euclid

CERN – Euclid meeting

Euclid Computing – Architecture and Challenges

M. Holliman, M. Poncet, M. Marseille, Q. Le Boulc'h

Euclid Summary

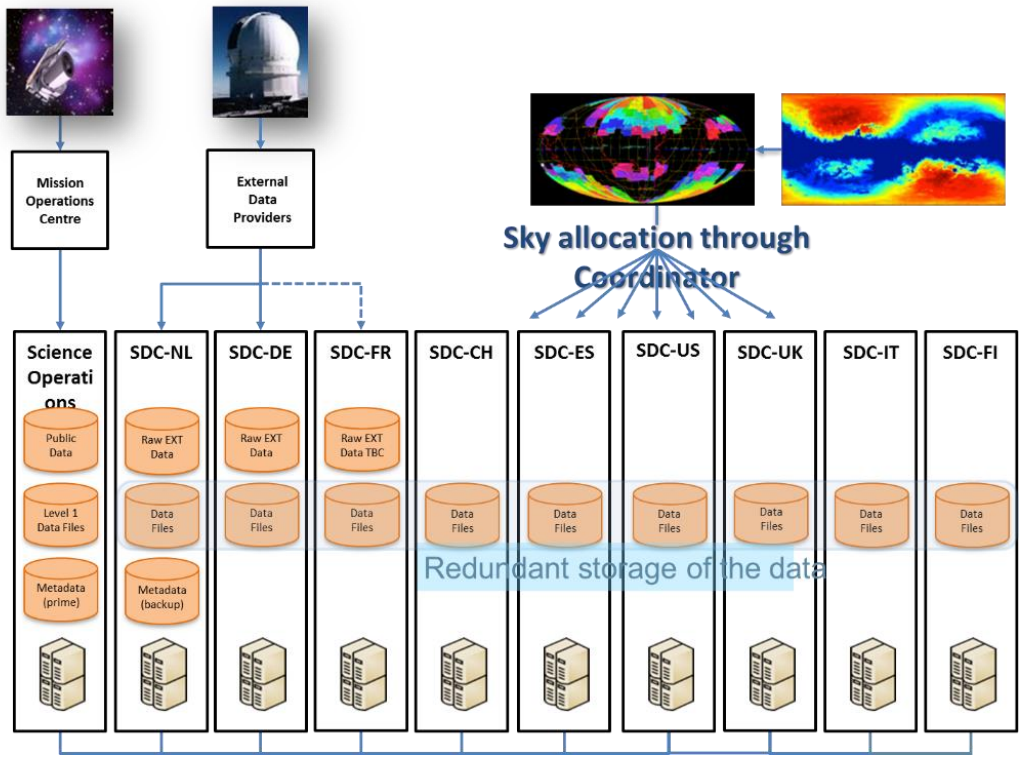
- ESA Medium-Class Mission
  - In the Cosmic Visions Programme
  - M2 slot (M1 Solar Orbiter, M3 PLATO)
  - Due for launch 2021
- Largest astronomical consortium in history: 15 countries, ~2000 scientists, ~200 institutes. Mission data processing and hosting will be spread across 9 Science Data Centres (SDCs) in Europe and US (each with different levels of commitment).
- Scientific Objectives
  - To understand the origins of the Universe's accelerated expansion
  - Using at least 2 independent complementary probes (5 probes total)
  - ***Geometry of the universe***:
    - Weak Lensing (WL) Galaxy Clustering (GC)
  - ***Cosmic history of structure formation***:
    - WL, Redshift Space Distortion (RSD), Clusters of Galaxies (CL)

Controlling systematic residuals to an unprecedented level of accuracy, impossible from the ground

