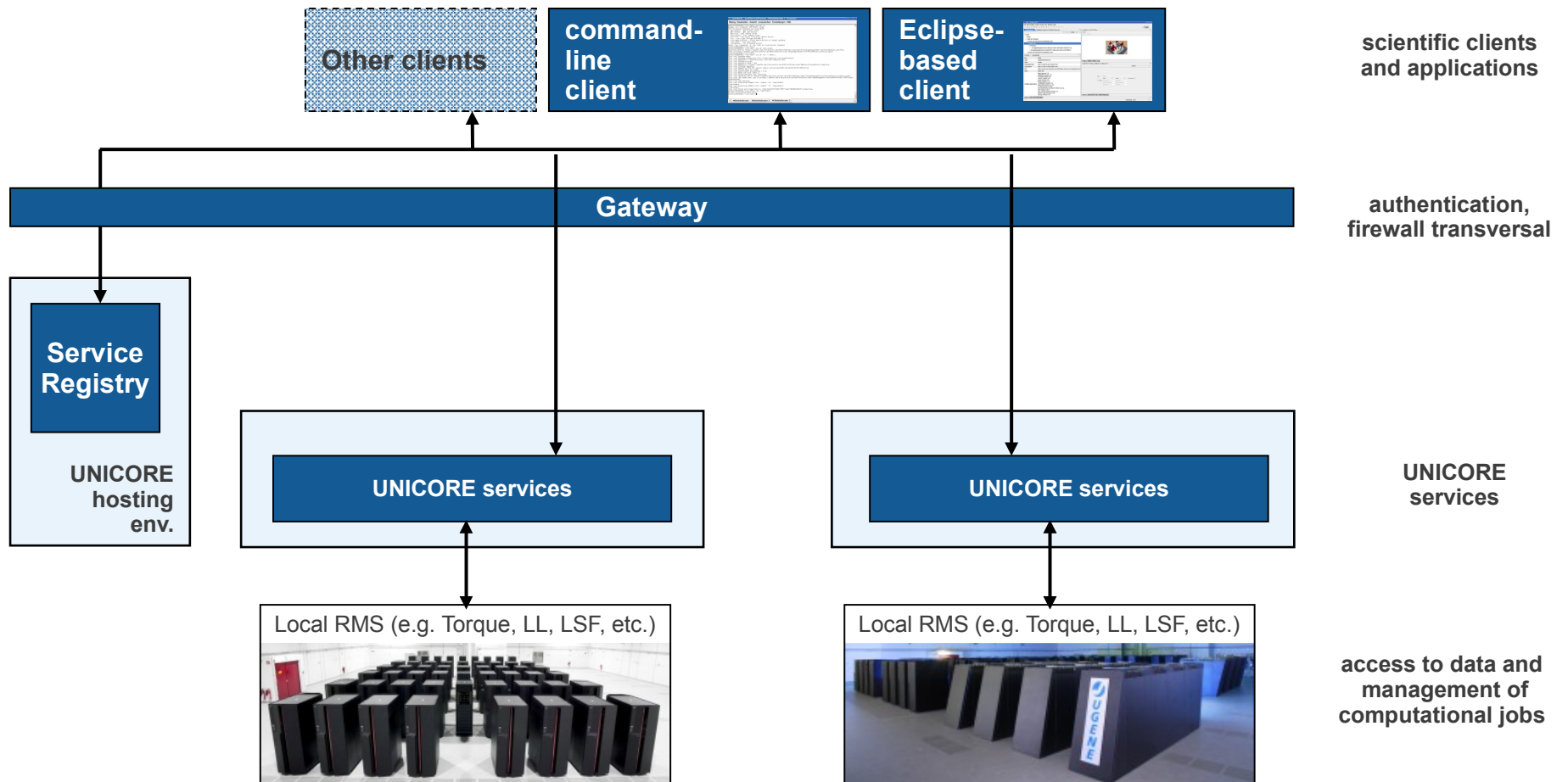


How can I ...

- ... use multiple, heterogeneous systems seamlessly,
- ... manage my job input data and results?
- ... across systems? Workflows?
- This was the original motivation for developing UNICORE (1997)

Integrated, complete Grid middleware stack

- Secure and seamless access to compute and data resources
- Excellent application and workflow support
- Suited for both high performance and high-throughput usage
- Wide variety of clients: GUI, commandline, APIs
- Java/Perl based, supports UNIX, MacOS, Windows and many resource management systems (Torque, Slurm, SGE, ...)
- Easy to install, configure, administrate and monitor
- Active developers, responsive to user wishes ;-), quick and efficient support
- **Open source, BSD licensed, visit <http://www.unicore.eu>**



Services

Basic services

- Job submission and management
 - Single jobs (stage-in, execution, stage-out)
- Storage system access
 - File/directory management (mkdir, rm, ls etc)
 - Access to job working directories
 - Access to admin-defined storages (HOME, TMP, ...)
- File transfer
 - Import/Export to local machine
 - Server-to-server (scheduled, managed)
- Metadata management
- Service registry (per container or shared)

