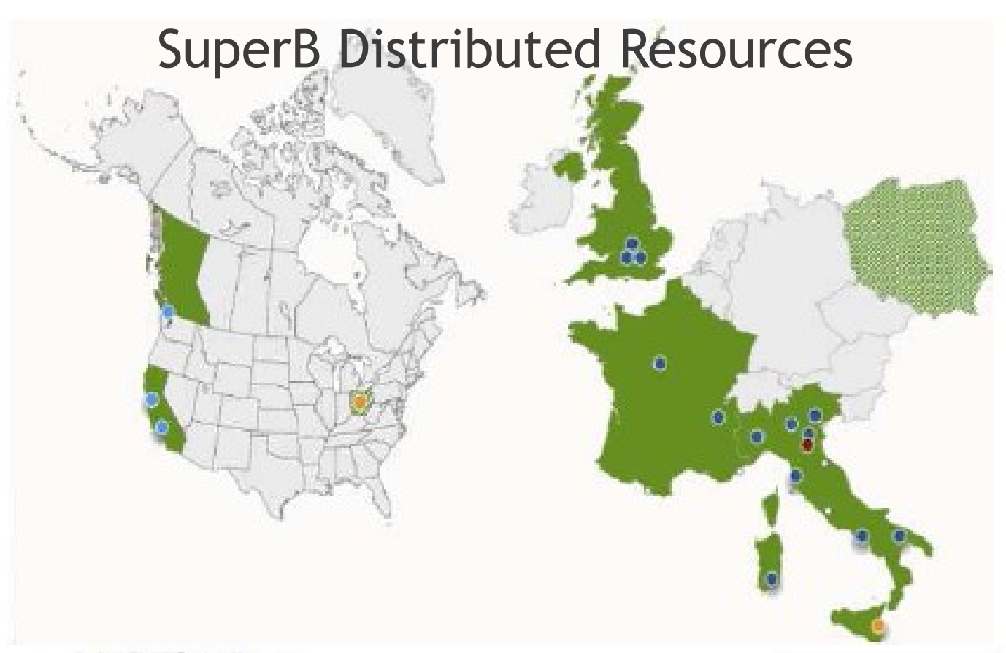
SuperB evaluation of DIRAC Distributed Infrastructure

Contributors

A. Fella[1], G. Donvito[2], B. Santeramo[2], A. Gianoli[3], E. Luppi[4], M. Manzali[4], L. Tomasetti[4], M. Rama[5], G. Russo[6], S. Pardi[6], D. Del Prete[6], R. Stroili[7], M. Corvo[8], S. Longo[9], A. Perez[10], A. Di Simone[11], F. Bianchi[12], S. Luitz[13], F. Giacomini[14], V. Ciaschini[14]

[1]CNRS, INFN Pisa and University of Ferrara, Pisa, Italy - [2]INFN Sezione di Bari, Bari, Italy - [3]INFN Sezione di Ferrara, Ferrara, Italy - [4]University of Ferrara and INFN, Ferrara, Italy - [5]INFN LNF, Frascati, Italy - [6]University of Napoli Federico II and INFN, Napoli, Italy - [7]University of Padova and INFN, Padova, Italy - [8]CNRS and INFN Padova, Padova, Italy - [9]INFN Sezione di Padova, Padova, Italy - [10]INFN Sezione di Pisa, Pisa, Italy - [11]University of Roma Tor Vergata and INFN, Roma, Italy - [12]University of Torino and INFN, Torino, Italy - [13]SLAC, CA, USA - [14]INFN CNAF, Bologna, Italy



- LHC Tier-1s (3)
  - infn-t1, in2p3-cc, ral-lcg2
- LHC Tier-2 (16)
  - uki-lt2-qmul, uki-southgrid-ralpp, uki-southgrid-ox-hep, grif, in2p3-lpsc, wt2(slac), cit-cms-t2b, victoria-lcg2, cyfronet-lcg2, infn-bari, infn-catania, infn-lnl-2, infn-milano, infn-napoli-atlas, infn-pisa, infn-torino
- Other (8)
  - infn-ferrara, infn-perugia, infn-cagliari, napoli-grisu, napoli-unina, in2p3-ires, cit-hep-ce, osc

