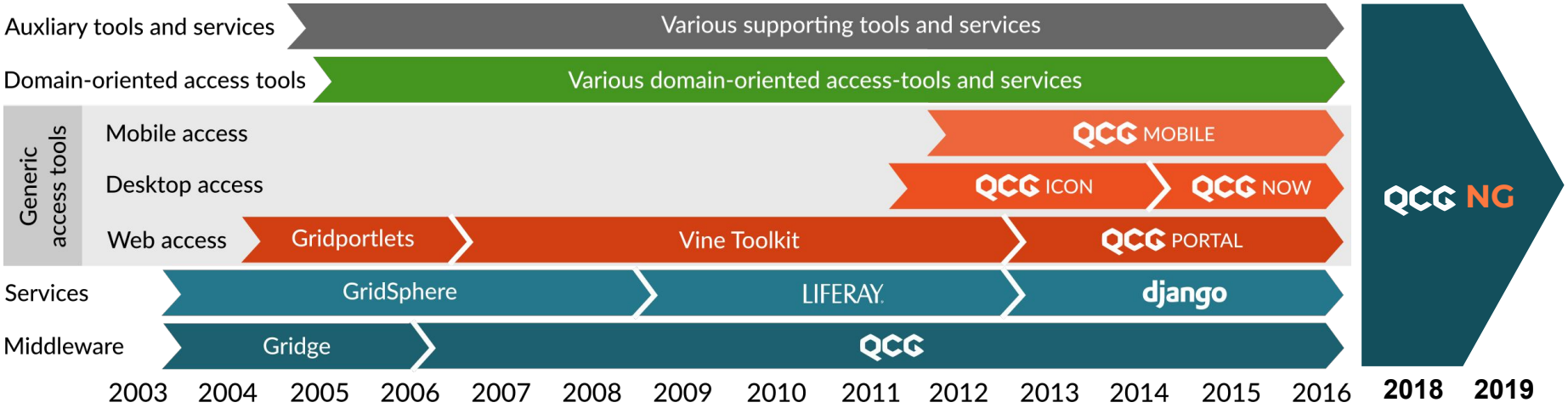
QCG quality in cloud and grid

Supporting advanced HPC/HTC scientific workloads with QCG services

PCSS ECMA

Bartosz Bosak, Piotr Kopta, Tomasz Piontek
{bbosak, pkopta, piontek}@man.poznan.pl
Poznan Supercomputing and Networking Center

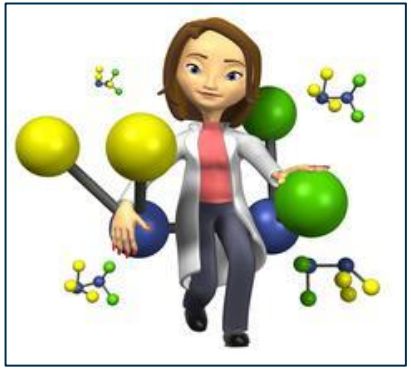
A bit of history

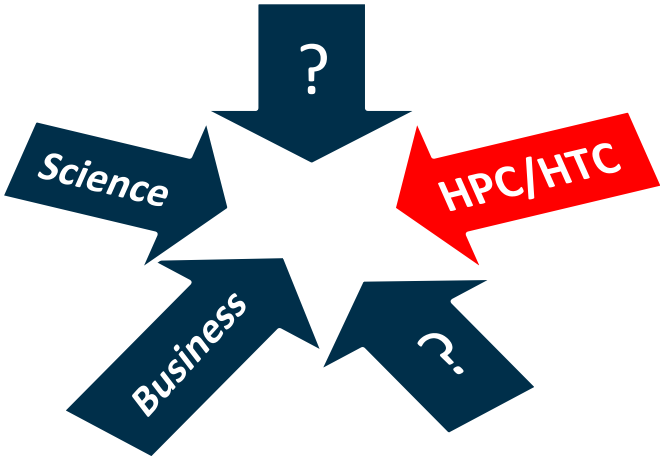


http://www.qoscosgrid.org

Our motivation

It is not enough to give scientist access to resources. They need tools and support.





QCG

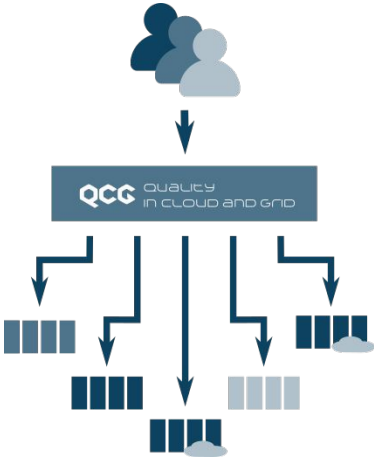
The world has changed… We cannot ignore this fact

Most of the new users need easy and intuitive tools to overcome the initial barrier in accessing the infrastructure

