Using CernVM-FS to deploy Euclid processing software on Computing Centres

Mark Holliman - ROE

Quentin Le Boulc'h - CC-IN2P3

Maurice Poncet - CNES

CernVM Users Workshop

Rutherford Appleton Laboratory

06.06.2016

Outline

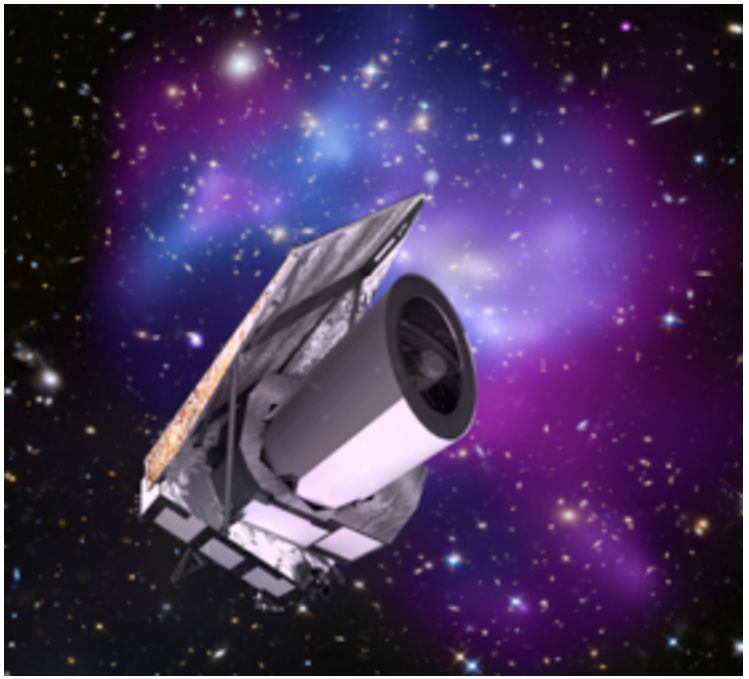
- Euclid data processing
- CernVM-FS architecture
- Successes and issues
- Conclusions

Euclid data processing

The Euclid mission

- Euclid is an ESA space mission that will help understanding the origin of Universe accelerating expansion

- Satellite with two instruments:
  - VIS : Visible Imager (600 MPix)
  - NISP : Near Infrared Spectrometer and Photometer (64 Mpix)

