Parallel jobs in UNICORE 6

Dr. Bernd Schuller
Jülich Supercomputing Centre

Outline

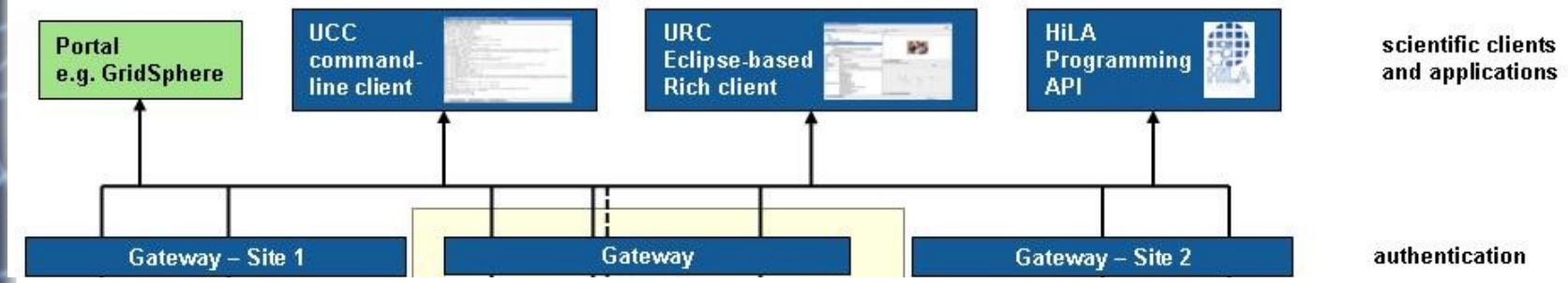
- UNICORE 6 overview and architecture
- Parallel jobs in UNICORE 6
  - Batch job generation
  - Job description
  - Execution environments
  - User side

UNICORE 6 Overview

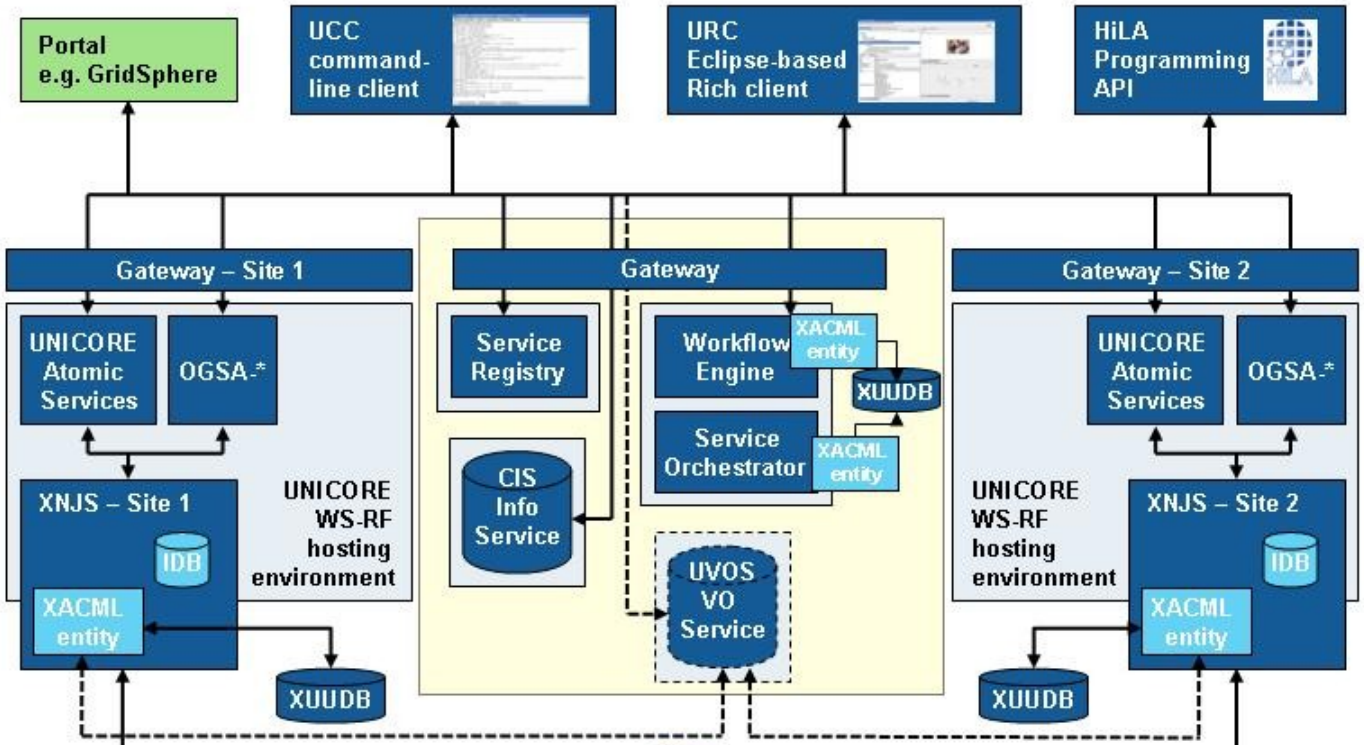
- Integrated, complete Grid middleware stack including graphical & commandline clients
- Focus on ease of use (both end users and admins)
- Lightweight: only Java + Perl
- Supports many resource management systems and operating systems (Linux/Unix, Mac OS X, Windows)
- Strong support for applications and workflows

