

Using Services and Service Compositions to Enable the Distributed Execution of Legacy Simulation Applications

Mirko Sonntag¹, Sven Hotta¹, Dimka Karastoyanova¹,
David Molnar², Siegfried Schmauder²

¹ Institute of Architecture of Application Systems (IAAS)
{sonntag, hotta, karastoyanova}@iaas.uni-stuttgart.de

² Institute of Materials Testing, Materials Science and Strength of Materials
{david.molnar, siegfried.schmauder}@imwf.uni-stuttgart.de

University of Stuttgart
Universitaetsstrasse 38
70569 Stuttgart
Germany



Phone +49 (0)711 685 88 202

Fax +49 (0)711 685 88 472

Agenda

- Introduction to scientific workflows
- Use case: simulation of solid bodies
 - Former situation
 - Implementation with services and BPEL
- Simulation environment
 - Architecture
 - Need for resource management
 - Need for data management
 - Providing Fortran applications as Web services
- Conclusion and outlook

Workflow Technology in Simulation

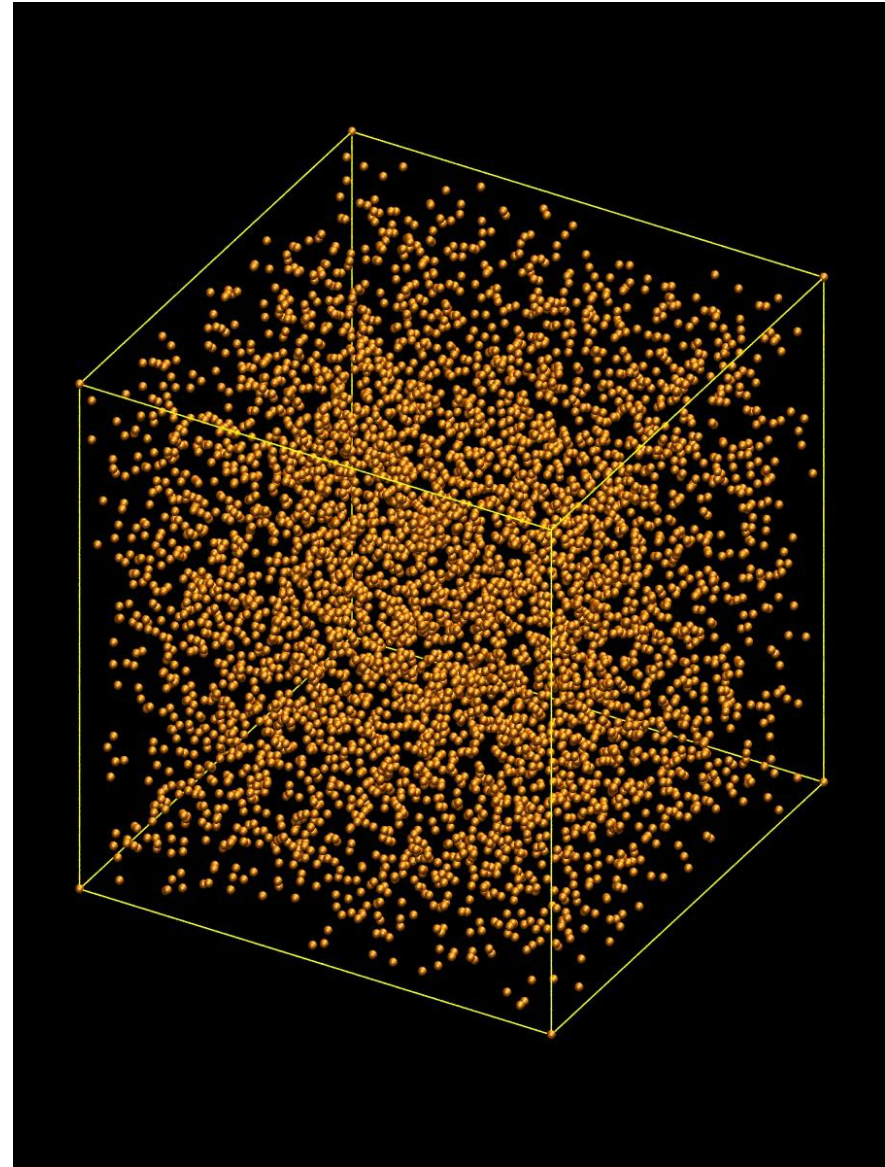
- Our major goal is to build a Workflow System that
 - uses services to implement scientific simulations and experiments
 - From experiment development through post-processing
 - meets general requirements
 - Robustness
 - Scalability
 - ...
 - meets special requirements given by the scientific domain
 - **Automation** as much as possible
 - Deployment, discovery, visualization, ...
 - Handling huge amounts of **data**
 - **Reuse** domain-specific algorithms
 - Support of **flexible**, evolving workflows
 - Support of **multi-scale and multi-physics simulations**