Green: EGI sites
Red: OSG sites

DIRAC community

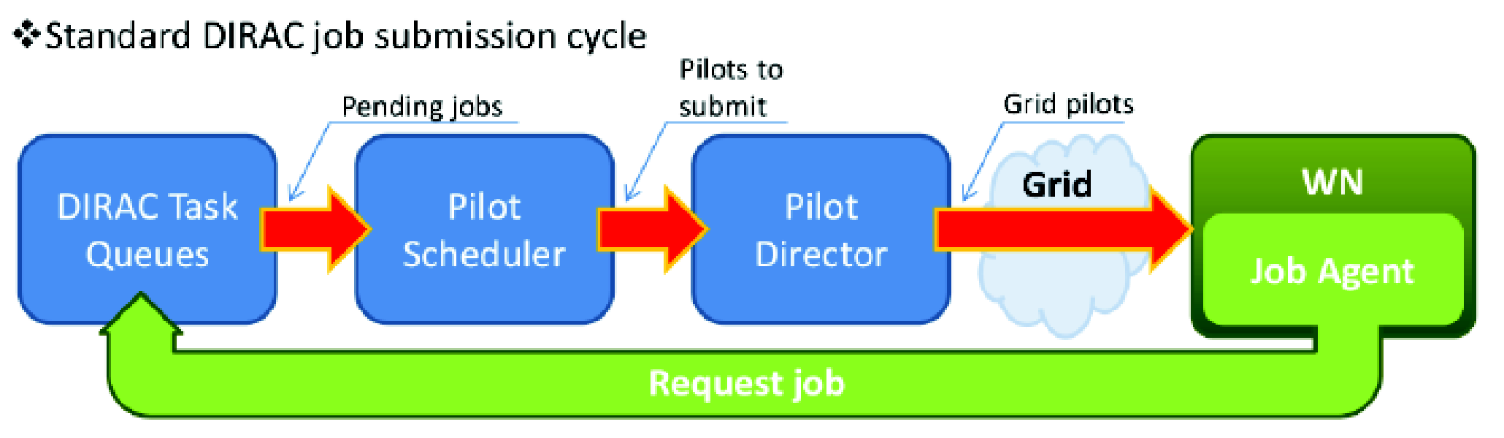
Belle II	experiment
BEPC	experiment
BES	experiment
BIOMED @ CREATIS, Lyon	community
CC/Lyon	community
CESGA	community
CTA	experiment
GISELA	community
GLAST	experiment
ILC	experiment
IOIT/Hanoi	community
LHCb	experiment
SuperB	experiment

The DIRAC (Distributed Infrastructure with Remote Agent Control) project is a complete Grid solution for a community of users needing access to distributed computing resources.

SuperB distributed computing group performed a detailed evaluation of DIRAC Distributed Infrastructure in terms of service capabilities, efficiency and reliability for two main use cases: end user analysis and Montecarlo simulation production.

DIRAC introduced the now widely used concept of **Pilot Agents**. This allows efficient Workload Management Systems (WMS) to be built. Pilot Jobs are nothing more than empty resource reservation containers that are sent to the available computing resources with the aim of executing the most appropriate pending payload in the central WMS queue.

❖Standard DIRAC job submission cycle



SuperB Resources estimation

- 5GByte/s of network bandwidth as experiment output
- 1000 PB of data collected in 5 years
- 50-60K CPU cores for first year data taking

Analysis test

real user application reading MC data performing anaylisis on $B^{+}B^{-}\rightarrow nn$ and $B_0\rightarrow k^{*}{}_{0}nn$ with $k^{*}{}_{0}\rightarrow\pi$ using standard ROOT libraries and SuperB libraries

exploited DIRAC parametric jobs, input and output saved on DIRAC File Catalog (DFC)

thanks to failover solution provided by DIRAC, no failures from user point of view

MonteCarlo production test

10 sites, 100 jobs per site, totally 1000 jobs submitted, 10.000 events simulated per jobs (about 2-3 hours per job)

Each job has 5 input files (about 3GB) saved in DFC and replicated on several Storage Elements.

