EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Six Years of MoSGrid

Jens Krüger
High Performance and Cloud Computing Group
Zentrum für Datenverarbeitung
Eberhard Karls Universität Tübingen

04.04.17
Granada

Overview

- Motivation and Background
- Infrastructure
  - Storage
  - UNICORE
  - Authentication and Security
  - Portal Instance
- Portal
  - Liferay
  - gUSE/WS-PGRADE
  - Portlet API
  - Simulation Portlets and Workflows
- Stability and Resilience
- Lessons Learned
- Future Work

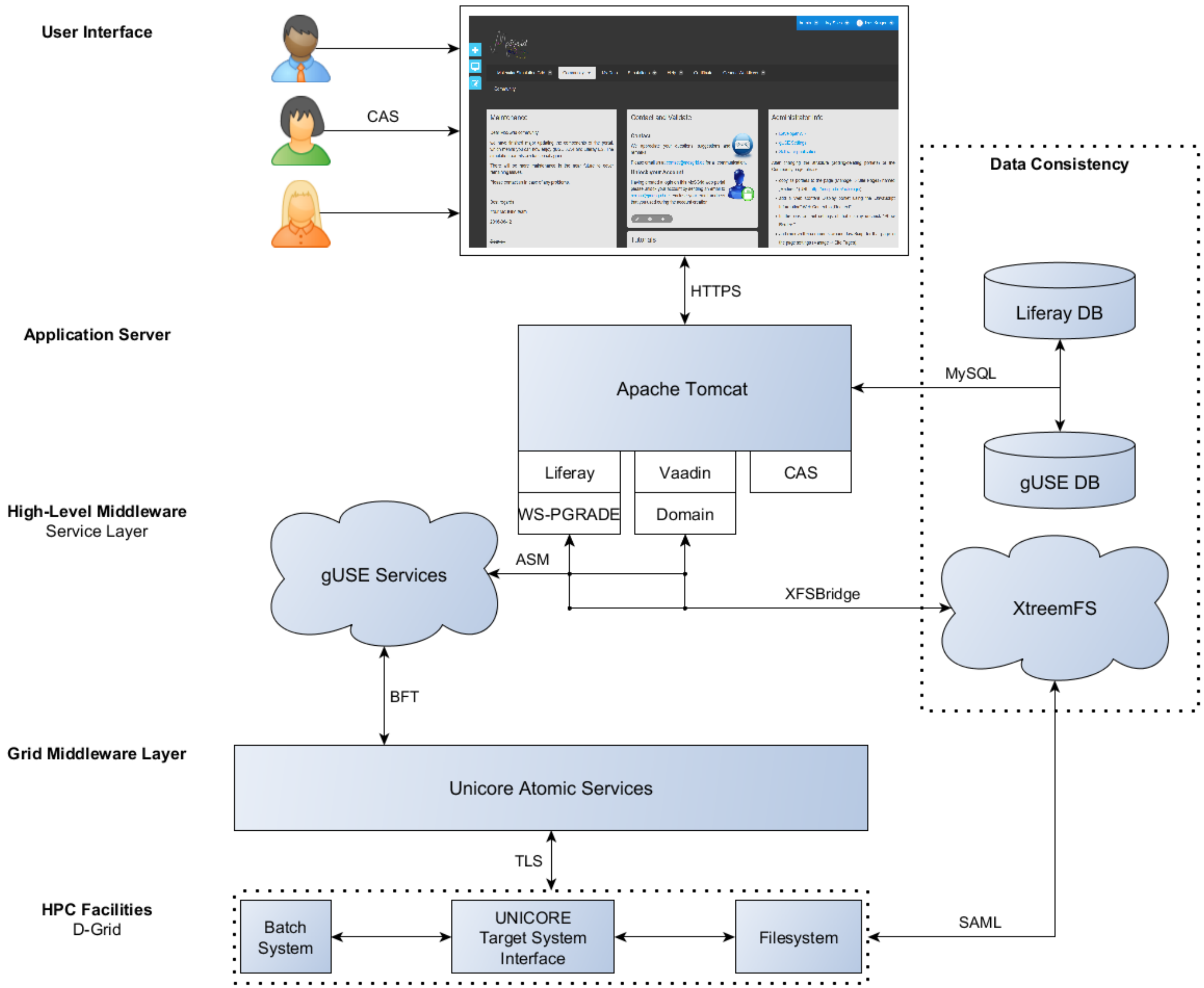
Motivation and Background

- MoSGrid was initially a BMBF funded project (since 2009 for 3.5 years)
- Intended for computational chemists and scientists from related fields
- Partners from academia and industry
