SG-DG Bridges

Zoltán Farkas, MTA SZTAKI

The EDGeS project receives Community research funding

Outline

- Introduction, aims
- SG features: EGEE
- DG features: BOINC (and XtremWeb)
- BOINC -> EGEE bridge
- 3G Bridge architecture
- EGEE -> BOINC bridge

Introduction, aims

- The EDGeS project aims to offer an infrastructure that integrates Service Grid (SG) and Desktop Grid (DG) infrastructures
- Users of one grid type should be able to make use of the other grid type in a transparent way and vice versa
- Thus, the integrated infrastructure will offer the advantages of the two grid type
- The core component of this infrastructure is the SG-DG bridge technology

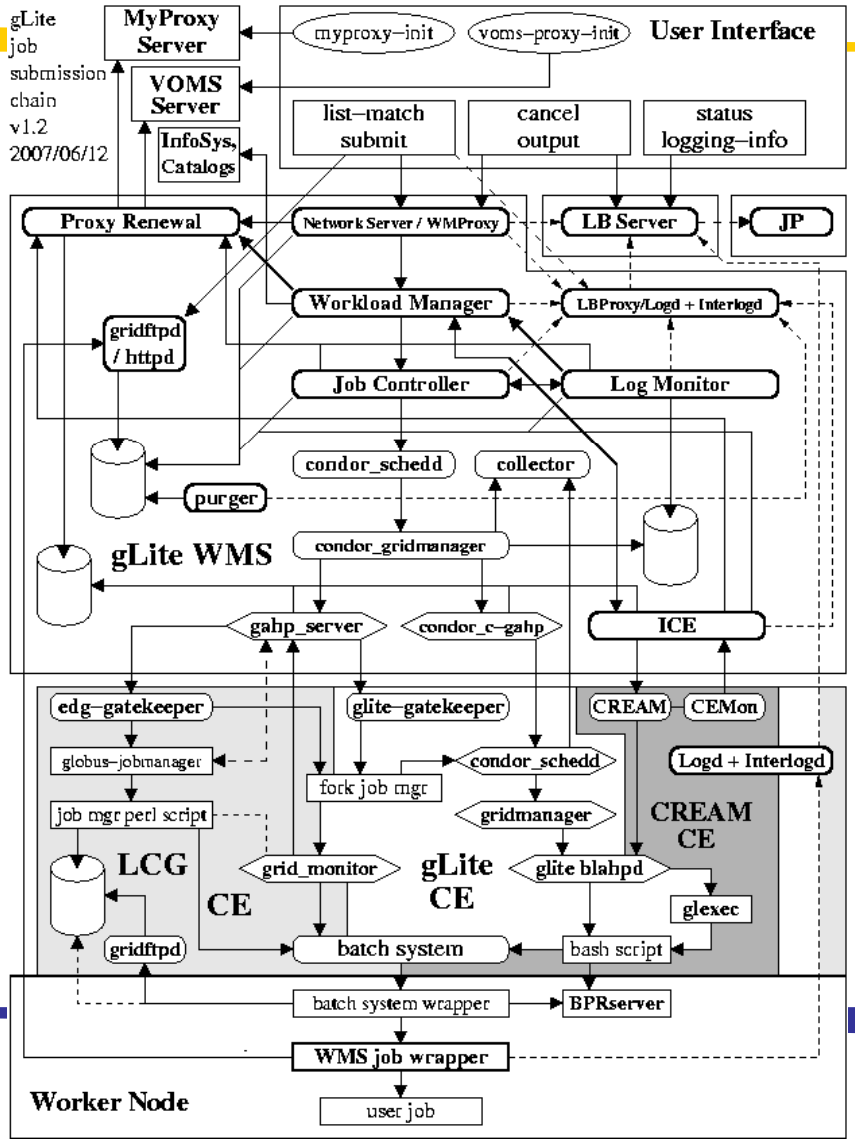
SG features – EGEE I.

- A big set of services:
  - WMS: broker, scheduling jobs to resources
  - LB: logging and bookkepping service
  - BDII: information system
  - WN: worker node, does actual job execution
  - CE: computing element, collects WNs in a queue using an LRMS
  - SE: storage element, used to store large files
  - LFC: file catalogue, files stored on SE can be organized into a directory structure
  - MyProxy: proxy certificate storage
  - VOMS: virtual organization membership handling component
  - R-GMA & APEL: accounting services

SG features – EGEE II.