Workflows for e-Science

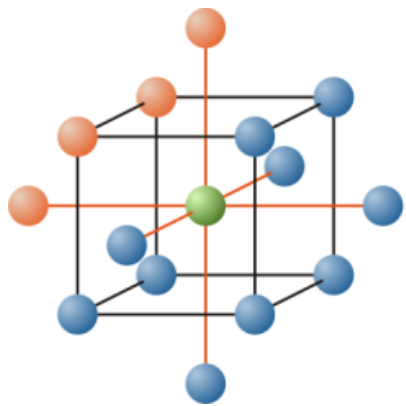
- Workflows are used in e-Science
 - Note, this is not (always) the conventional workflow technology
- Examples of existing workflow systems for e-Science:
 - Kepler
 - Taverna
 - Triana
 - Pegasus
 - Microsoft Trident
 - e-BioFlow
 - WASA
 - SEGL
 - Simulink
- Most of these workflow systems can be used for simulations

Use Case: Simulation of Ageing Process of Steel

- Atom lattice of copper-alloyed steel
- Single vacancy in the lattice
- In each Monte Carlo step, the vacancy changes the position with a randomly selected atom
- Atom clusters form over many time steps
- They negatively influence the material properties

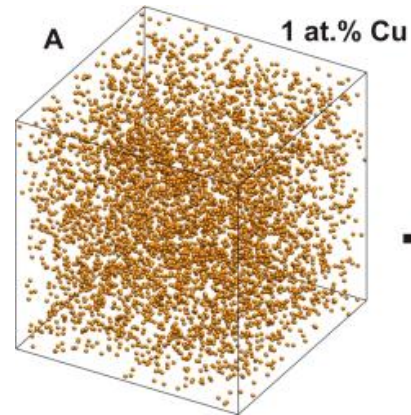


Use Case: Simulation of Ageing Process of Steel (2)

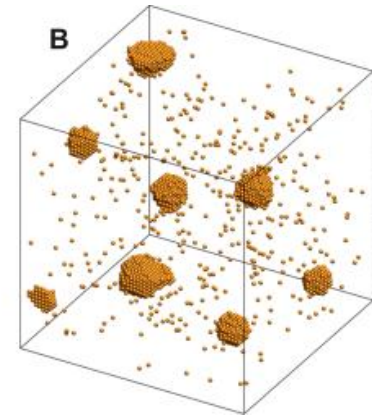


- vacancy
- iron atoms
- copper atoms

a)