- The portal is online since 2010
- FP7-EU projects ER-flow and SCI-BUS followed (till end of 2014)

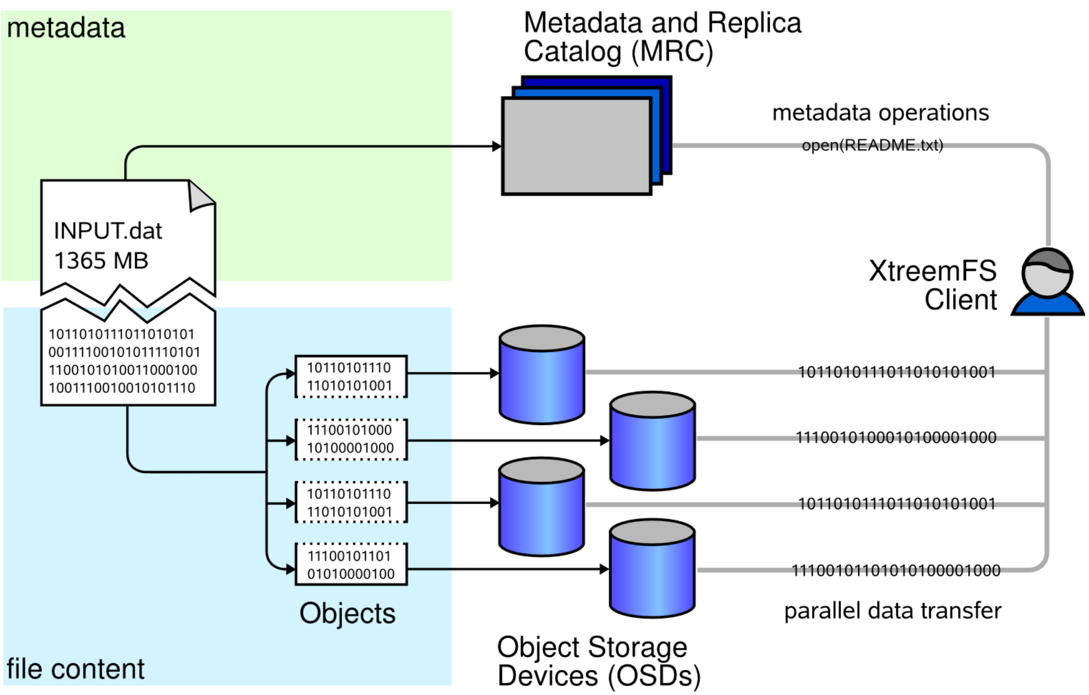
- Currently 493 users and more than 120 different workflows

mosgrid.de

Infrastructure



Storage



XTreemFS

Distributed object based file system

DIR directory service (1x)

MRC stores metadata (N x)

OSDs store pure file (N+ x) content

Client provides file system interface

xtreemfs.org

UNICORE

- UNICORE is a middleware facilitating the connection between portal and compute resources
- It offers a set of high-level services for job and workflow handling as well as for data movement
- For MoSGrid UNICORE offers a uniform layer easing operation and maintenance