- Mostly institutes provide the computing resources
- Resources are organized into Virtual Organizations
- Users with a registered certificate accepted in some VO can use the infrastructure
- Basically any kind of job can be executed, with some restrictions

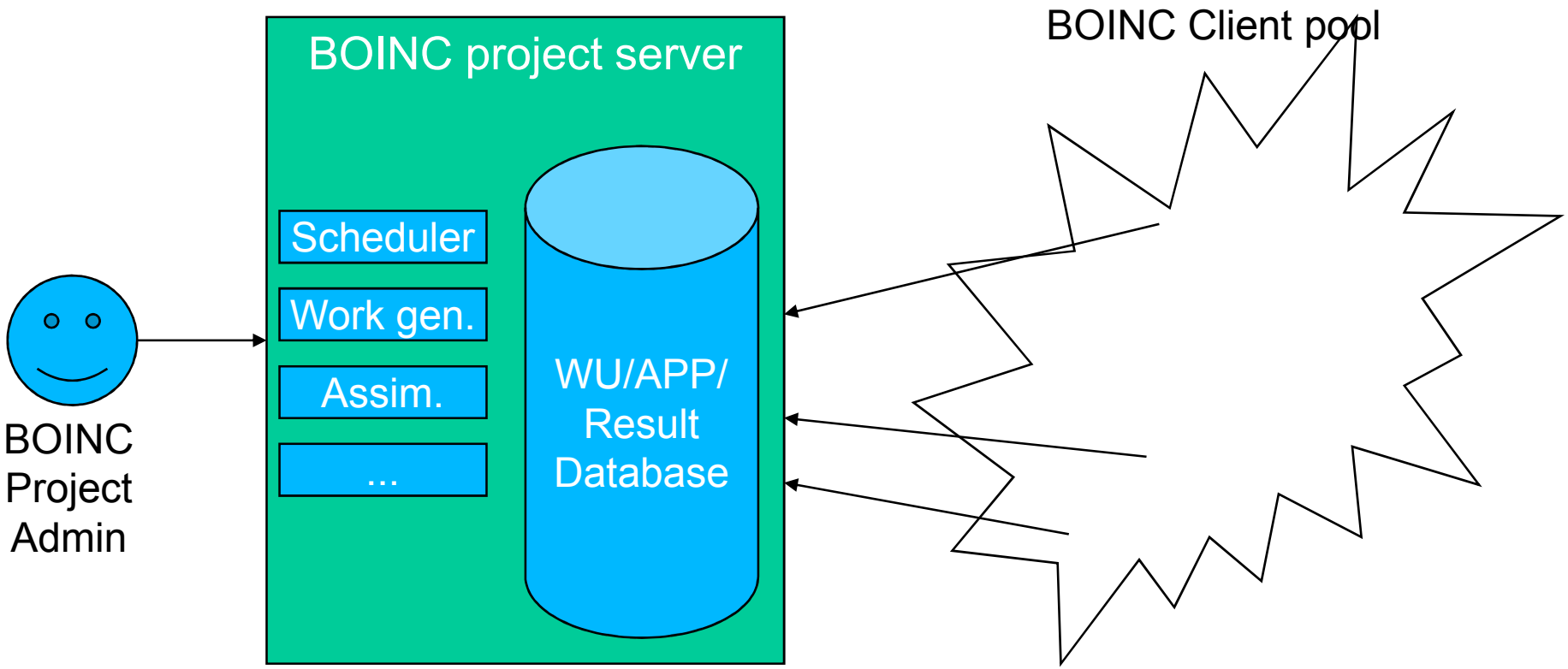
SG features – EGEE III.



Grid features – BOINC I.

- One central service per project with limited access that stores work to be processed
- Desktop PCs connect with a simple client application and offer their free CPU cycles
- Client application fetches workunits, processes them, and uploads results to the server
- Mostly the same application is run with many input data sets (parameter study applications)

Grid features – BOINC II.