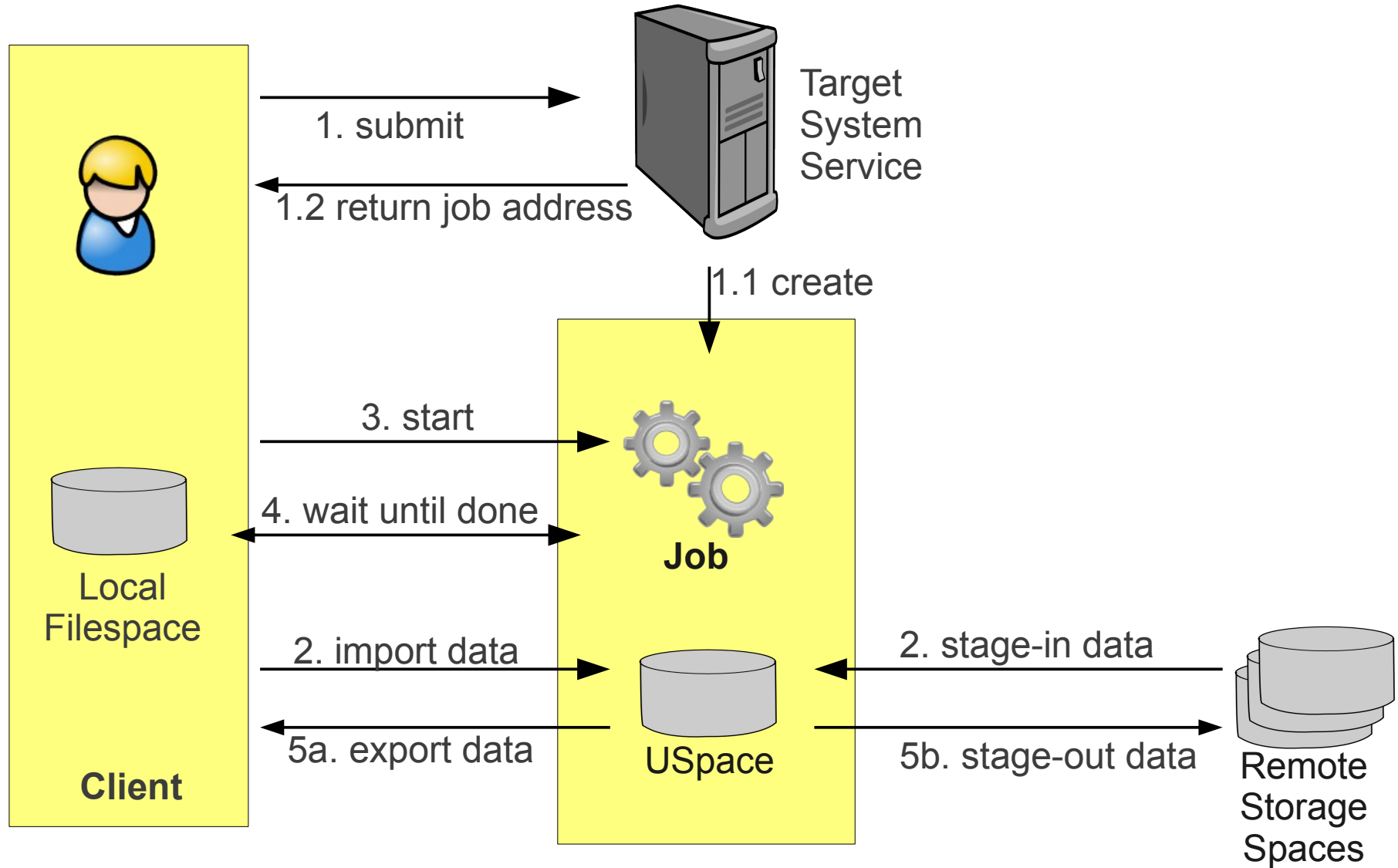
Job model

Job description

JSDL 1.0 (OGF standard)

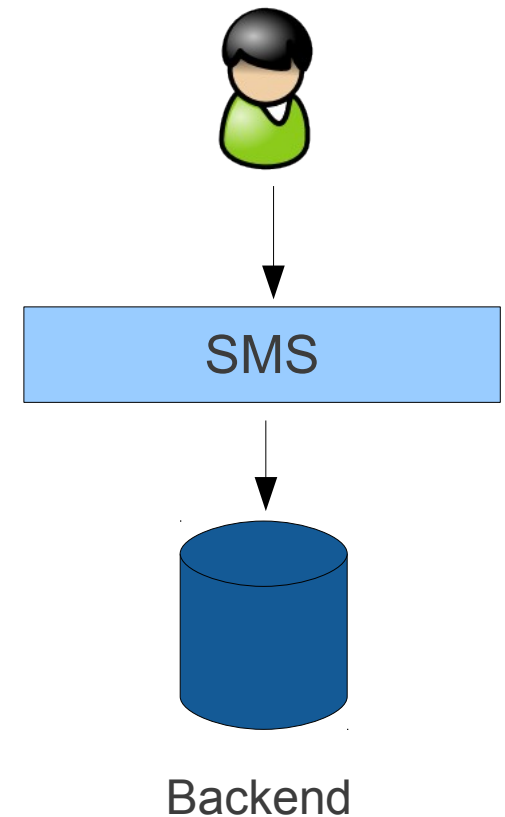
- What to execute
 - Application name / version (mapped to executable by UNICORE)
 - or: Executable path
 - Arguments
 - Environment variables
 - Optional stdin/stdout redirect
- Data staging specification
 - Into job directory from URL
 - From job directory to URL
- Resources requested (number of CPUs, etc)