Challenges

- Renewal of service certificates
- Different UNICORE versions on different HPC resources
- Curation of user list and user credentials
- Synchronization of information over all instances and sites

unicore.eu

Authentication and Security

- MoSGrid relies on personnel user certificates
- The user delegates the right to submit jobs and workflows to the portal instance using a SAML assertion
- Same applies for the secure connection to the storage containing simulation data and metadata information

Challenges

- Yearly renewal of user certificates
- Maintenance of the Certificate Portlet which allows the creation of SAML assertions (Java Applet → plain Vaadin portlet)
- Synchronization between all instances

Portal Instance

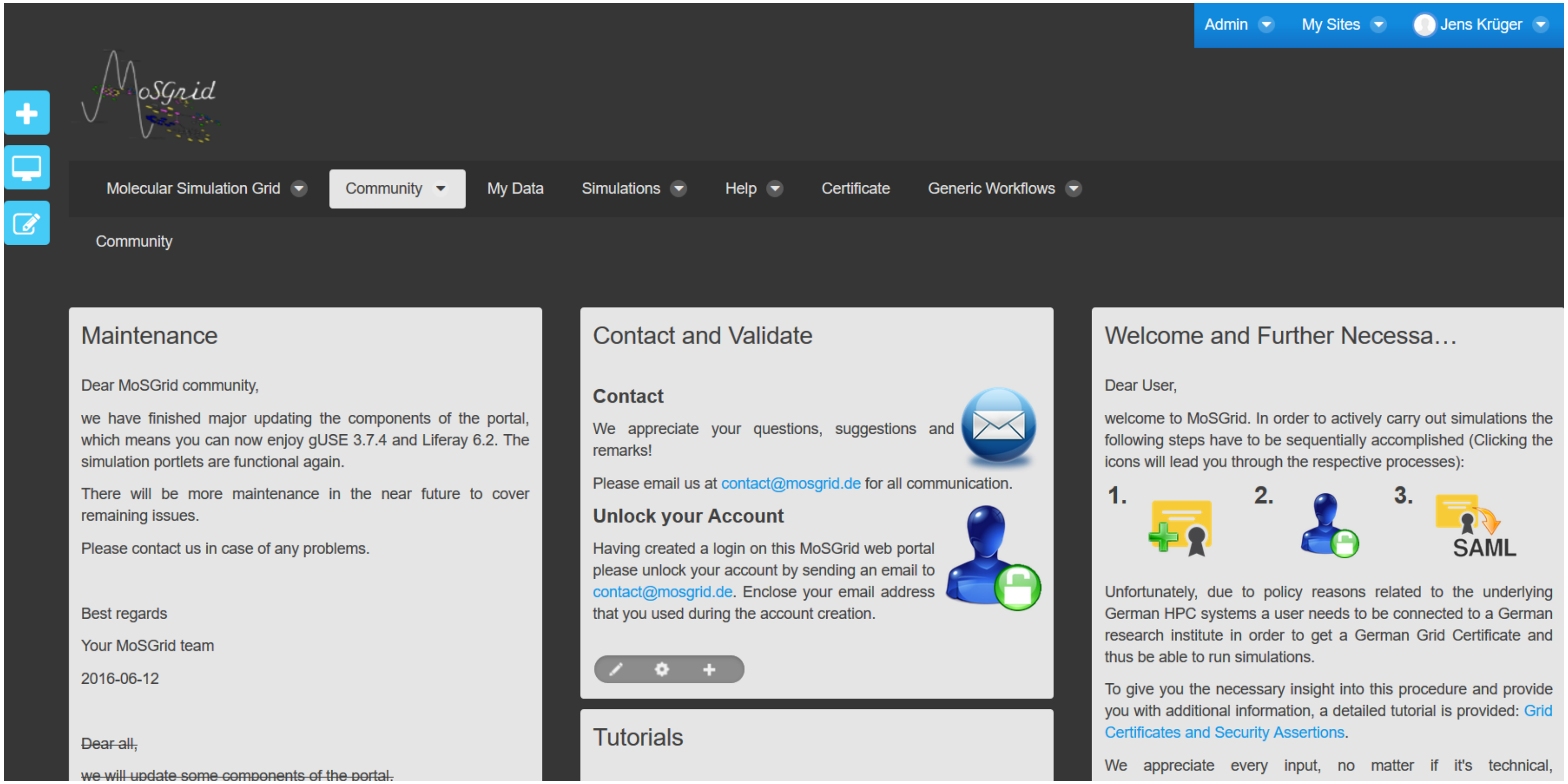
- Running as VM on a shared host
- Scientific Linux 6.7
- Java Virtual Machine 1.7
- MySQL 5.1.73 for Liferay and gUSE databases
- gUSE 3.7.4 including Liferay 6.2 GA2 and Apache Tomcat 7.0.55

