ILCDIRAC

A few words on future plans

S. Poss

CERN, LAPP

Feb. 1, 2013

Outline

Whizard 2 support: process factory?

Idea:

- Propose a production service to users
- WHIZARD 2 can take a process definition and provide a binary
- Integrate this in the DIRAC production system (or interface to it)

W2: What would be needed?

- Web interface for fanciness, and CLI for gurus
- DIRAC service to listen to the queries
- Agent to run WHIZARD2 (submit jobs to dedicated resources?) and produce the binary
- Agent to monitor everything
- Agent to submit integration jobs (for the grid files) to dedicated resources w/o CPU time limit (ex. desktop grid)
- Notification to notify the user about completion (already in DIRAC)
- Interface to DIRAC Transformation service to produce the events
- Some state machine

W2: Proposed interface

What would be required from the user:

- Process definition (a la WHIZARD 2, in sindarin), also the Machine for lumi spectrum
- Generator level cuts (if any), either in whizard or using the StdHepCut framework
  - Will have to find a way to include user code easily (maybe include it in the whizard compilation procedure)
- Simulation parameters
  - Detector model (sidloi3, ILD_o1_01, CLIC_ILD_CDR, etc.),
  - Steering files
  - Software version (detector type should be enough to get the software type: Mokka, SLIC, etc.)
- Reco parameters
  - Steering files
  - Software version: again, Marlin/LCSIM chain can be determined from detector model
  - Adding overlay: need to add check of availability of background events for machine/detector/energy

W2: How would it work?

- The service stores the process request: what has to be done, how, etc.
- Submission Agent picks up any New request, checks if the process is already there (implies clever look up in DB)
  - If YES, then create transformation (or extend existing one, which needs storing also Transformation IDs), send notification to the user about the prod ID
  - if NO: compile the whizard binary (with whizard2), compile cuts (svn up, compile), add them as a software in the DIRAC CS, create the transformation, notify the user about TransformationID to monitor.
  - Notify the user in case of (compilation) failures
  - Could use submitter credentials for the jobs
  - Need a way to “guess” the time it takes to generate N events to maximize CPU efficiency

W2: How would it work?

- Monitoring Agent (on Running requests):
  - When number of events requested in given Request achieved, notify the user that the job is done, and give him the LFN directory or meta data query corresponding (by mail).
  - Extend remaining productions to reach desired number of events: needs knowledge of current state
  - Should be able to pick up cases when the CPU time per job is reached, and create splitting tasks as well as merging tasks
  - Check failure rate and stop if threshold reached (+ notify user/admin)
- Cleaning Agent (on Cleaning, deleted, etc):
  - Remove the whizard binary from the GRID
  - Issue the proper request to the Transformation System (Clean, Archive)
  - Clean tables accordingly.

Of course all those are ONLY ideas, most likely far from final product

File catalog(s)