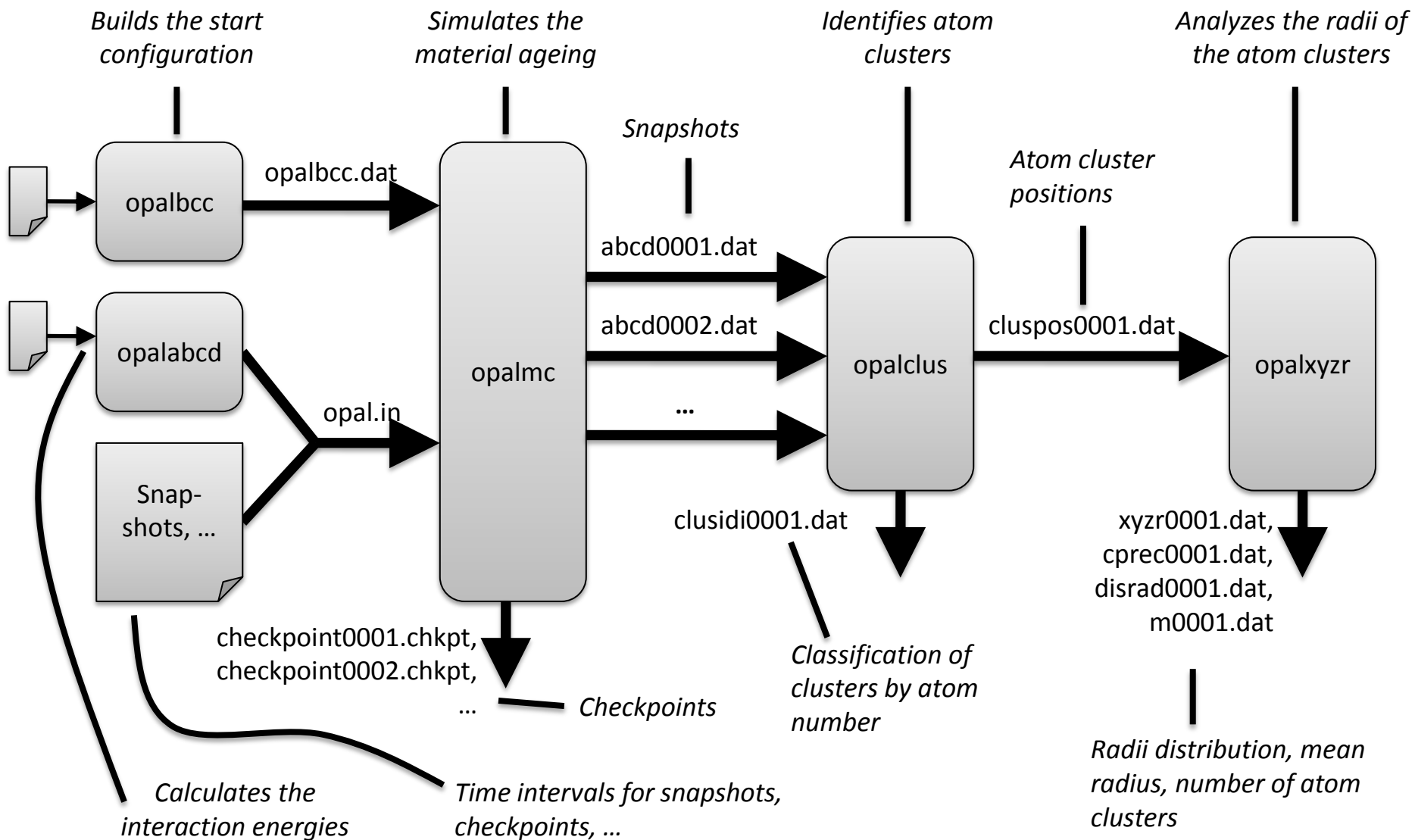


thermal ageing

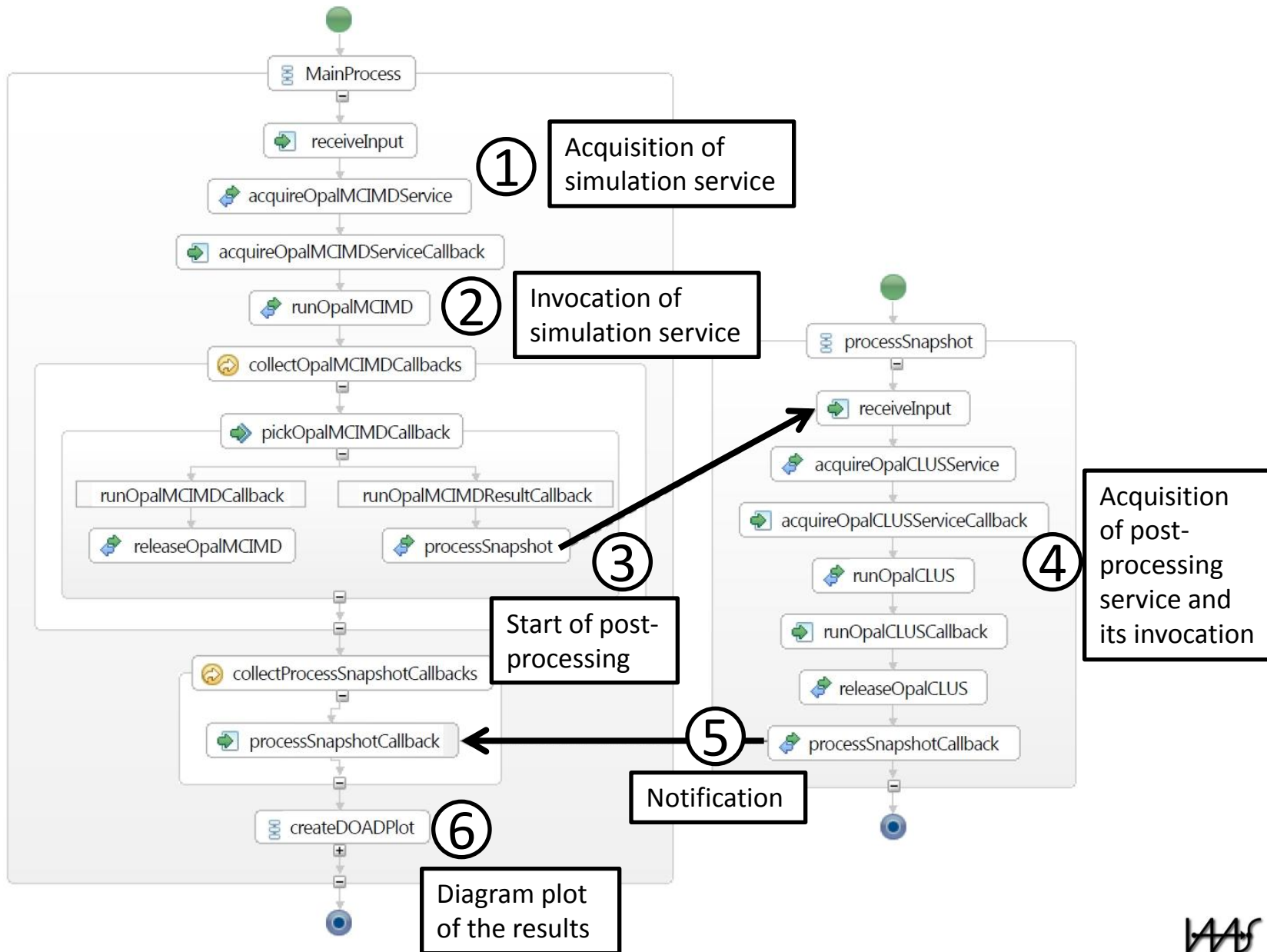


b)