- No foreign software repositories are used
- No automated updates

Challenges

- Changes to the basic infrastructure like network or VM host
- Possibility to continuously upgrade to recent versions without breaking compatibility

Liferay



Challenges

- Incompatibilities between different versions of Liferay (6.x→7.x), gUSE/WS-PGRADE, Vaadin, …

liferay.com

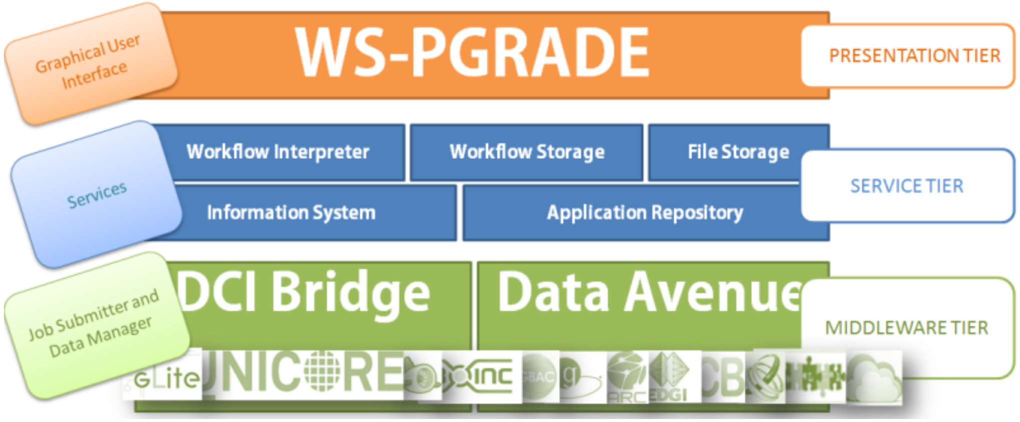
gUSE/WS-PGRADE

Generic-purpose gateway framework

- Based on Liferay
- Workflow-oriented gateway framework
- Supports the development and execution of workflow-based applications
- Supports the fast development of domain-specific gateways by a customization technology
- Most important design aspects are flexibility and robustness

Challenges

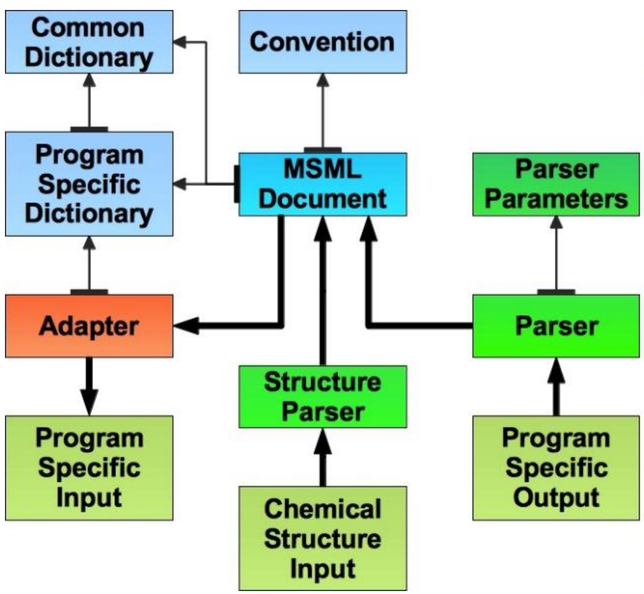
- Academic project
- Rather steep learning curve



guse.hu

Metadata Annotation and Search

- Molecular Simulation Markup Language (MSML)
- CML compliant
- Template for each and every workflow
- Molecular input
- Domain specific tools
- Job configuration
- Optimized structures, trajectories, energies, …
- Semantic search

