Euclid:
Big Data from “Dark” Space

Guillermo BUENADICHA (ESA/ESAC)
Maurice PONCET (CNES)

and C. Dabin, J.J. Metge, K. Noddle, M. Holliman,
M. Melchior, A. Belikov, J. Koppenhoefer

on behalf of Euclid Science Ground Segment System Team

The Euclid Mission

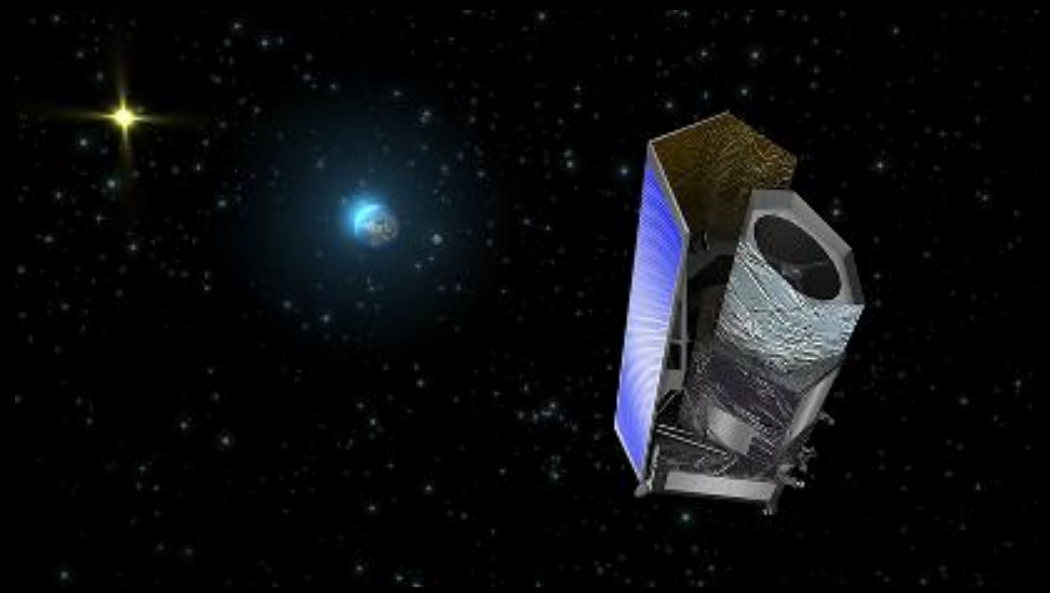
M2 mission in the framework of the **ESA Cosmic Vision Programme**

Euclid mission objective is to map the geometry and understand the nature of the dark Universe **(dark energy and dark matter)**

Actors in the mission: **ESA** and the **Euclid Consortium** (institutes from 14 European countries and USA, funded by their own national Space Agencies)

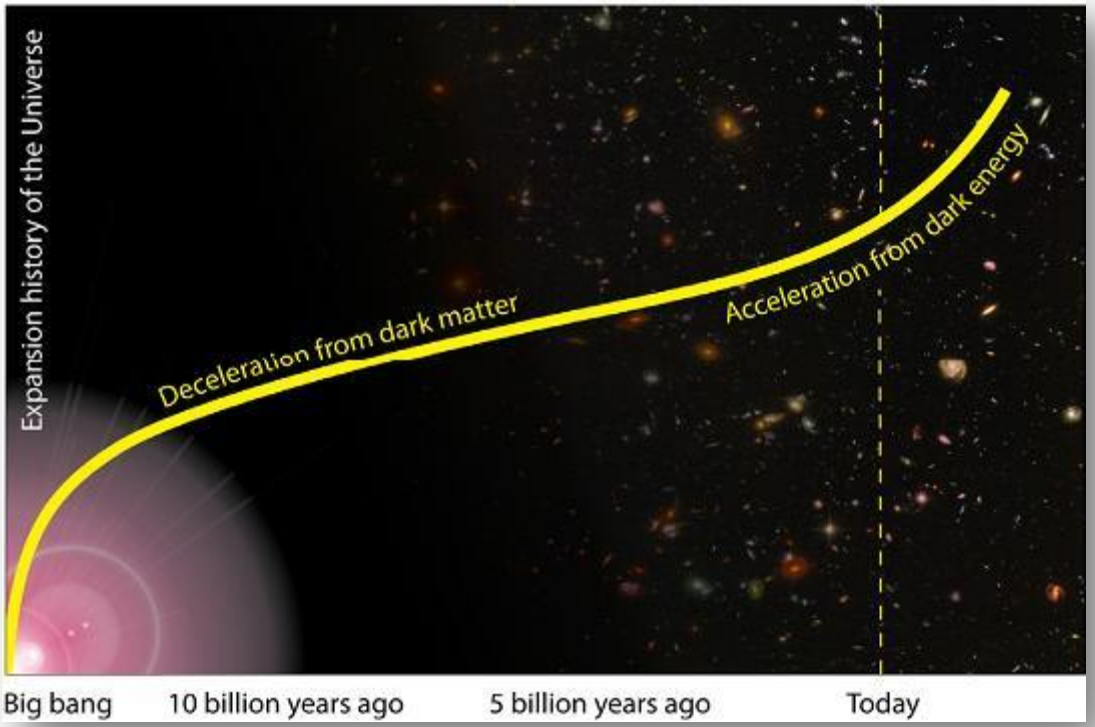
Euclid Consortium:
15 countries
100+ labs
1200+ members

Biggest collaboration!