Parameters used during test:

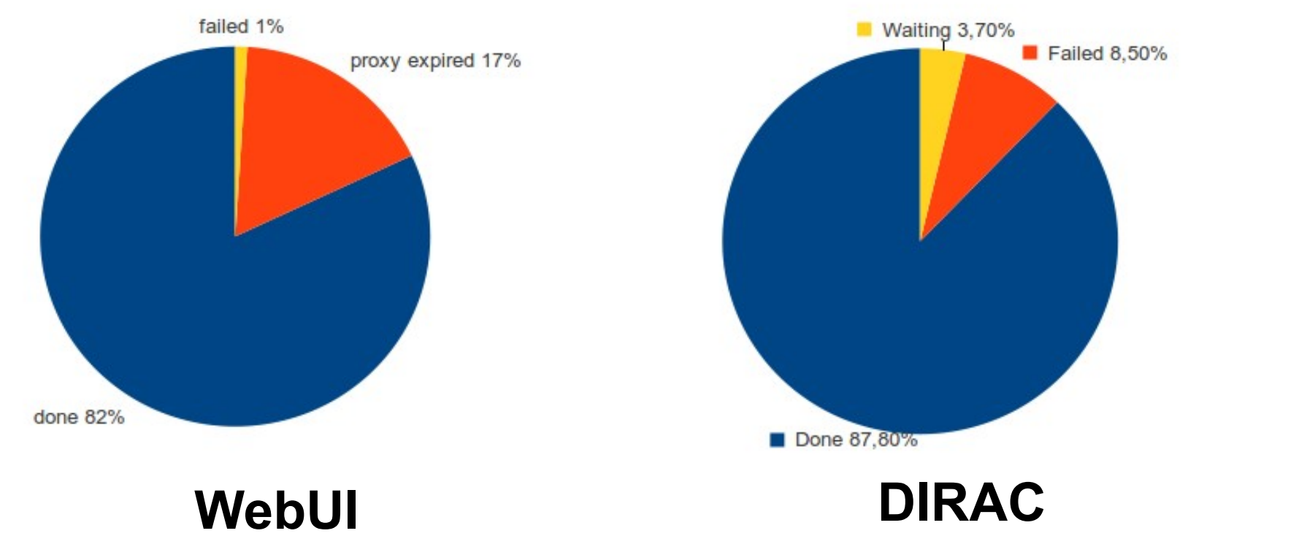
production:	2010_July_test
analysis:	BtoKNuNu
generator:	B+B-_K+nunu
geometry:	DG_4
background:	MixSuperbBkg
events:	10,000

Output data stored at INFN-T1 and registerd in DFC according to SuperB data model

LFN:/superbvo.org/test_dir/output_testbed/<job_output_dir>/<file_name>.root

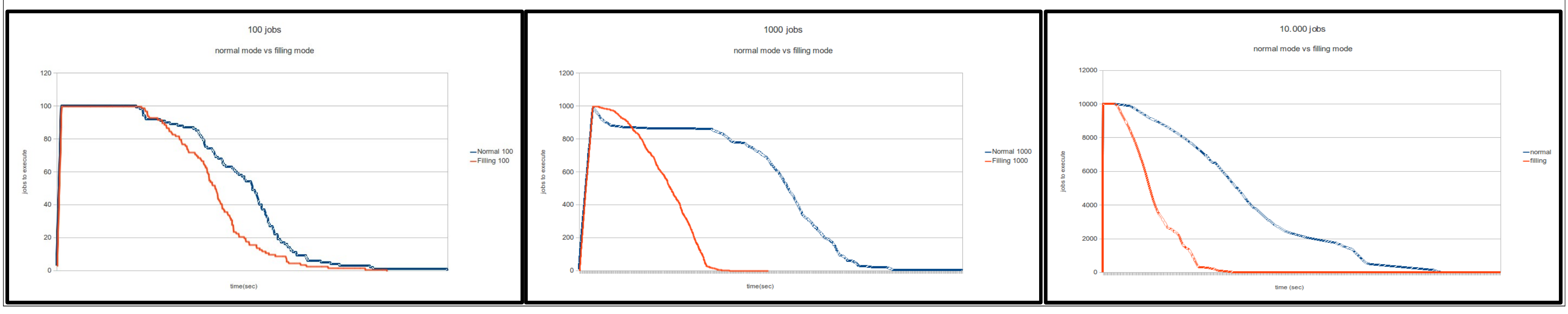
In case of failures at INFN-T1 storage element, otuput files automatically stored on a fallback storage element