QCG

The idea

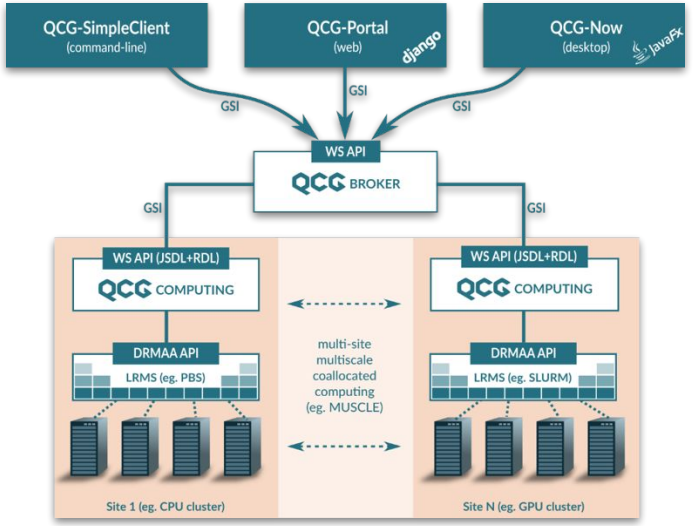
- QCG is a set of services, tools and libraries providing uniform and efficient access to HPC/HTC resources.
- Integrates many computing resources, but preserves simplicity of execution on a single machine.



- Provides highly efficient mapping for a variety of applications, such as parameter sweep, workflows, MPI or hybrid MPI-OpenMP, GPGPU, and multiscale.
- **Useful and applicable for both experienced and novice users.**

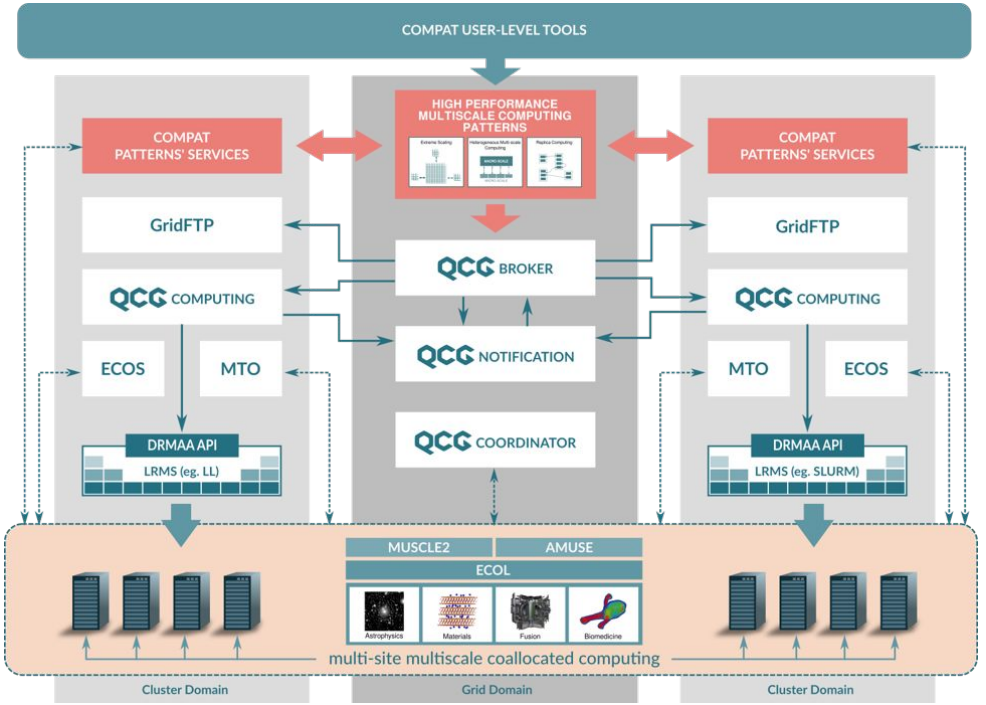
http://www.qoscosgrid.org

Architecture



- QCG consists of three basic layers:
  - Client tools
  - High-level services (common)
  - Cluster-level services

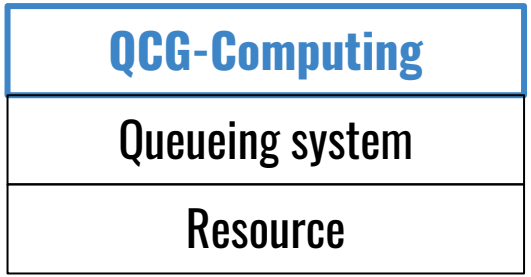
Architecture (2)



Basic services:

QCG-Computing

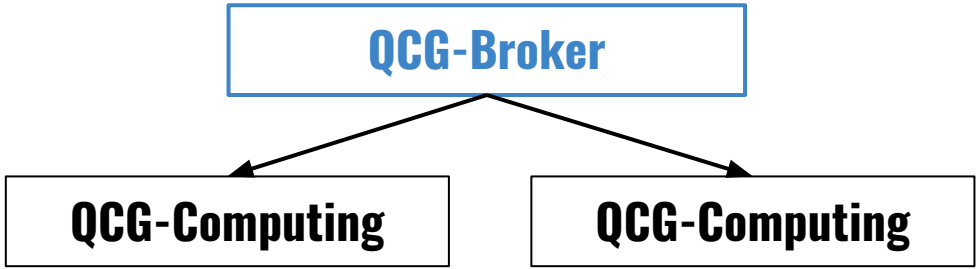
- Deployed on access nodes of the batch systems (Slurm, SGE, Slurm, torque/maui, LoadLeveler, PBS Pro, Condor, Apple Xgrid)
- Provides remote access to task submission and advance reservation capabilities of LRMS via DRMAA interface (the own DRMAA implementations)
- Compatible with the OGF HPC Basic Profile specification (JSDL and BES)



QCG

Basic services:
QCG-Broker

- Offers capabilities for scheduling and brokering of jobs
- supports multi criteria approaches taking into consideration many criteria like energy and resources consumption as well as time-to-finish
- Controls the execution of whole experiments (including workflows and parameter sweep tasks)
- Provides requested QoS and co-allocates resources
- Stages in/out files and directories (the main transport mechanism based on gridFTP)



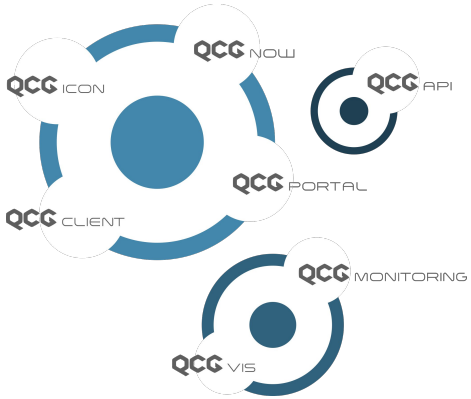
Basic services:

QCG-Notification

- Supports the topic-based publish/subscribe pattern for asynchronous message exchange
- Serves as the main message bus between the services, applications and the end-user
- Is capable of sending notifications using variety of transport mechanism, including SOAP, SMTP and XMPP (notifications for users)
- All major QCG clients indirectly use QCG-Notification to support tracking of statuses and progress of executed application (mail, XMPP, QCG-Monitoring)

QCG

Client tools & libs



Users are interested in tools not services ...

- Scientists have different preferences in accessing the infrastructure what results in the necessity to provide them with various tools tailored to their habits and expectations. (Swiss army knife approach often fails).
- Users very often can accept the fact that their jobs need long time to be finished but they would like to know when they finish.
- For long running simulations users need a possibility to see partial results and to control the execution.

Client tools:
QCG-Client

- Set of commands modelled on queuing system tools
- QCG-Simple, JSDL, QCG-XML description dialects
- Detailed information about running and historical tasks
- Support for interactive tasks
- Automatic staging in/out files with gridFTP
- Single access point to all resources