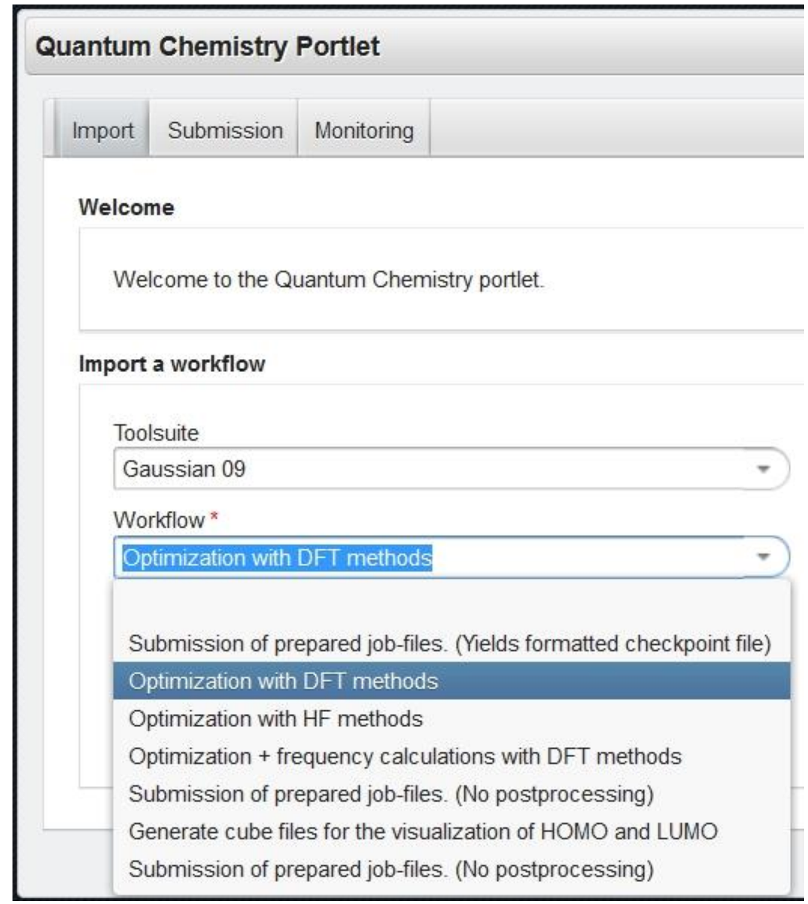



```
<?xml version="1.0" encoding="UTF-8"?>
<cml convention="convention:compchem">
        <module dictRef="compchem:jobList">
                <cmlx:parserConfiguration/>
                <module dictRef="compchem:job" id="Job1">
                        <module dictRef="compchem:environment">
                                <propertyList/>
                        </module>
                        <module dictRef="compchem:initialization">
                                <parameterList/>
                                <cmlx:adapterConfiguration/>
                                <cmlx:parserConfiguration/>
                        </module>
                        <module dictRef="compchem:finalization">
                                <propertylist/>
                        </module>
                </module>
                <module dictRef="compchem:job" id="Job2"/>
                ...
        </module>
</cml>
```