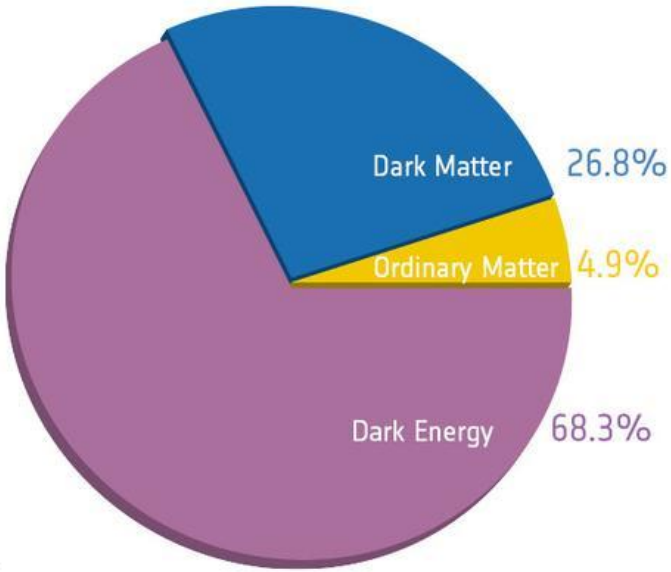


For more information see :
http://sci.esa.int/science-e/www/area/index.cfm?fareaid=102
http://www.euclid-ec.org

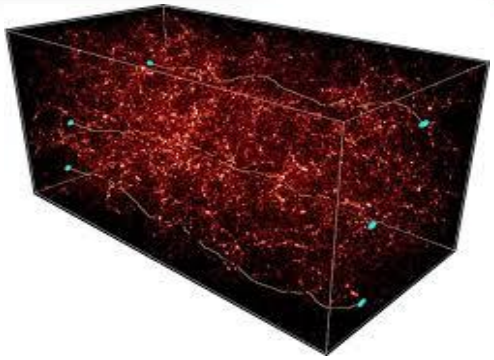
The Dark Universe



The expansion of the Universe is accelerating !



The acceleration of the Universe is produced by a new component called « Dark Energy »

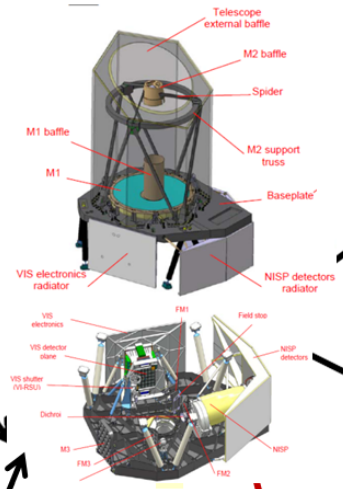


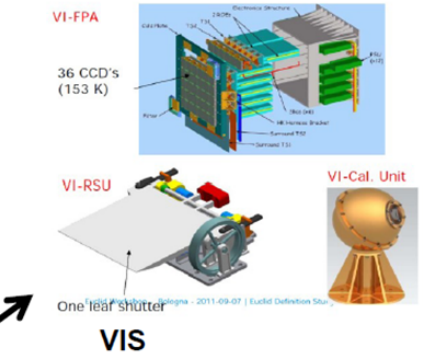
Euclid mission at a Glance

Soyuz@Kourou

Q1 2020

PLM+SVM: 2010-2019



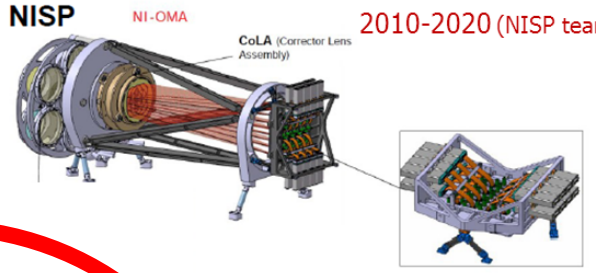


VIS imaging: 2010-2020

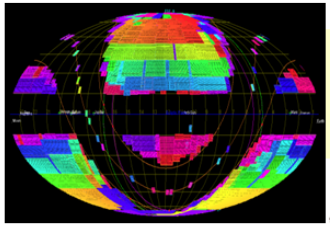
(VIS team)

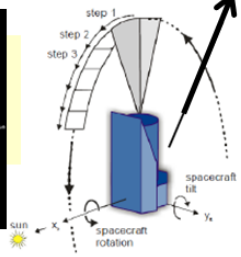
NIR spectro-imaging

2010-2020 (NISP team)



Surveys: 2010-2028 (Survey WG)





6 yrs - 15,000 deg^2

Commisionning – SV

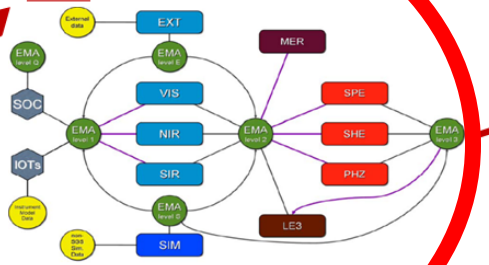
Euclid opération:

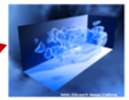
5.5 yrs:Euclid Wide+Deep

+: SNIa, mu-lens, MW?