Former Situation in Opal



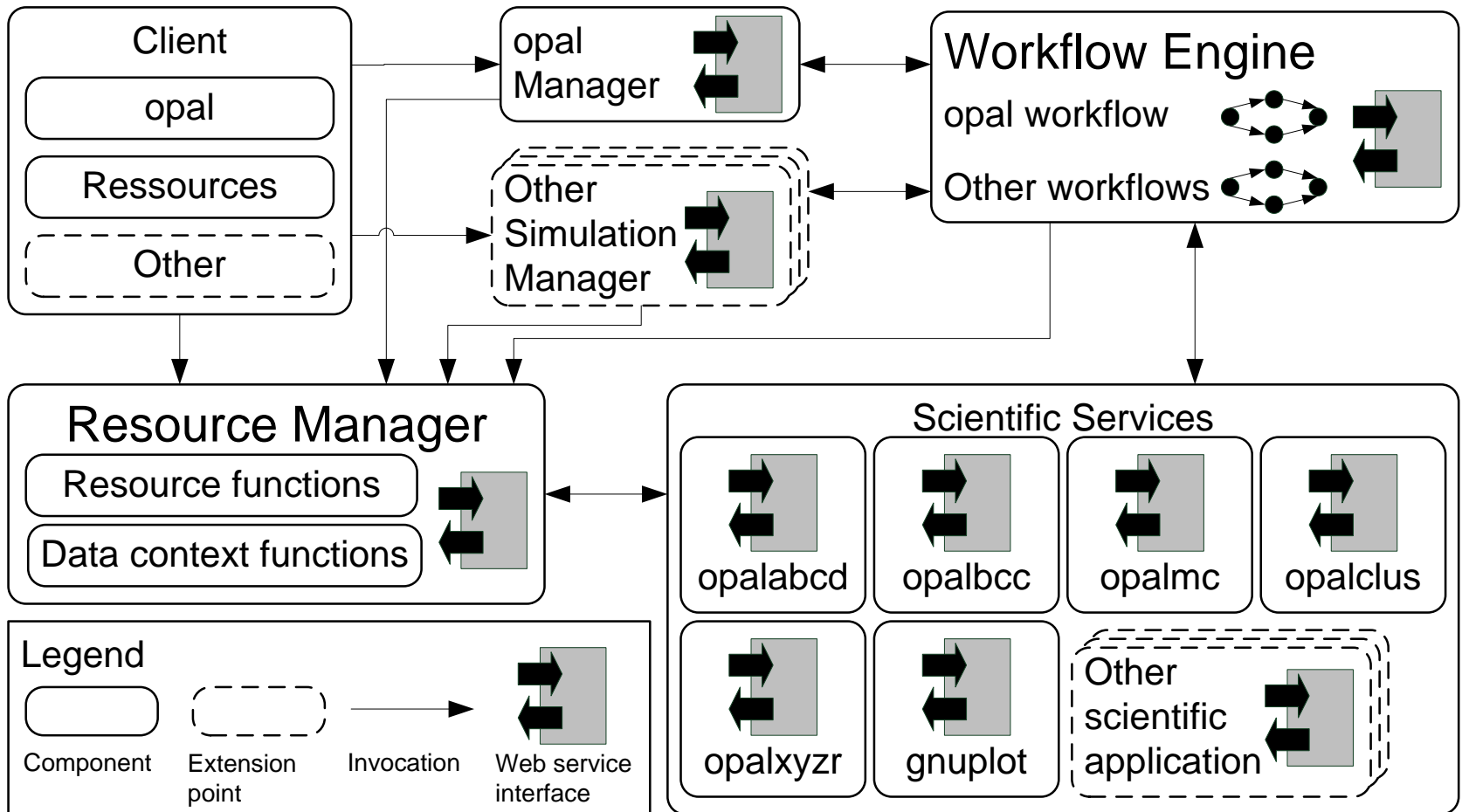
BPEL Processes to Implement the Simulation