CCPE 2013, 26(10):1744-1759

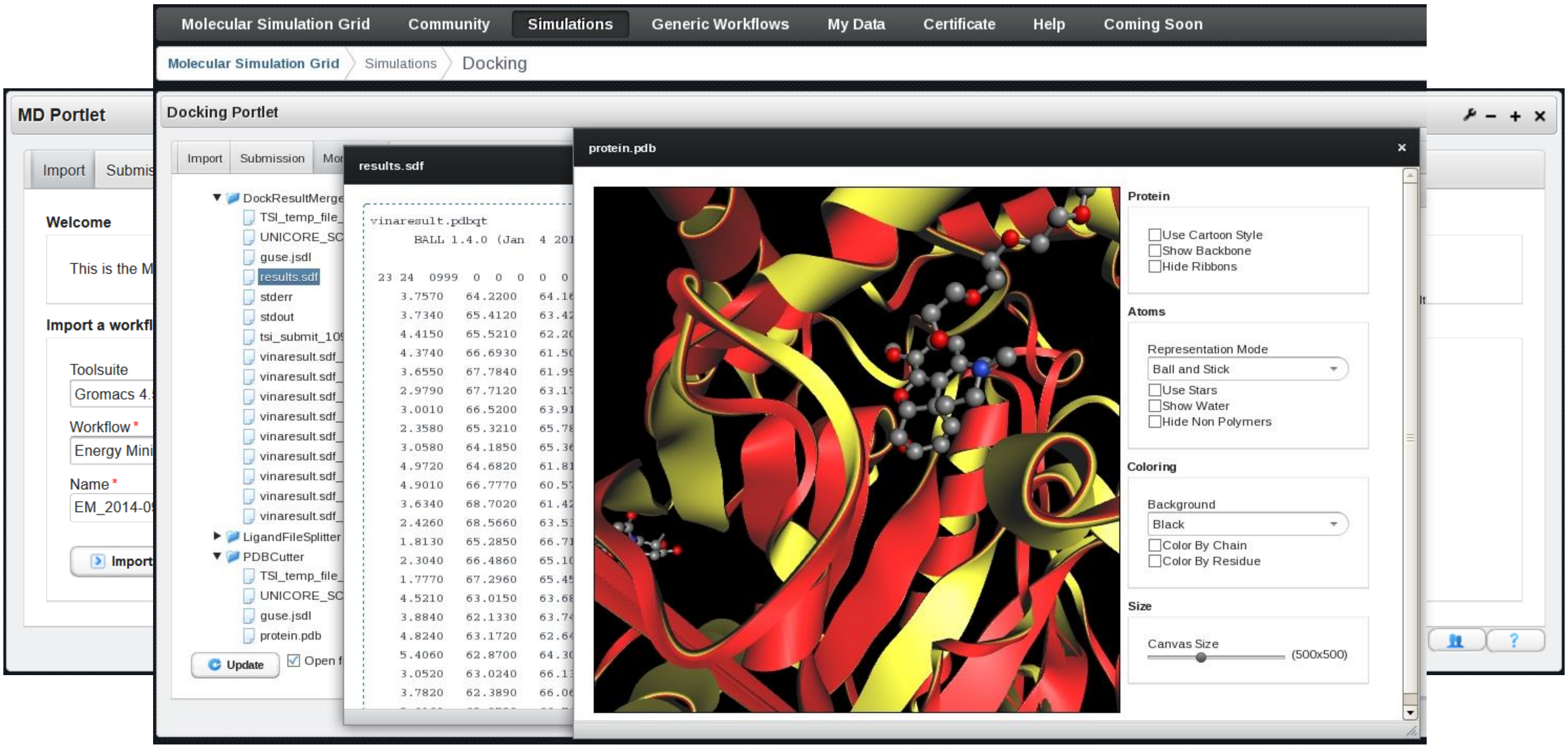
Portlet API

```xml
<module dictRef="compchem:initialization">
    <parameterList>
        <parameter dictRef="g09:loglevel">
            <scalar dataType="xsd:string" units="si:none">p</scalar>
        </parameter>
        <parameter dictRef="g09:jobtype">
            <scalar dataType="xsd:string" units="si:none">opt</scalar>
        </parameter>
        <parameter dictRef="g09:hf.theory" cmlx:editable="true">
            <scalar dataType="xsd:string" units="si:none">hf</scalar>
        </parameter>
        <parameter dictRef="g09:basisset" cmlx:editable="true">
            <scalar dataType="xsd:string" units="si:none">6-31G</scalar>
        </parameter>
        <parameter dictRef="g09:formal.charge" cmlx:editable="true">
            <scalar dataType="xsd:integer" units="si:none">0</scalar>
        </parameter>
        <parameter dictRef="g09:spin" cmlx:editable="true">
            <scalar dataType="xsd:integer" units="si:none">1</scalar>
        </parameter>
        <parameter dictRef="g09:checkpointfile">
            <scalar dataType="xsd:string" units="si:none">job.chk</scalar>
        </parameter>
    </parameterList>
    <cmlx:adapterConfiguration adapterID="g09adap"
        fileExtension="com" portName="job.com"/>
</module>
```