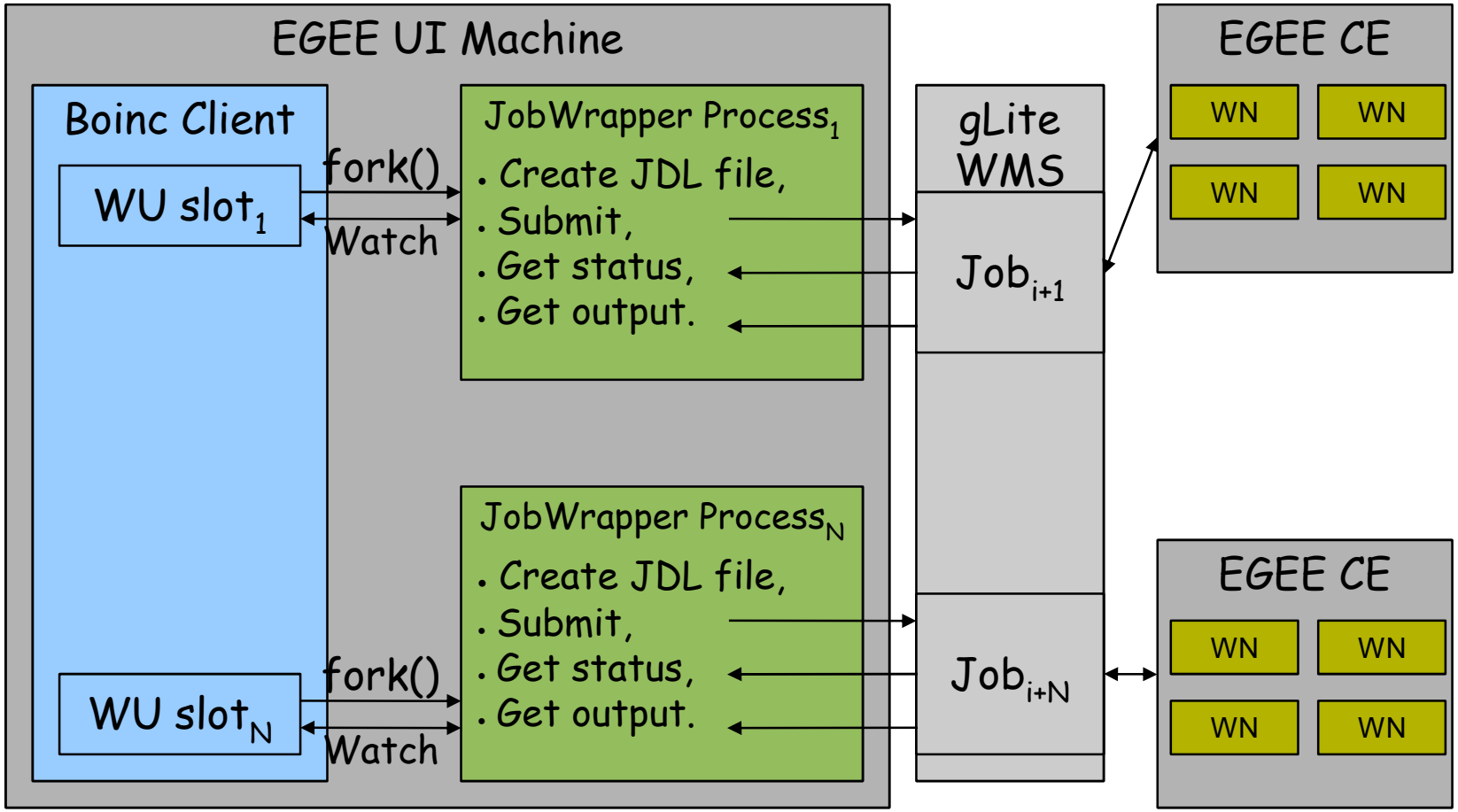
BOINC -> EGEE I.

- Task to be solved:
  - Process BOINC workunits
  - In the EGEE infrastructure
- Develop a bridge that:
  - Can handle BOINC workunits
  - And is able to create EGEE jobs from the workunits, and run them in EGEE

BOINC -> EGEE Possible solutions

- Agent-based execution:
  - Send BOINC clients to EGEE
  - BOINC client connects to BOINC server to fetch work and report results
- Wrapping workunits execution:
  - Send BOINC applications to EGEE
  - Fetch BOINC workunits, and execute them in an EGEE job, finally report results