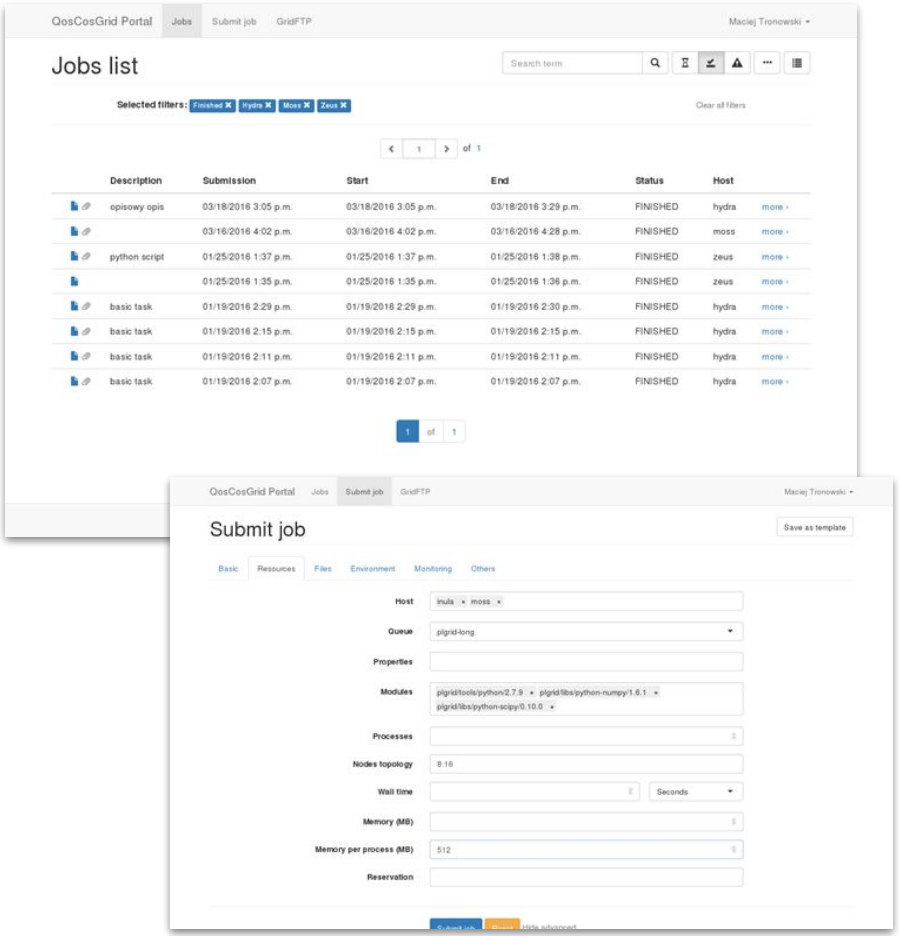
```
#QCG note=NAMD apoa1
#QCG host=hydra.icm.edu.pl

#QCG walltime=PT10M
#QCG queue=plgrid
#QCG nodes=1:12:12
#QCG output=apoa1.output
#QCG error=apoa1.error
#QCG application=NAMD
#QCG argument=apoa1/apoa1.namd
#QCG stage-in-file=apoa1.zip
#QCG preprocess=unzip apoa1.zip
#QCG stage-out-dir=. -> results
#QCG notify=xmpp:tomasz.piontek@plgrid.pl
#QCG watch-output=mailto:tp@mail,20,ENERGY
```

Client tools:
QCG-Client (2)

- Submission and control of tasks:
  - **qcg-cancel** - cancel task(s)
  - **qcg-clean** - clean the working directories of given tasks
  - **qcg-connect** – establish interactive session to the task
  - **qcg-info** - display detailed information about the given tasks
  - **qcg-list** - list tasks in the system
  - **qcg-peek** - display ending of (stdout, stderr) streams
  - **qcg-refetch** - retry/repeat the transfer of output files/directories
  - **qcg-sub** - submit batch or interactive tasks to be processes by QCG
- Reservation and control of resources:
  - **qcg-rcancel** - cancel reservation(s)
  - **qcg-reserve** - reserve resources
  - **qcg-rinfo** - display information about the given reservation(s)
  - **qcg-rlist** - list reservation in the system
- Information about resources:
  - **qcg-offer** – provides detailed information about all managed resources, e.g. list of available applications, list of available modules, current load

Client tools:
QCG-Portal

- A web client to QCG, based on the Django framework
- Offers functionality needed to submit, monitor and manage jobs over QCG middleware
- Authentication/authorization based on OpenID
- Includes built-in support for the gridFTP protocol
- Support for job templates
- In principle easily extendable and adaptable to certain needs
- Several ready application / domain based solutions built on top of QCG-Portal



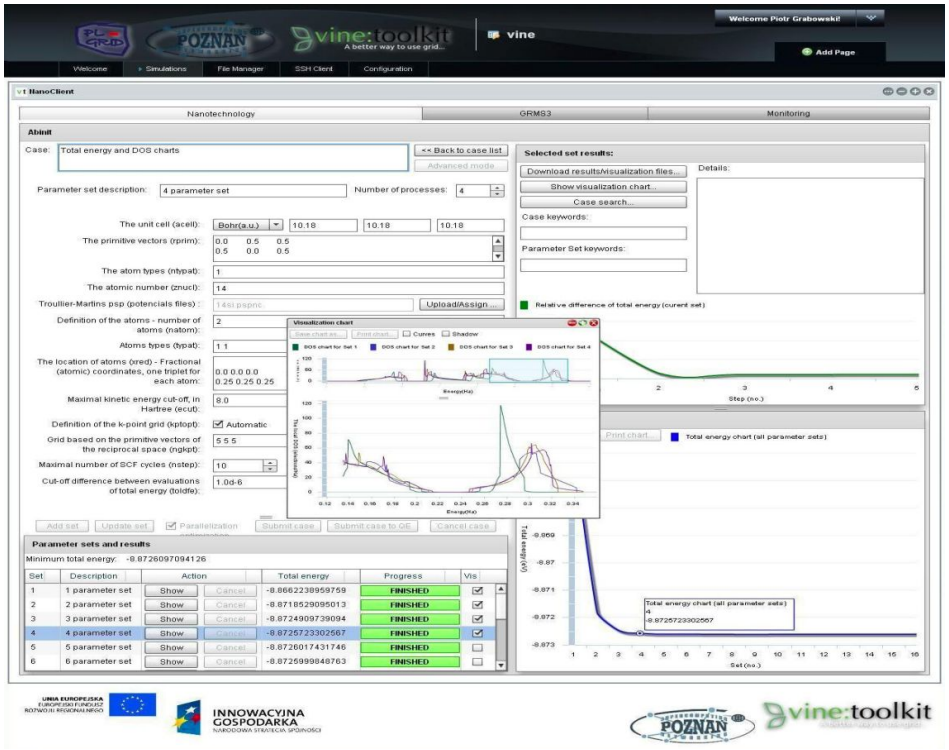
Client tools:
QCG-Portal (2)

