Improved 3G Bridge scalability to support desktop grid executions

Zoltán Farkas
zfarkas@sztaki.hu

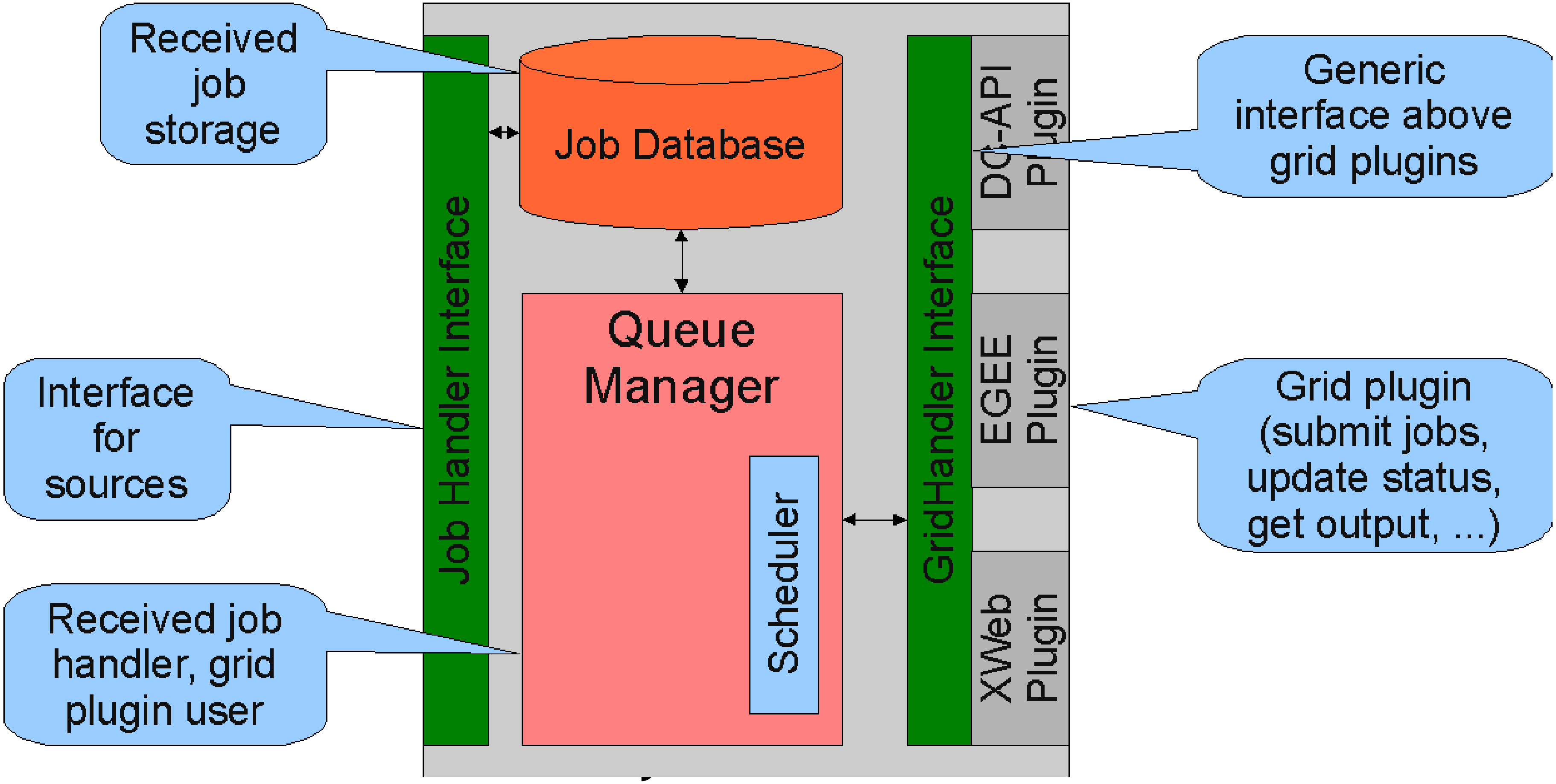
MTA SZTAKI LPDS

09/01/2010

Outline

- Introduction
- The scalability problem
  - Limited number of jobs
  - Unnecessary data transfer overhead
- Job scalability improvement
- Eliminate unnecessary data transfers
- Summary, future work