BOINC -> EGEE First version



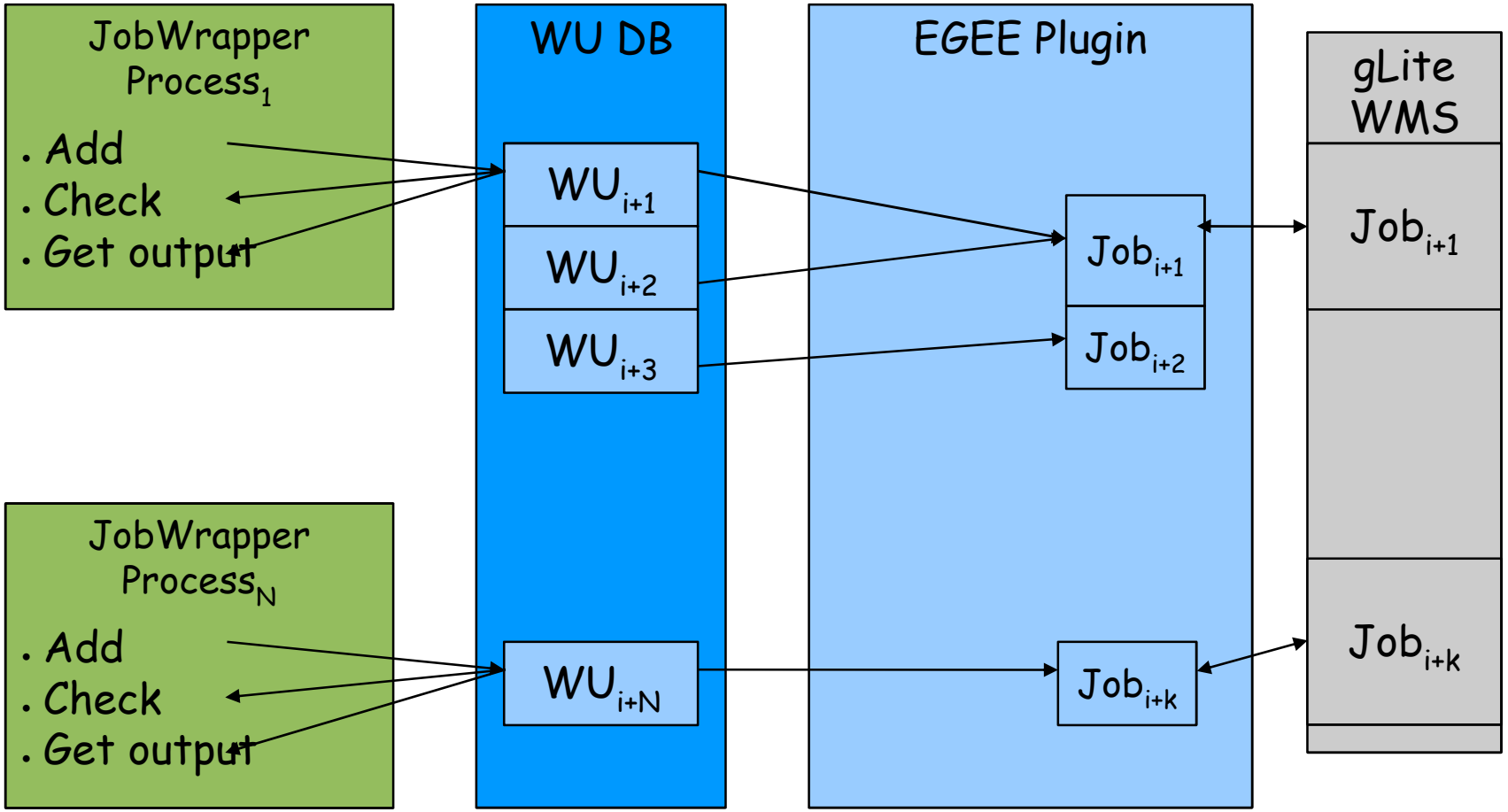
Lessons learnt

- EGEE WMS doesn't like periodic interaction (needs restart every x hours)
  - Workunits should be gathered
- EGEE Operations sometimes fail
  - If there is a failure, retry the operation at most three times
- Ways to improve:
  - Interact with WMS the least possible times
  - Handle pack of jobs instead of individual jobs

Improved BOINC → EGEE bridge

- Collect jobs originating from BOINC:
  - Place them in a queue
  - New jobs in the queue are periodically handled by an EGEE plugin, that
  - Uses Collection possibilities of EGEE to submit many jobs in one request
- This way the usage of the WMS is reduced