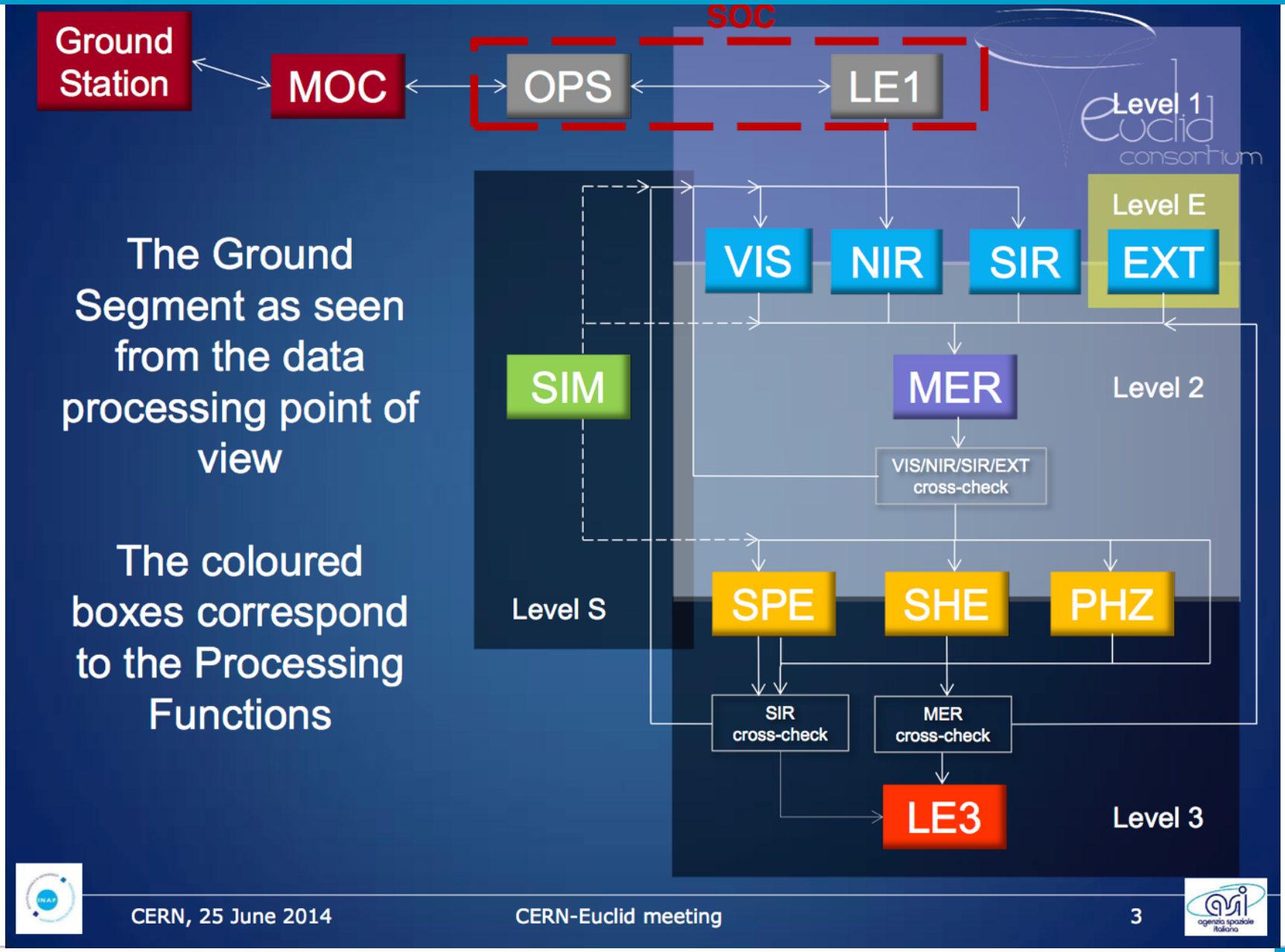


- Cosmological probes:
  - Weak lensing
  - Galaxy clustering (BAO,…)
  - Structure formation
  - …

- Launch in 2020 (6 years mission)

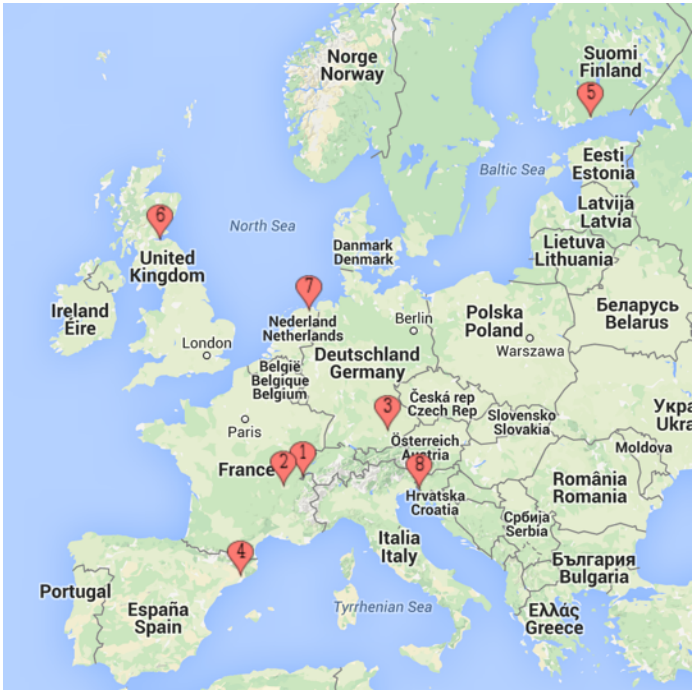
- Euclid is a CERN recognized experiment
- We have a cooperation agreement that includes the use of CernVM-FS