Introduction



Received job storage

Interface for sources

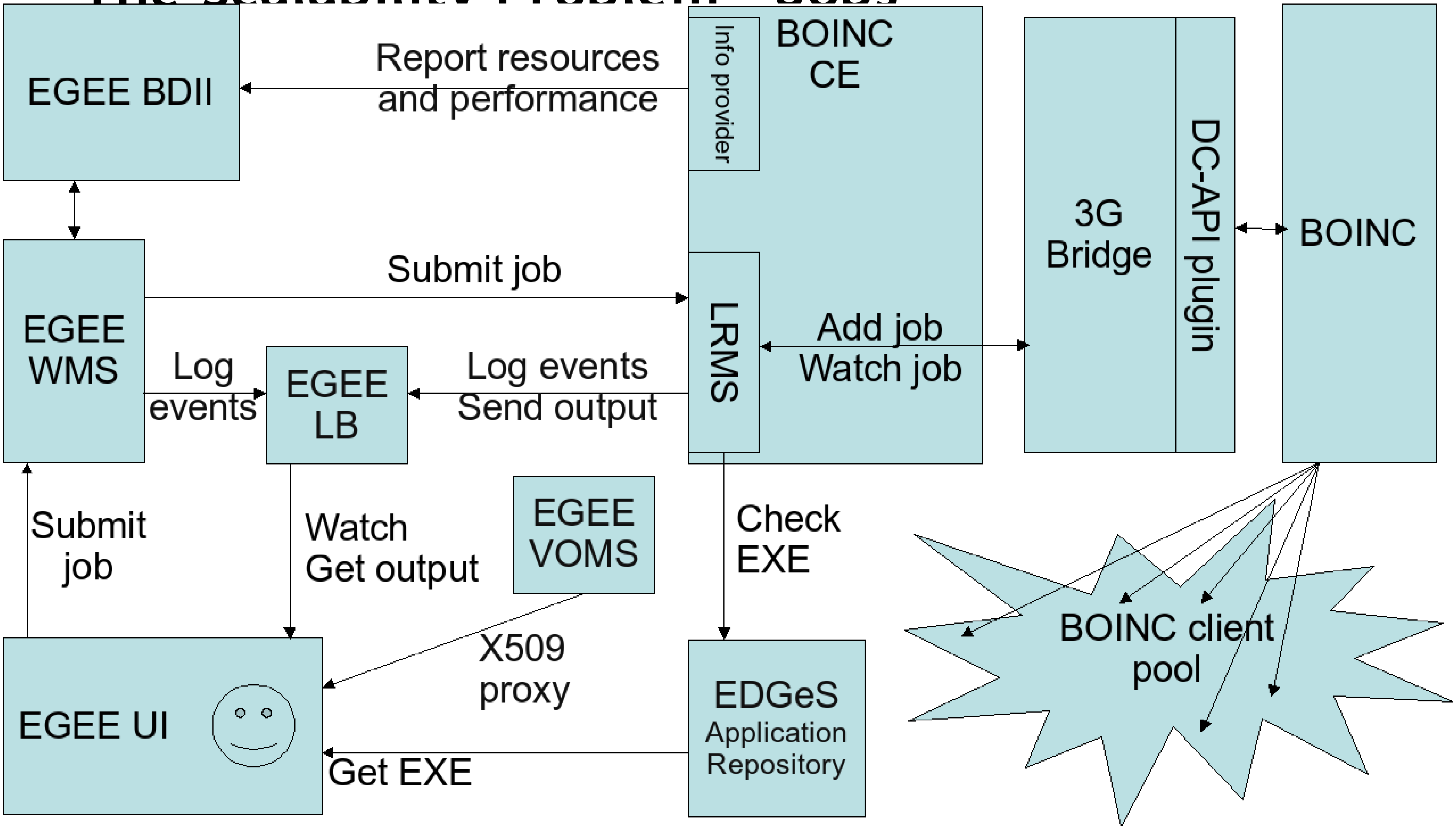
Received job handler, grid plugin user

Generic interface above grid plugins

Grid plugin (submit jobs, update status, get output, ...)