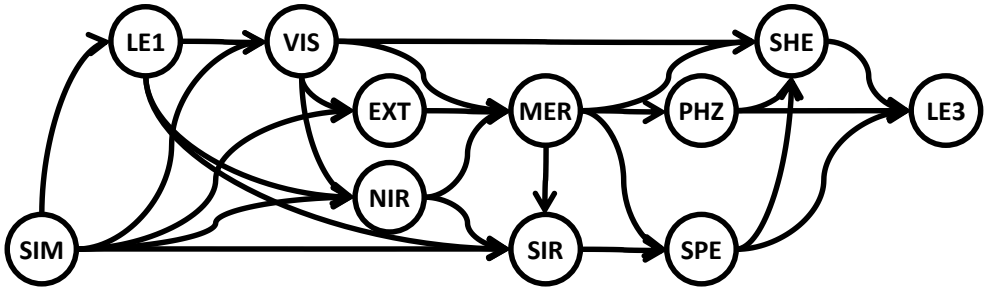
Euclid Computing Architecture

Euclid Data Flow

1. Euclid data arrives from the satellite to the Mission Operations Centre in Spain. There it undergoes level 1 processing.
2. Level 1 data is transferred to the 9 SDCs based on sky area allocations
3. EXT data transferred to SDCs
4. SDCs run the full Level 2 pipeline on their designated sky area
5. Science ready Level 2 data is transferred to the Science Operations Centre in Spain
6. Level 2 products needed for Level 3 processing are transferred to select SDCs
7. Select SDCs run the Level 3 pipeline on the full sky
8. Science ready Level 3 data is transferred to the SOC.