The Euclid processing pipeline



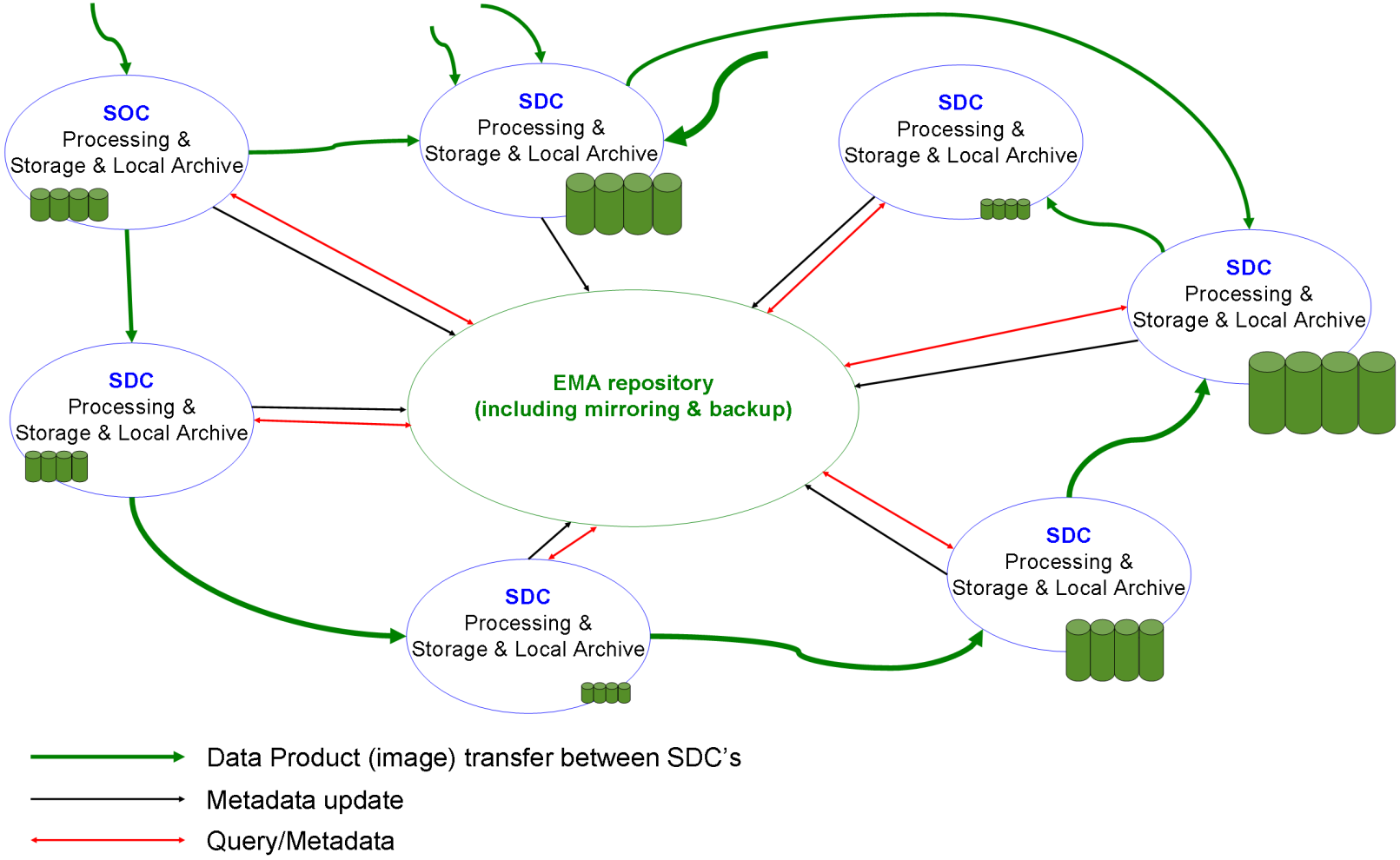
Euclid data processing

- 9 Science Data Centers
  - Finland : University of Helsinki
  - France : CC-IN2P3, Lyon
  - Germany : Max Planck Institut, Garching (MPE)
  - Italy : Astronomical Observatory of Trieste (INAF)
  - Netherlands : University of Groningen (RUG)
  - Spain : PIC, Barcelona
  - Switzerland : University of Geneva (ISDC)
  - UK : Royal Observatory, Edinburgh (ROE)
  - US : IPAC, Caltech



- Some numbers:
  - RAW data : ~ 300 TB (6 years) = 150 x Planck
  - Total (including intermediate and external data) : ~ 150 PB (10^10 objects)
  - Processing : ~ 20 000 CPU cores at maximum

