3G-Bridge: Bridging the Gap between BOINC and Other Systems

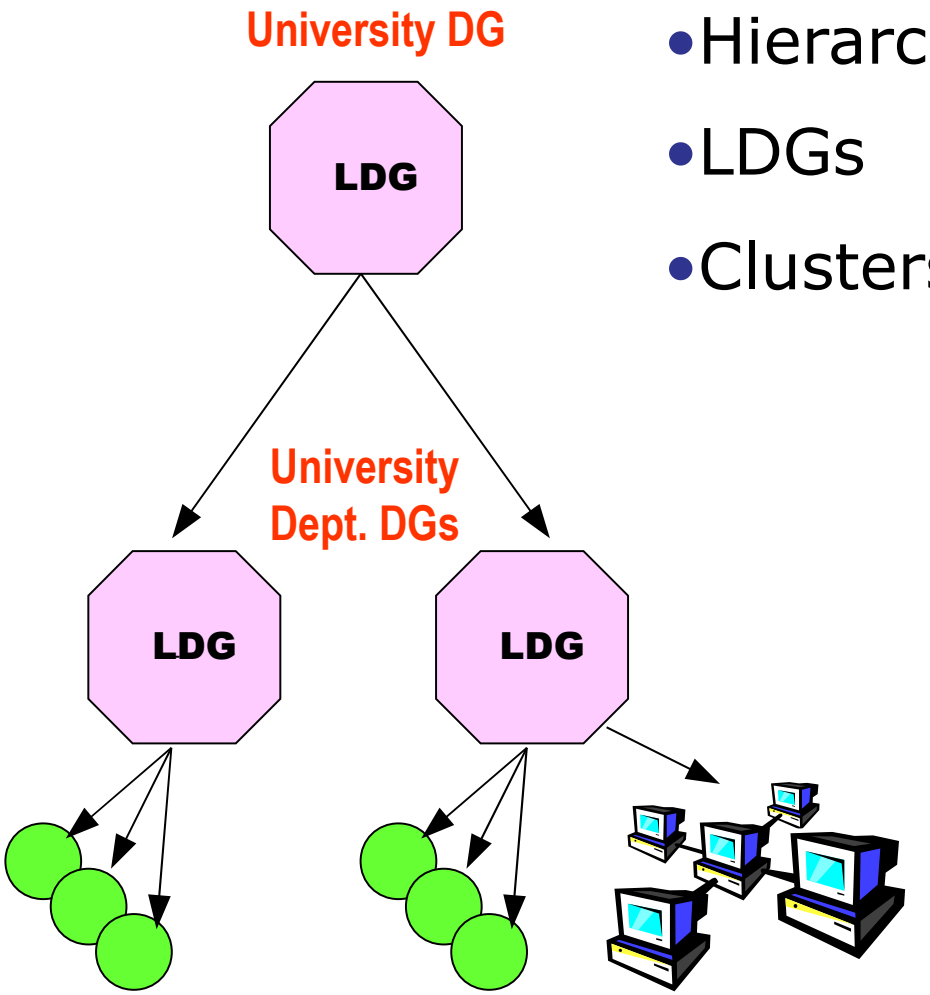
BALATON Zoltán
balaton@sztaki.hu
MTA SZTAKI LPDS

22 October 2009

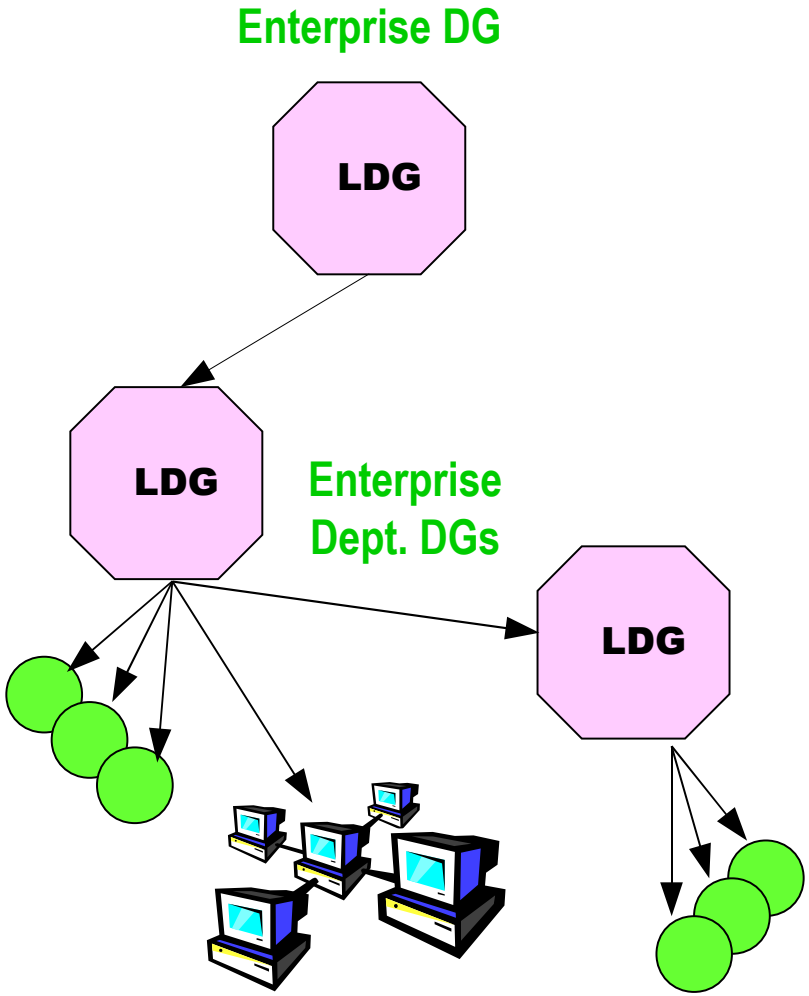
SZDG: SZTAKI Desktop Grid

- SZDG (www.desktopgrid.hu) extends BOINC for Desktop Grids and local usage:
  - Easy setup: Debian packages for the server
    - For Volunteer Computing: Global (used in public projects)
    - For Private/Local Desktop Grids: Local (LDG skin)
  - Simplify writing applications: DC-API
  - Support porting legacy applications: GenWrapper
  - Security enhancements: certificates, virtualisation
  - Allow setups peculiar to DGs:
    - clusters as resources, hierarchy, bridging to/from other systems, connecting portals