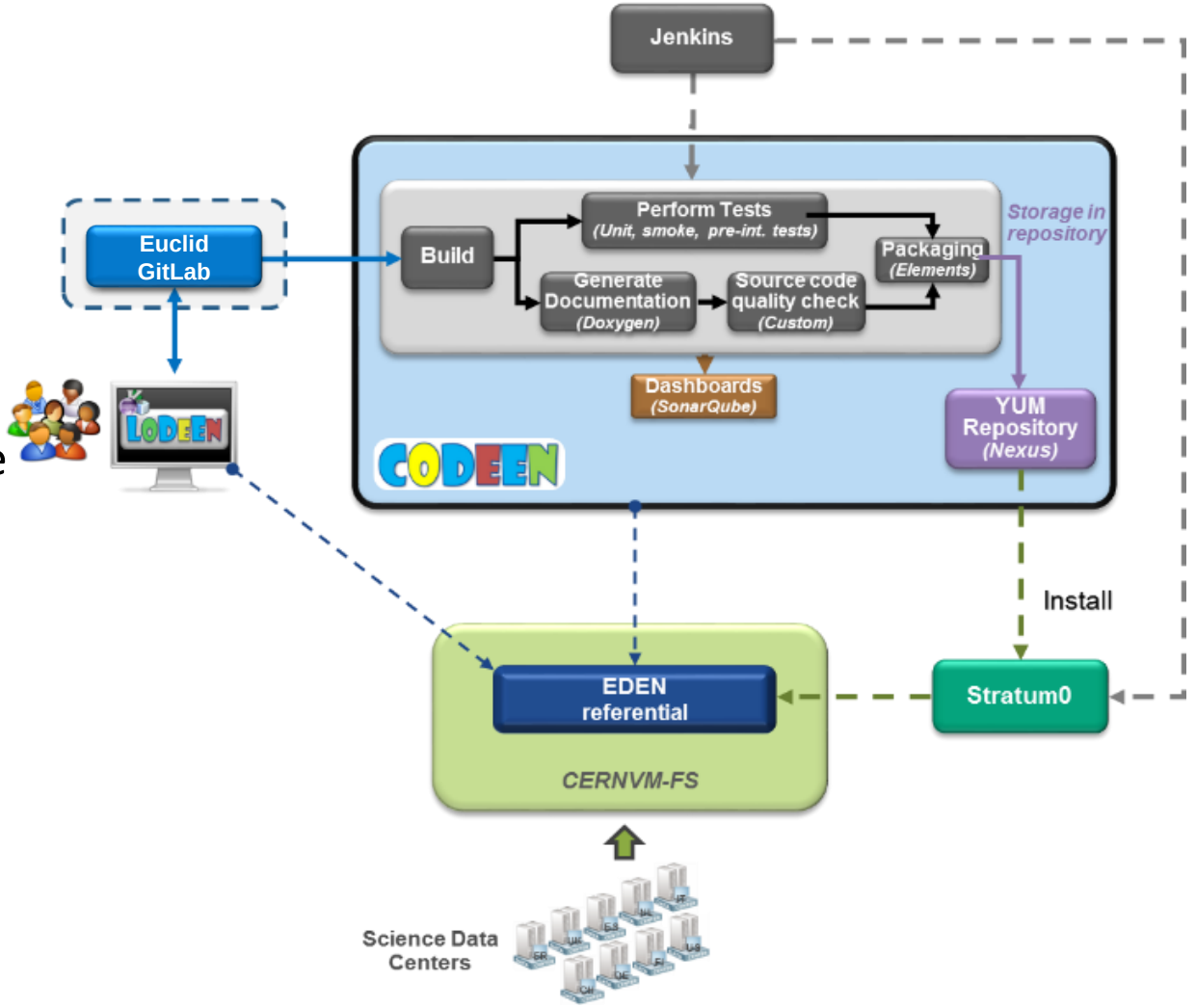
Level 2 Pipeline



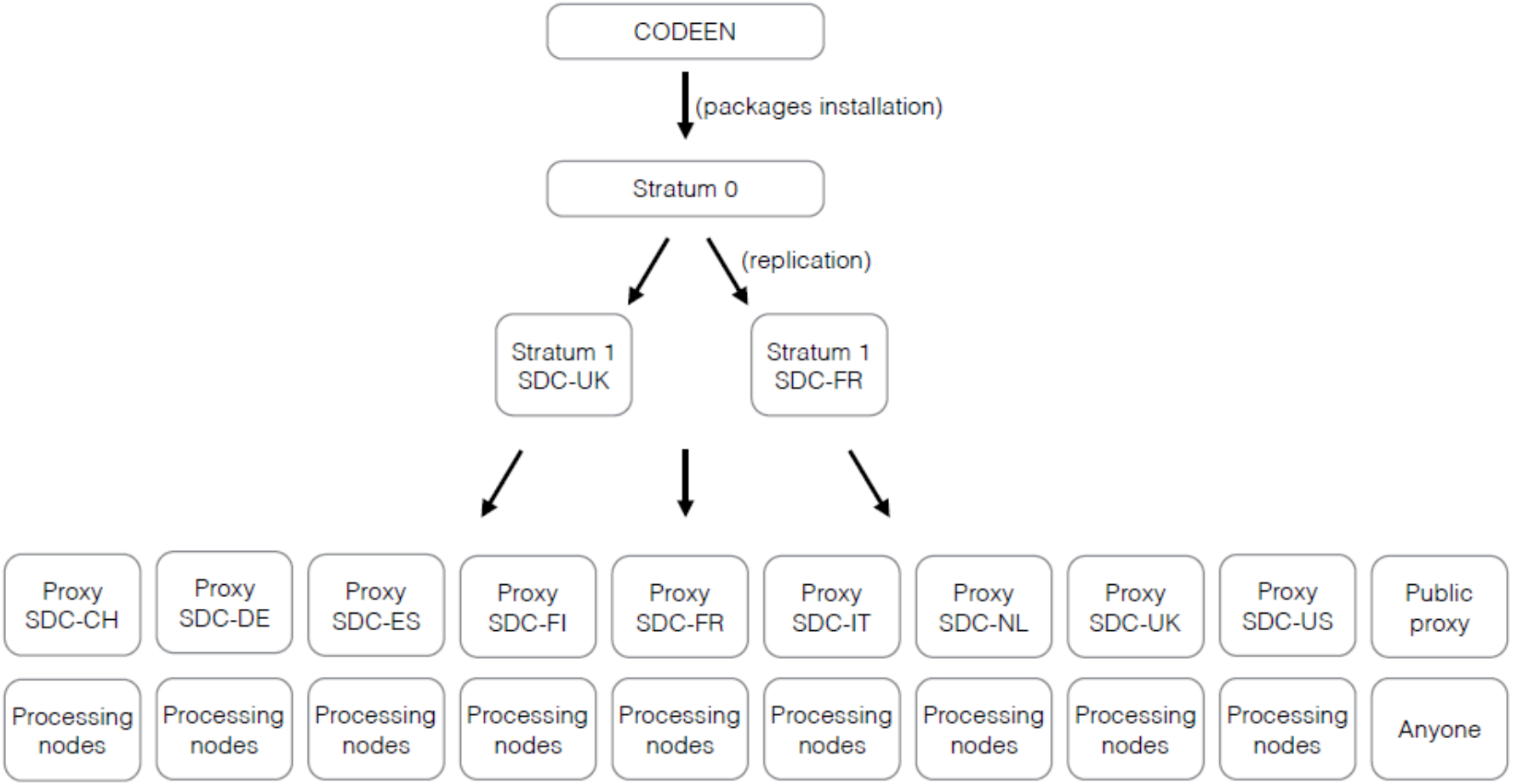
- 11 Processing Functions (PFs) developed by widely distributed teams of scientists and developers
- Data products from PFs often feed into multiple PFs later in the pipeline -> this makes it infeasible to run each PF separately at designated SDCs and then ship the intermediate data to the next SDC for the next PF
- So…ship the code, not the intermediate data!
  - Level 1 data products divided by sky region and assigned to individual SDCs
  - PF code/executables are distributed through CVMFS
  - Developers use a Common Development Environment with set OS/compilers/libraries (Provided as a pre-installed virtual machine known as LoDEEN)
  - Continuous integration methodology followed to allow rapid testing and deployment across all SDCs simultaneously
  - All SDCs run all PFs on their designated sky patches