Performed as comparative test between DIRAC and currently used production stack, after 48 hours results are better using DIRAC

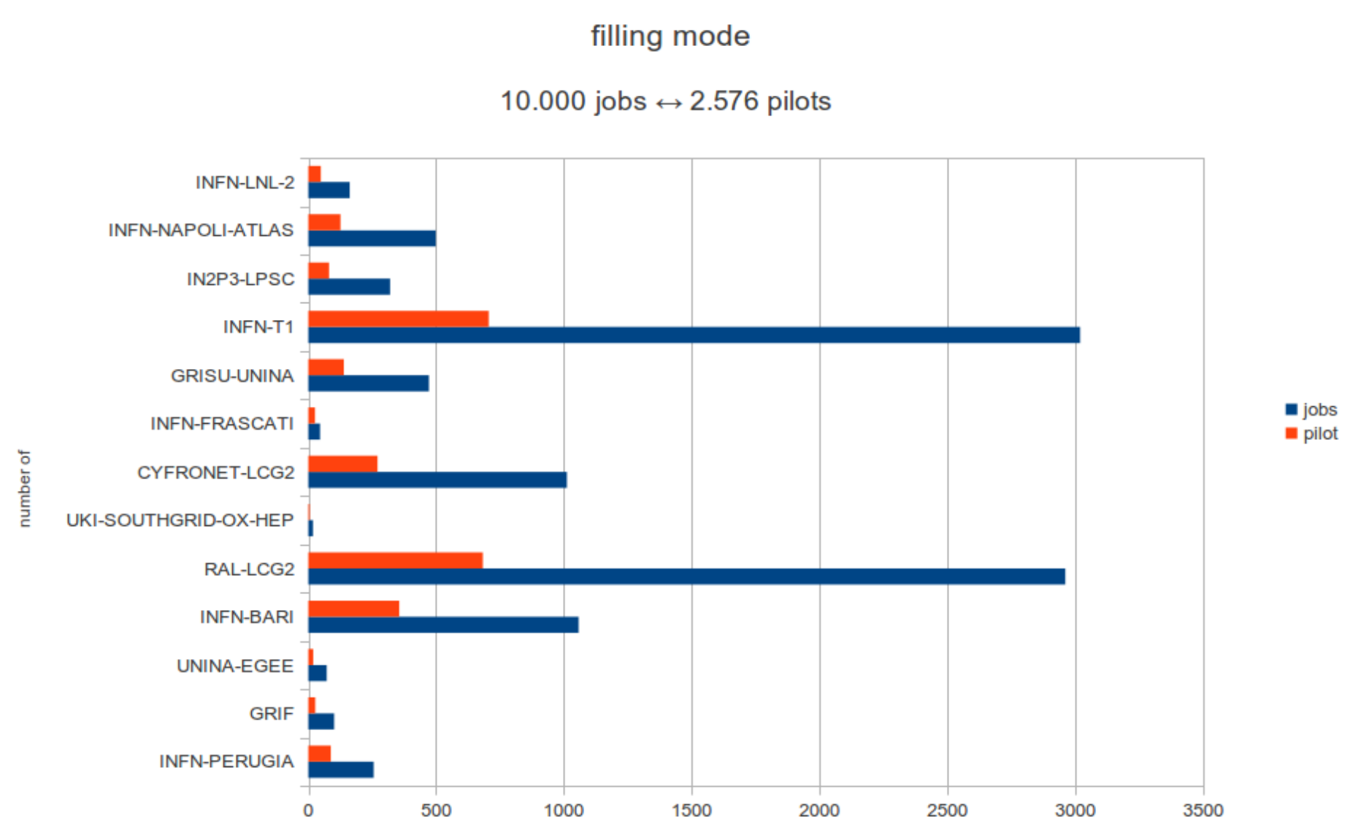


Filling mode allow a single pilot job to execute more than one job payload.

