

# Software Distribution Framework for SuperB

## Distributed Computing

Dr. Silvio Pardi  
INFN-Naples



# People Involved



Armando Fella – INFN-Pisa

Silvio Pardi – INFN-Naples

Guido Russo – INFN-Naples

**Incoming people are welcome!**

Goal: Guaranteed that the SuperB code is correctly installed and configured in each site involved in job analysis.

- Software Installation
- Software Testing and tag Management in the IS
- Framework for job tracking and task management
- Monitoring

**Crucial task**



# Software distribution



## in g-Lite based GRID Environment

In each site, Worker Nodes share a storage area (default /opt/exp\_soft/vo\_name)

Each virtual organization defines a set of software-manager among the member who have the privilege to write in the exp\_soft area in all sites that support the VO

Software manager can write in the software area through Grid-Installation job or through a direct access to Grid-FTP / UberFTP.

# SOFTWARE INSTALLATION

## gLite/EMI Legacy solution

Installation Job



CE  
Computing  
Element

UberFTP

EXP\_SOFT\_AREA

WN

WN

WN

WN

Software Manager  
sends a GRID Job  
With the Installation  
script

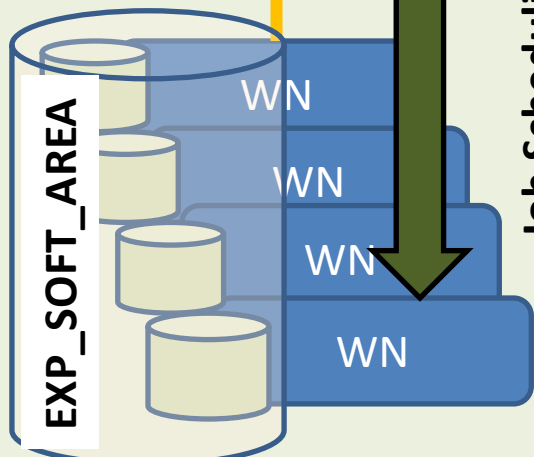
# SOFTWARE INSTALLATION

## gLite/EMI Legacy solution

Installation Job



Job Scheduling

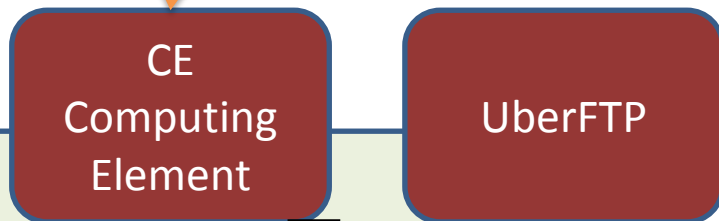


Job is scheduled on a Worker Node choice by the LRMS. This choice is out of the SWManager Control