Ground data

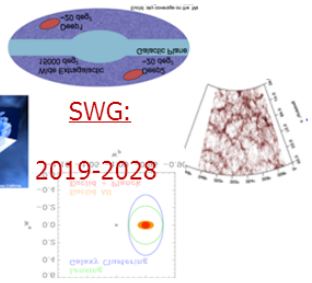
SGS: 2010-2028





SWG:

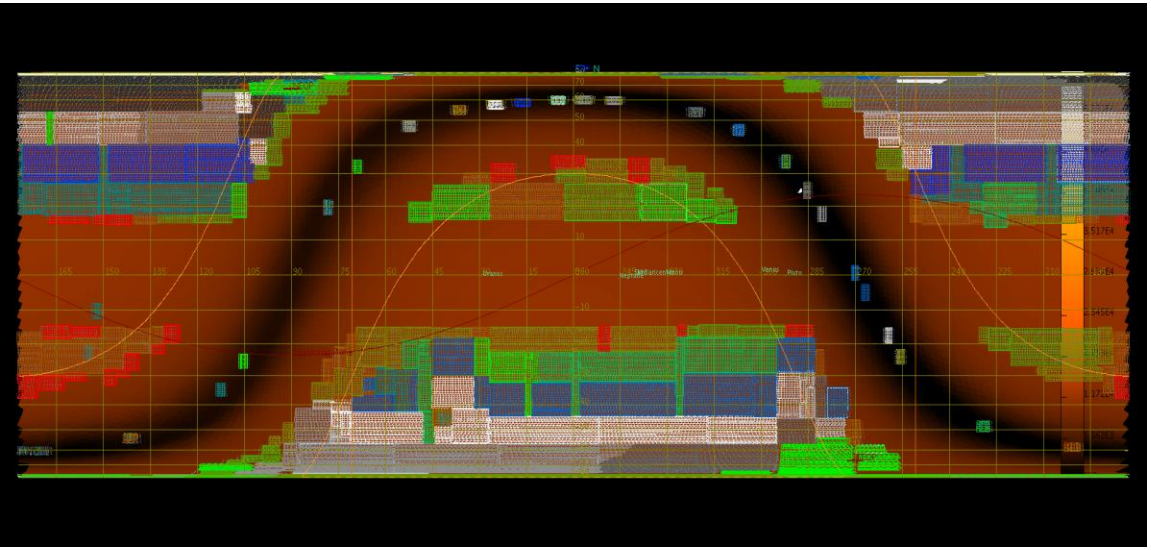
2019-2028

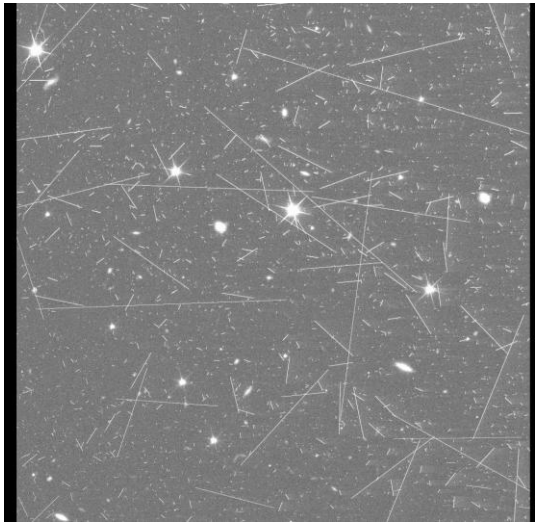


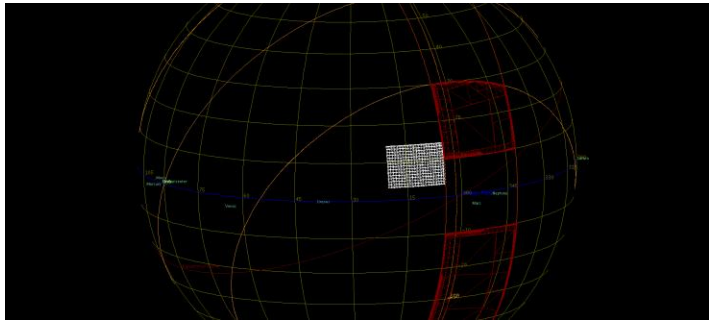
20-30 PB data processing (EC SGS team) – Science analyses