Develop/Deploy Main Loop

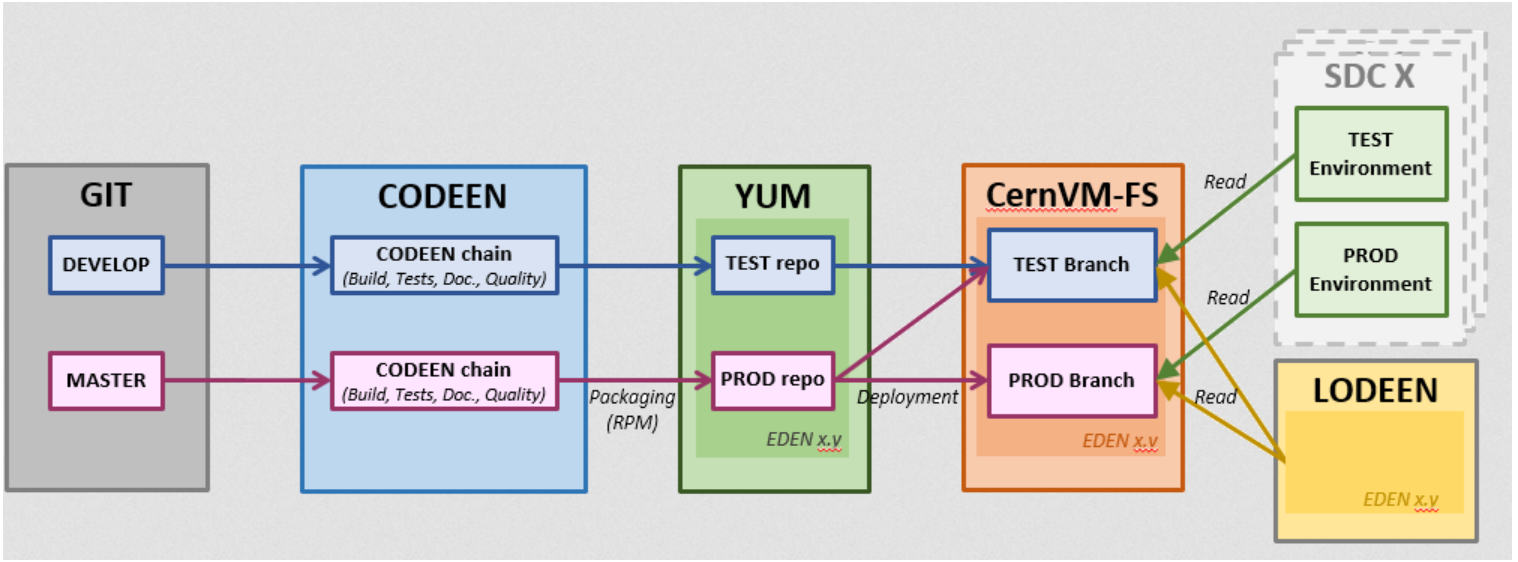
1. Modify your code
2. Commit changes
3. CODEEN runs
4. Packages are deployed via CernVM-FS
5. Packages are accessible on CernVM-FS clients
   - LODEENs
   - CODEEN
   - SDCs