UNICORE 6 Architecture

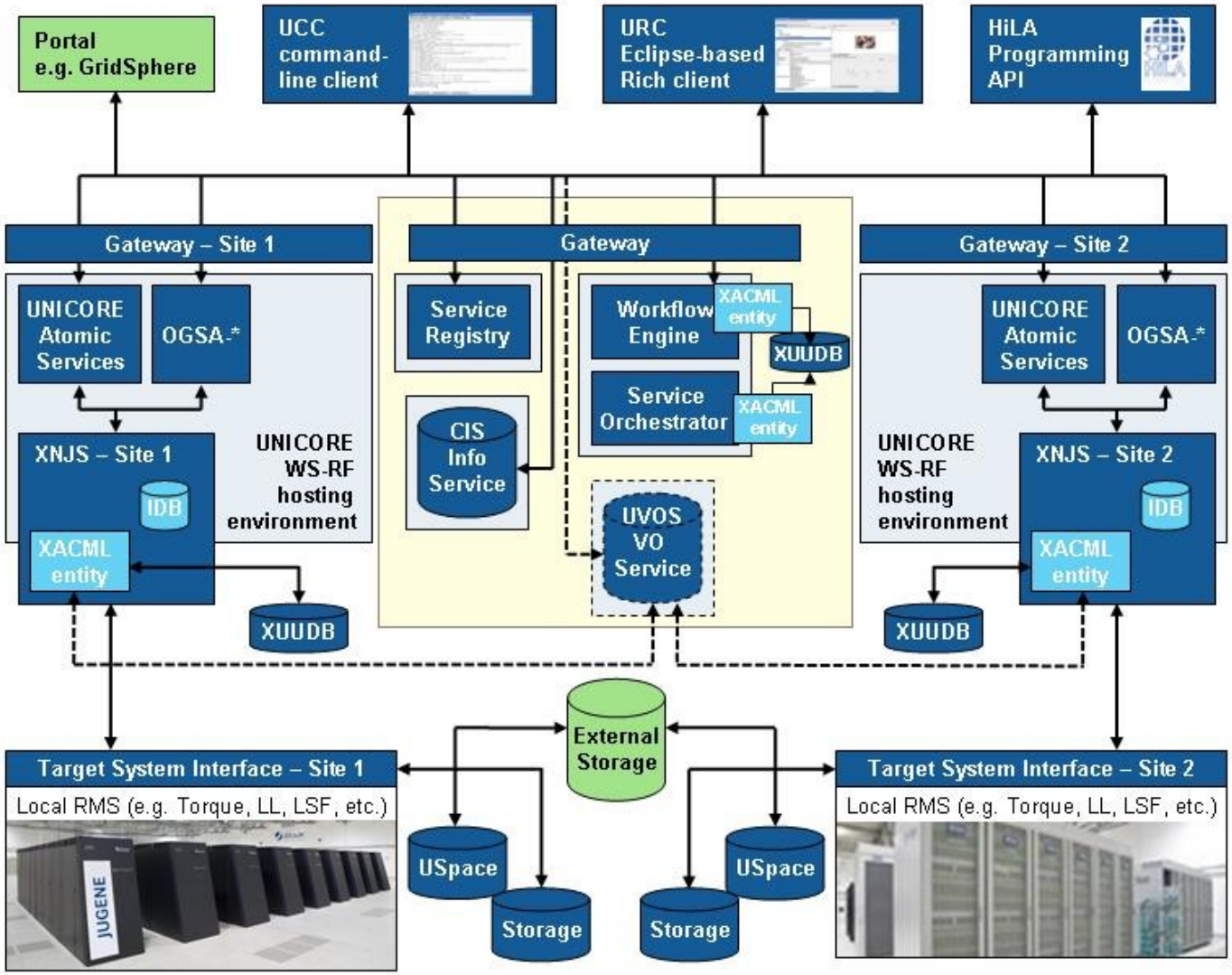
UNICORE 6 Architecture



UNICORE 6 Architecture



UNICORE 6 Architecture



scientific clients and applications

authentication

emerging standard interfaces

Grid services hosting

job incarnation & authorisation

parallel scientific jobs of multiple end-users on target systems

UNICORE 6

Parallel jobs in UNICORE 6: A bit of history...