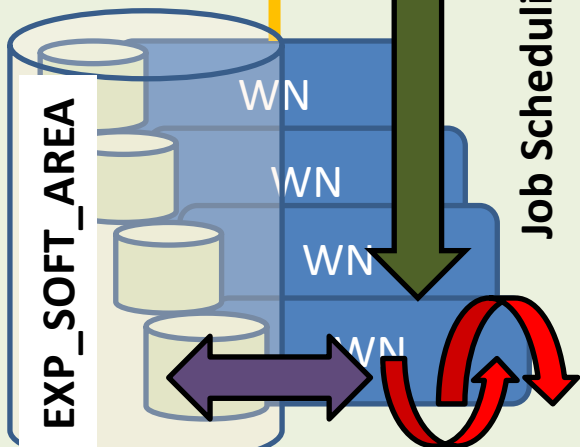
# SOFTWARE INSTALLATION

## gLite/EMI Legacy solution

Installation Job



Job Scheduling



Copy software in  
The Soft Area

JOB install Collaboration  
Software in the Shared  
Area between the nodes

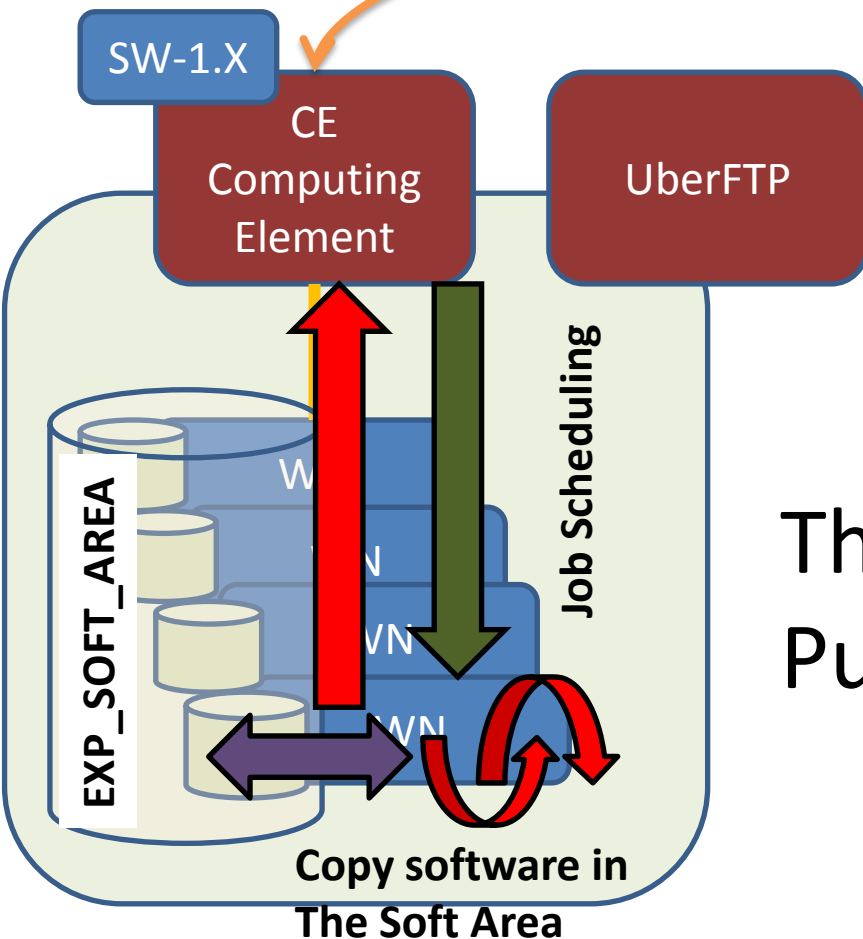
# SOFTWARE INSTALLATION

## gLite/EMI Legacy solution

Installation Job



The Computing Element  
Publish a tag in the IS







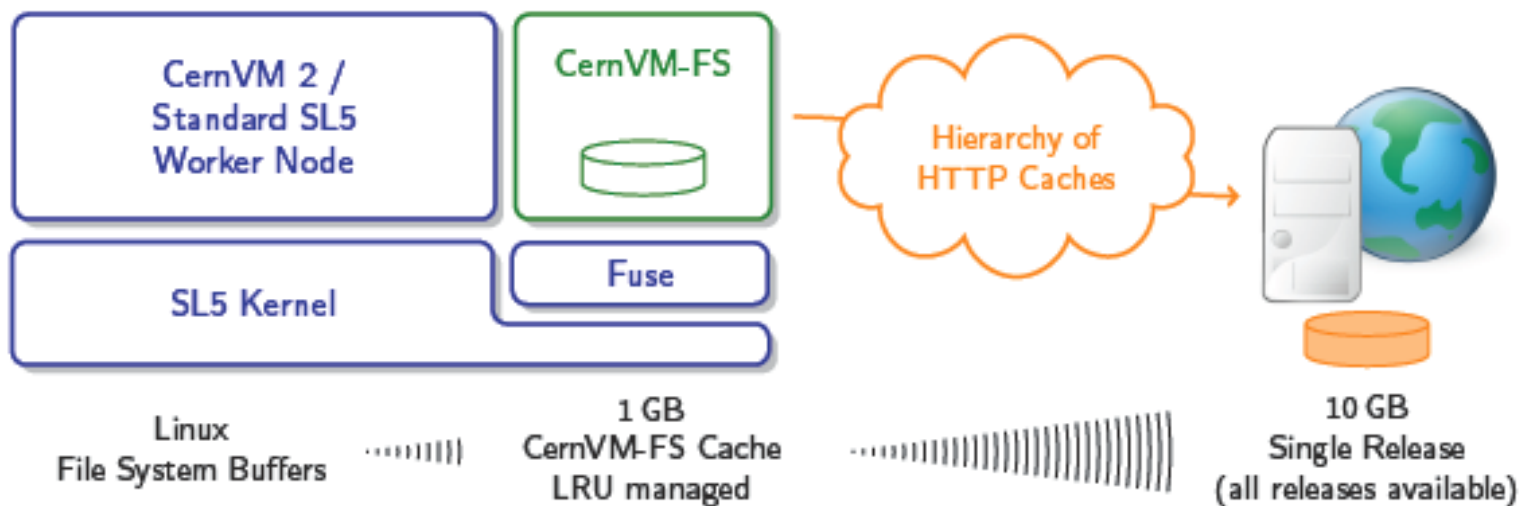
# Software in SuperB



gLite/EMI Legacy solution is the strategy currently adopted in SuperB to distributed the FastSim software needed for Simulation on the GRID infrastructure.