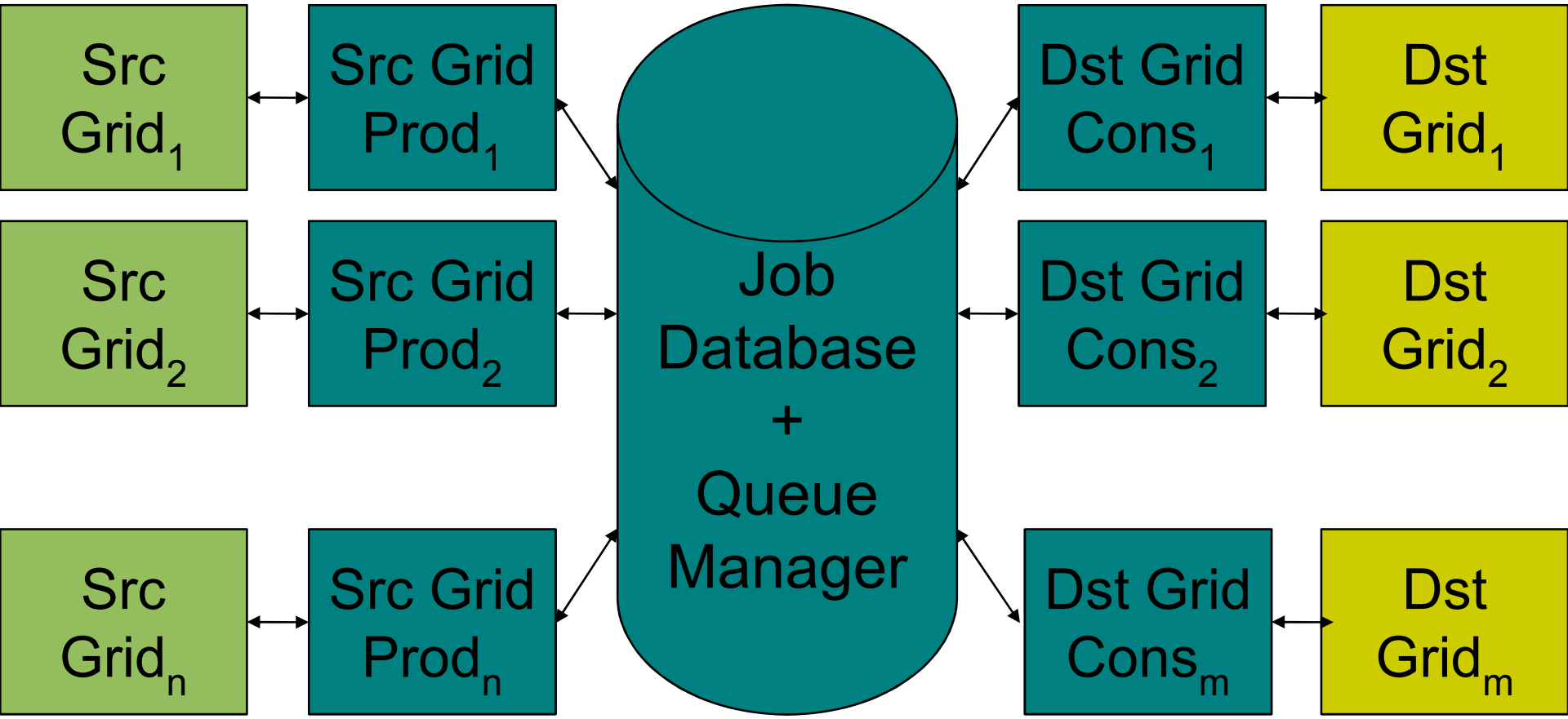
Improved bridge architecture



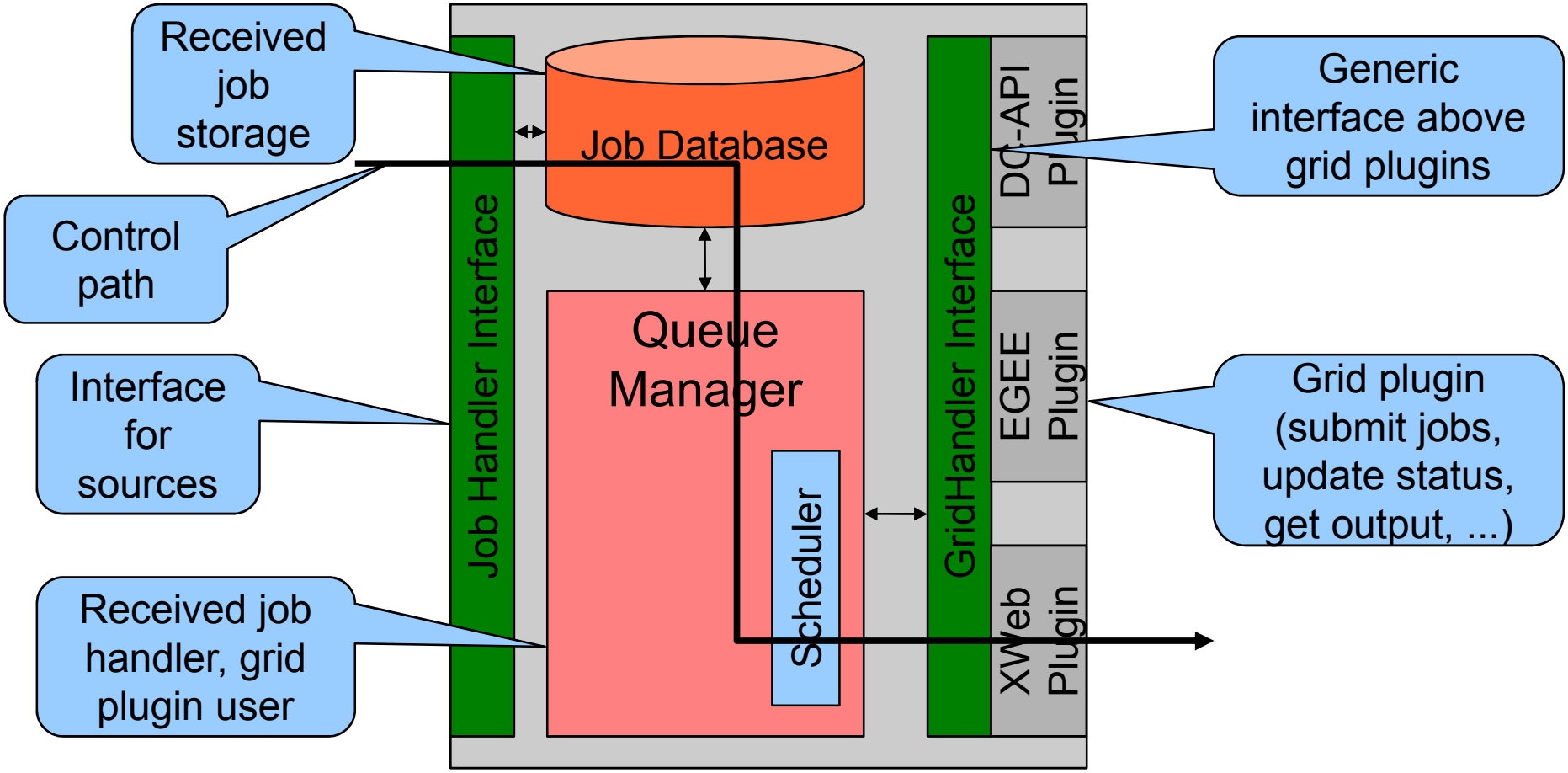
Bridge generalisation

- Jobwrapper → Source grid producers
  - Produce jobs originating from source grids
- WU DB → Job database + Queue Manager
  - Stores job produced by source grid producers