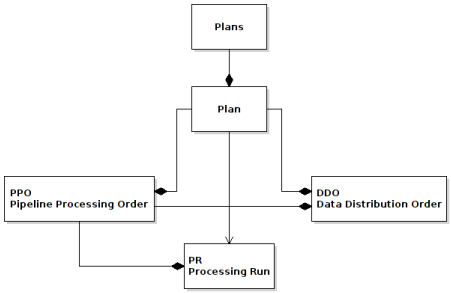
1

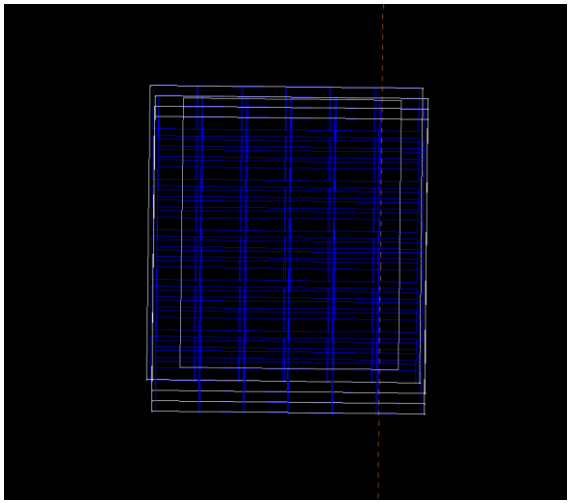
Euclid Survey

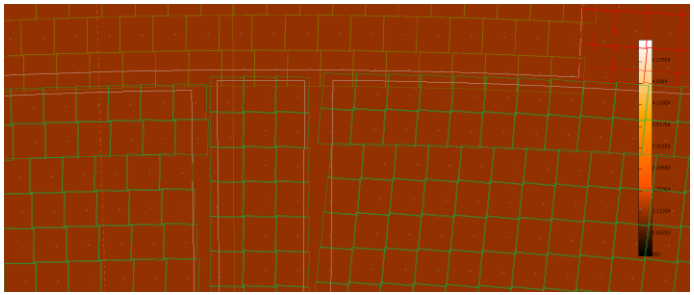




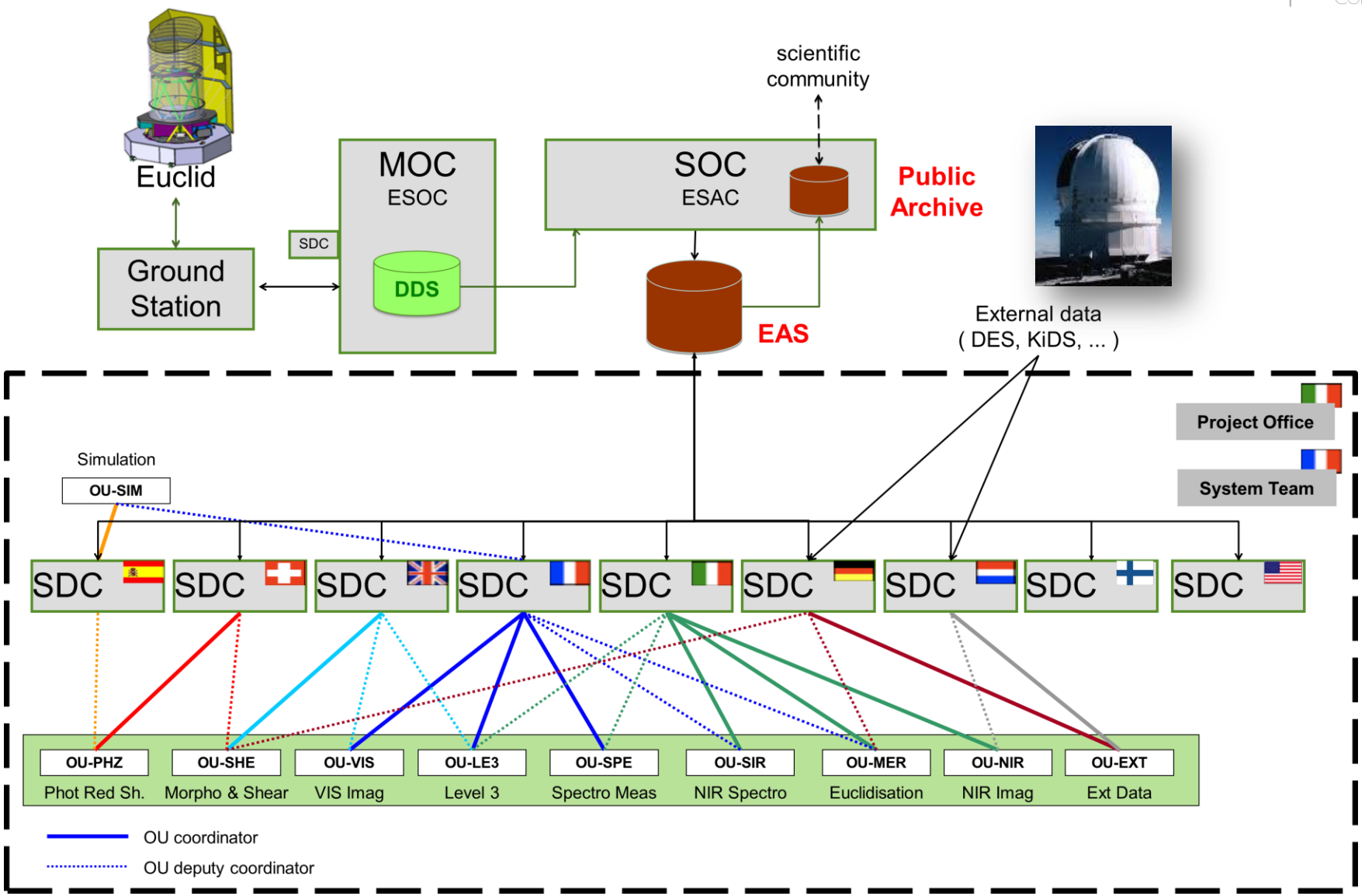




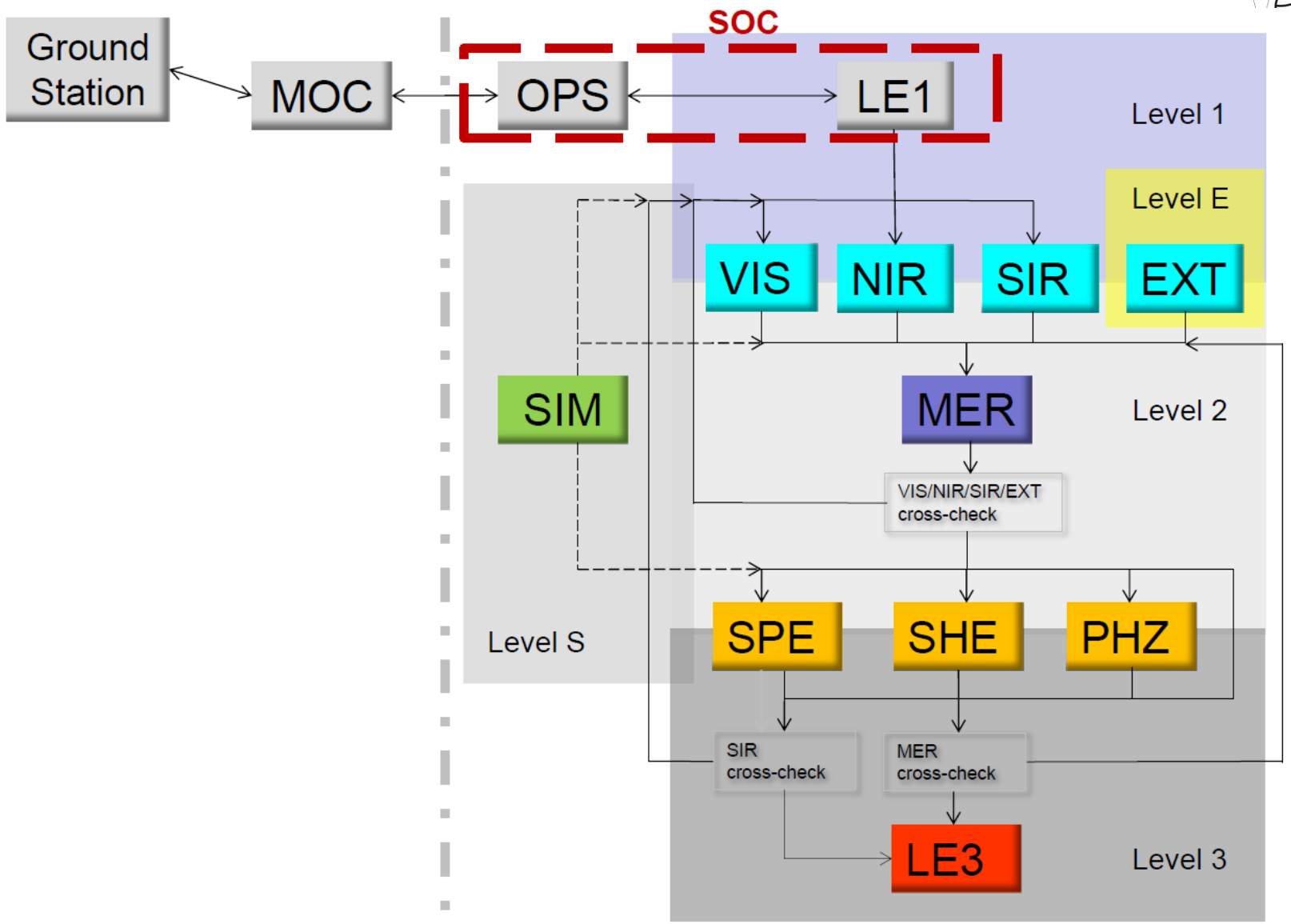




Euclid Ground Segment



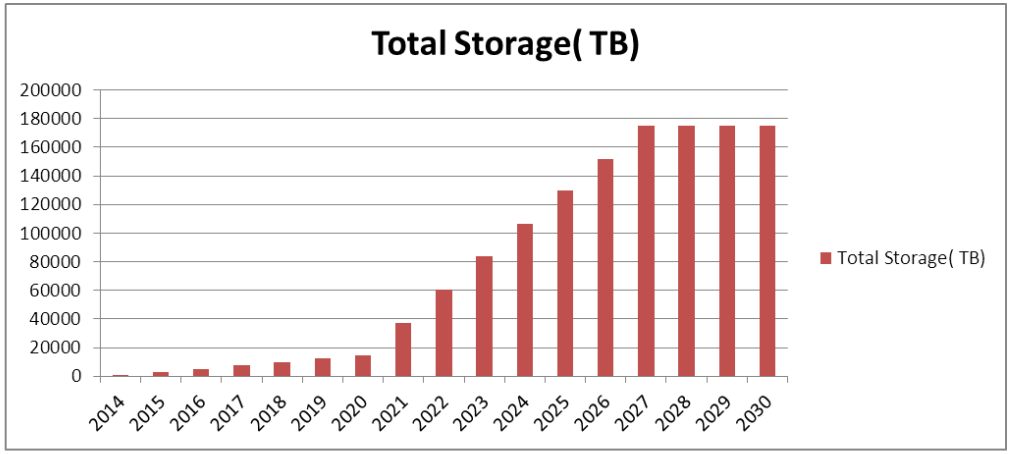
Euclid – Processing Overview

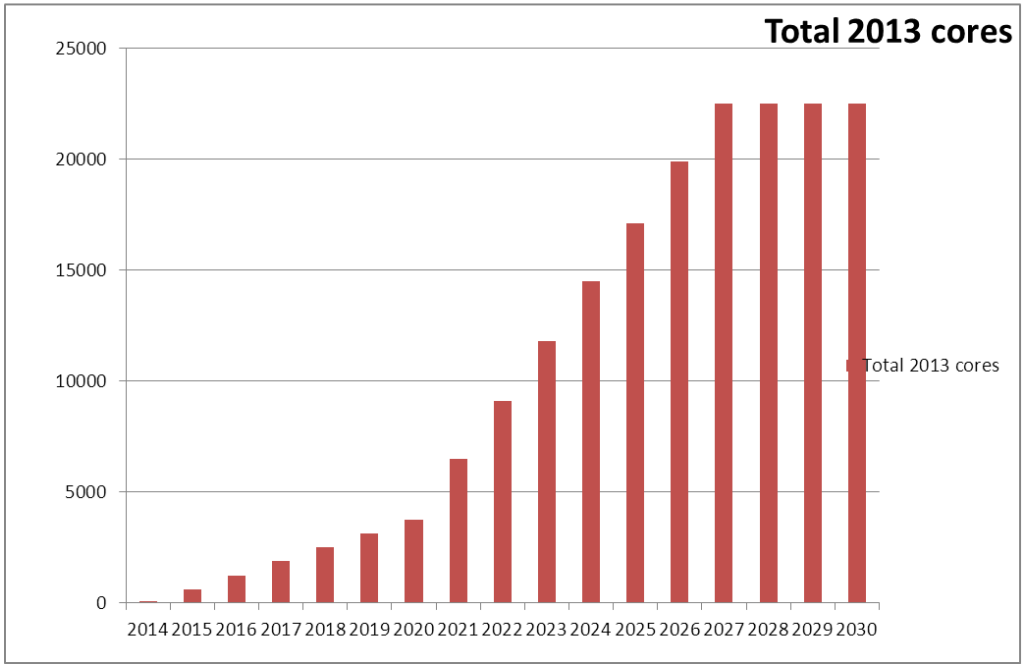


Key Challenges

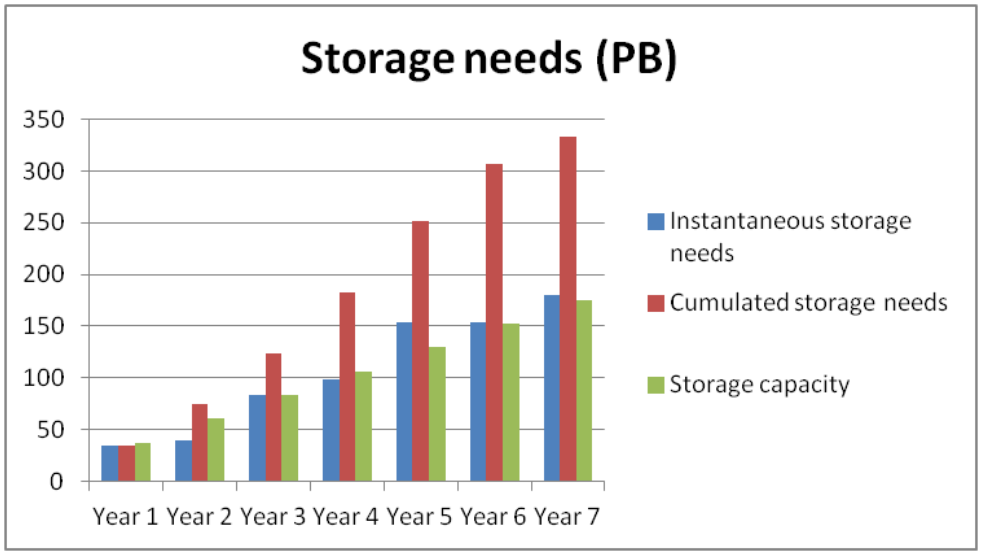
- **Federation** of 8 European + 1 US SDCs (Science Data Centers) + SOC (Science Operation Center)
- Heavy **simulations** needed before the mission