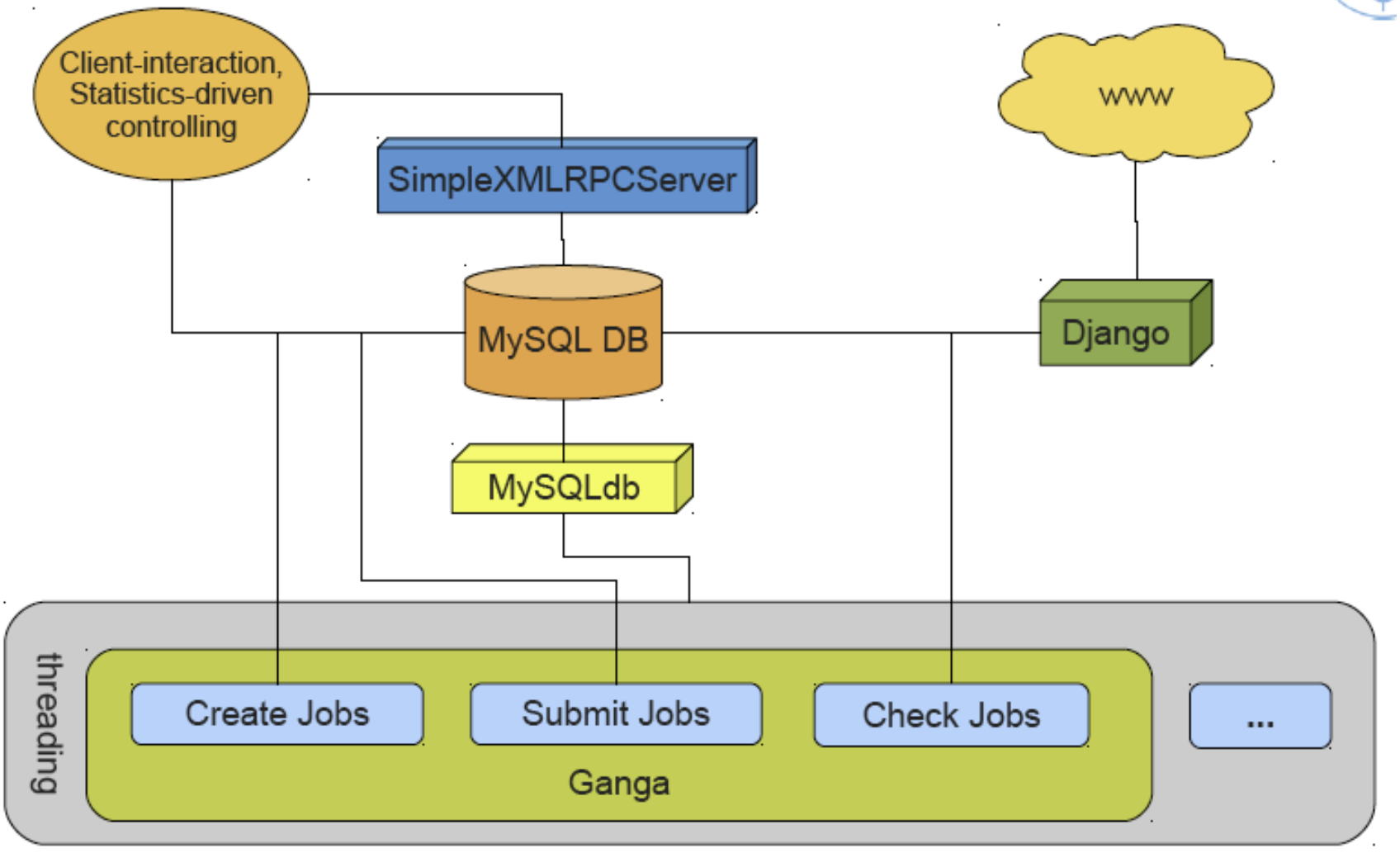
Currently for replica:

- Lcg File Catalog: used for ILD DBD studies, being more or less dropped by the LHC experiements (lead dev retired). Fairly slow as usually the server is shared between multiple VOs
- DIRAC File Catalog: Used for CLIC CDR and SID DBD, supported by the community. Quite fast as dedicated to only ILC (and CALICE)