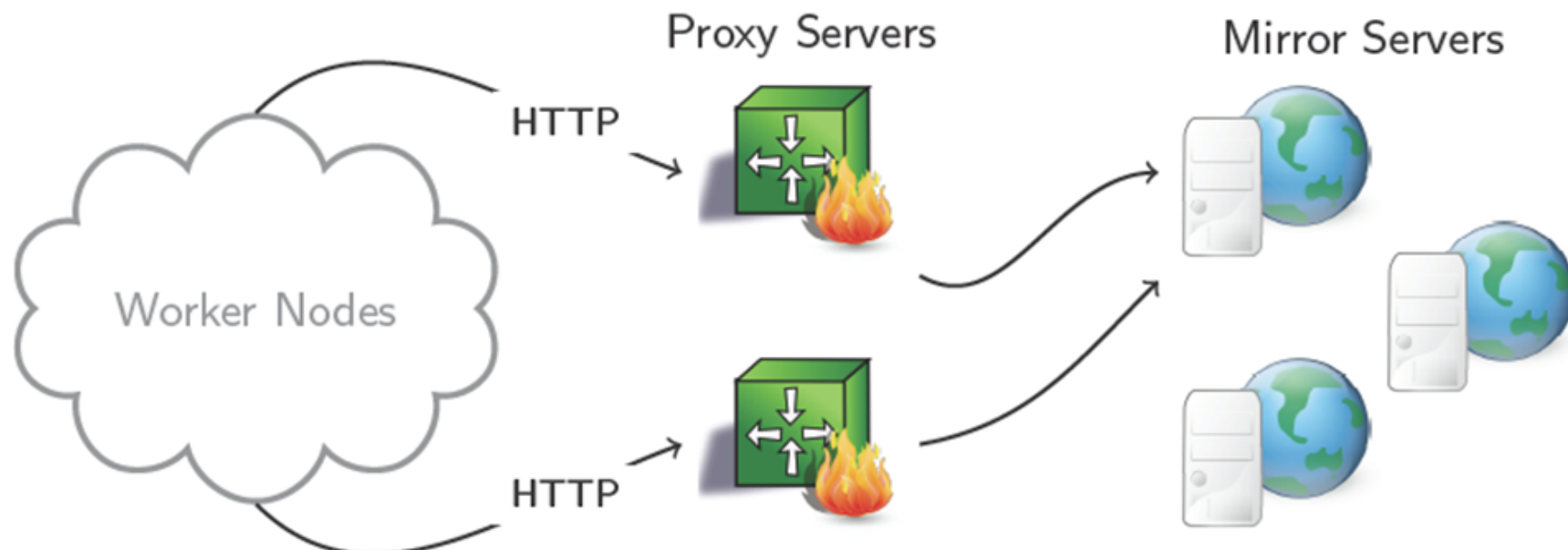
In the LHC world they are some R&D activities aims to simplify the software distribution through advanced file system able to manage automatic replica.

- ATLAS and LHCb moving to a dynamic software distribution model via CVMFS (CernVMFileSystem)
  - Virtual software installation by means of an HTTP File System



- Distribution of read-only binaries
- Files and file meta data are downloaded on demand and locally cached
- Self-contained (e. g. /cvmfs/atlas.cern.ch), does not interfere with the base system