- Heavy (re)**processing** needed from raw data to science products (volume multiplied by dozens),
- Large amount of **external data** needed (ground based observations)
- Amount of **data** that the mission will generate per full release
  - 26 PBytes of data (including external data) => ~175 PB grand total
  - 1.10^{10} objects
  - **=> not achievable with classical architecture**
- **accuracy and quality** control required at each step

Resources estimations



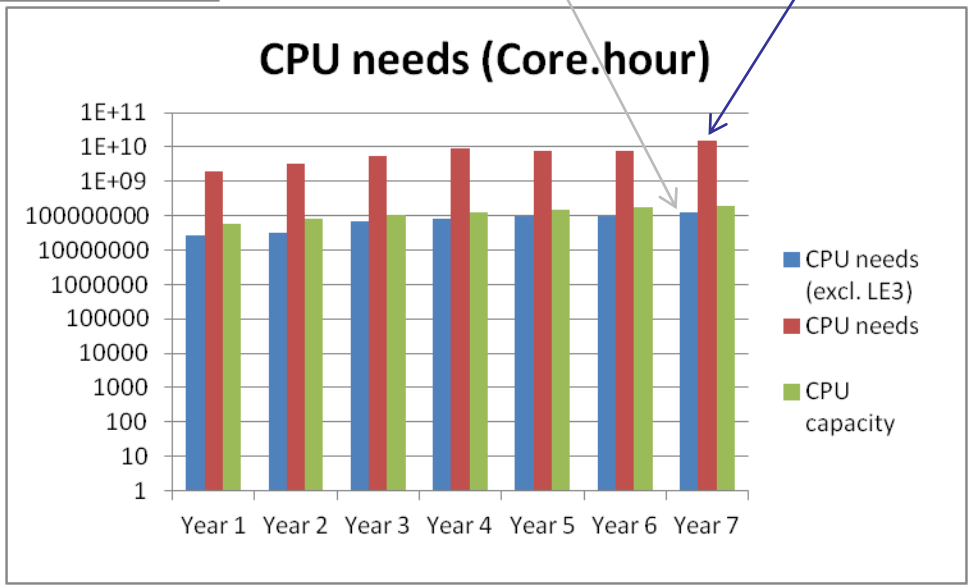


Production resources estimation



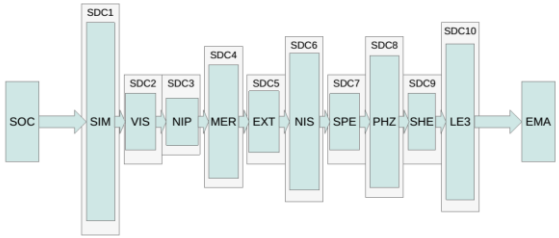
***~10 000 cores** on continuous use all year*

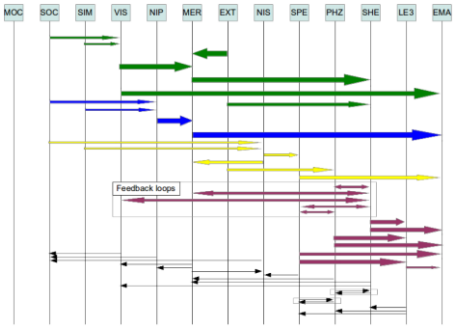
***~1 000 000 cores** on continuous use all year*



Architecture definition

- "Traditional" **Processing Centric** SGS architecture would be: "each SDC runs the code it writes" and the output data from one SDC is then transferred as the input of the next one
- But this schema **implies** some **issues**:
  - **Unequal load** between SDCs
  - How to deal with a new SDC ?
  - How to deal with the loss of an SDC ?
  - Each SDC = SPOF
  - How to set up and fund redundancy ?
  - **Data Volumes** over WAN (plenty of PBs) !