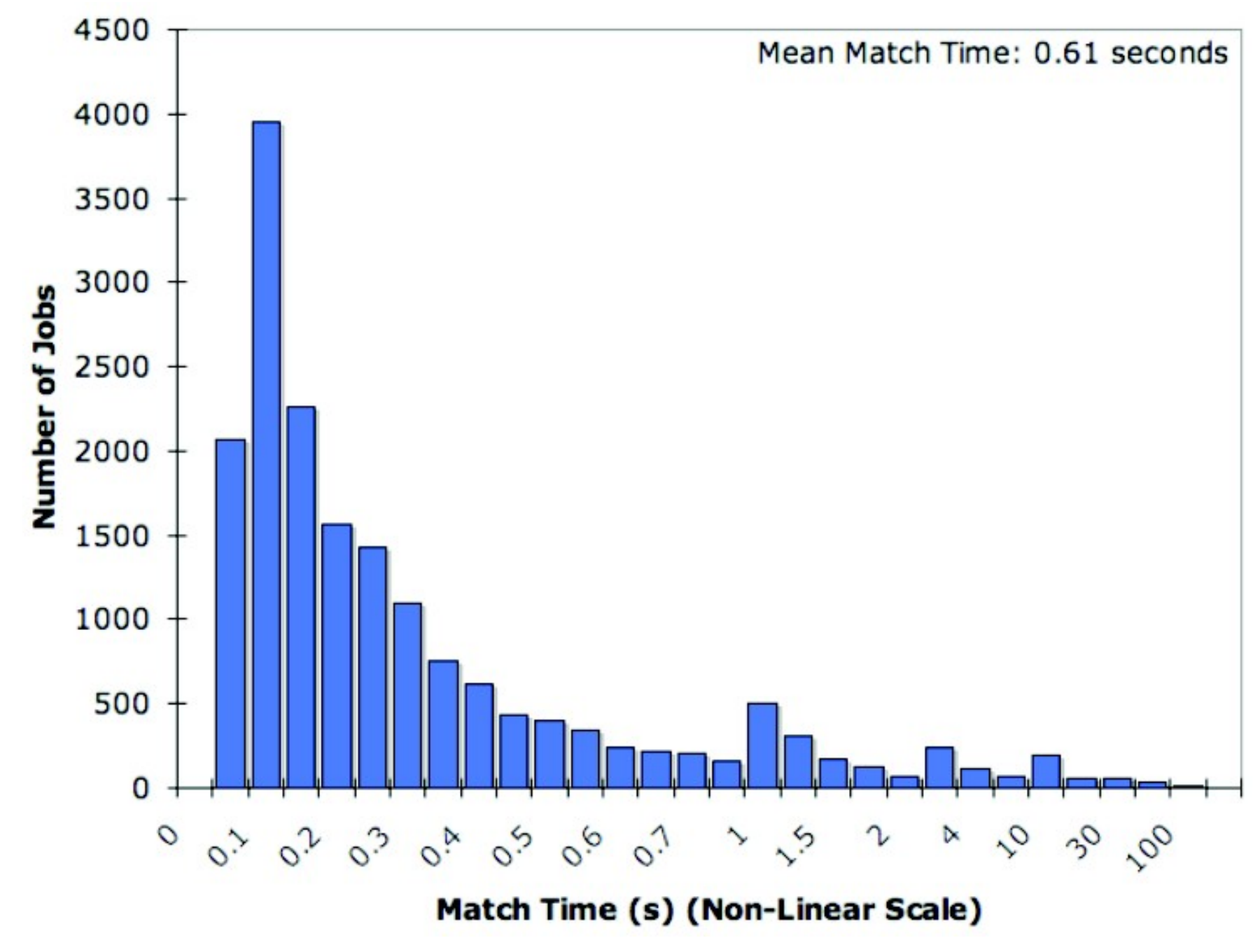
Filling mode permits a significant decrease in total execution time of jobs and this effects is more evident increasing the number of submitted jobs