How Services And Service Compositions Can Help

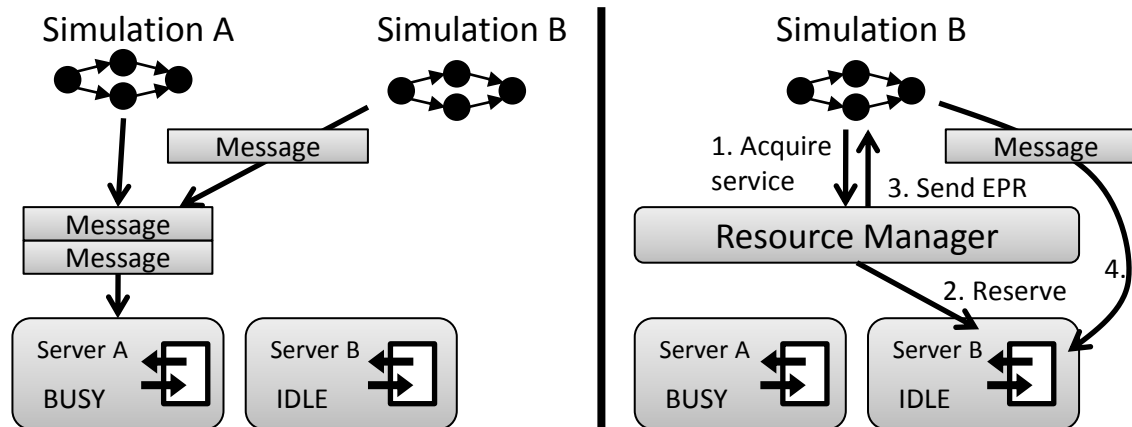
- Distributed computing
- Graphical programming/programming in the large
- Automation of manual steps
- Robustness by fault handling and compensation features
- A complete re-engineering of the tool is often not needed
- Documentation of the simulation logic
- Documentation of simulation runs via audit trail
- Monitoring of simulations
- Commodity hardware can be used

Architecture of the Service-Based Simulation Environment



Need for a Resource Manager

- Scientific services can be long-running and resource demanding
 - As opposed to most business services
 - E.g. most Opal services allocate a complete processor core
 - Natively invoking scientific services can lead to busy, non-responding services
 - i.e. load balancing is a major issue
 - A service bus reacts to non-responding services → loss of time
- A resource manager (RM) is needed that distributes work based on the processor load of services