# CVMFS Web Caches

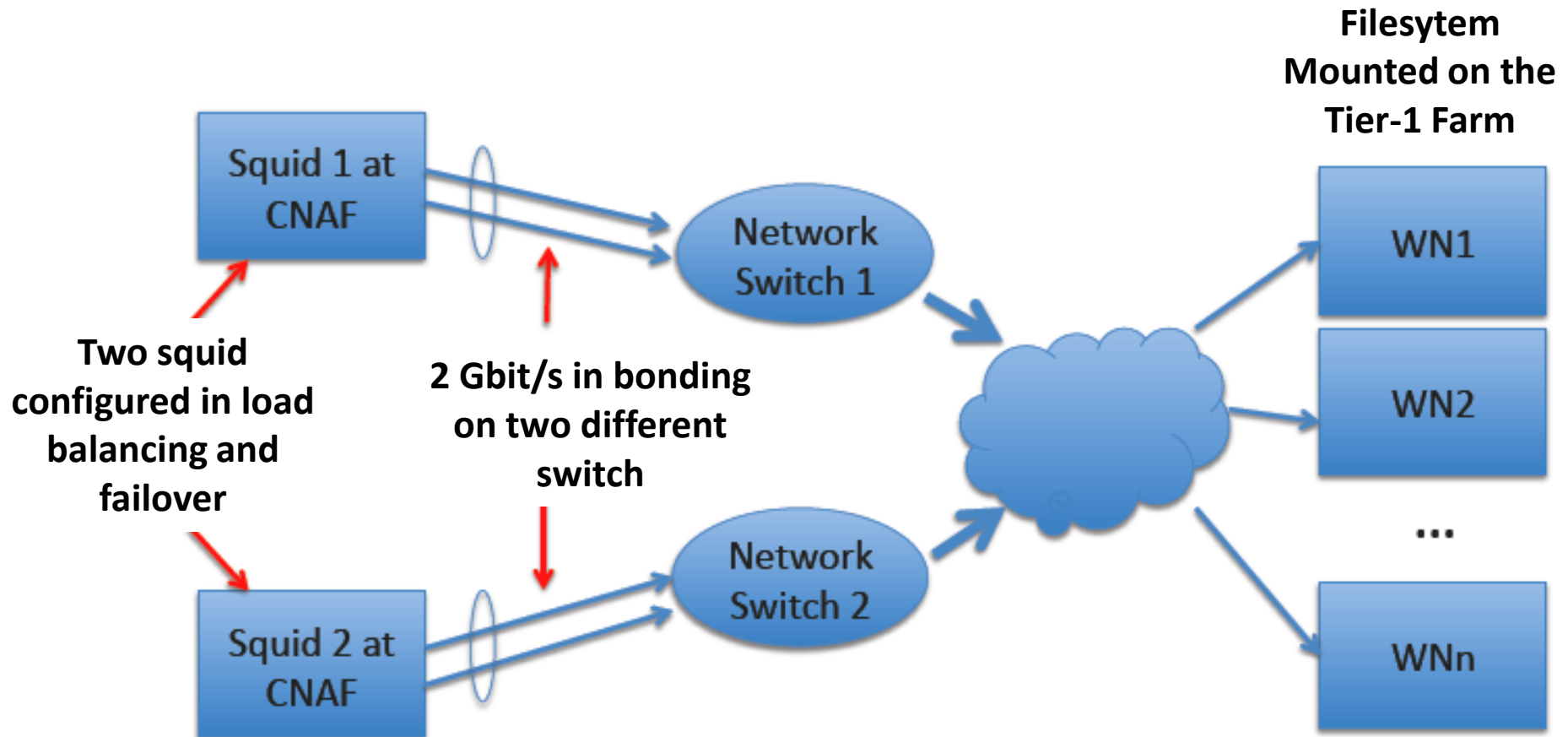


- **Proxies SL5 Squid, Load-Balancing + Fail-Over**
  - e. g. `CVMFS_HTTP_PROXY="A|B|C"`
  - as of next revision: semantics like Frontier/Squid
- **Mirrors Fail-Over, set order manually in**
  - `/etc/cvmfs/domain.d/cern.ch.local`
  - Mirrors at CERN, RAL, BNL operational by the end of the month

## CVMFS setup in grid sites

- **Sites migrating to CVMFS should provide**
  - A local squid cache for the CVMFS clients
  - A local disk cache for CVMFS on each node
    - Suggested minimum: 8 GB
  - Installation of the CVMFS clients on each node
  - No mount points already using /cvmfs/atlas.cern.ch !
  - One of the following setups for the local settings (to be discussed)
    - Manual settings (static model)
      - Site admins are in charge of setting the needed local settings in their environment and local disks
      - Frontier settings, dcache/DPM fixes, DQ2\_LOCAL\_SITE\_ID set by the system admins
    - Centrally managed local setup area (semi-dynamic model)
      - Site admins define a variable ATLAS\_LOCAL\_AREA in the WNs, pointing to a directory writable by the SGM users where to store the overrides to the global local configuration in CVMFS
      - The Installation System will take care of updating the local settings each time a new PFC must be rebuilt
      - All jobs will still have to source the <sw-area-path>/local/setup.sh file, with overrides from the local area
    - Full dynamic model
      - Site admins do not care of anything and we obtain all the settings from AGIS
      - Probably an heavy load on the central system!
- **The preferred way would be to start with a semi-dynamic model, to possibly migrate to the fully dynamic model in the future**

# CVMFS test at CNAF (LHCb)





# Software Installation Framework



Framework for software installation management and monitoring is an other crucial aspect of Software Distribution Problem.