Job execution



Accessing storage systems

- The UNICORE Storage Management Service („SMS“) provides a filesystem-like view of data
- Typical functions
 - mkdir, delete, ls, chmod etc
- Start tile transfers
 - Import/export of data from/to the user's local machine
 - Send/receive of data from other servers
 - Different supported file transfer protocols

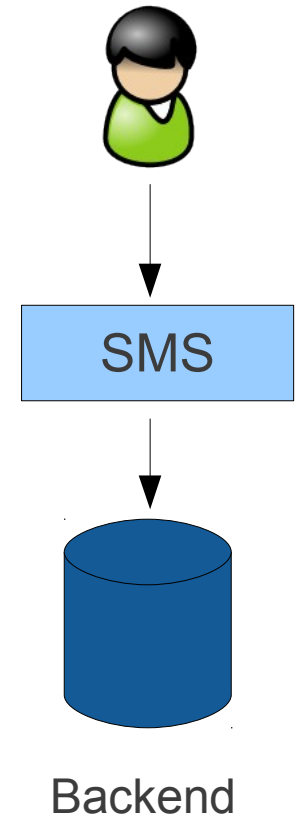


Existing SMS implementations

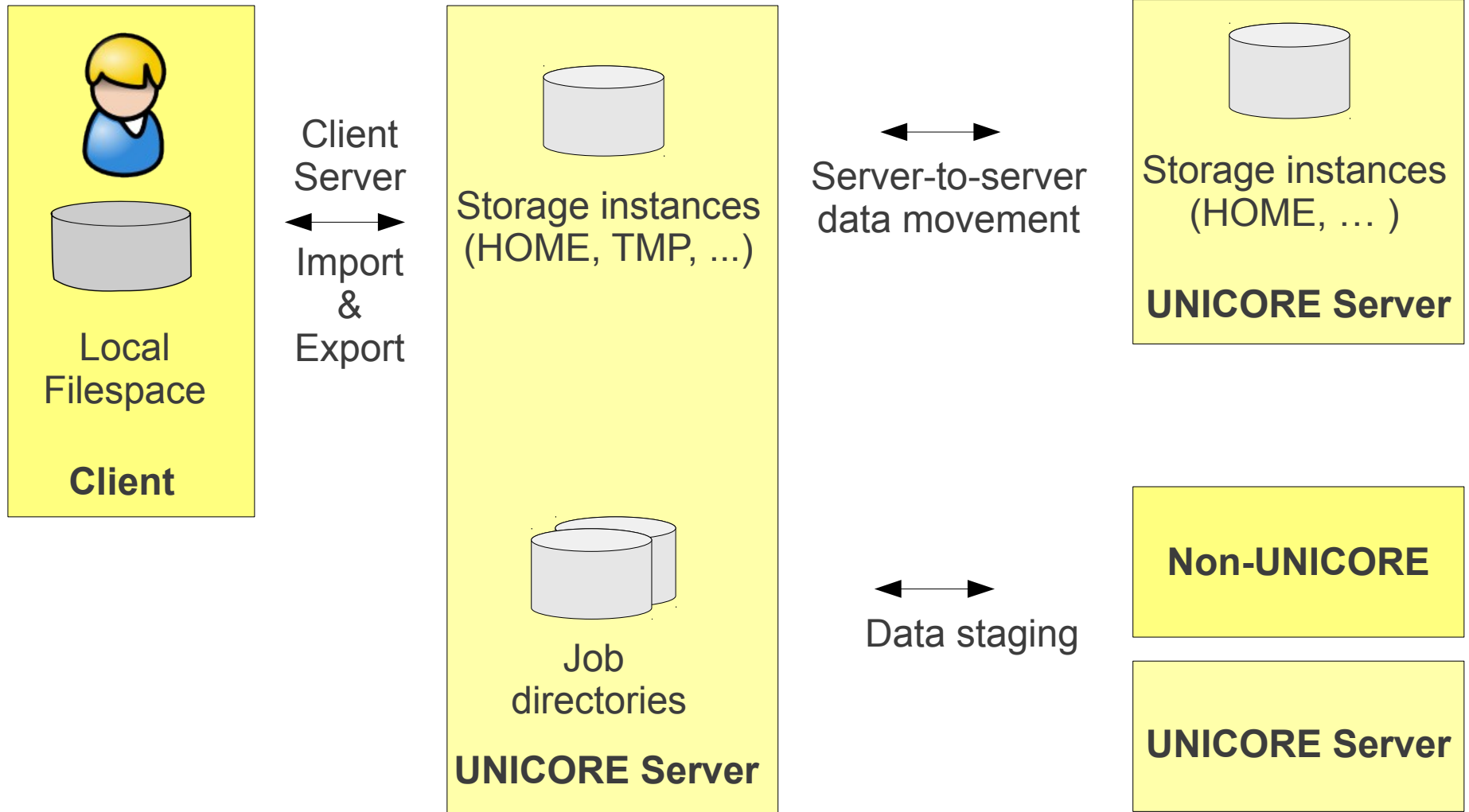
- File system
- Apache HDFS (Hadoop distributed file system)



- iRODS (prototype)