QCG

Client tools:

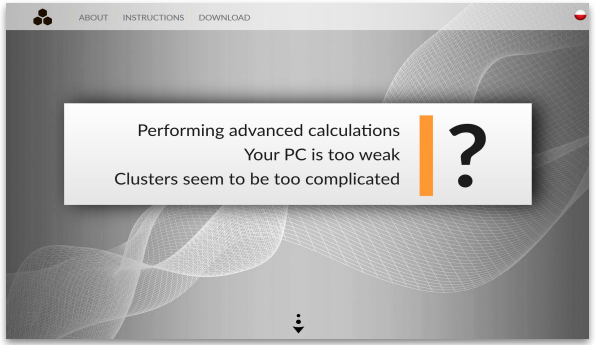
QCG-Science-Gateways

- Domain or problem oriented, highly specialized portal solutions



Client tools:
QCG-Now

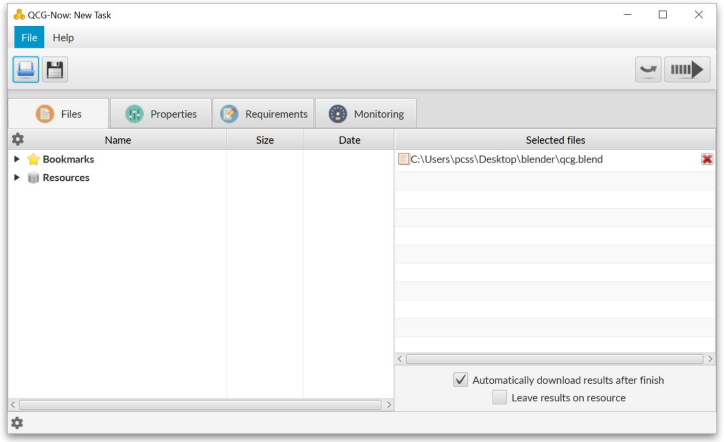
- A desktop (GUI) tool for submission of jobs to computing resources made available with QCG middleware.
- Available as an installable package for Windows, Linux and OS X.
- Integrated support for data transfer and application monitoring.
- Support for job templates and command-line submission (easy integration with other applications).



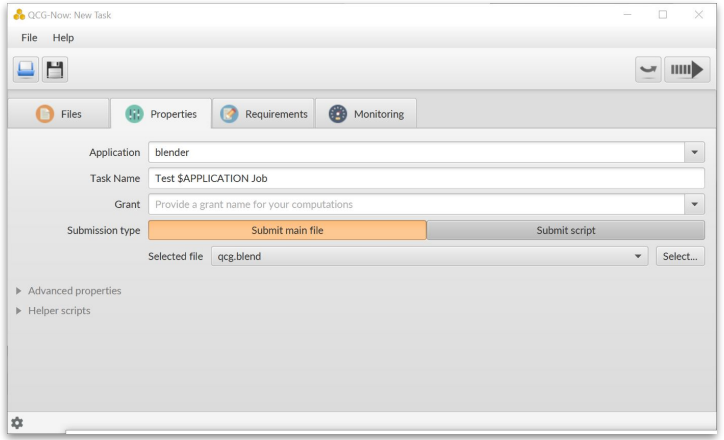
http://www.qoscosgrid.org/qcg-now

Client tools:
QCG-Now (2)

Task definition view - files



Task definition view - properties



Client tools:

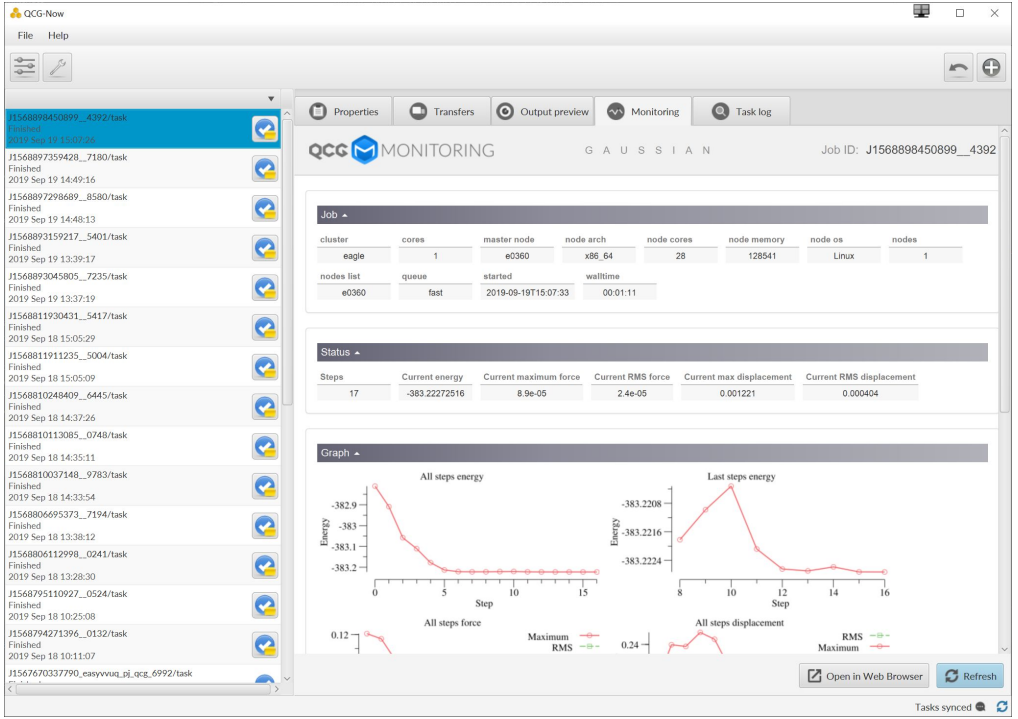
QCG-Monitoring

- Monitoring progress of long-lasting jobs
- Cyclic scanning of output files
- Current status presented with text messages, tables, graphs and pictures
- Predefined schemes for different types of applications (e.g. Gaussian)
- Integrated with QCG-Client, QCG-Portal and QCG-Now

QCG-Monitoring

An example monitoring scheme in QCG-Now

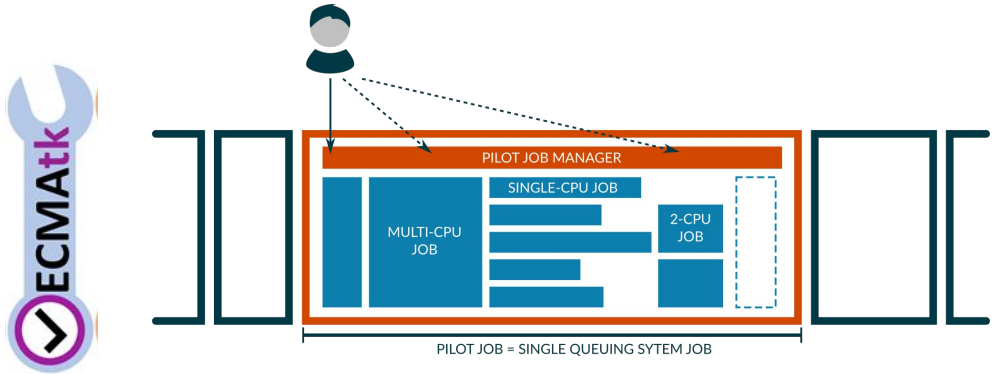
- Integration of QCG-Monitoring with QCG-Now (in the testing phase)



QCG-PilotJob Manager

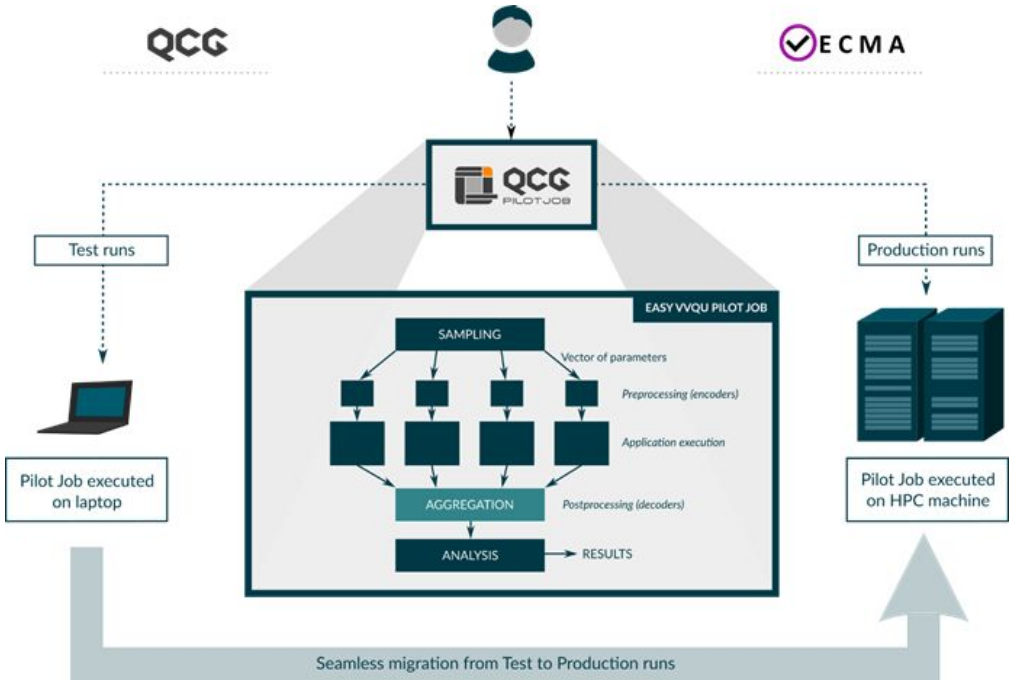
HTC scenarios are also important ...