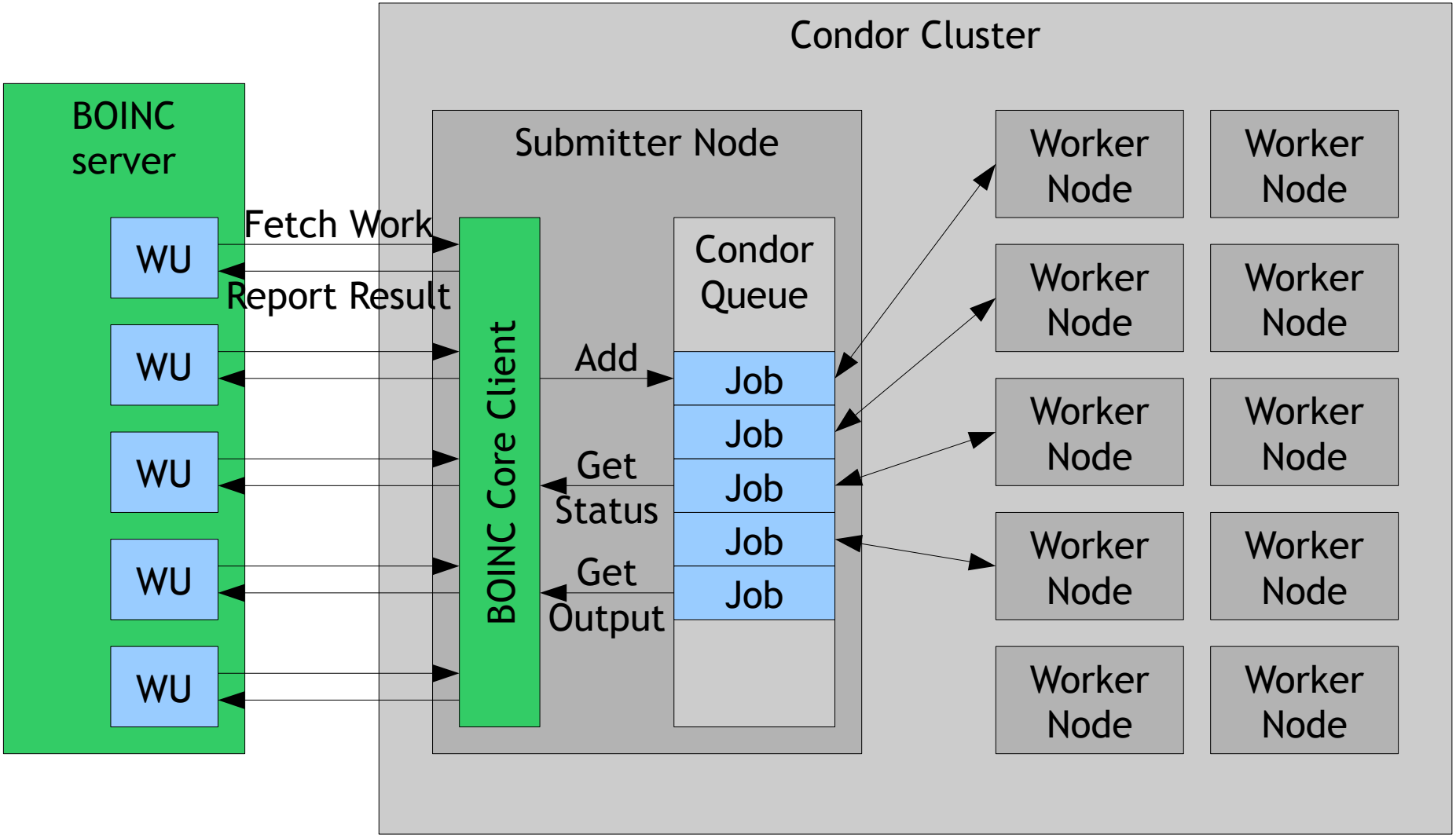
SZDG Setups



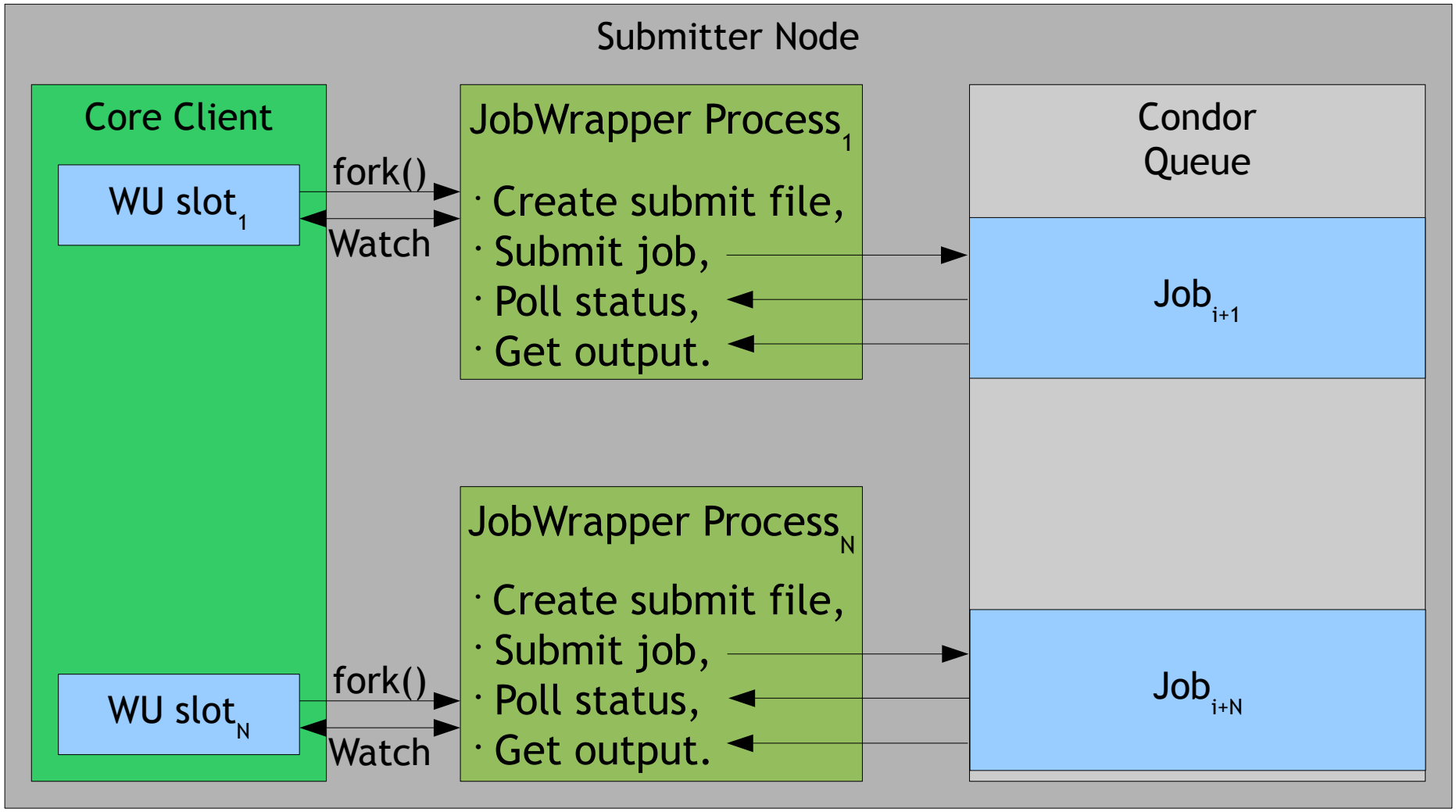
- Hierarchy
- LDGs
- Clusters



Clusters in SZDG

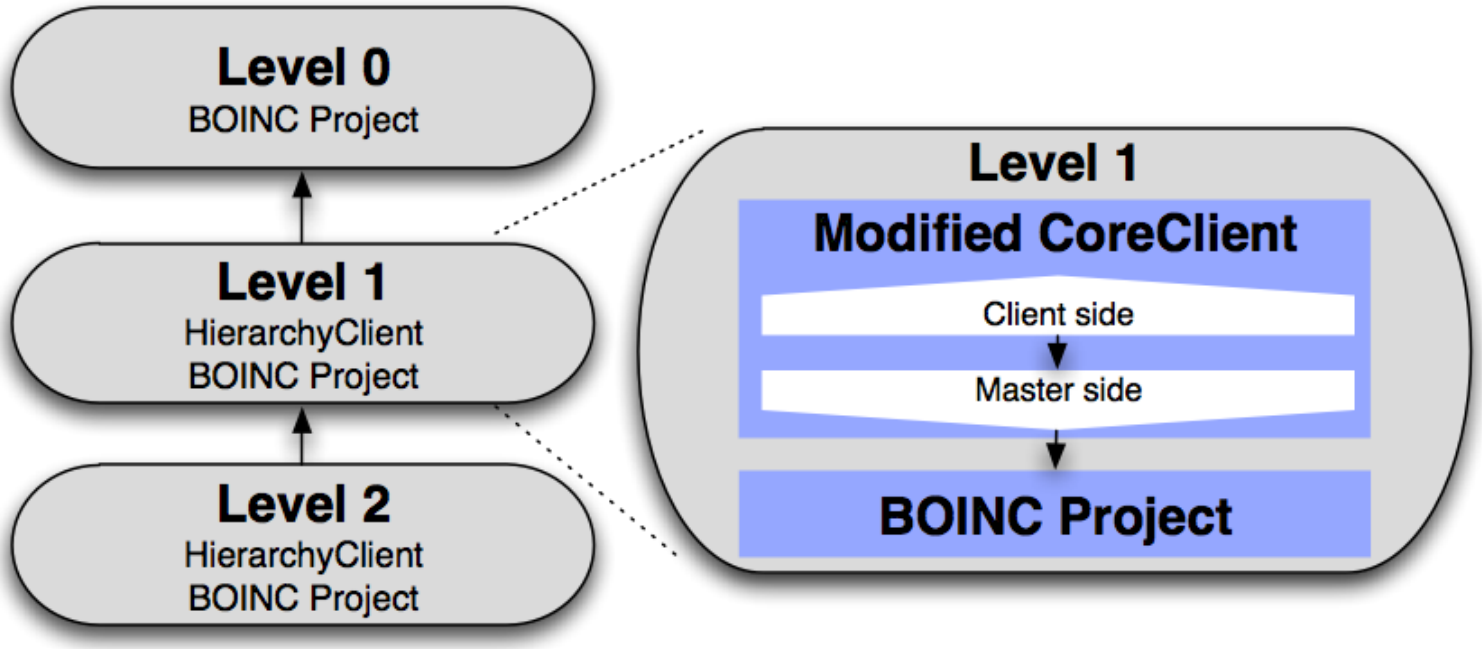


The JobWrapper Client



Hierarchy in SZDG

- The basic idea is very similar
(apart from additional security and application registration related changes)
Also implemented in a modified Core Client