Storages and data movement



File transfer

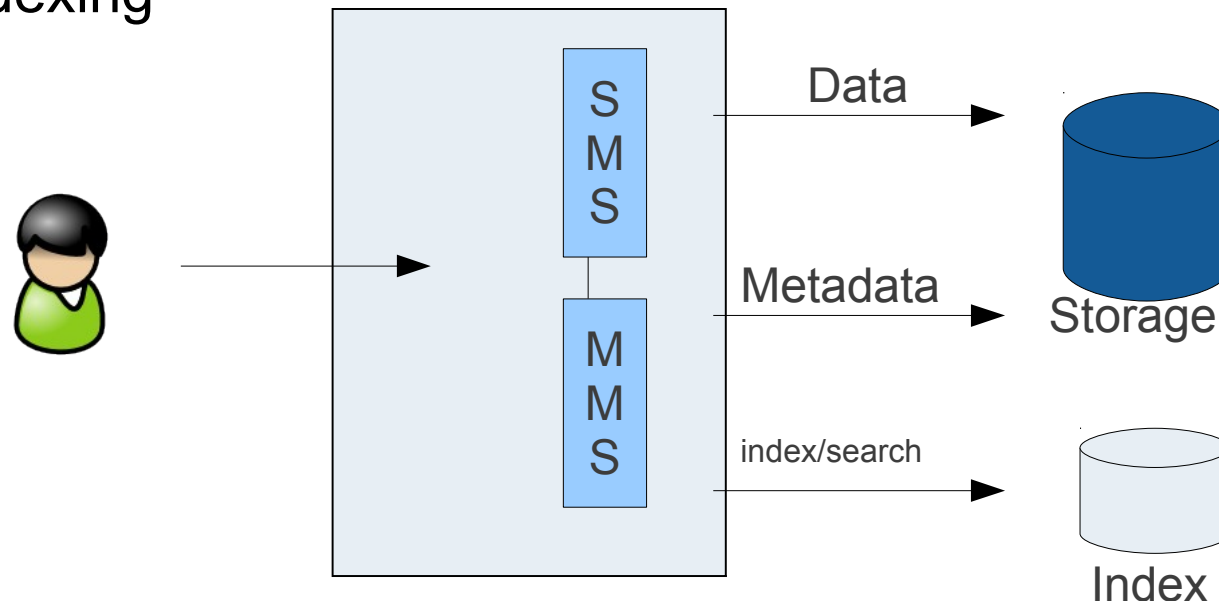
- Both client-to-server and server-to-server FT available
- Builtin: BFT transfer (based on HTTPs)
 - Single open port needed, (almost) full UNICORE security
 - Simple interface (bulk write, read supports byte ranges), fast (several MB/sec.)
- Builtin: OGSA ByteIO (uses SOAP messages)
 - Single port, full UNICORE security
 - Rich interface (POSIX-like, block read/write, etc), slow ~400kB/sec)
- High-performance UFTP solution is available as a separate download

Additional options for data staging

- GridFTP
 - Uses existing globus-url-copy
 - Proxy generated on the client and sent with the job
- Plain HTTP and HTTPS
- FTP and SCP (including client credentials)
- „mailto“ for stage-out :-)

Metadata management

- Decentralized approach: „metadata management service“ (MMS) associated with each storage service („SMS“)
- Schema-free: metadata is key-value pairs
- User can create, edit, delete metadata
- Metadata indexing



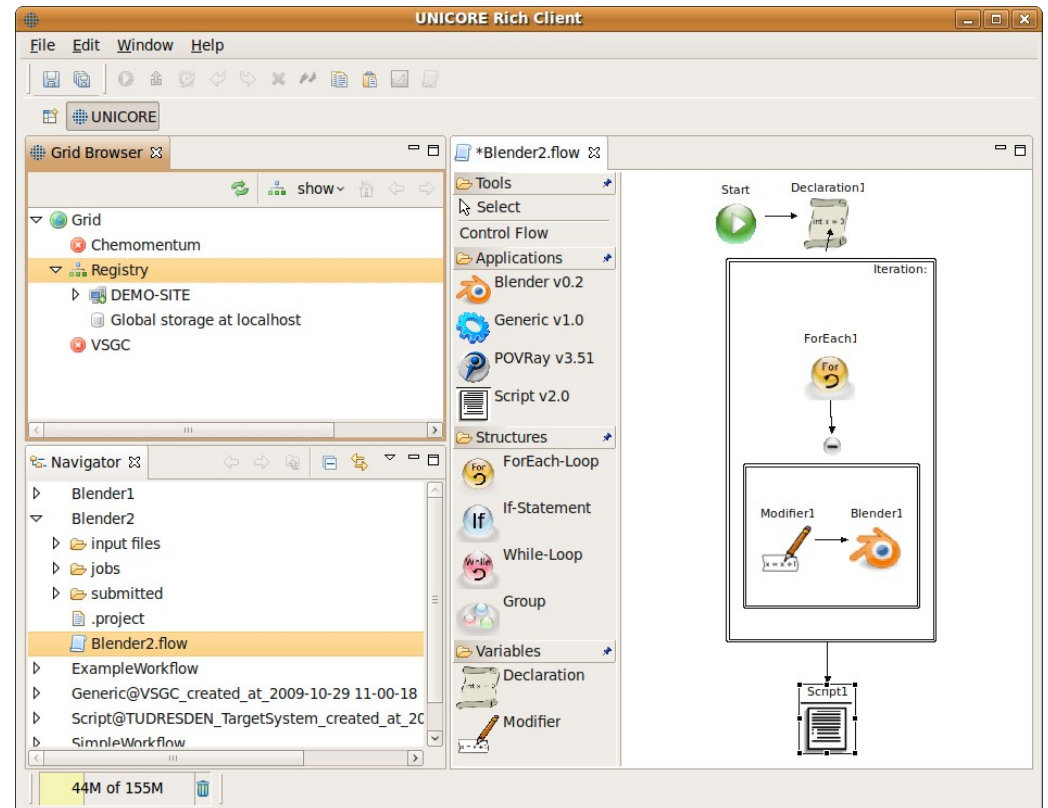
- Metadata storage directly as files on the storage
- Uses well-known open source libraries
- Indexer and search engine: Apache Lucene
- Metadata extraction framework: Apache Tika
- Client tools currently only available in UCC
 - Example: list file properties including metadata

```
schuller@zam994-t400:/$ ucc-vsgc ls -l u6://VSGC-2/Home/Documents/refcard-hadoop.pdf -m
-rw-          1670701 2011-02-24 09:26 /Documents/refcard-hadoop.pdf
{
  "Content-Type": "application/pdf",
  "Creation-Date": "2010-09-23T16:25:05Z",
  "Last-Modified": "2010-09-23T16:25:11Z",
  "created": "Thu Sep 23 18:25:05 CEST 2010",
  "creator": "Adobe InDesign CS5 (7.0)",
  "producer": "Adobe PDF Library 9.9",
  "resourceName": "/Documents/refcard-hadoop.pdf",
  "trapped": "False",
  "xmpTPg:NPages": "6"
}
```