- Schedules and manages a set of tasks on already allocated resources:
  - to eliminate waiting of every single task in a queue
  - to better utilize resources (better resource granularity)
  - to „overcome” local administrative policies
- Can read requests from various sources: file or network socket (static vs dynamic scenario)
- Contains API (Python) for communication with user programs



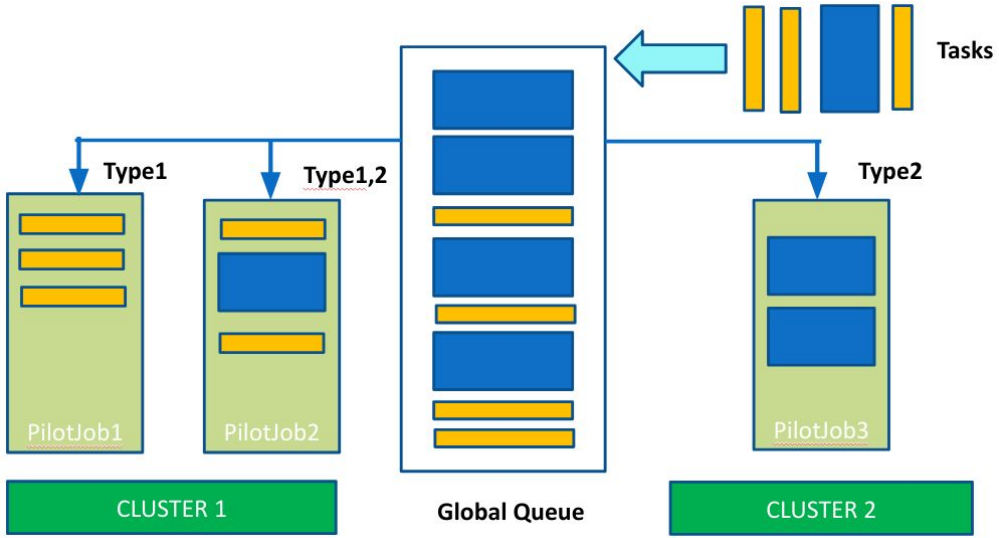
QCG-PilotJob Manager

- Integration of QCG-PJM with EasyVVUQ for uncertainty quantification in the VECMA project



QCG-PilotJob Manager (future plans)

- Ongoing work on introducing the concept of a global queue
  - to use resources of many clusters
  - to adjust the amount of resources to the current need of the application by adding new allocations when needed
  - to support exascale examples
- Integration with the QCG-Monitoring system (almost done)