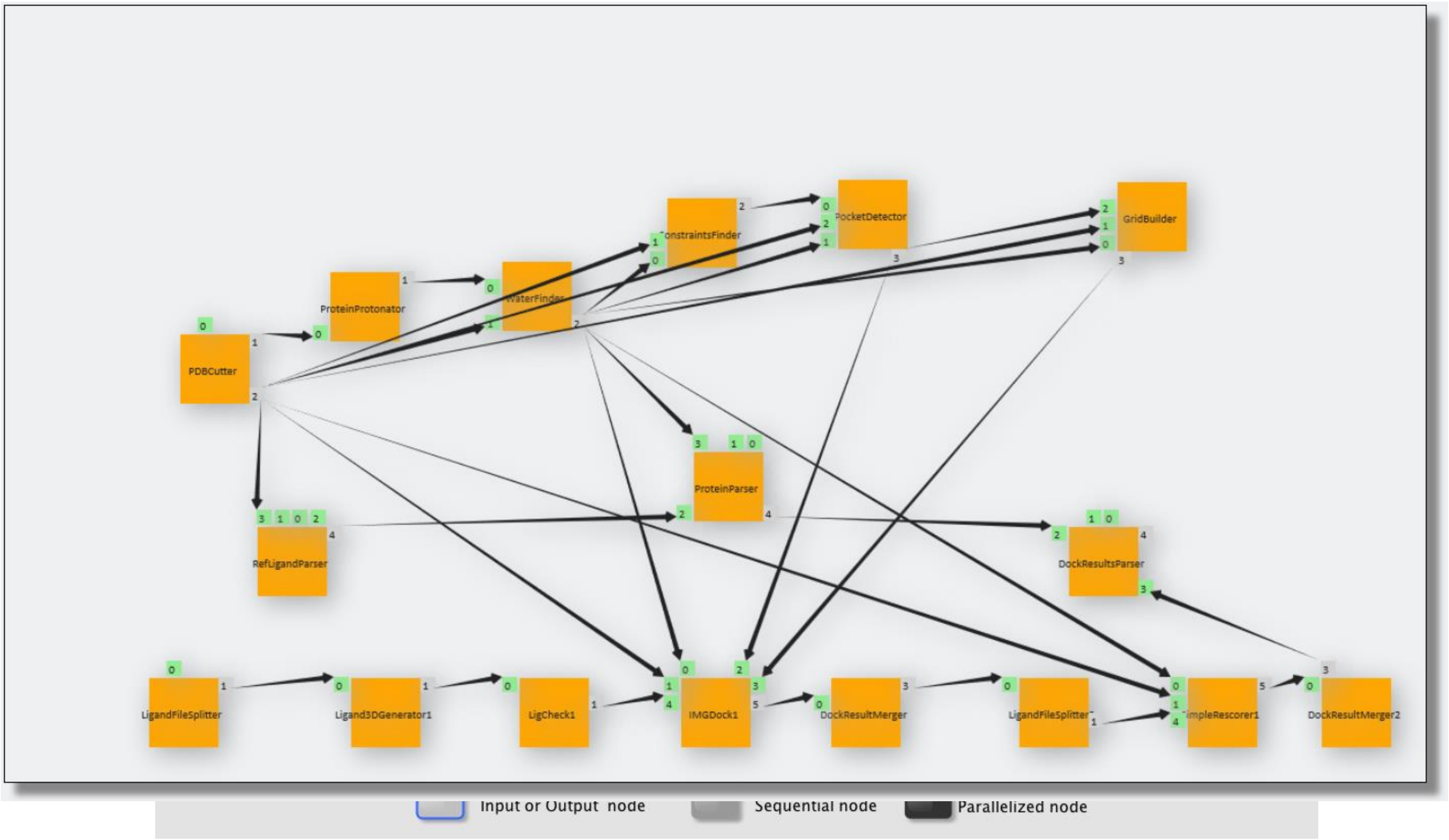


JCTC, 2014, 10 (6), 2232–2245

Simulation Portlets



Workflows


PDBCutter
ProteinProtonator
WaterFinder
ConstraintsFinder
PocketDetector
GridBuilder
ProteinParser
RefLigandParser
DockResultsParser
LigandFileSplitter
Ligand3DGenerator1
LigCheck1
IMGDock1
DockResultMerger
SimpleRescorer1
DockResultMerger2
Input or Output node
Sequential node
Parallelized node

Stability and Resilience

- MoSGrid relies on the support of various university compute centers across Germany
- Multiple loosely coupled layers

Challenges

- Communication about maintenances
- Correct detection of errors and problems
- Changing policies
- Decommission of resources
- Availability of dedicated personnel in an academic environment

User Behavior

95 % of all users are smart and hard working people
Same applies for administrators, developers, …

… but everybody may have a weak day.

Be prepared for every imaginable stupidity.

Lessons Learned

Challenges

- MoSGrid relied on the availability of D-GRID services
- Migration of a whole science gateway
- Consistency over multiple levels and multiple sites

Solution

- Constant communication with the compute centers
- Careful planning and plenty of manpower
- Careful planning and good testing procedures

Take home message

- Choose an actively maintained science gateway framework
- Enrich with features which truly add an extra value
- Keep the number of used technologies, interdependencies and overall complexity as low as possible
- Ensure that direct access to remote compute and data instances is possible