Workflow system

Workflow features

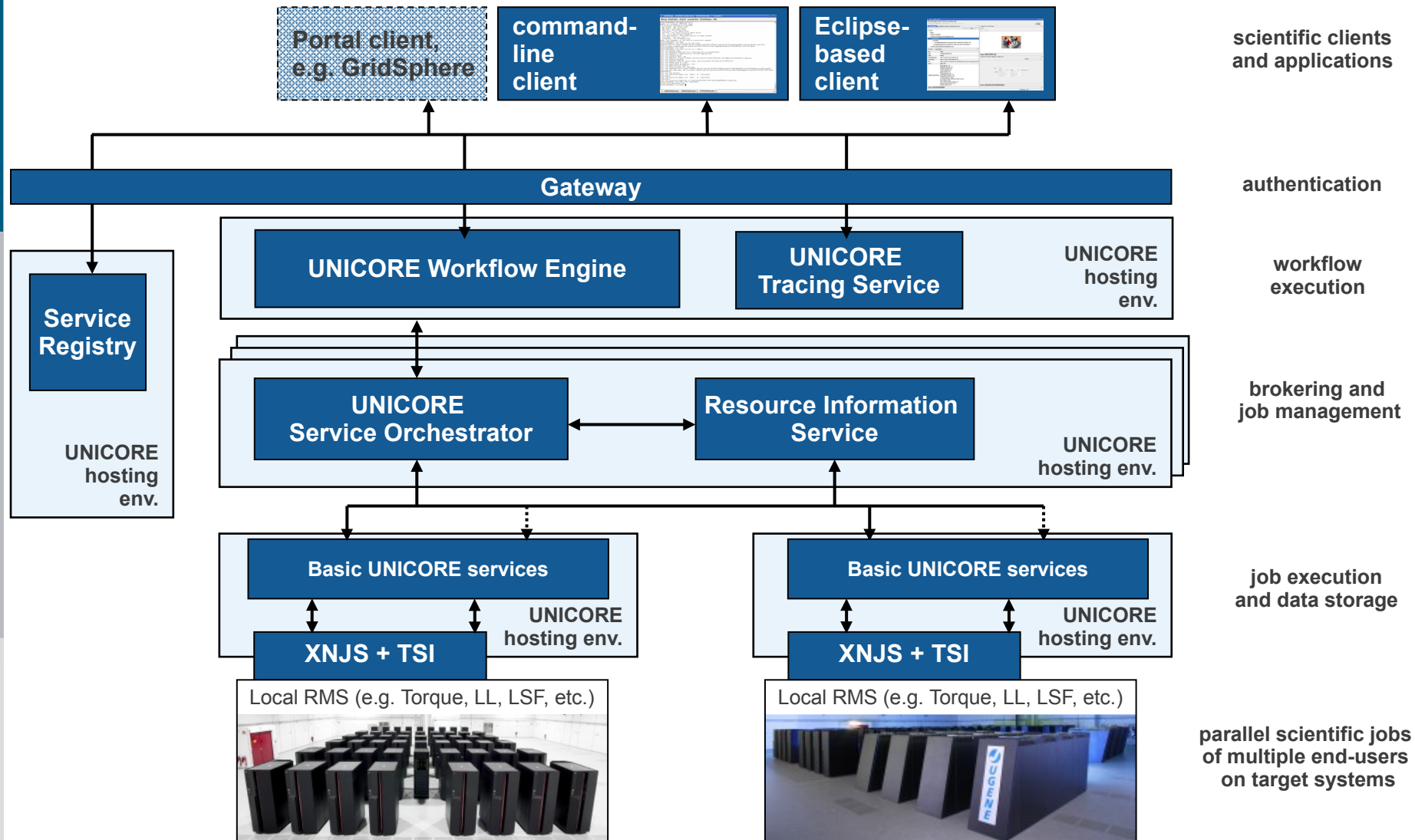
- Simple graphs (DAGs)
- Workflow variables
- Loops and control constructs
 - while, for-each, if-else
- Conditions
 - Exit code, file existence, file size, workflow variables
- Clients
 - UNICORE Rich client
 - Commandline client