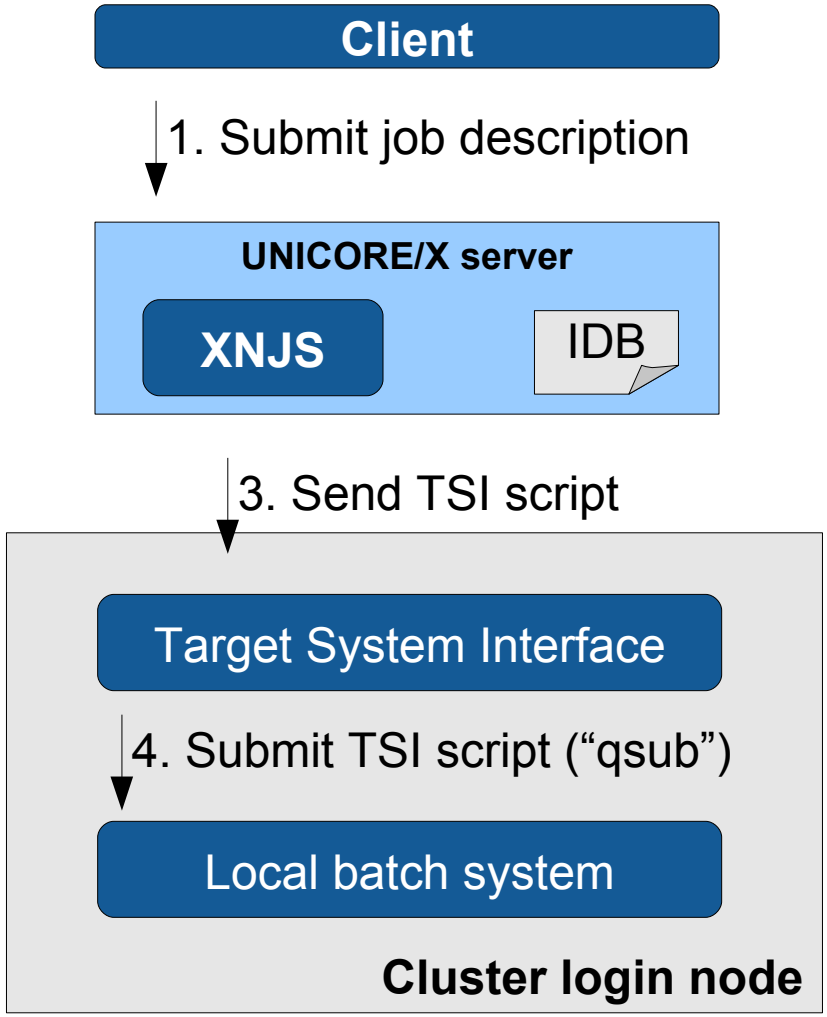
- UNICORE was initiated in 1997 as a means for simplifying and unifying access to German supercomputer centers (in Jülich, Stuttgart, Munich, ...)
- Back then, almost every job was a parallel job (excluding e.g. compile jobs), so it was necessary to specially mark serial jobs :-)
- Nowadays, systems and jobs are much more heterogeneous, and UNICORE's parallel job support has changed to reflect that