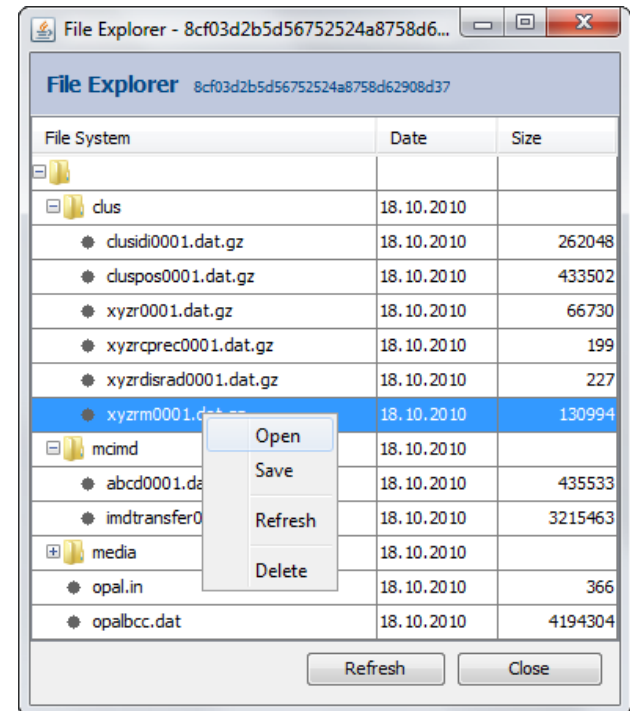
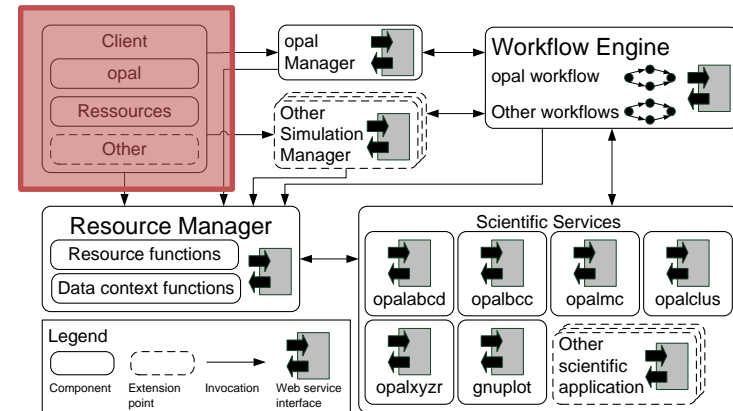
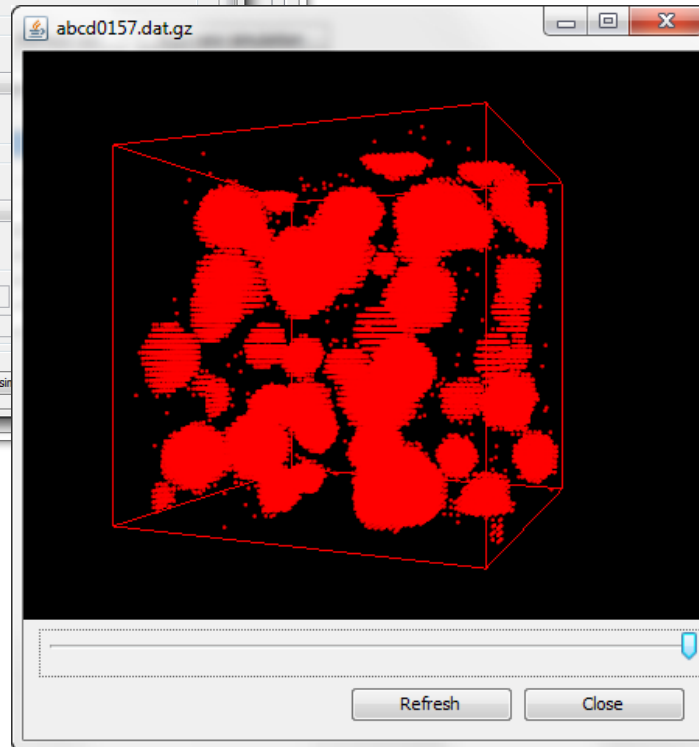
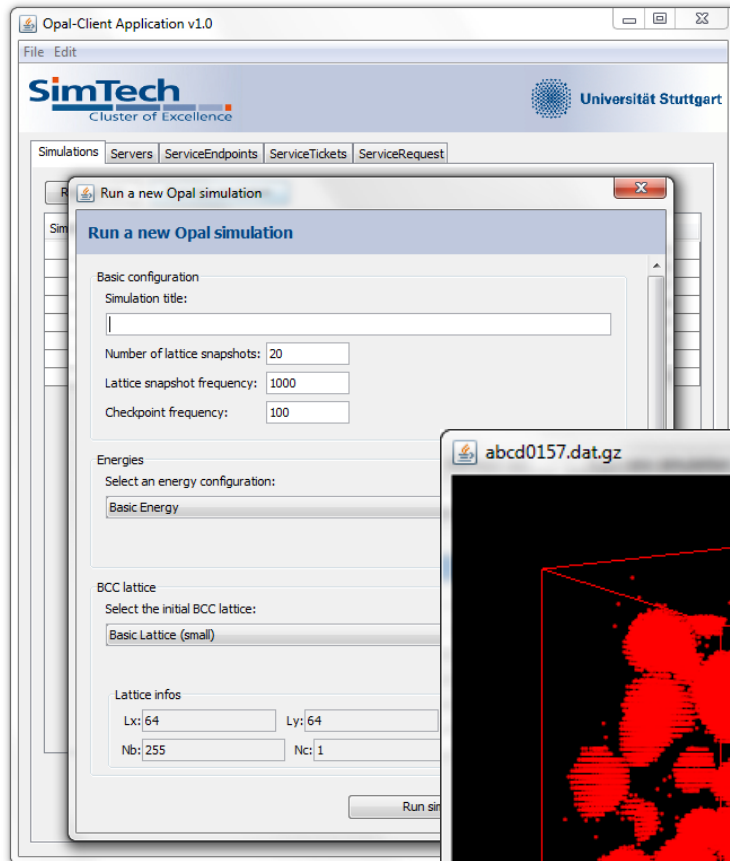
Need for a Data Manager

- Legacy simulation tools have a restricted data management
 - E.g. files are simply put into the working directory of the tool
 - Simulations overwrite former result files
 - Files have to be copied manually if many tools participate in the simulation
 - Data has to be correlated by the scientist
- A data manager is needed that ...
 - ... stores result files
 - ... organizes files in a simulation context
 - ... allows to pass files by reference

Providing Fortran Applications as Web Service

- Java-based WS wrapper with JNI or JNA
 - Invocation overhead: Java → C → Fortran
- C-based WS wrapper with gSOAP
 - Direct invocation of Fortran code possible
- Problems with parallel execution of Fortran program
 - Failed write operations even on different files:
 - Data was written in wrong lines
- Sandboxing solved problem
 - Java-based WS wrapper with command line invocation

New GUI to Manage the Infrastructure and the Simulations



Conclusions

- Automation and integration of scientific simulations based on Web services and service composition
- Resource management is part of the middleware
- Improved simulation process through parallelization and load balancing
- Decreased learning burden for scientists
- Implementation for a Fortran77 simulation of solids

- Realization of further simulations
- Installation of scientific services on demand

Thanks for your attention!

Do you have questions?