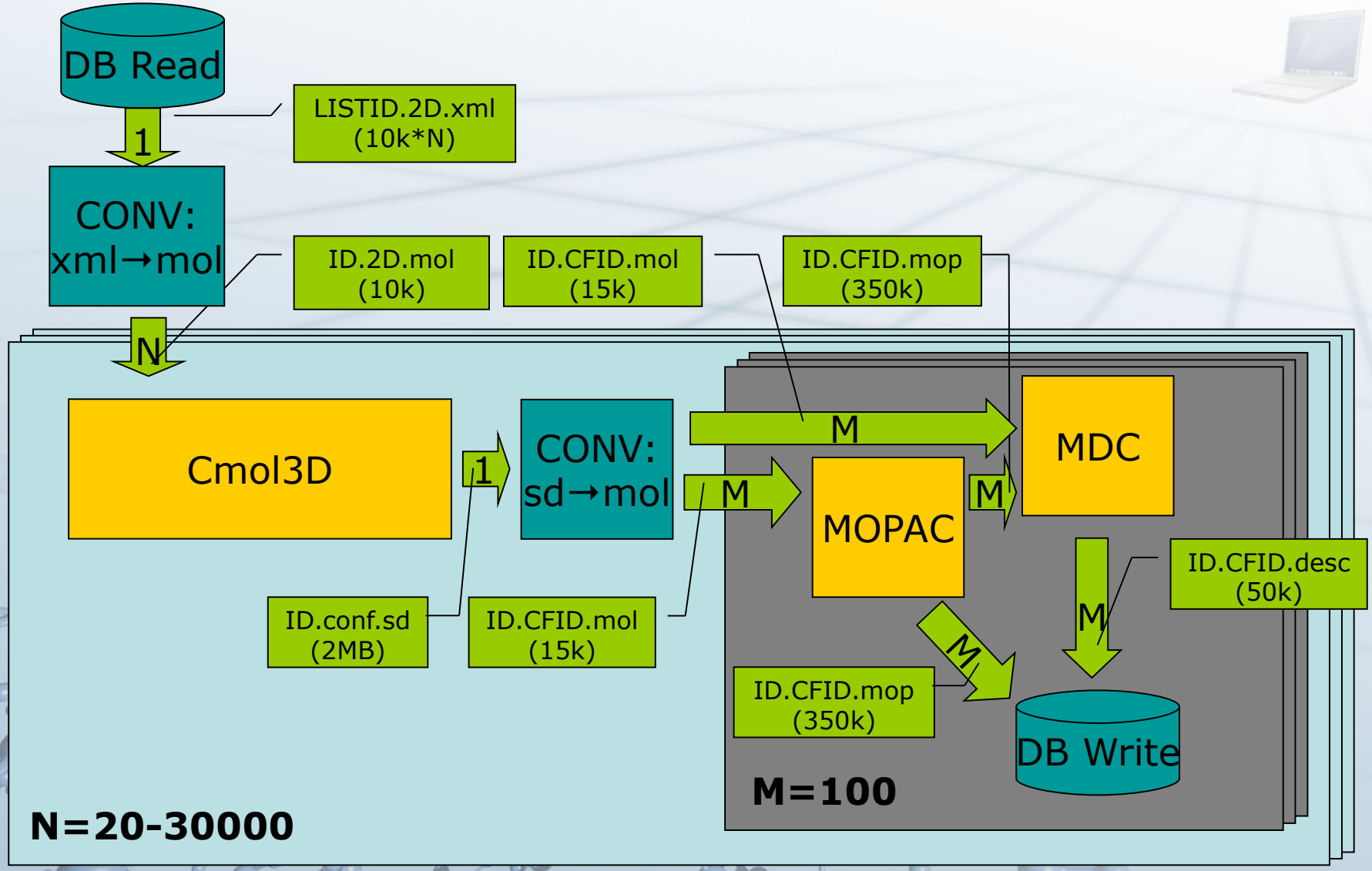
The CancerGrid project

CancerGrid

- EU FP6 STREP
- Title: **Grid Aided Computer System for Rapid Anti-Cancer Drug Design**
- Project period: 2007 – 2009
- Partners: 6 academic and 5 industrial from 7 countries
- Goals:
  - Developing **focused libraries** with a high content of anti-cancer leads, building **models** for predicting various molecule properties
  - Developing a **computer system** based on grid technology, which helps to accelerate and automate the **in silico design** of libraries for drug discovery processes

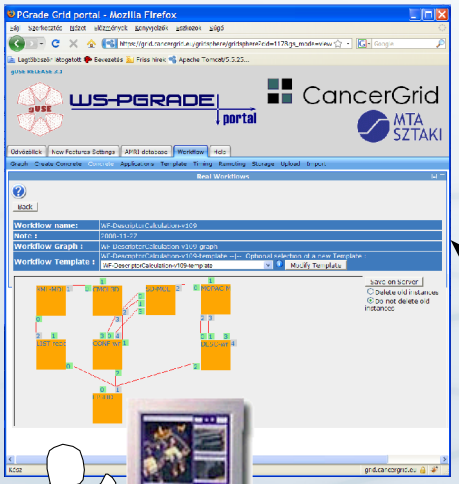
Sample CancerGrid Workflow: Molecule Descriptor Calculation

CancerGrid

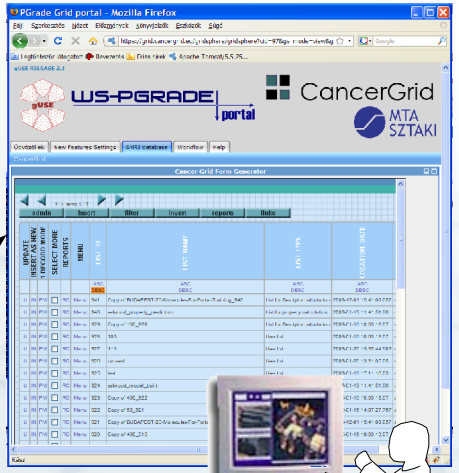


CancerGrid Overview

CancerGrid



executing workflows



browsing molecules

gUSE

DG jobs

Local jobs

Queue Manager

Job 1
Job 2
Job N

Local Resource

BOINC server

WU 1
WU 2
WU M

Portal Storage

Portal and Desktop Grid server

BOINC client

GenWrapper for batch execution

WU X
WU Y

Legacy Application

Private DG of partners

molecule database

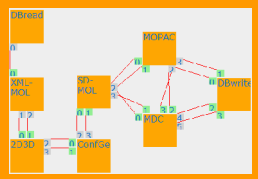
Molecule database server

The CancerGrid Architecture

CancerGrid

DG Server

gUSE WS-PGRADE (User IF)



(WF repre-sentation)

gUSE WFI (Workflow interpreter)

gUSE DG Submitter

Job descr.

Job Database
(**Queues per Application**, Description of Jobs: Args, I/O files)

Job descr.

Queue Manager

Scheduling policy

Batch creation

DC-API master

WU

BOINC Task DB

BOINC Server Components

Sche-duler

Data server

BOINC client

DC-API cl.