Parallel jobs

- Batch job generation process
- Job description
- Execution environments
- Clients

EMI INFSO-RI-261611

UNICORE 6
Batch job generation process



2. The XNJS generates a TSI script from the job description and local configuration stored in IDB file

UNICORE 6
Job description

- JSDL 1.0 (OGF standard)
  - Application name / version (mapped to executable by UNICORE) or executable
  - Arguments, environment variables, in/out/err redirects
  - Data staging specification
  - Resources requested (number of nodes, number of CPUs, etc)
- UNICORE-specific extension
  - Execution environment
- "Standardized" JSDL extension: SPMD
  - Covers parallel applications
  - Mapped to execution environments

Execution environments: motivation

- Many different parallel environments
- (MPI, OpenMP, Hybrid, ...) and implemenations exist
- Different execution modes (debug, testing, production, ...)
- Additional configuration parameters
- Needed: abstraction, nice user interface

UNICORE 6

Execution environments

- Administator
  - Knows available tools and how to set them up
  - Configures available options
- User
  - Selects execution environment
  - Chooses parameters and options
  - Customises via pre/post commands

UNICORE 6
Simple example: IDB entry

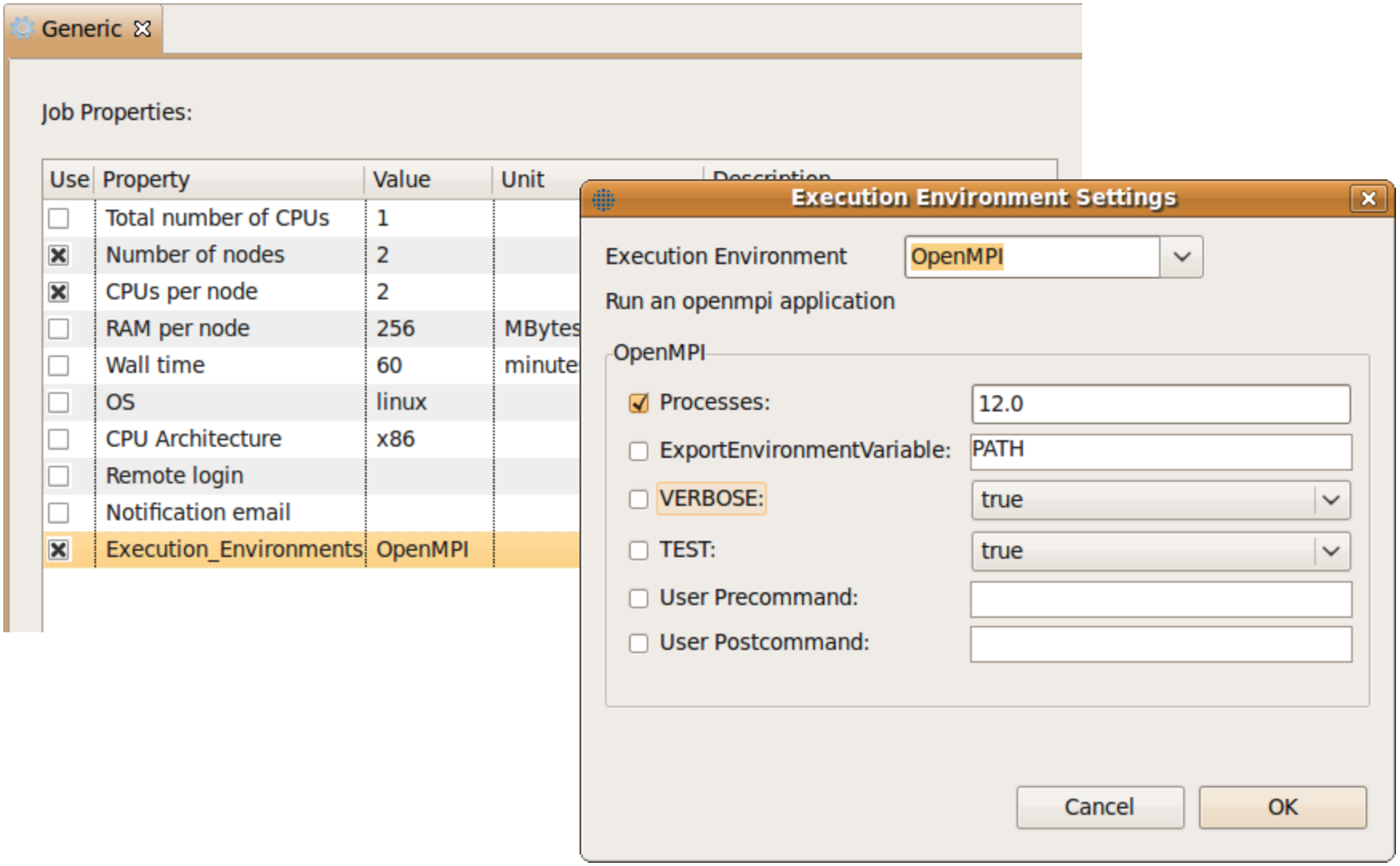
```
<ee:ExecutionEnvironment xmlns:ee="...">
  <ee:Name>OpenMPI</ee:Name>
  <ee:Version>1.0</ee:Version>
  <ee:ExecutableName>
   /vsgc/software/openmpi/bin/mpiexec
  </ee:ExecutableName>

  <ee:Argument>
    <ee:Name>Processes</ee:Name>
    <ee:IncarnatedValue>-np </ee:IncarnatedValue>
    <ee:ArgumentMetadata>