Euclid data processing

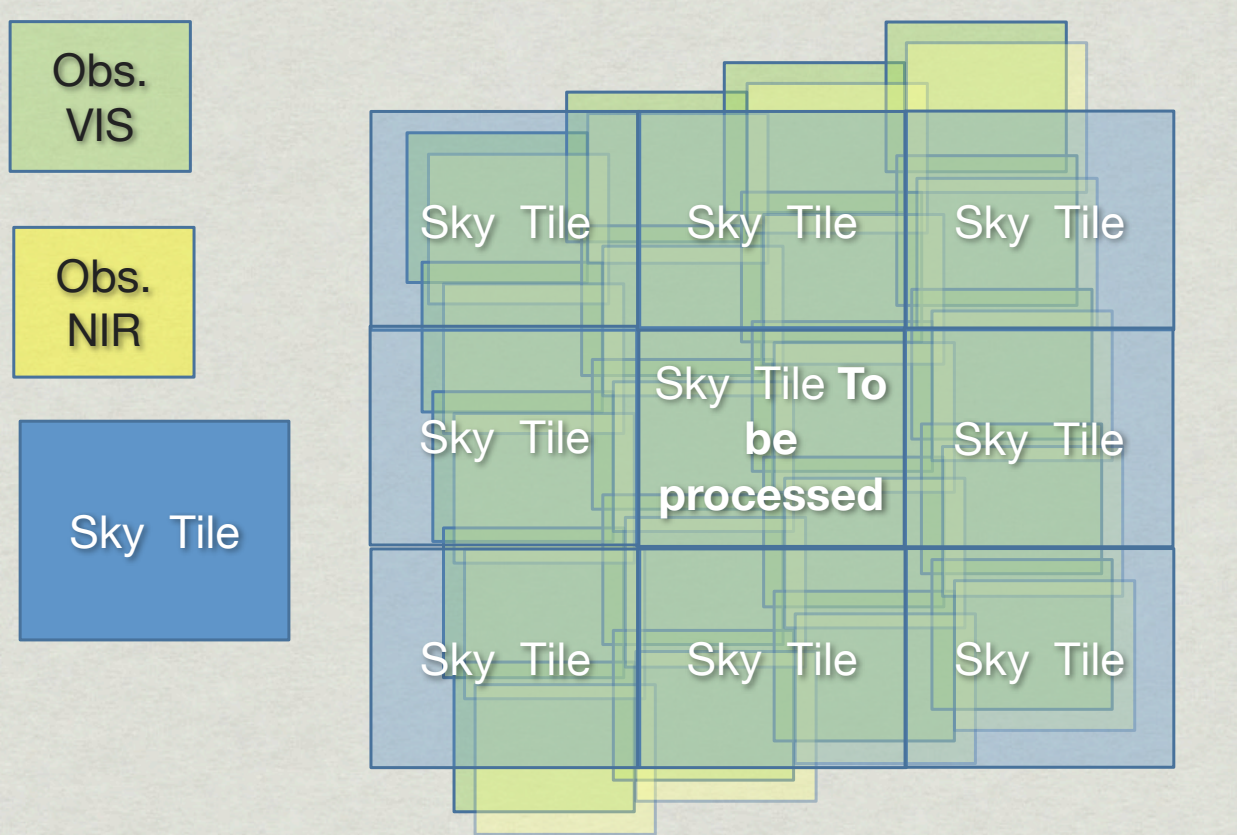


- Data distributed on the 9 computing centers (redundancy)
- Centralized metadata database

Euclid data processing

« Move the code, not the data »

Data storage and processing distributed in computing centers following a sky division:

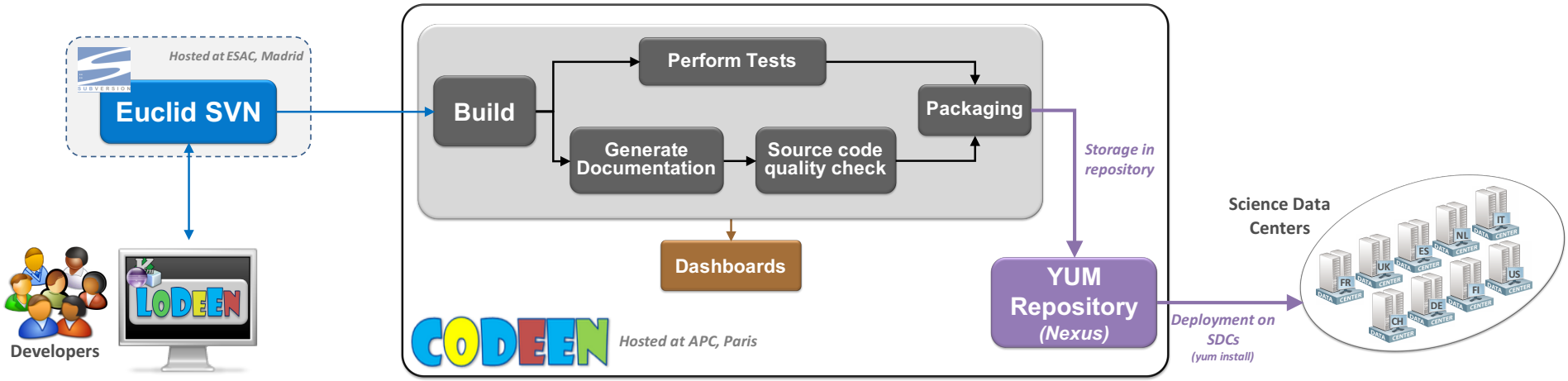


All kind of processing (Processing Functions) must run on all computing centers

CernVM-FS architecture

Software deployment: CODEEN

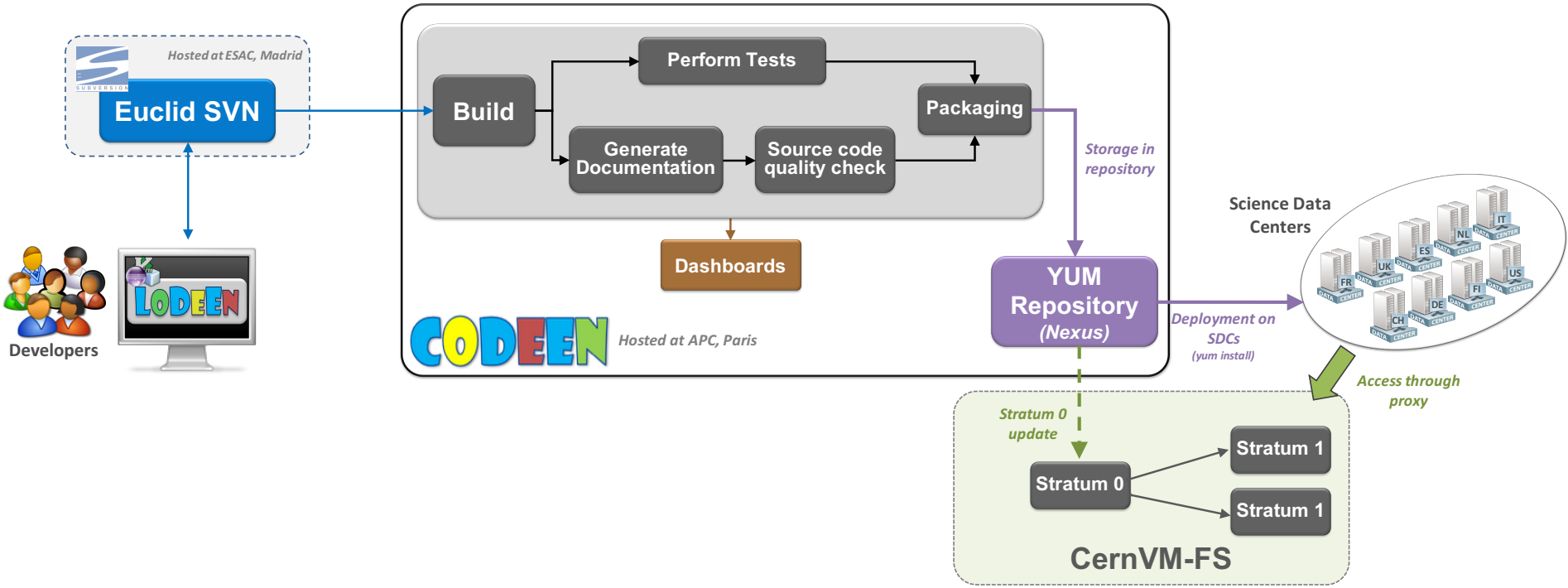
- Euclid software is integrated in a development platform called CODEEN (« COllaborative DEvelopment ENvironement »):