- Close communication with the user community

Future Work

- Migration of key services to a RHEV environment
- Containerization of applications using Docker or Singularity
- Support of mobile devices
- Virtualization possibilities such as microservices enabling scientists to move compute resources to their research data and not vice versa.

Summary

- A Science Gateway can ease the access to computational and storage resources for a specific community

- MoSGrid is successfully operated for more than six years
- Several hundred users have used MoSGrid successfully
- Regularly used in teaching higher education classes
- Experience gathered for the computational chemistry community could provide useful insights for other communities

- A lean, robust and open-source technology stack is advised when making design decisions for future science gateways.

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Acknowledgement

Core team

- Sandra Gesing, Sonja Herres-Pawlis and Richard Grunzke

Contributors

- Lars Packschies, Andreas Zink, Lukas Zimmermann, Patrick Schäfer, Zoltan Farkas, Peter Kaczuk, Alexander Hoffmann, Sebastian Breuers, Gregor Fels, Georg Birkenheuer, André Brinkmann, Luis de la Garza, Oliver Kohlbacher, Martin Kruse, Wolfgang Nagel, Ralf Müller Pfefferkorn, Thomas Steinke, Tobias Schlemmer, Klaus Dieter Warzecha, …

Compute Centers

- ZIH Dresden, ZDV Tübingen, PC² Paderborn, RRZK Cologne, ZIB Berlin, …

Funding

- MoSGrid (BMBF), SCI-BUS (EU-FP7), ER-flow (EU-FP7) and MASi (DFG)

https://scholarspace.manoa.hawaii.edu/handle/10125/41918