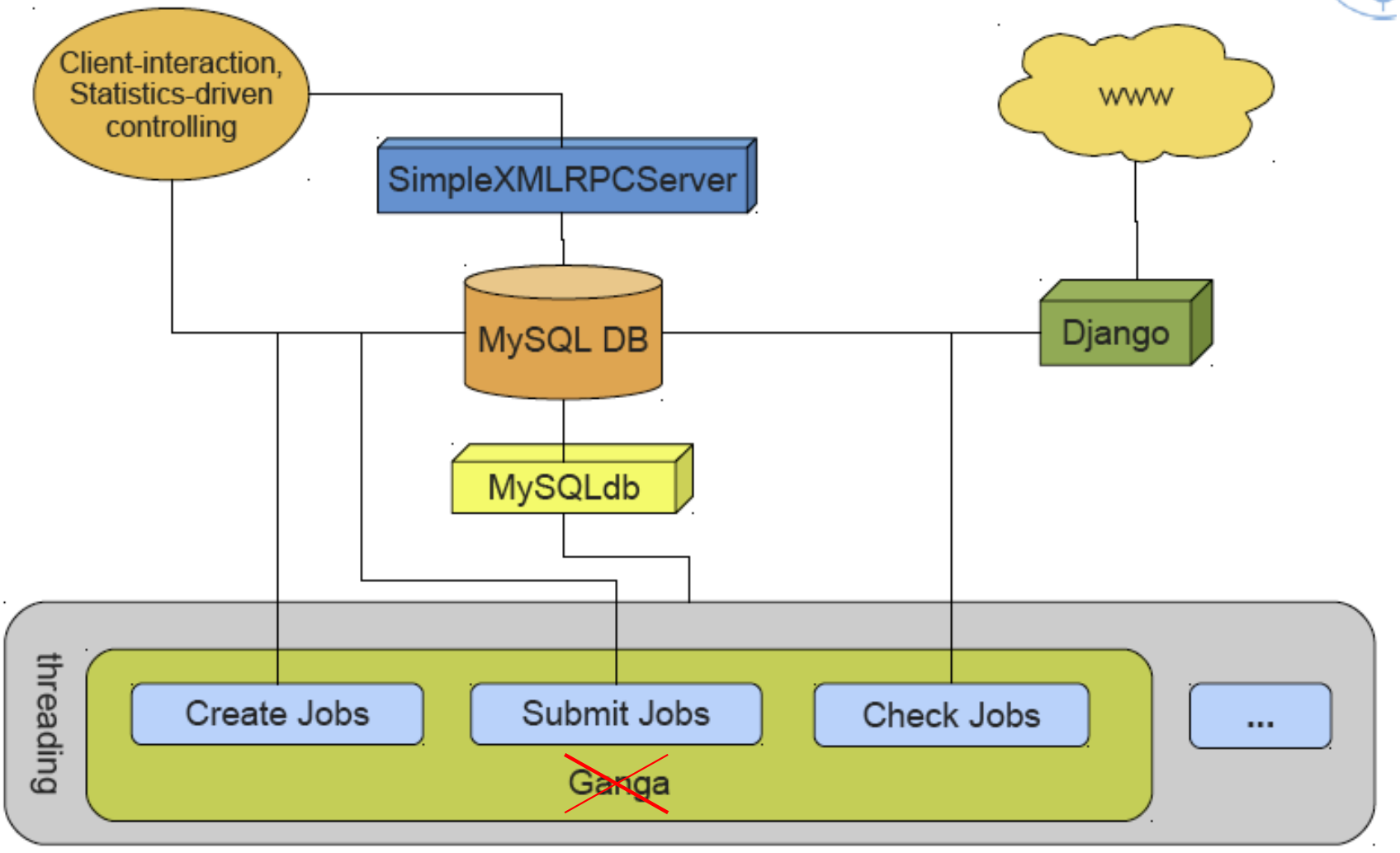
For meta data:

- FC from DESY: used for ILD DBD, access with DJANGO interface, file level meta data
- DFC again: used for CLIC CDR and SID DBD, file and directory meta data