Euclid CVMFS Setup



Euclid CVMFS & Continuous Deployment

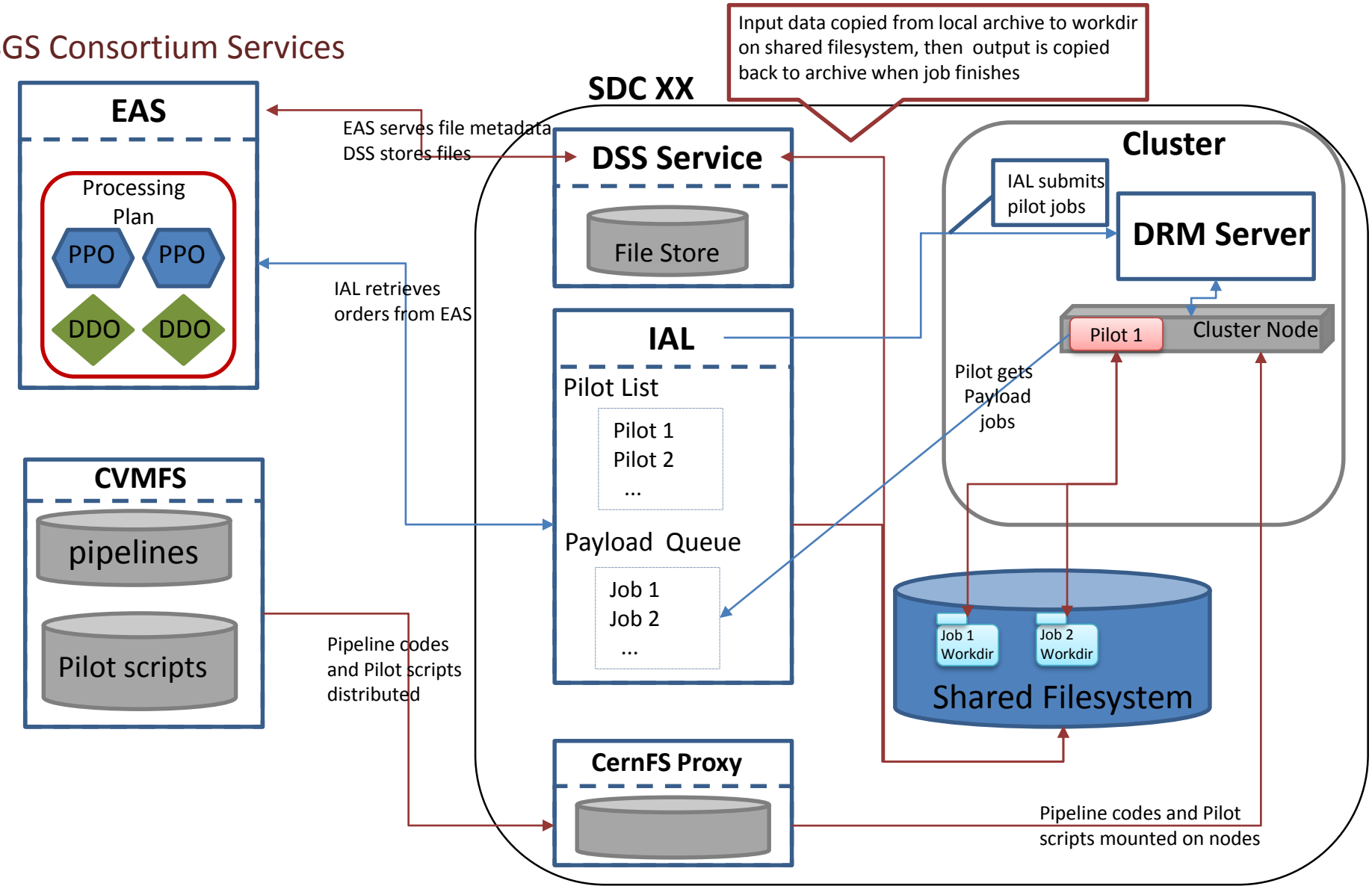


- Both Develop/Test and Production branches
- Few min latency between
  - Installation on stratum 0
  - Availability on any cvmfs client: SDC, LODEEN…