  - Selects jobs for execution
- EGEE plugin → Destination grid consumers/plugins
  - Execute jobs in the job database in the supported destination grids

Generic Grid-Grid Bridge (3G Bridge)



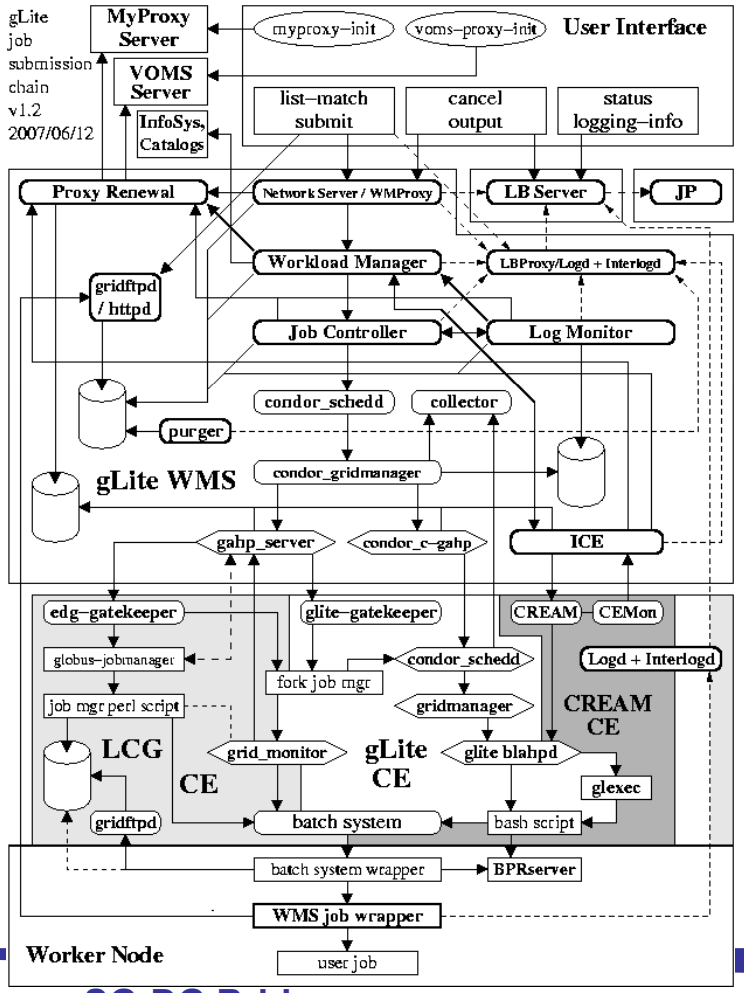
Job Database + Queue Manager



EGEE -> BOINC I.