The screenshot displays the UNICORE Rich Client interface. On the left, a 'Grid Browser' shows a hierarchy of resources, with 'Workflow engine at zam025s02.zam.kfa-juelich.de' selected. Below it, the 'Properties' panel for a 'ForEach1' task is visible. The 'Iterate over:' dropdown is set to 'Workflow Variable Values'. The 'Files' table lists 'FILE_1' with source type 'Other'. The 'Variable Iterators' section shows a 'counter' variable of type 'Integer' with an initial value of 1 and a value modifier of 'counter+=10;'. The 'While variable value' is set to '<= 100'. On the right, a visual flow diagram for '*Blender2.flow' shows a 'Start' node leading to a 'Declaration1' node (int x = 0). This is followed by a 'ForEach1' loop containing a 'For' node, a 'Modifier1' node (x = x + 1), and a 'Blender1' node. The loop ends with a 'Script1' node.

Iterate over files or variables

Data files can be local or remote



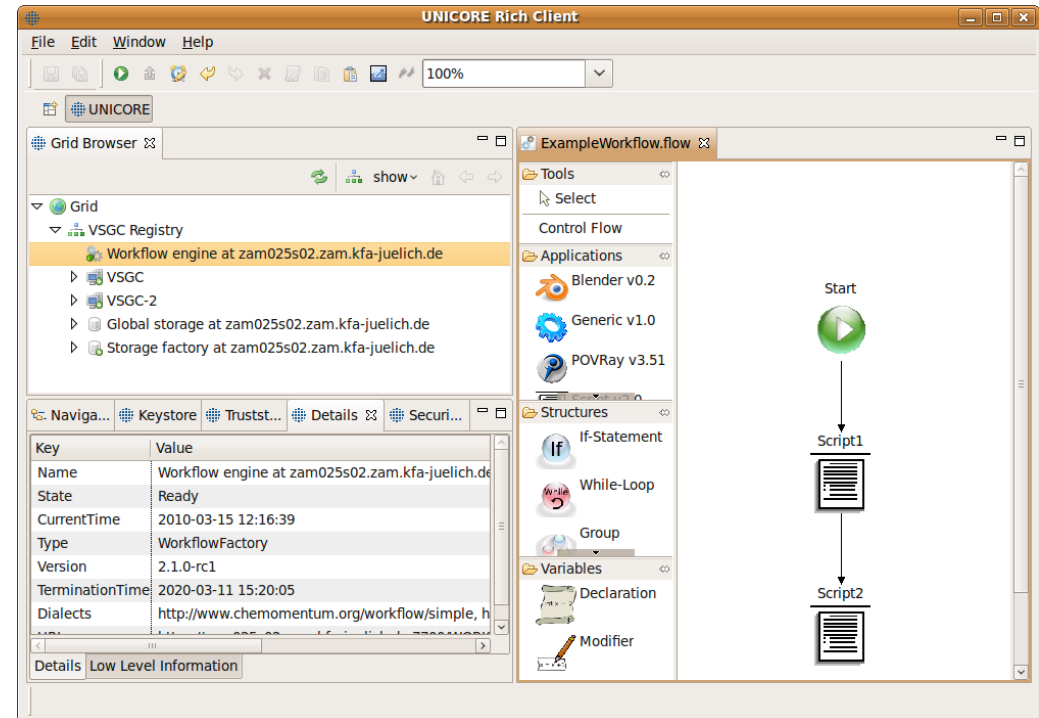
Workflow System summary

- Workflow engine
 - High-level process enactment
 - Pluggable, domain-specific workflow languages
- Service orchestrator
 - Resource brokering based on pluggable strategies
 - Low-level Grid job execution and monitoring
 - Multiple instances can be deployed for scalability
- Tracing service
 - Logs workflow events with timestamps
 - Useful for performance evaluations and reporting

Clients

Rich client

- Building, submitting and monitoring jobs and workflows
- Integrated data and storage management
- Pluggable credentials
- “Simple view” for novice users
- Based on the Eclipse framework
- Extensibility through plug-ins
 - Examples: Shibboleth, CIS view, Admin Interfaces (UVOS, XUADB, dynamically deploy/undeploy services), etc.
- Installation/update mechanism for plug-ins and GridBeans



The screenshot displays the UNICORE Rich Client interface. The main window is titled "UNICORE Rich Client" and features a menu bar (File, Edit, Window, Help) and a toolbar with various icons and a zoom level of 100%. The interface is divided into several panes:

- Grid Browser:** Shows a tree view of the grid resources. The selected item is "Workflow engine at zam025s02.zam.kfa-juelich.de". Other visible items include "VSGC", "VSGC-2", "Global storage at zam025s02.zam.kfa-juelich.de", and "Storage factory at zam025s02.zam.kfa-juelich.de".
- Workflow Editor:** The main workspace shows a workflow diagram for "ExampleWorkflow.flow". The workflow starts with a "Start" node (a green play button), followed by "Script1" and "Script2" nodes, each represented by a document icon. Arrows indicate the flow from Start to Script1 and then to Script2.
- Toolbox:** Located on the right side, it contains several categories of workflow elements:
 - Tools:** Select
 - Control Flow:** (empty)
 - Applications:** Blender v0.2, Generic v1.0, POVRay v3.51
 - Structures:** If-Statement, While-Loop, Group
 - Variables:** Declaration, Modifier
- Details Panel:** Located at the bottom, it shows a table of key-value pairs for the selected workflow engine.