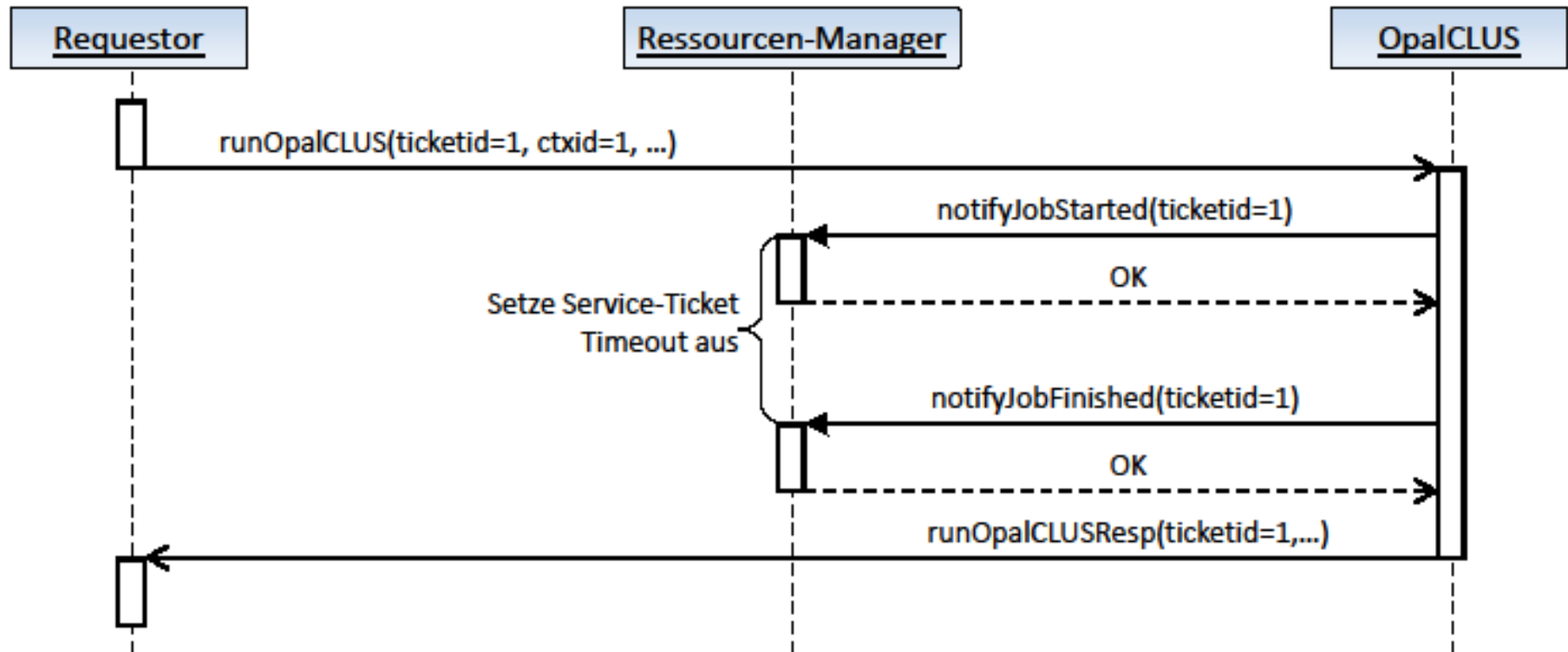
End of Document

Computer Simulations in Scientific Research

- Simulations are programmed by scientists
 - Simulation logic is buried in the code
- Legacy programming languages
 - Do not benefit from multi-core processors
 - Opal is written in Fortran77
- Monolithic applications
 - Do not benefit from distributed computing
 - Opal: consists of 5 programs without GUI running locally
- Scientists have to conduct many manual steps
 - E.g. starting programs, adapting and copying files
- Restricted data management
 - Often, data is simply located in the working directory of the application
- No or low-quality documentation due to lack of time
 - Stands in conflict to the high fluctuation of employees in scientific institutes
- No monitoring functionality
- Grid resources and computing hours are rare; commodity hardware is available