- DevOps approach: continuous integration, delivery and deployment
- Continuous integration: Jenkins
- Continuous delivery: RPMs, yum & Nexus
- Continuous deployment: no obvious solution
  - Only way to deploy software on the computing sites was to ask admins to run a « yum install » command
  - As we had a very positive experience with CernVM-FS at CC-IN2P3 we suggested to start testing it

Software deployment: introducing CernVM-FS

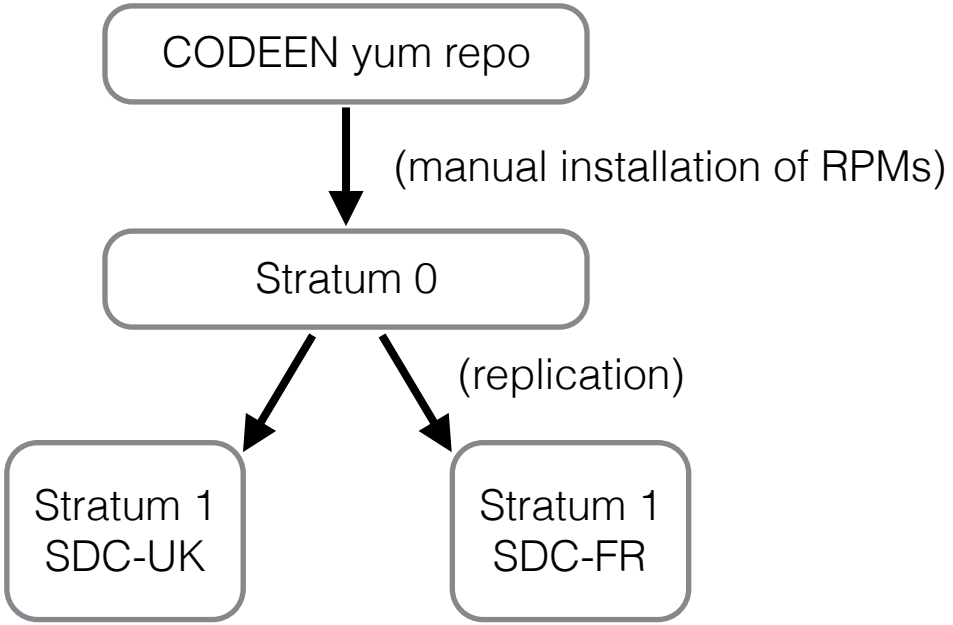
- The idea was to keep the RPM repository and introduce CernVM-FS as an alternative way to access the software:

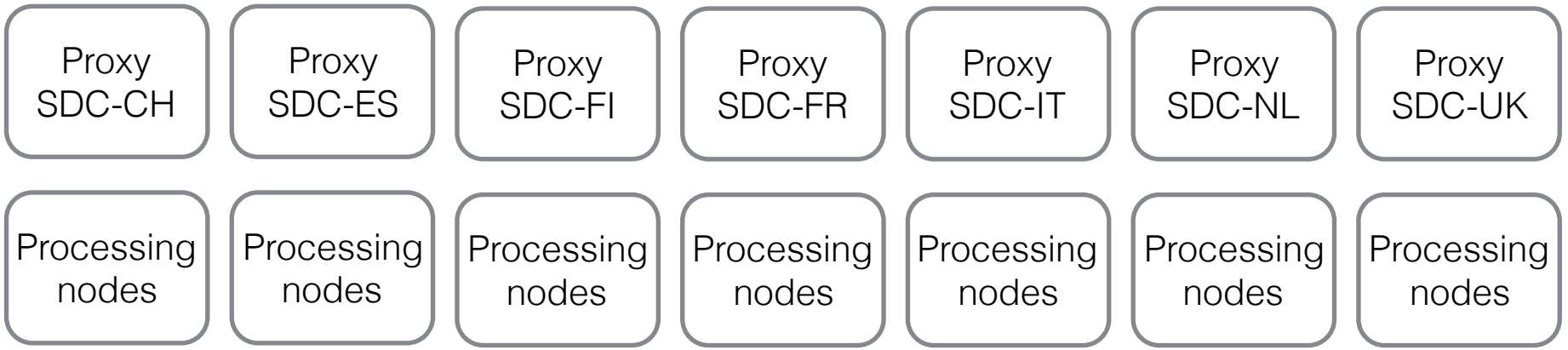


- We started with a simple testbed: a Stratum 0 in the CODEEN infrastructure, and a Stratum 1 + proxy + clients at CC-IN2P3
- We learnt how to deploy some codes in the repository and running them on processing nodes with a CernVM-FS client

CernVM-FS architecture

- We then setup a second Stratum 1 in SDC-UK, proxies and clients in others SDCs (currently 7 out of 9):





Software deployment: environments

- EDEN (Euclid Development ENvironment) is an environment with defined versions of O/S, libraries, and tools that is used to develop and run Euclid software.
- Our software in the CernVM-FS repository is structured in the following way:

```
/cvmfs/euclid.in2p3.fr/
                        SL6/
                        CentOS7/
                                 EDEN-1.0/
                                 EDEN-1.1/
                                 EDEN-1.2/
                                          etc/
                                          usr/
                                          opt/
                                          var/
                                 EDEN-1.0_env.sh
                                 EDEN-1.1_env.sh
                                 EDEN-1.2_env.sh
```

- Environment variables need to be customized: a script is sourced before the job

Successes and issues

Successes

- Every year we run one “Infrastructure Challenge” to update and test the distributed architecture design across all SDCs
- For the « Infrastructure Challenge #6 » this year, CernVM-FS was the recommended solution to distribute the software
- Most of the SDCs used CernVM-FS and the Challenge was successful
- CernVM-FS has been selected as the software deployment solution for the Euclid architecture!

Issues

- CentOS 7 not yet supported for Stratum 0: deploying CentOS 7 packages using a SL6 server
  - Feasible but not ideal
  - Start testing CentOS 7 Stratum 0 with updated kernel

- Deploying RPMs from a yum repository into a CernVM-FS repository can be tricky:
  - Using « installroot » option to relocate packages into /cvmfs/euclid.in2p3.fr/CentOS7/EDEN-1.1/…
  - Some packages are not relocatable, manual deployment for these specific cases
  - installroot create a lot of hardlinks, but repository configured to accept (transform) them
  - installroot create lot of files with root-only permissions : we just delete them
  - installroot allows us to install all dependancies to be independent from the local environment, but system libraires can conflict, e.g. in the case of different glibc versions: we need to remove the glibc package.

Issues

- Overall deployment procedure is complex. We developed a script to deploy a RPM in the CernVM-FS repository using one command line

- Need for continuous deployment:
  - Developers want to be able to test their code on computing sites right after they committed changes on SVN
  - Not only tagged versions, the trunk should be deployed
  - Under work: code compiled in CODEEN will be automatically deployed into a second, dedicated CernVM-FS repository (bypassing RPMs)

- Latency (~ 30 minutes) sometimes frustrating when doing frequent tests
  - Especially important for a continuous approach
  - Need to tune Stratums and proxies configuration
  - Future cvmfs releases will be interesting

Issues

- Repository can be accessed only from sites having a proxy
  - Need to setup a public proxy
  - Interesting for developers working on their local lab infrastructure
- Changes in clients configuration not very practical and prone to errors: will create a cvmfs-config package
- Long-term improvement:
  - Will need to work on performance aspect at some point: Stratums and proxies tuning, nested catalog, etc.
  - Setup automated test of the code before publication
  - Define a clear deployment strategy including environment initialization

Conclusions

Conclusions

- CernVM-FS is a very efficient tool adapted to our needs
- Works already well with a basic architecture and configuration
- We will start to look deeper into functionalities and configuration
- We hope it can be useful in a continuous deployment approach (some work to do)
- It is also foreseen to use it to deploy other kind of data like static calibration and configuration files
- Deployment part still complex: is our approach relevant?

Conclusions

This work has been done within the Euclid SGS System Team

Thanks to all the collaborators, in particular from the Common Tools and Architecture groups

Thank you for your attention!