Key	Value
Name	Workflow engine at zam025s02.zam.kfa-juelich.de
State	Ready
CurrentTime	2010-03-15 12:16:39
Type	WorkflowFactory
Version	2.1.0-rc1
TerminationTime	2020-03-11 15:20:05
Dialects	http://www.chemomomentum.org/workflow/simple, h

Integrated storage management in the UNICORE Rich client Grid browser

- Create files
- Drag and drop from/to desktop environment
- Copy and paste
- Remote file editing

The screenshot displays the UNICORE Rich Client interface. The main window is titled "UNICORE Rich Client" and contains several panes:

- Grid Browser:** Shows a hierarchical view of the file system. The path is: Grid > VSGC > Workflow engine at zam025s02.zam.kfa-juelich.de > VSGC > VSGC-2 > VSGC-2_TargetSystem > VSGC-2_TargetSystem Home. A context menu is open over the "VSGC-2_TargetSystem Home" folder, listing actions such as "add Registry", "create file", "create directory", "destroy", and "refresh".
- Navigator:** Shows a tree view of the current project, including folders like "input files", "jobs", "submitted", and "project", and a file named "Blender2.flow".
- Workflow Editor:** Displays a visual workflow diagram. It starts with a "Start" node, followed by a "Declaration1" node (containing "int x = 3"). This leads to an "Iteration:" block containing a "ForEach1" loop. Inside the loop, there is a "Modifier1" node (containing "x = x + 1") and a "Blender1" node (containing the Blender logo). The workflow ends with a "Script1" node.
- Tools Panel:** Located on the right, it lists various tools and applications, including "Blender v0.2", "Generic v1.0", "POVRay v3.51", and "Script v2.0".

Client Plugins / GridBeans

- Application specific graphical interface
 - Generates job description from user input
 - Provides graphical user interface for input data
 - Provides graphical user interface for output data
- Consist of
 - Job description generation code
 - One or more user interface modules
- Developer's Guide:
<http://www.unicore.eu/documentation/manuals/GridbeanDevelopersGuide.pdf>

Example: Rich client extensions

Display results for the MOPAC application

The screenshot shows a web-based interface for MOPAC results. The main window has a title bar with 'Navigator', 'Grid Browser', and 'Output for Job MOPAC1'. Below the title bar are buttons for 'Save as' and 'Export structures'. A navigation bar includes a '#' button and the text 'Showing results 1-24 of 24'. The main content is a table with the following data:

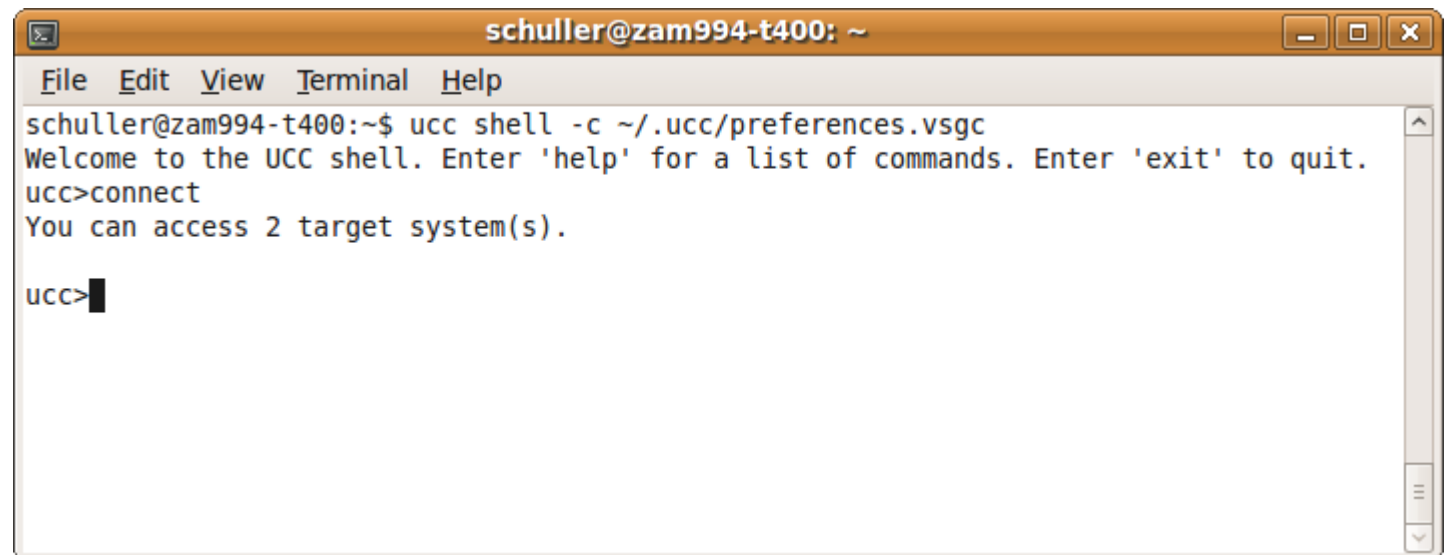
Structure Id	3D structure	MOPAC output
2	View	View
3	View	View
4	View	View
6	View	View
7	View	View
8	View	View
9	View	View
10	View	View
11	View	View
12	View	View
14	View	View
15	View	View