GenWrapper for batch execution

Legacy Application

BOINC client

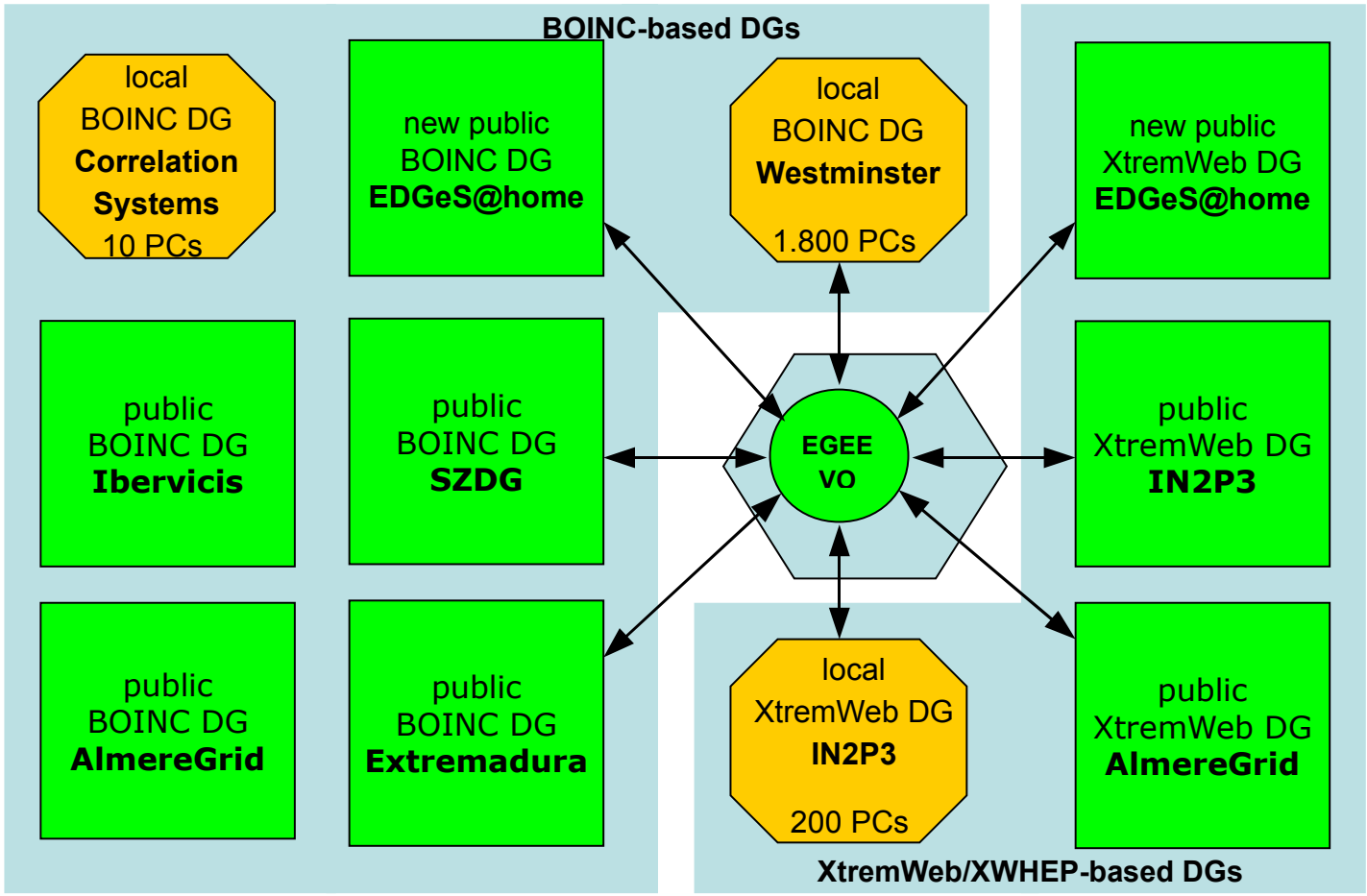
DC-API cl.

GenWrapper for batch execution

Legacy Application

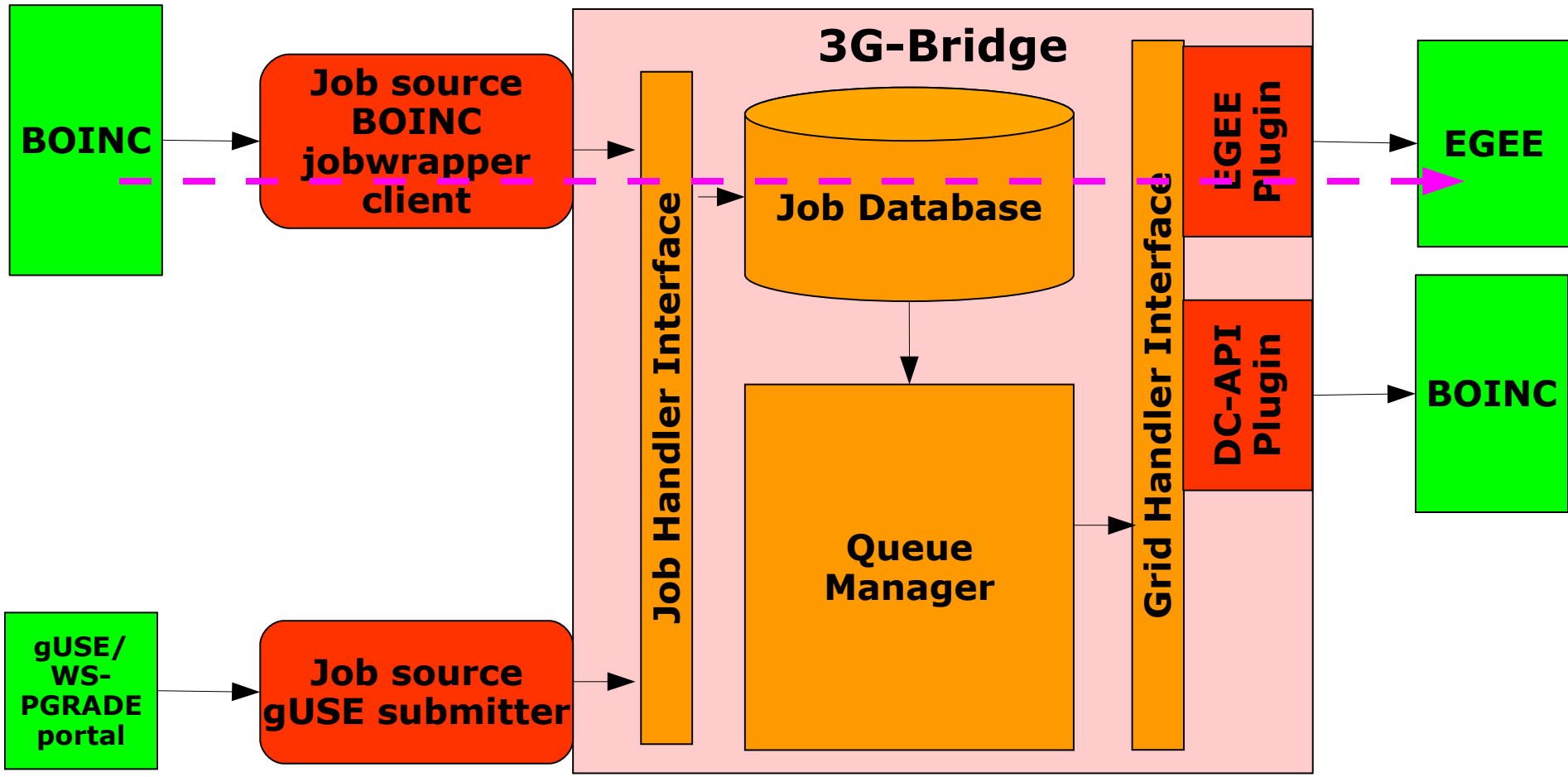
gUSE | gUSE-DG integration | DesktopGrid