QCG

QCG-Comp-NG

QCG Next Generation services

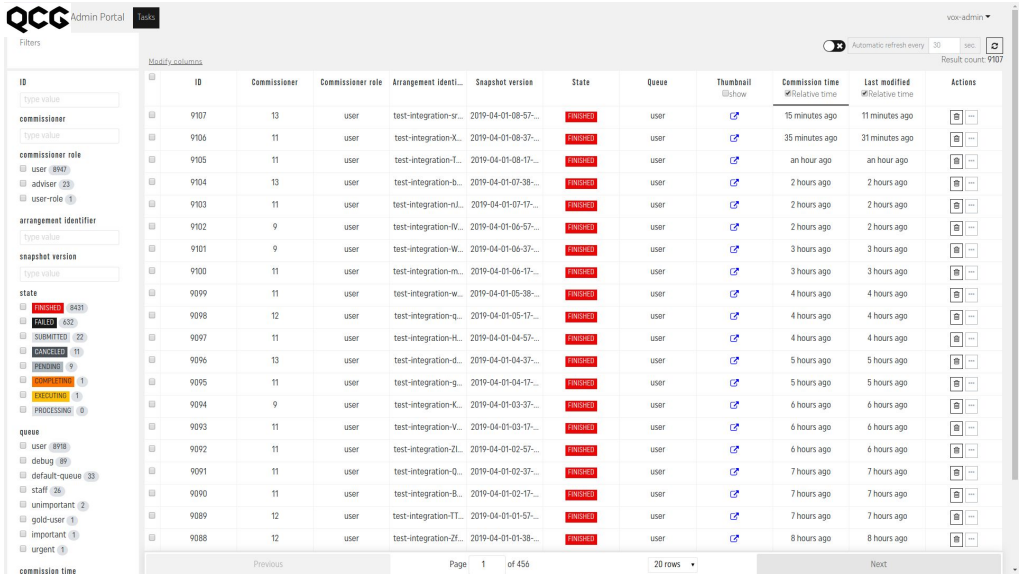
Transition to new technologies

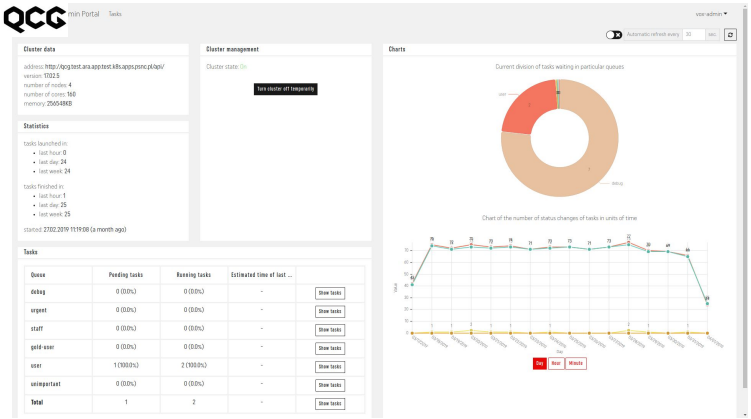
- Mico-services approach for HA
- Unified interface on cluster and ‘grid’ levels
- Implemented in Python with modular approach
- REST interfaces
- New authentication and authorization
- First version deployed and validated at PSNC as a part of external commercial system

QCG

QCG-Comp-NG

QCG Next Generation Tools - Portal





QCG

Success stories

PL GRID

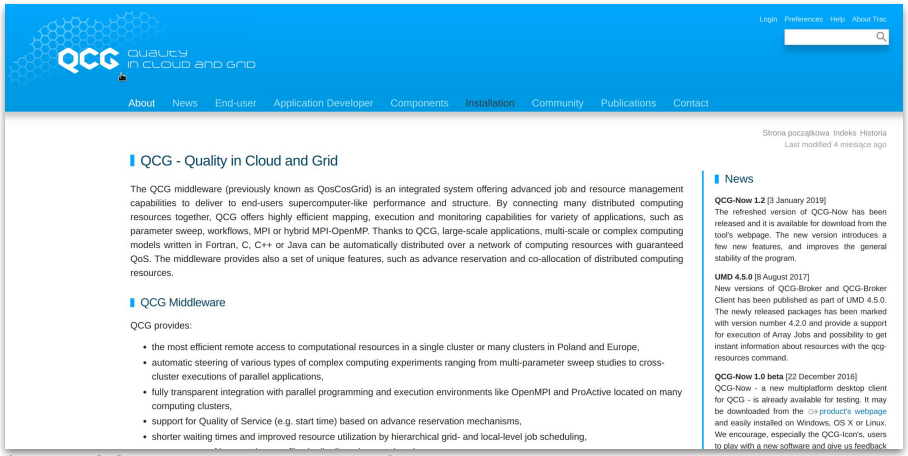
pracelab

QCG is the primary middleware in Poland

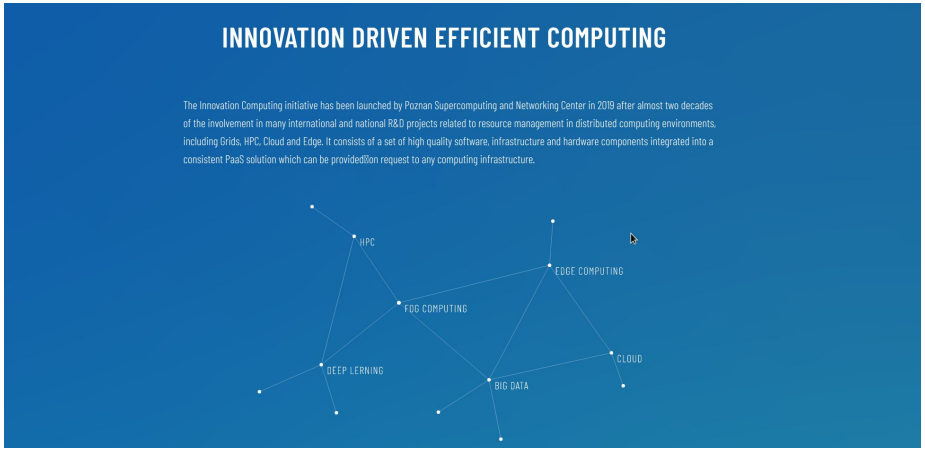
QCG

Success stories

Project web page



http://www.qoscosgrid.org



http://www.computing-innovations.org

Thank you!

Contact:
piontek@man.poznan.pl