At the bottom of the main window are buttons for 'Calculation summary', 'Visualise structures', 'Stdout', 'Stderr', and 'Log'. A secondary window titled 'Structure (ID=11)' displays a 3D ball-and-stick model of the molecule, showing a complex organic structure with various atoms represented by different colors (grey, white, red, blue, yellow, green).

UNICORE 6 commandline client (UCC)

Provides access to all the UNICORE functionality from the commandline

- Over 50 commands from all areas
- Scriptable in the Groovy programming language
- Extensible with custom commands
- Shell autocomplete, Emacs integration

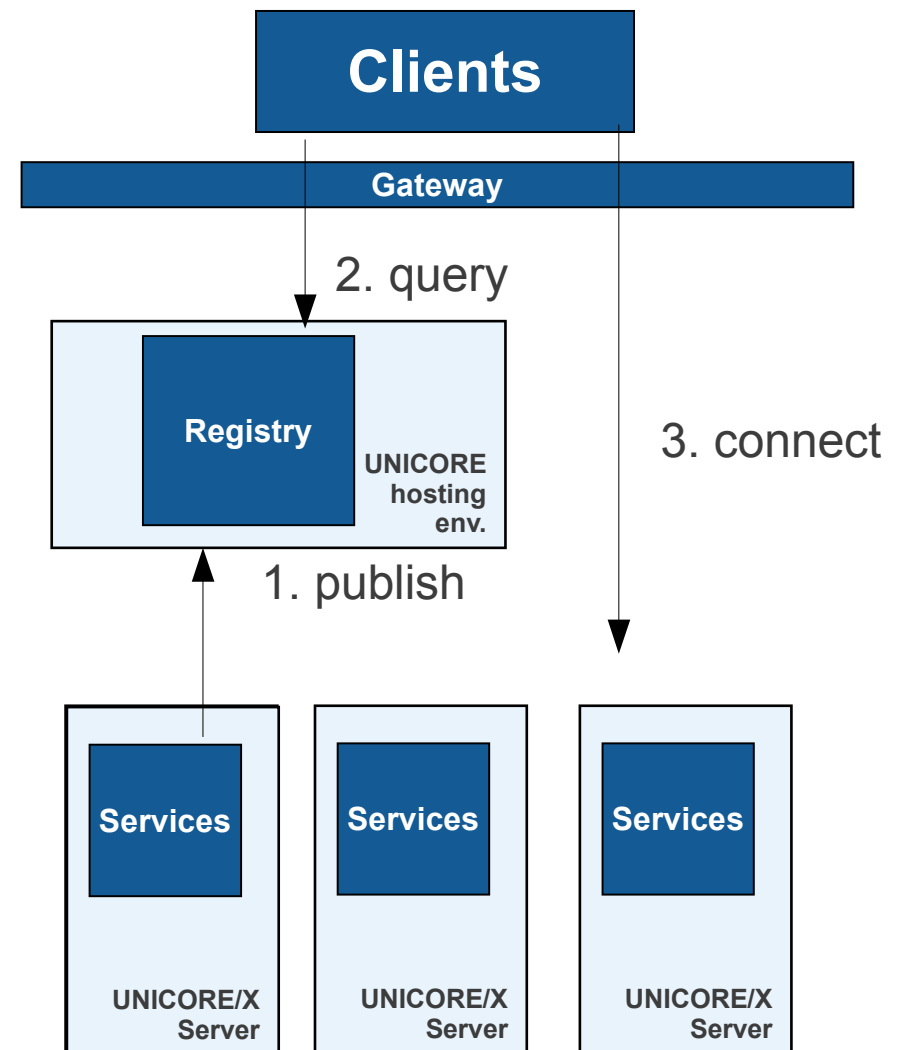


```
schuller@zam994-t400: ~  
File Edit View Terminal Help  
schuller@zam994-t400:~$ ucc shell -c ~/.ucc/preferences.vsgc  
Welcome to the UCC shell. Enter 'help' for a list of commands. Enter 'exit' to quit.  
ucc>connect  
You can access 2 target system(s).  
ucc>
```

Information systems

UNICORE Registry („global registry“)

- Aggregates service information from multiple UNICORE sites
- An Entry contains at minimum:
 - Service address
 - Service type
 - Server DN
- Content is held up-to-date
 - Services register with local registry
 - Local registries push info to global registry
 - Entries in global registries expire
 - ...if not re-published in time



Outlook to future developments

Outlook

- Strong(er) focus on data features
 - Data oriented processing
 - Data-aware scheduling
 - More metadata features
- Simpler usage
 - Web portal under heavy development
 - Certificate-less access via SAML authentication
- Better integration of virtual systems / clouds
 - Manage user-defined VM images

Thank you!