The EDGeS integrated SG–DG infrastructure

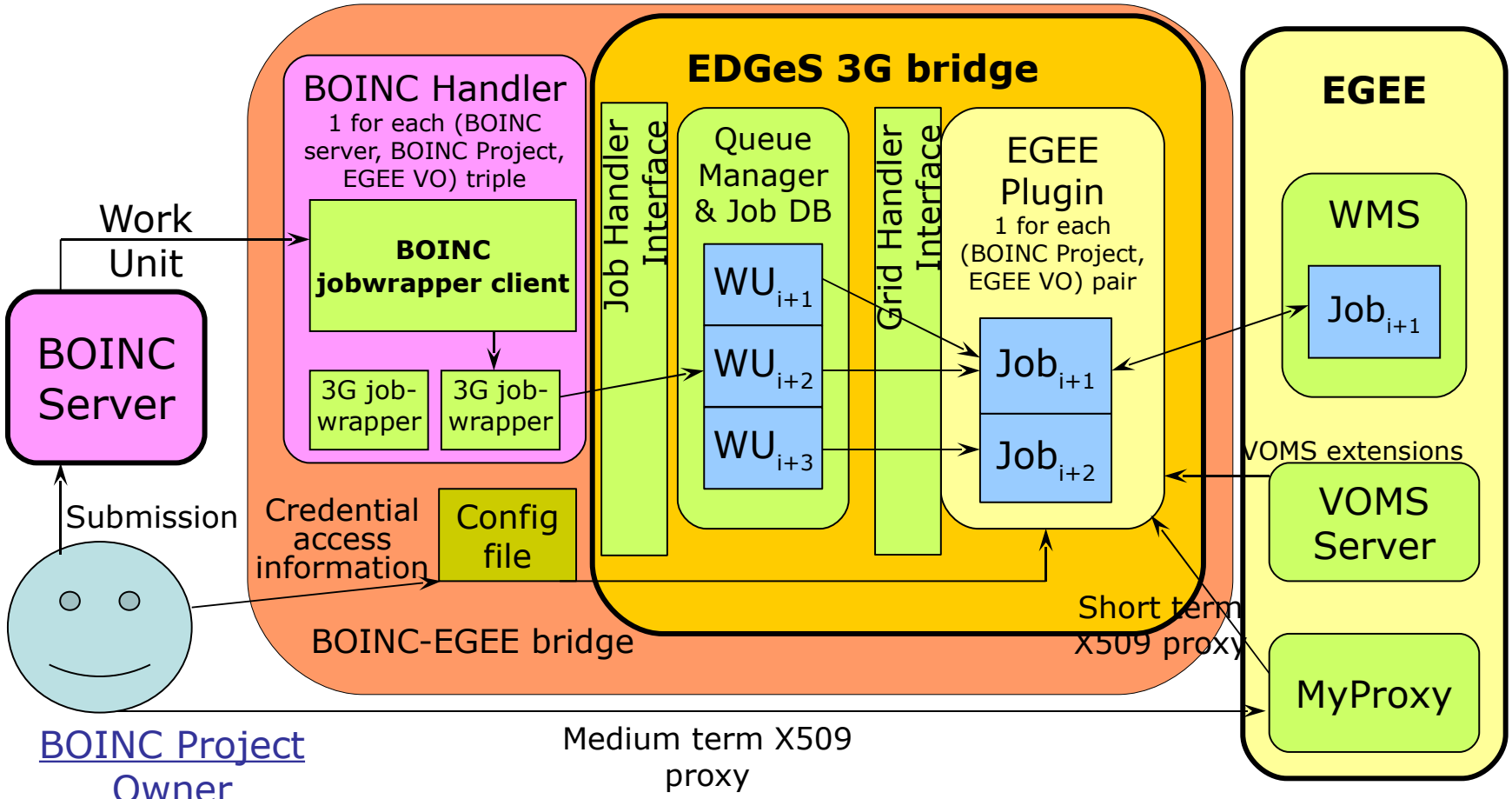


CancerGrid

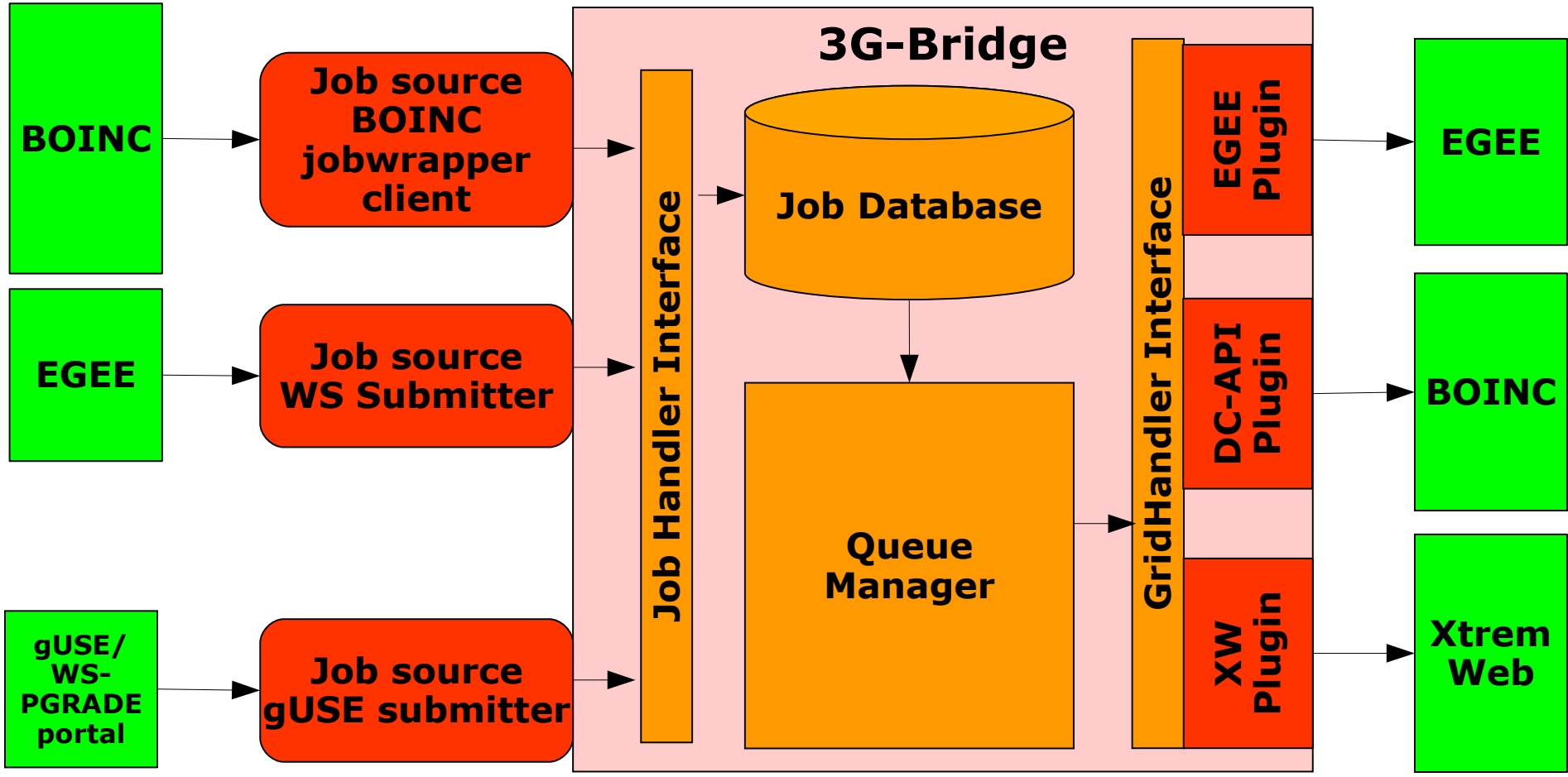
The Generic Grid-Grid (3G) Bridge



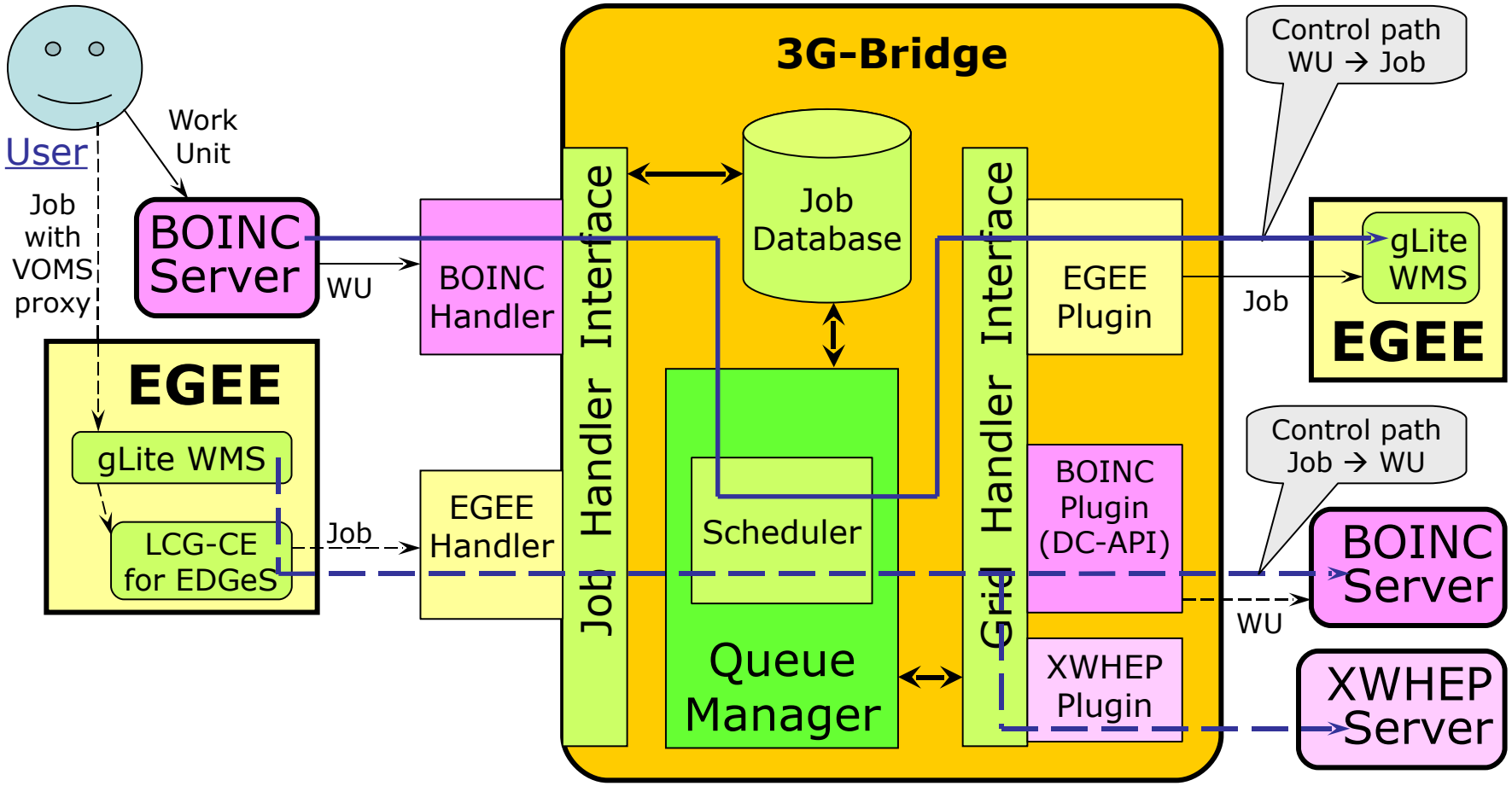
Bridging BOINC ⇒ EGEE



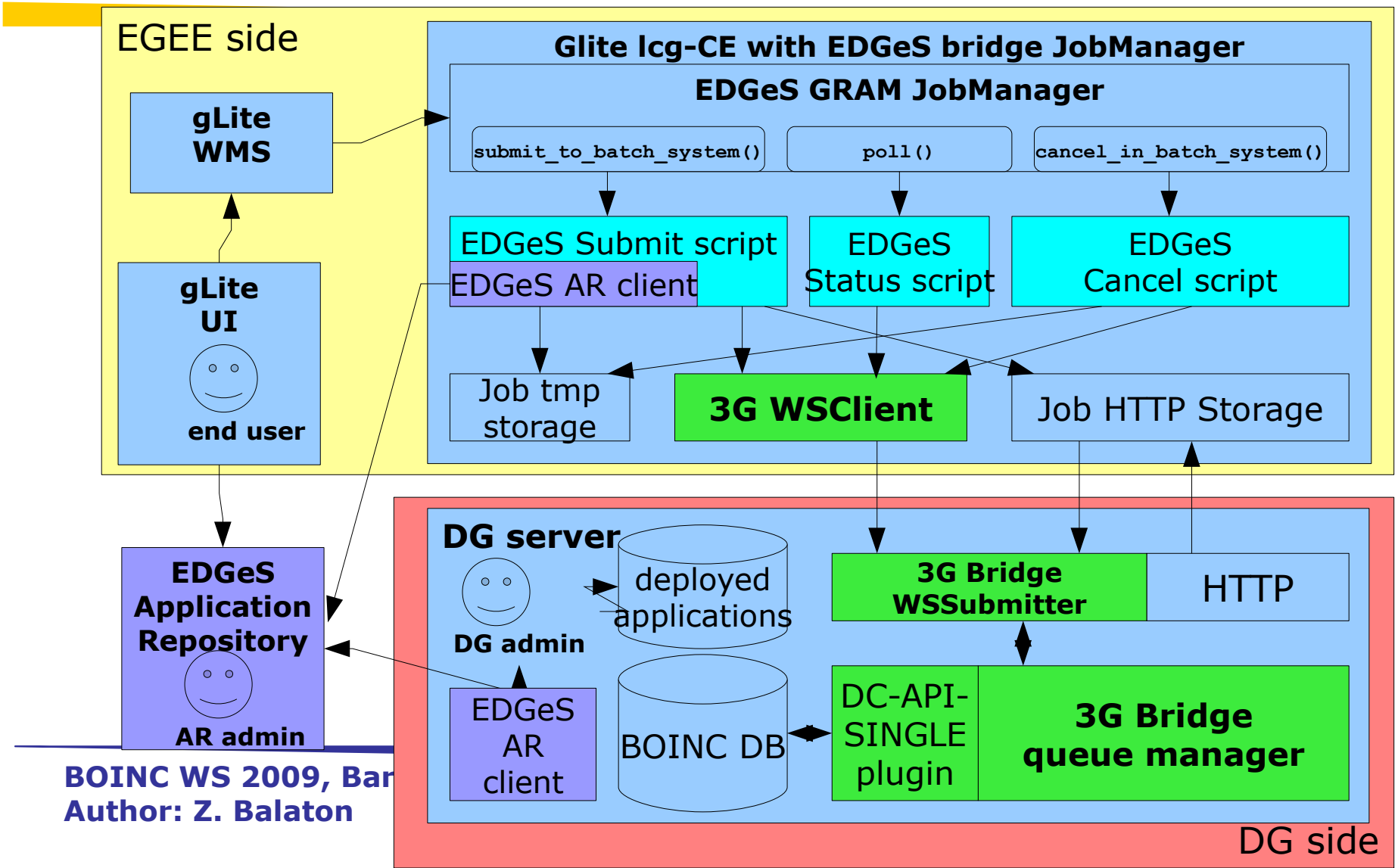
The Generic Grid-Grid (3G) Bridge



Bridging between EGEE, BOINC and XW
Architecture of the EDGeS 3G-Bridge



Bridging EGEE ⇒ BOINC



Using the 3G wsclient

Usage:
wsclient [OPTION...] - Simple client for the 3G Submitter service

Application Options:

-e, --endpoint=URL	Service endpoint
-m, --mode=(add|status|delete|output|finished)	Operation mode

- Add: submit jobs, Status: get status, Delete: remove job, Output: get results, Finished: Get list of jobids in finished state

Query/manipulation options:

-j, --jid=UUID	Job identifier
-f, --jidfile=FILE	Input file holding the job identifiers. '-' for STDIN

Using the 3G wsclient