Generally LHC experiment use a self implementation of software manager framework.

An interesting tool is represented by **LJSFi**



# LJSFi - The Light Job Submission Framework for Installation



LJSFi is a fully automatic software installation system, originally developed for LCG/EGEE and then **extended to other grid flavors**. The system is based on the Light Job Submission Framework, an independent job submission framework for generic submission and job tracking in EGEE.

**The ATLAS experiment currently uses LJSFi in production for both EGEE and OSG sites**





# LJSFi - The Light Job Submission Framework for Installation



LJSFi is composed by

- > A central DB that contain information about, Software Release, Grid-Site, VO-User and the status of the Software Installation Procedure on each Site.
- > A Backend component that perform all the main action (SW Installation, Testing, Canceling)
- > A Web Interface as GUI for all the Query and Actions procedure

**LJSFi can interact with CVMFS system**



# ATLAS Installation Pages

## Actions



- Home page
- User registration
- Request an installation
- Pin a release
- Mail subscriptions
- Show requests
- Tags matrix
- Site map
- Firefox plugin (experimental)
- Architectures
- InfoSys
- Releases
- Sites
- Targets
- Tasks

Powered by



FORM TO QUERY THE  
DATABASE

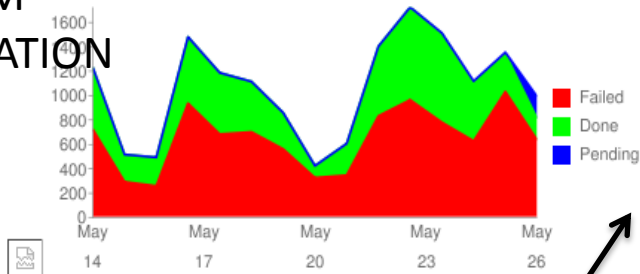


ACTIONS  
ALLOWED BY  
THE SYSTEM

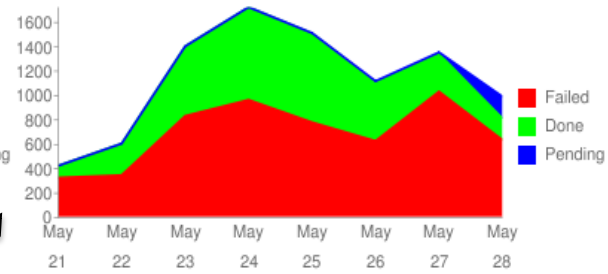
Release	- any -
Grid name	- any -
Site name	- any -
Site arch	- any -
Computing Element	- any -
User	- any -
<input type="button" value="Search"/> <input type="button" value="Reset"/>	

SYSTEM  
CONFIGURATION

ATLAS installation jobs in the last 14 days



ATLAS installation jobs in the last 7 days



INSTALLATION MONITORING



# LJSFi - The Light Job Submission Framework for Installation



LJSFi seem well documented and they are instruction to install.

LJSFi is a general purpose tools but now is strongly customized on the ATLAS Framework.

<https://atlas-install.roma1.infn.it/twiki/bin/view/Main/LJSFiDocumentation>



# Software in SuperB



The SuperB software is structured in a set of rpm packages accessible via *yum* suite.

The procedure can be performed by simple user local execution or via Grid job submission.

The installation is managed directly by the software-manager, right now we don't use any framework ad hoc.



# Issue in SuperB



Individuate a Framework to simplify and automate as much as possible the software distribution procedures.

Individuate a medium/long term solution to support the simulation production and the future requirements of SuperB (EMI Legacy? CVMFS or Similar approach?)

# ACTION

Investigate the possibility to configure LJSFi as software distribution tools for SuperB code.

(Alessandro de Salvo can give a basic support for this test)

Evaluating CVMFS for GRID Software Installation  
(The Atlas Naples-Tier2 will migrate soon)

Investigate other possible solution to simplify the software distribution among the sites.

# Some References

## LJSFi

<https://atlas-install.roma1.infn.it/twiki/bin/view/Main/LJSFiDocumentation>