- Thus a **Data Centric** rather than a Processing Centric approach is **more relevant**:
  - Allocate the data and not the processing to the SDCs
  - Run AMAP the "whole" pipeline on any SDC on the smallest meaningful processable bundle of data (QoD: Quantum of Data)

Architecture key concepts

- **No Dedicated Processing SDC**: Any pipeline should run on any SDC (with some exceptions, e.g. Level 1, EXT ingestion, LE3)

- **Distributed Data and Processing**
  - Each SDC is both a processing and a storage « node »

- **Move the code, not the data**
  - Run the pipeline where the main input data is stored

- **Separation** of **metadata** (inventory) from **data** (storage)

- Kind of home made "*Map/Reduce*"
  - Lower level of processing on QoD (minimal processable set of data covering a given sky area), constituting catalogs of objects
  - Higher level of processing based on data cross-matching/correlation: need to colocate reduced set of data (whole catalog)

Logical Architecture

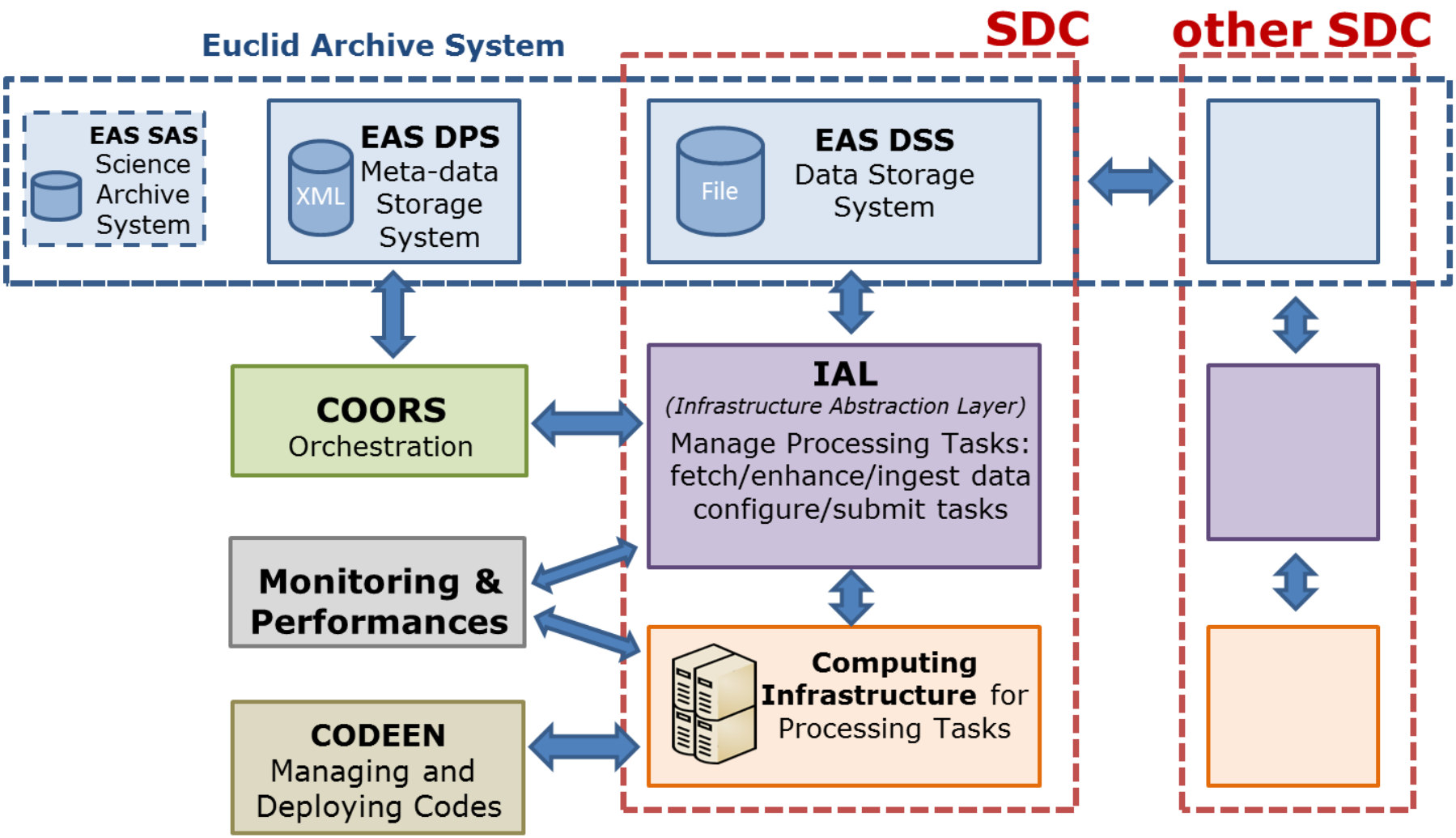
- A set of **Services** which allows a low coupling between SGS components : e.g. metadata query and access, data localization and transfer, data processing M&C, …