      <ee:Description>The number of processes</ee:Description>
      <ee:Type>double</ee:Type>
      <ee:ValidValue>[1,8192]</ee:ValidValue>
    </ee:ArgumentMetadata>
  </ee:Argument>

  <ee:Option>
    <ee:Name>VERBOSE</ee:Name>
    <ee:IncarnatedValue>-v</ee:IncarnatedValue>
  </ee:Option>
</ee:ExecutionEnvironment>
```

UNICORE 6

Example: commandline client job file

```
{

  Executable: "./hello.mpi",

  Imports: [
    {From: "/myfiles/hello.mpi", To: "hello.mpi" },
  ],

  Resources:{ CPUsPerNode: 2, Nodes: 2, },

  Execution environment: {
    Name: OpenMPI,
    Arguments: { Processes: 12,  },
  },

}
```

UNICORE 6
Example: rich client



Example: generated TSI script (sent to TSI and executed by the batch system)

```
#!/bin/sh
#TSI_SUBMIT
#TSI_TIME 3600
#TSI_MEMORY 256
#TSI_NODES 2
#TSI_PROCESSORS_PER_NODE 2
#TSI_TOTAL_PROCESSORS 4
#TSI_HOST_NAME none
#TSI_QUEUE batch
#...

/vsgc/software/openmpi/bin/mpiexec -np 12 ./hello.mpi
```

Example: generated TSI script

```
#!/bin/sh
#TSI_SUBMIT
#TSI_TIME 3600
#TSI_MEMORY 256
#TSI_NODES 2
#TSI_PROCESSORS_PER_NODE 2
#TSI_TOTAL_PROCESSORS 4
#TSI_HOST_NAME none
#TSI_QUEUE batch
#...

/vsgc/software/openmpi/bin/mpiexec -np 12 ./hello.mpi
```

Summary

- UNICORE 6 execution environments provide user-friendly way to run parallel jobs without the need to know site specific details
- Behind the scenes
  - Administrator configures site-specific details and UNICORE publishes them to Grid clients
  - Batch job generation is customised through execution environments

Thank you!

EMI is partially funded by the European Commission under Grant Agreement RI-261611