- Transparent method for running EGEE jobs on BOINC DGs
- User interacts with EGEE using EGEE tools
- 3G Bridge used to transfer jobs to BOINC
- Special CE created to catch EGEE jobs
- EDGeS AR is used to check validity of applications

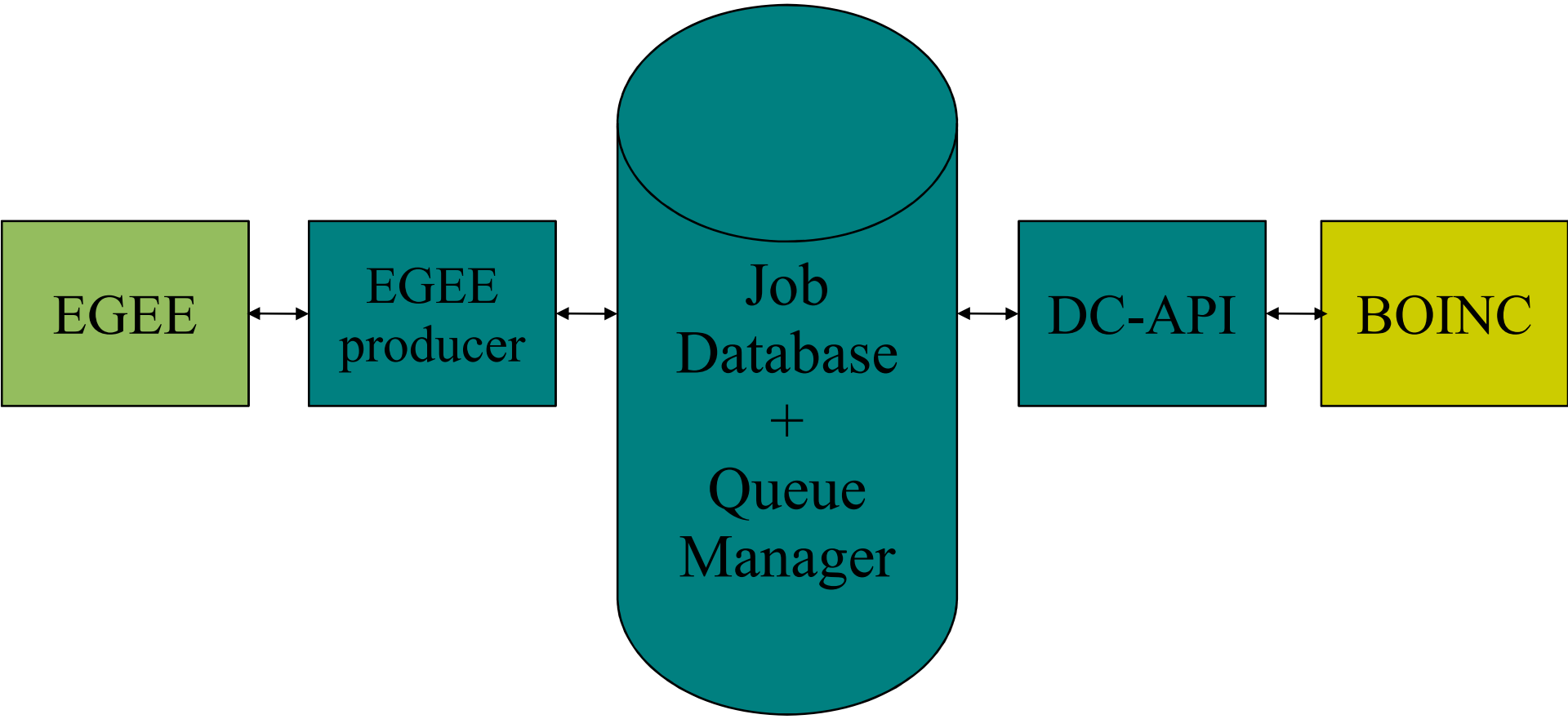
EGEE job submission



EGEE -> BOINC extension key concept

- Create a new GRAM jobmanager/LRMS:
  - For every job, we get the job info (executable name, input files used) from the wrapper script submitted by the EGEE WMS
  - Add the job to the 3G Bridge
  - Report logging using DGAS/glite-lb-logevent
  - The 3G Bridge uses a DC-API plugin to run the job on BOINC