- → In 3G Bridge and/or DC-API

The Scalability Problem - Jobs



- → EGEE user receives job results here (at least 5 min)

Job Numbers - Possibilities

- Job batching
  - User creates a package of jobs
  - Submitted to the SG/3G Bridge as a single job
  - Results in multiple jobs on the target grid
- Pilot submission
  - User runs some kind of pilot service somewhere
  - Submits a big number of pilot jobs to the target grid
  - Adds jobs to run to pilot service

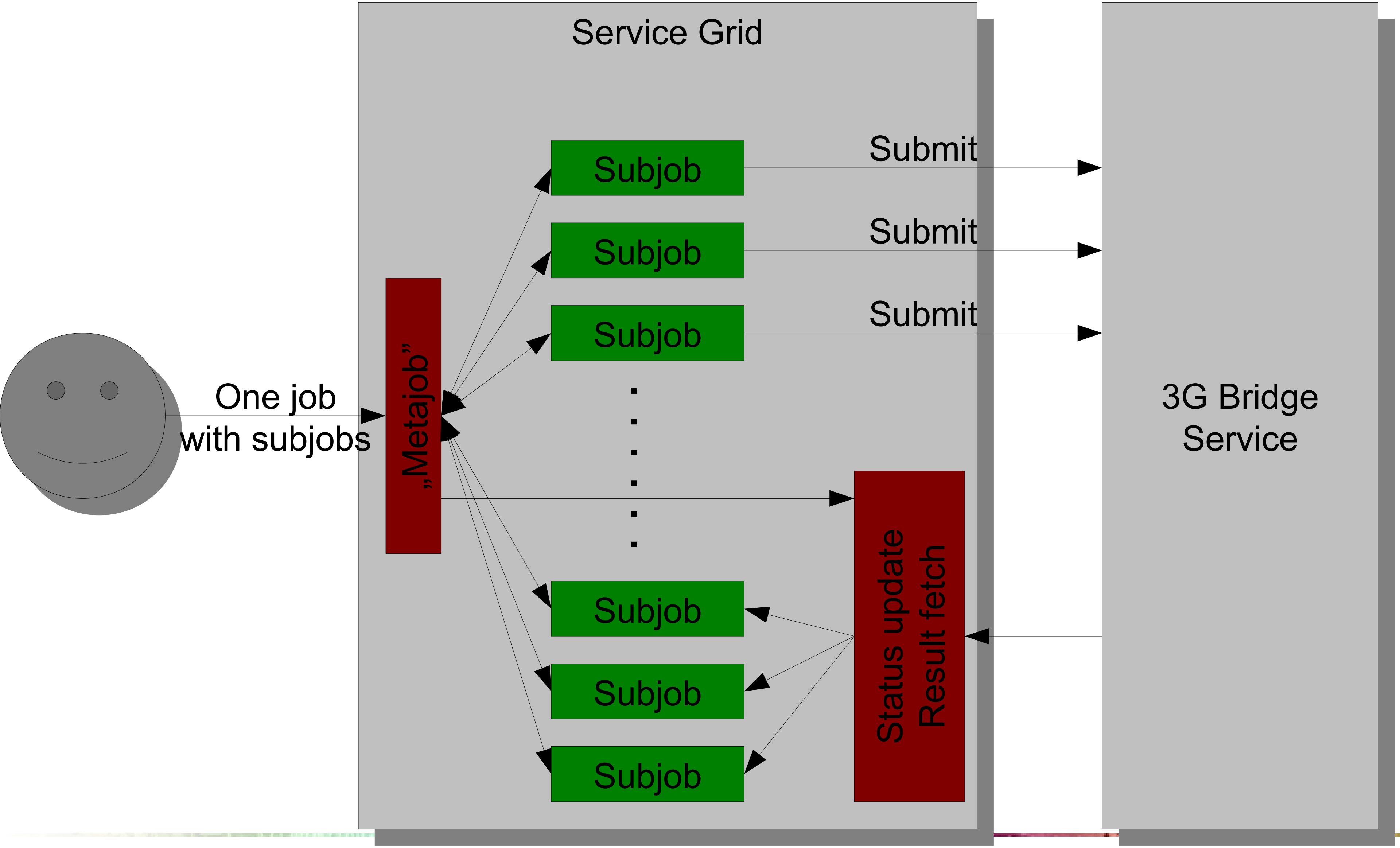
Job Numbers - Pilot Features

- Cons:
  - Needs an additional service (design, implementation)
  - Problematic application checking against AR in EGEE → DG
  - Number of pilot jobs is limited (as sent through the SG)
  - Users have to be aware of how to use the pilot service, have to submit pilots, …
  - Not transparent at all
- Pros:
  - „Immediate” results