Job submission options:

-n, --name=NAME	Name of the algorithm to execute (app)
-g, --grid=NAME	Grid where to submit the job to (queue)
-a, --args=ARGS	Command-line arguments for the job
-i, --in=NAME=URL	Input file specification
-o, --out=NAME	Output file names
--repeat=NUM	Repeat the operation this many times

Examples:

```
$ wsclient -m add -e http://server:8084 -g DG -n dsp -a "-f 22 -i 20 -p 723 -n pools.txt" -i pools.txt=http://dataserver/p/pools.txt -o cost.txt
$ wsclient -m status -e http://server:8084 -j 5fae62b7-d91c-4e94-ac9d-b08998224a54
$ wsclient -m output -e http://server:8084 -j 5fae62b7-d91c-4e94-ac9d-b08998224a54
```

Summary of 3G-Bridge

- Generic, extensible solution for bridging different grid systems
- Can handle multiple queues and plugin instances
- Multiple job sources available:
  - Job Handler Interface (mysql API with DB schema)
  - Web Service interface (SOAP+HTTP, used by the EDGeS EGEE ⇒ DG bridge)
  - CLI (local: injector, remote: wsclient)
  - BOINC via jobwrapper client and 3G jobwrapper
  - gUSE submitter

Summary of 3G-Bridge (cont.)

- Multiple output plugins available:
  - BOINC via DC-API and DC-API-SINGLE
  - XtremWeb/XWHEP via XTREMWEB plugin
  - EGEE through WMS via the EGEE plugin
- Currently in use for production in:
  - CancerGrid and University of Westminster:
    - gUSE/CLI ⇒ BOINC (with batched WUs)
  - EDGeS:
    - BOINC ⇒ EGEE
    - EGEE ⇒ BOINC and XtremWeb/XWHEP

Future and in progress work

- We plan to extend 3G-Bridge with:
  - BES job source for standardised grid middlewares
  - ARC and Unicore grid middleware support
  - Output plugin for clouds (in prototype)
  - Output plugin for OurGrid P2P middleware (in progress)
  - Improve existing job sources and plugins (e.g. better implementation of the BOINC job handler to avoid jobwrappers)