3G Bridge: EGEE → BOINC



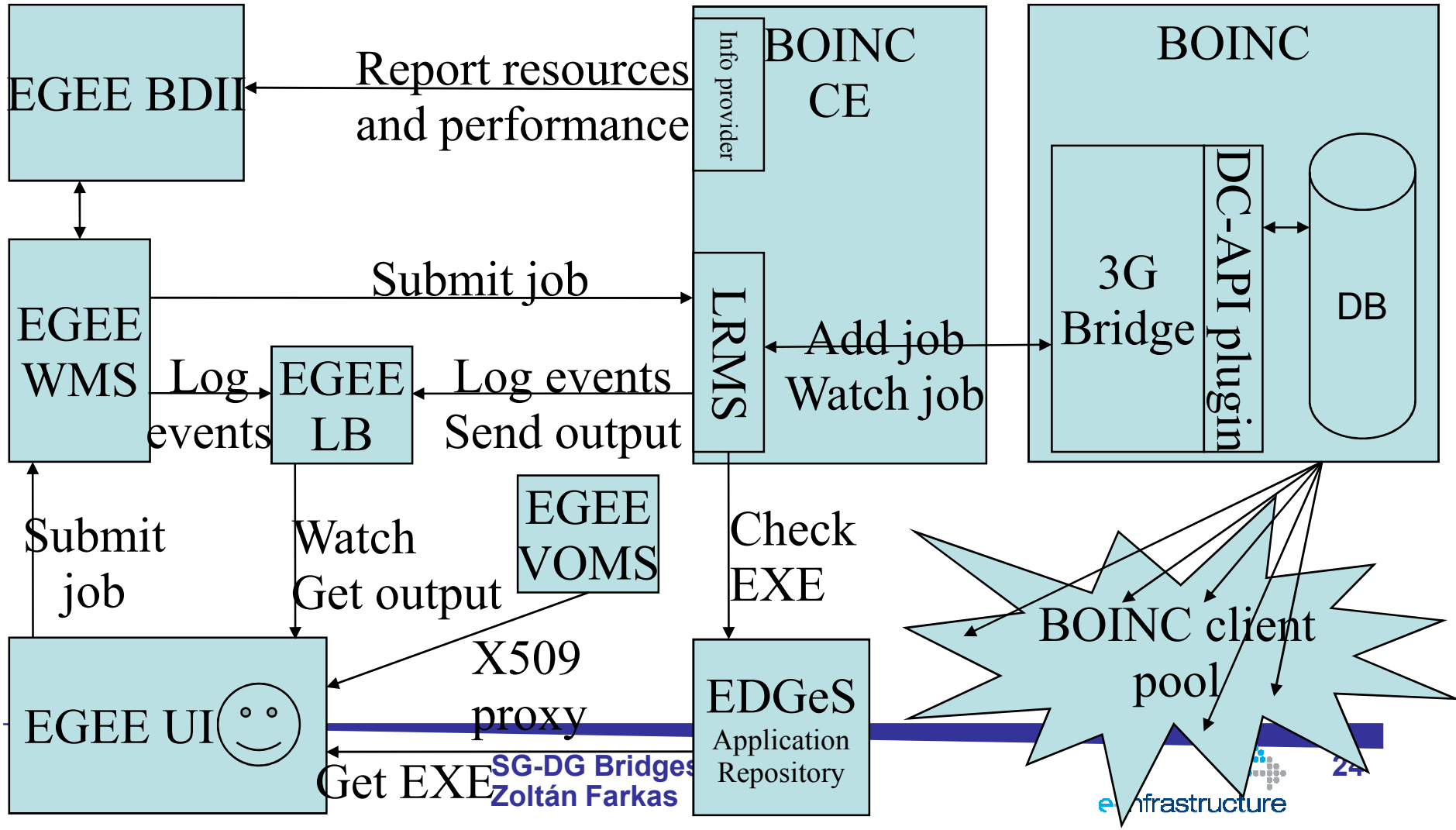
EGEE producer Overview

- A new GRAM jobmanager
- Gets job information from the WMS wrapper script
- Checks if exe is a validated one
- Checks if exe is supported by one of the attached BOINC (or XtremWeb) projects
- Gets files from WMS
- Adds job to 3G Bridge job DB
- Polls status of jobs in 3G Bridge DB
- Gets results from 3G Bridge and uploads to WMS

DC-API plugin

- Use DC-API to generate BOINC WUs
- Jobs are read from the 3G bridge DB
- 3G DB entries are updated on events
- The plugin has already been implemented for the CancerGrid system

EGEE -> BOINC: Overview of the system



3G Bridge Data Handling Issues

- EGEE applications might use huge input files
- For data distribution, ADICS/ATTIC can be used (developed by Cardiff University)
- 3G Bridge uses ATTIC to publish selected files (recent development)
- ATTIC support in DGs (BOINC/XtermWeb) is work in progress

Conclusions

- The 3G Bridge architecture:
  - Offers transparent way for running jobs on BOINC for EGEE users
  - Offers transparent way for running BOINC jobs on the EGEE infrastructure
- Has been extended to support P-GRADE Portal parameter study applications (thus special case of remote file handling is solved)
- Initial support for handling large amount of data
- Existing 3G Bridge plugins: EGEE, DC-API, XtremWeb
- Future 3G Bridge plugins: OurGrid