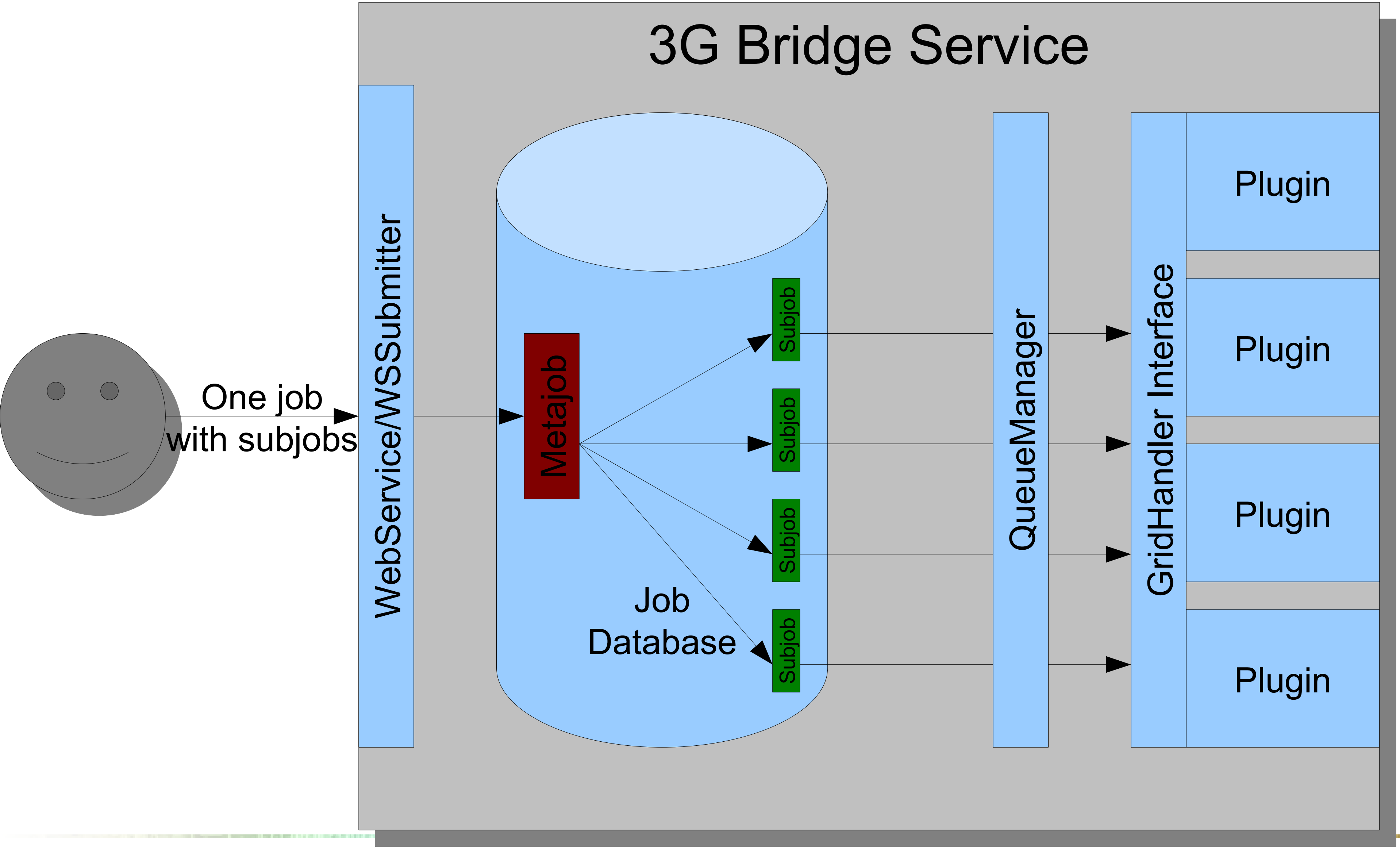
Job Numbers - Batch Features

- Cons:
  - Not really transparent (but still better than pilot)
  - Subresults aren't available as long as at least one subtask is still running
- Pros:
  - Relatively easy implementation on SG side
  - Minor additional user tasks
- Two ways to implement:
  - SG side
  - 3G Bridge side

Job Numbers - Batch (SG Side)



Job Numbers - Batch (3G Bridge Side)



Job Numbers - Batch (Metajob setup)

- Following properties:
  - Executable: filename, URL, MD5, size
  - Input$_1$: filename, URL, MD5, size
  - …
  - Output$_1$: filename(, URL)
  - Arguments: args
- URL:
  - /foo/bar/in – file contents sent using DIME
  - http://foo.bar/in - normal URL
  - x-3gb-list+http://foo.bar/in - contents is list of parametric file URLs, MD5s and sizes used by batch submission

Job Numbers - Batch (Metajob features)

- Supported only through the web service interface (not through MySQL)
- Metajob's status: sequence of subjob statuses
- Any number of inputs may be parametric, the cross product of enumerated files is used to create subjobs

Job Numbers - Batch (Metajob examples)

- Input file 'input1' is parametric one:
  - URL: x-3gb-list+http://foo.bar/input1
  - Contents:
    - http://foo.bar/ins/in1, 2a9....be, 1230
    - http://foo.bar/ins/in2, 4ef.....dd, 4096
    - ...
- Metajob will result in as many subjobs as many input file entries are in x-3gb-list+http://foo.bar/input1
- Subjobs will use different entries that are x-3gb-list+ http://foo.bar/input1