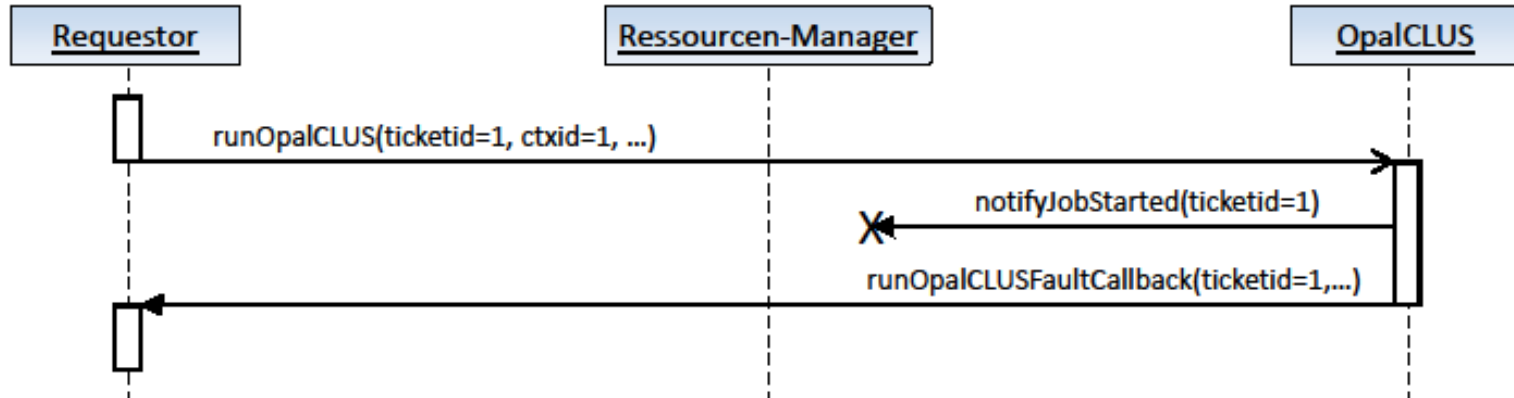
RM: Dealing With Network Partition

- Availability of services is checked
 - RM frequently polls the installed services on a server
- Successful usage of a scientific service:

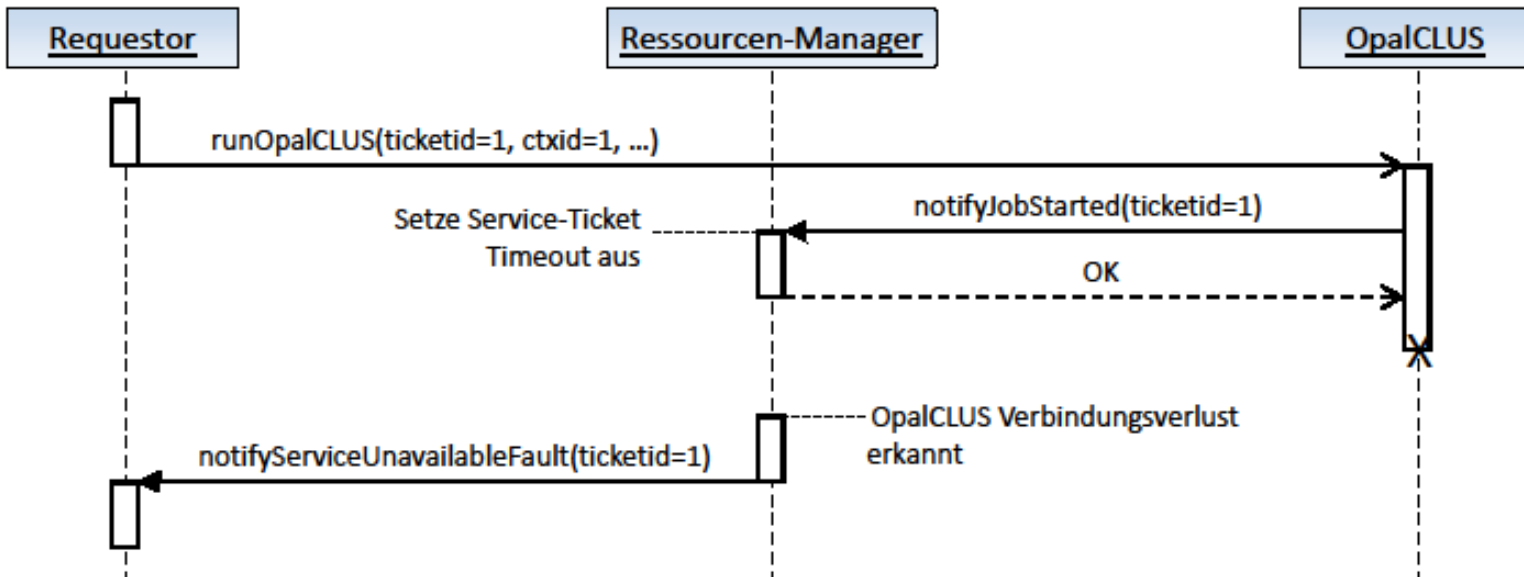


RM: Dealing With Network Partition (2)

Network partition before start of a scientific service



Network partition after start of a scientific service



Related Work Excerpt

■ OGSA and WSRF

- Grid resources are made available as Web services
- Advanced reservation techniques and life time of a resource had an impact on our ticket acquisition mechanism
- But a resource manager for load balancing and data management is not foreseen

■ Grid middleware, e.g. Globus Toolkit, Unicore

- Provides similar features to our approach especially in combination with Grid workflows
- Load balancing and distributed computing is possible
- Management of simulation data is not provided