Sites where filling mode works properly can can increase significantly jobs executed from VO point of view.



User jobs with many varied requirements present the biggest challenge for the PULL paradigm. DIRAC matcher times for 18K real user jobs submitted between January and August 2007. 92% of jobs are scheduled in under 1 second. (LHCb team)



Data Management

Storage Elements SRM interfaces configured.

tested majority of DIRAC data management capabilities: file upload, file replication, file deletion

tested DIRAC File Catalog (DFC) which offers interesting features like file provenance, metadata on directories and files. synchronization with LFC

Testbed Setup

In order to evaluate DIRAC capabilities and its matching with SuperB computing model needs, a test environment was configured at INFN-T1 and INFN-BARI.

Three servers were configured:

bbrbuild01.cr.cnaf.infn.it @ INFN-T1

64 bit VM, 2GB ram, Scientific Linux 5.4, gLite UI 3.2.0

sb-server-04.cr.cnaf.infn.it @ INFN-T1

64 bit VM, 2GB ram, Scientific Linux 5.4, gLite UI 3.2.0

gridtest-05.ba.infn.it @ INFN-BARI

64 bit VM, 2GB ram, Scientific Linux 5.7, gLite UI 3.2.0

SuperB resources configured in DIRAC testbed:

- 23 gLite sites
- 1 OSG site
- 50 Computing Elements
- 19 Storage Elements
- 1 DIRAC File Catalog (DFC)

Main goal of the testbed is the evaluation of job and data management DIRAC capabilities. The following DIRAC components have been configured:

- **Framework**: Core system of DIRAC
- **DataManagement**: Storage elements and data management, file catalog, FTS transfers
- **WorkloadManagement**: Jobs management
- **Accounting**: Monitoring functionalities
- **Configuration**: Computing Elements automatic discovery and configuration

Mass Data Transfer:

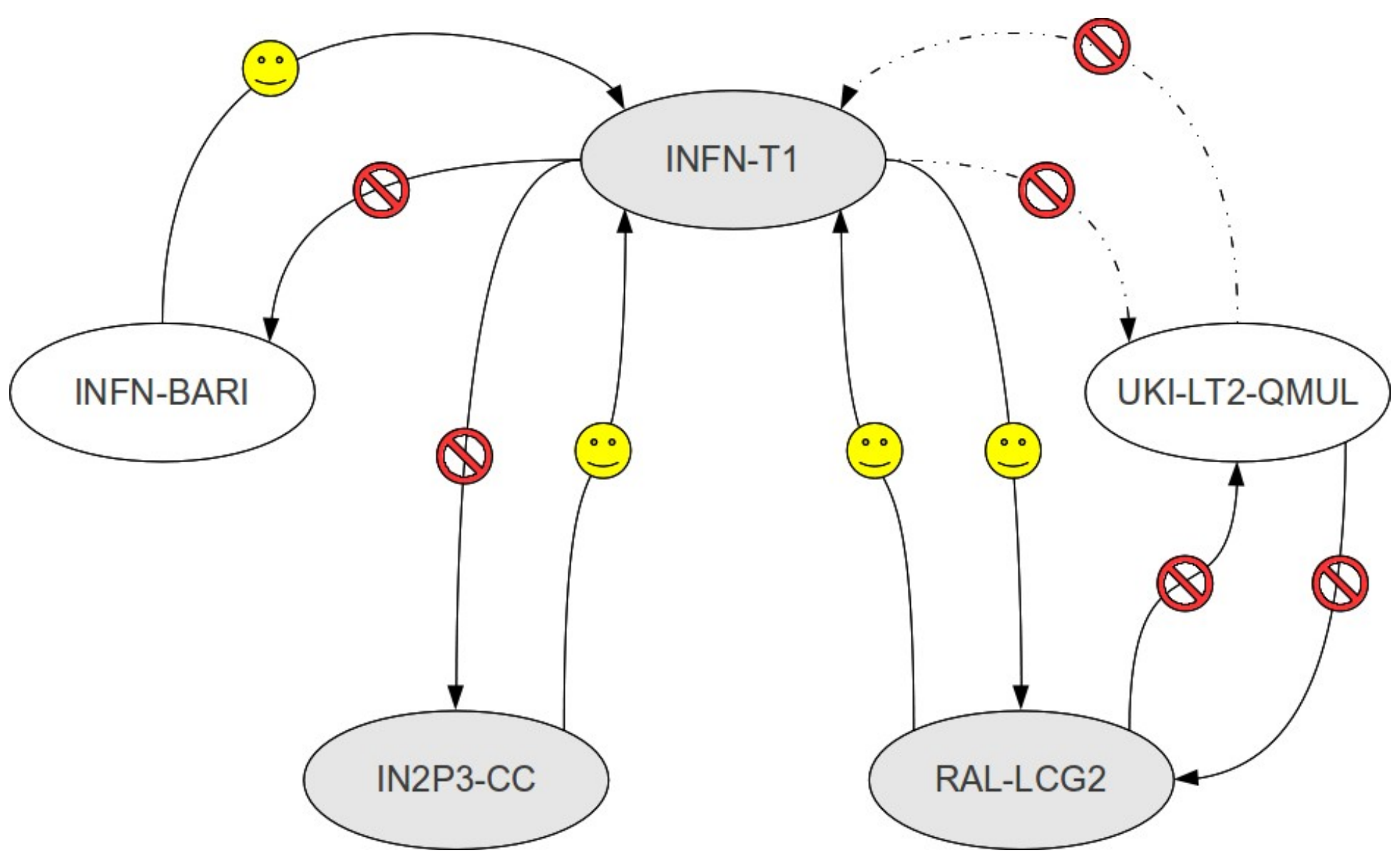
Tested FTS data transfers:

- catalog automatically updated
- no need to upload long-lived proxy on myproxy server
- FTS servers configured in DIRAC (user do not need to remember them)
- target and source SE by name, not by url
- source file by lfn, not by pfn

dirac fts transfer example:

```
dirac-dms-fts-submit /superbvo.org/data/datasetsXXX.root SourceSE TargetSE
```

All enabled FTS channel for superbvo.org tested



Final considerations:

The SuperB experiment at the moment is in an intese R&D phase, that will provide the needed experience and information to write a Computing TDR, that is expected to be relased around middle of 2013. The activities carried on within the SuperB experiment in this phase is directed both in the direction of writing missing piece of code and in evaluating already available solution that could be of help in order to solve SuperB use-cases. In this phase of the life of a new collaboration it is important to evaluate all the available solutions as there are at least the LHC collaboration that has similar use-cases and that are producing huge amount of good solution for many of the problems that SuperB is facing. The job submission and data management handling is surely two of the areas where already available software like DIRAC could be of help. Moreover the DIRAC framework is used at the moment by LHCb experiment and the roadmap of the tool is surely interesting for a Virtual Organization like SuperB that will use a geografically distributed grid infrastructure in order to solve the computational problems of an HEP Experiment that will produce a huge amount of data in the next years.

Giving those consideration, the work that we have carried on, already give us a good feeling on the capabilities of the DIRAC framework, and we will go on testing all the features that it make available, in order to have the opportunity to understand if this tool could solve the use cases of the DIRAC community.

One of the most important future test will be the DIRAC capability to handle massive data transfer among grid sites. In this case for example DIRAC will be compared to PHEDex that is the CMS tool for massive data transfer.