Euclid Infrastructure Services

- Data Management – All data is managed by the Euclid Archive System (EAS) which consists of a central metadata database and a distributed network of storage services known as the DSS (one DSS service at each SDC).
  - **Central metadata database**: The main database contains all the metadata for the mission, including file names and locations across the SDCs.
  - **DSS**: Each SDC runs an instance of the DSS service, which acts as a data manager for ingesting and retrieving files from the EAS. The DSS interfaces with local storage using POSIX, SFTP, iRODS, or XRootD.
- Job Control – Data processing is managed in a 2-step process through the Coordination and Orchestration System (COORS) and the Interface Abstraction Layer (IAL).
  - **COORS**: Operators use COORS to construct Processing Plans (PPs) using the pipelines distributed through CVMFS and input files listed in the EAS. These Processing Plans consist of Pipeline Processing Orders (PPOs), which are effectively a single unit of processing assigned to an SDC. PPOs specify the pipeline codes to run and the input files to use. COORS automatically maps the PPOs to their target SDCs based on the sky allocations. Both PPs and PPOs are stored in the EAS.
  - **IAL**: The IAL acts as the job controller and payload queue at each SDC. It retrieves PPOs from the EAS, determines what files are needed, and retrieves then through the DSS. IAL builds a work directory on the shared filesystem for each PPO with these input files, and it breaks down the PPO into a list of payload jobs. It then submits pilot jobs to the cluster, which run the payload jobs using the work dirs for inputs and outputs. Once all payload jobs for a given PPO are complete, the IAL ingests the output products back into the EAS (using the local DSS for files) and cleans up the work dirs.
- SDC Resources
  - **Compute**: All SDCs are required to provide a batch queue that can run our pilots. The execution environment needs to meet EDEN rules, which can be done as direct installs or through containers (both Docker and Singularity have been used).
  - **Storage**: All SDSs must have a DSS service running and attached to permanent local storage (which serves as a portion of the full Euclid Archive). The clusters are expected to have a shared file system with reasonable IO rates which is visible to all our pilots and the IAL. Traditional file staging was originally supported by the IAL, but this led to a number of job control and IO/bandwidth issues, so it has since been dropped.

SGS Consortium Services



CVMFS Questions

- What is the best architecture for roaming clients?
  - Direct Stratum 1 access?
  - Dedicated public proxies?
- How to access CVMFS through local proxies (authenticated or not)?
  - Is this possible: e.g. CVMFS_HTTP_PROXY="http://<login>:<password>@<localProxyUrl>:<ProxyPort>
- Can CVMFS clients be configured dynamically?
  - e.g. add a stratum 1, add a list of proxies, etc
- What do the proxies look like at CERN?
  - Resources: Virtual Machines or bare metal? CPUs, RAM? Hard drives or SSD, and size?
  - What is the proxy->worker ratio?

CVMFS for Data?

- Problem: Multiple PFs within the pipeline use identical files (like calibration files) for different jobs. Currently the IAL uses symlinks to create a cache for these files to eliminate wasted bandwidth/space, but this can strain shared storage systems due to thousands of open file handles all pointing to a single file.
- Solution: CVMFS for data?
  - We plan on testing CVMFS for data over the next 6 months, using a selection of PFs and files to see if this can improve performance and/or reliability
  - The plan would be to publish certain high-use, low/medium volume files on CVMFS infrastructure to be called as inputs in our standard data processing workflows
  - What infrastructure do we need for these tests?

CVMFS for Data Setup

- Stratum 0 – euclid.data.roe
  - Server at SDC-UK currently configured to host the Stratum 0 data.
  - All data is stored on low-latency, high IO SSD (12TB available), with 10GBps network connectivity
  - Stratum 0 is currently accessible through the local SDC-UK CVMFS proxy, with the repo mounted on a select group of worker nodes
- Infrastructure questions – what are the recommended architectural setups/decisions for building a CVMFS for data system?
  - How much disk (and what speed) at Stratum 0, Stratum 1, and proxies? Do these also require more cores/memory to handle the increased load?
  - Does using CVMFS for data require a higher proxy-> worker ratio?
  - How much disk is needed for the local cache on the workers? Should the client be set to not stream files to local cache, or should this be determined on a case by case basis?
  - Are there any important tuning parameters to implement on the clients?
  - Is WAN distribution performance reasonable, or do we need to keep Stratum 1 level resources local to all SDCs?
  - Are there any good examples of CVMFS for data similar to our use case we can mimic? Or more details on the regional setups used by the ligo and cms examples listed here: https://cvmfs.readthedocs.io/en/stable/cpt-large-scale.html

CVMFS for Data con't

- Repo management
  - What tools/methods are used for ingesting data into a CVMFS for data repo?
    - And what do the turn-around times look like from file generation to ingestion to availability through the Stratum 1s?
  - Are there existing plugins enabling username/password access for securing the data?
  - What storage backends are used for the ligo and cms examples?