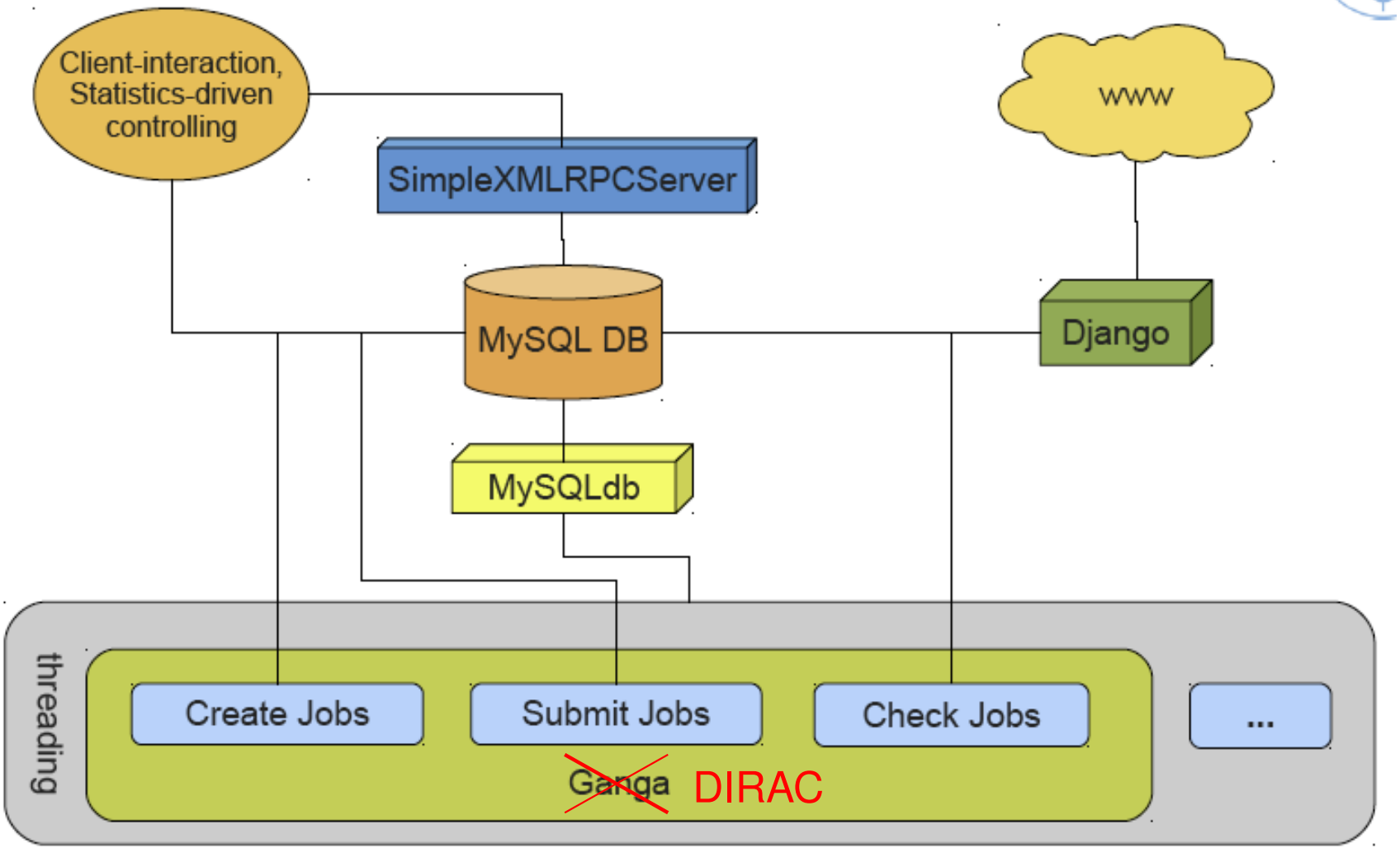
What can we do?



What can we do?



What can we do?



Then use DIRAC utilities to get the files, register the outputs, etc. No other changes needed.

Using OSG resources

- US part of the ILC VO relies on Open Science Grid
- Status information of sites not “properly” published in the Status service (BDII) according to gLite
- Faced many complains by sites because the info was not respected by gLite
- Solution: get rid of gLite, use OSG client. Implies coding pilot submission tool specific to OSG.
- Testing would be done by Belle2 people

Conclusion

Some long term ideas, some short term ideas.
Unifying the ILDGridProd with DIRAC can be done in a relatively short term.
Using properly OSG resources will be possible soon.

Backup Slides

Resource sharing

Groups:

- All ILC members do not share the same activities (ex. ILD, SID, TPC, etc.)
- Many groups in the VO to match those activities
- Users are not aware!

Roles:

- Several Roles for every group (Production, lcgadmin, etc.)
- Users do not know what they mean. . .

Resources:

- Should be used properly
- Should be shared between groups

DIRAC/ILCDIRAC can allow that!

- Create different JobTypes and assign them to different groups, then set site priorities per JobTypes.
- Define JobType automatically given a user proxy.

What about /ilc/Role=Production?