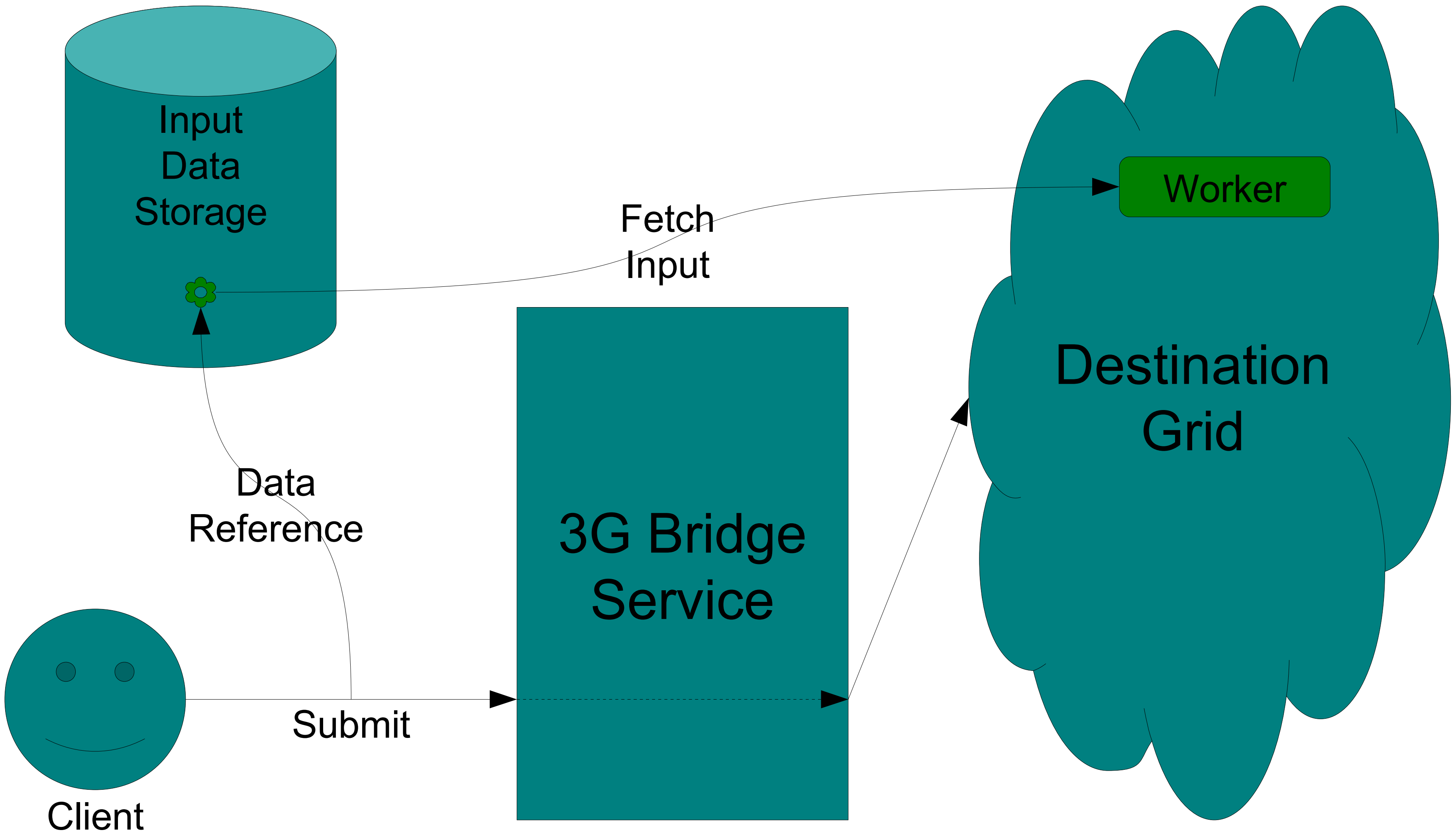
The Scalability Problem - Data transfer

- In case of parametric job submission jobs are likely to use same files:
  - Executables,
  - Common input files, ...
- 3G Bridge fetches and stores all input files:
  - Even if the given file has already been fetched
  - Even if the file's URL could be handled by the target plugin:
    - EGEE, BOINC, XtremWeb: HTTP
- Improvements:
  - Conent-based caching
  - URL passthrough whenever possible

Data Transfer Improvements - URL passthrough

- Additional plugin property: supported URLs
  - 'http', 'ftp', 'gsiftp'
- Change in behavior: download files only if the plugin doesn't support the protocol
- Modified component: WSSubmitter, plugins
- Gain: files are fetched only if really needed
- Requirement: job's owner is responsible for public availability of data

Data Transfer Improvements - URL passthrough



Data Transfer Improvements - Content caching

- Do not fetch/store the same file multiple times
- Fetch:
  - Check the file's MD5 before fetching (if possible)
  - Do not fetch if a file with the given MD5 hash already exists
- Store:
  - If a file has been fetched, check its MD5 hash
  - If a file with the same MD5 hash already exists, use the existing one

Data Transfer Improvements - DC-API/BOINC

- Used by the 3G Bridge for BOINC
- Supports physical input/output files
- Improvement:
  - Add support for HTTP URLs in case of WU input files
  - Modified components:
    - DC-API
    - BOINC: tools/backend_lib.cpp – accept URL, MD5, nbytes in input template

Data transfer improvements measurements

- 3G Bridge WS interface URL passthrough tested
- Scenarios:
  - 10000 jobs, {4/512b, 2/10k, 1/1M files}
- Old version (CPU time/elapsed time):
  - 32s/124s, 30s/91s, 60s/108s (for 1000 jobs for 1M)
- New version:
  - 3s/31s, 3s/30s, 2s/20s (for 10000 jobs for 1M)
- Througput of WS interface increased notably

- TODO: measure DC-API/BOINC throughput

Summary

- Future work: finish implementation :)
- Job throughput increase in EDGeS/EDGI through:
  - URL passthrough wherever possible
  - Contant-based data caching
  - Metajob support in 3G Bridge
- Affected components:
  - 3G Bridge (WS, plugins)
  - DC-API (to support “remote” files)
  - BOINC (to support extended workunit input template)

Thank you for your attention!

Questions?