- A Euclid **Archive** System (EAS)
  - ✦ A **central Metadata Repository** which inventories, indexes and localizes the huge amount of distributed data,
  - ✦ A **Distributed Storage System (DDS)** of the data over the SDCs (ensuring the best compromise between data availability and data transfers), with redundancy

- **M&C** and **Orchestration (COORS)** layers responsible for distributing data and processing among the SDCs, according to a distribution policy

- An **Infrastructure Abstraction Layer** (**IAL**) allowing the data processing software to run on any SDC independently of the underlying IT infrastructure, and simplifying the development of the processing software itself (e.g. takes care of I/O and I/F)

Architecture components



EuclidVM principles

- Any Euclid Processing Function (PF) should run on any SDC, but the **SDCs are not homogeneous**:
  - Hosting O/S, compilers, libs and versions, …
  - Kind of infra: cluster, cloud, shared storage or not, …

- Thus concept of **EuclidVM**
  - A **Processing node** VM appliance for Euclid
  - Relies on **virtualization** at any SDC: independence from hosting O/S
  - Allows to deploys the **same guest** processing **VM everywhere**
  - Develop, test, integrate, validate "once" on a **reference platform**
  - EuclidVM:
    - **Lightweight VM** with core O/S (SLx,CentOS 7) and most stable core S/W
    - "Dynamic" **Deployment** of libs and PF S/W in push or pull mode

- Candidate technos
  - CernVM ecosystem (µCernVM, CernVM-FS, …)
  - Docker

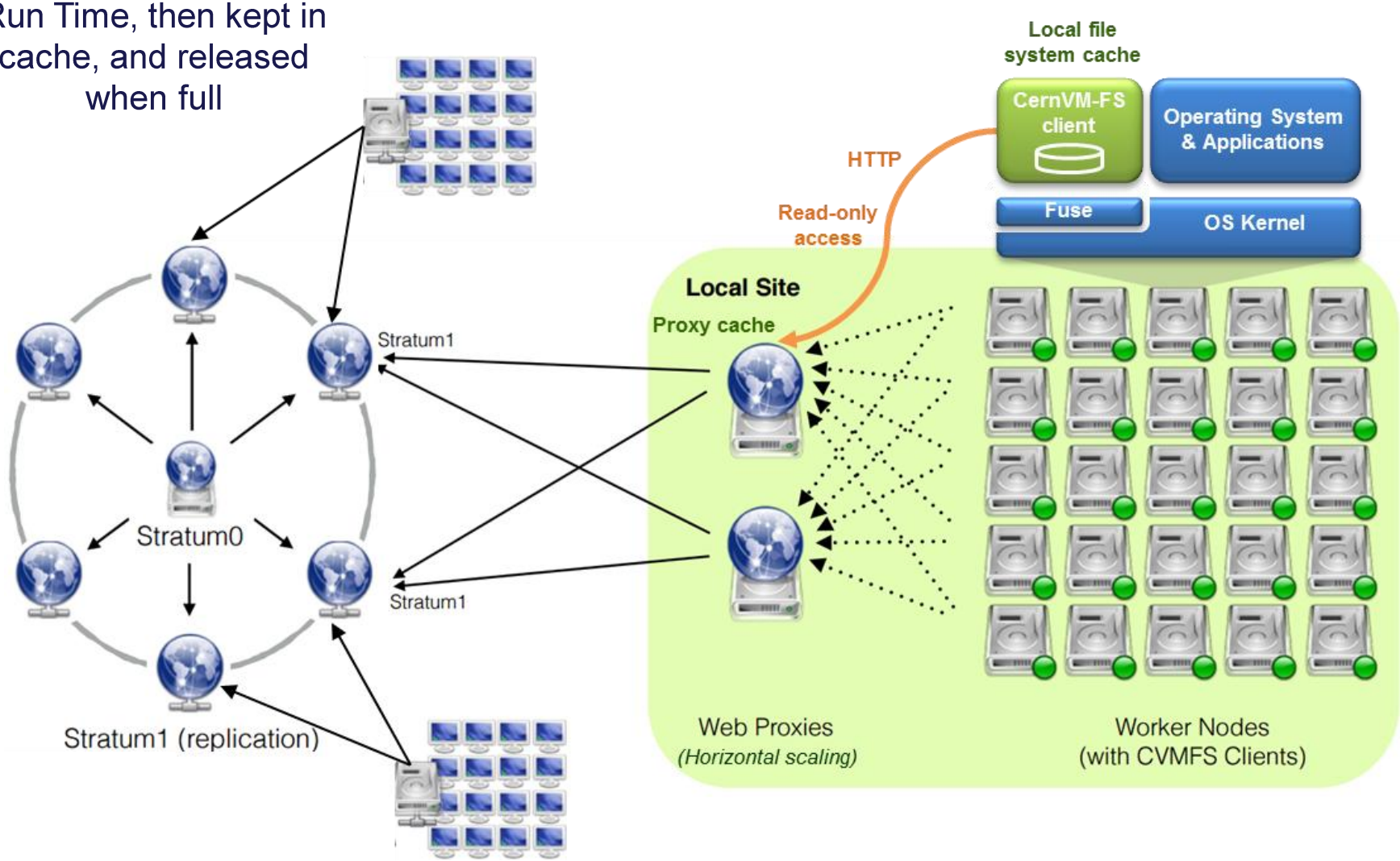
Continuous Deployment

- Need to symplify chain from development to deployment (DEVOPS)
  - allow early tests & improvements on target SDCs

- Candidate Solution: CernVM-FS
- Principles:
  - A central repository of Software (Stratum 0) => unique reference
  - A set of distributed replicas (Stratum 1) => scalability and availability
  - SDC Local Squid proxies => performances
  - CernVM-FS client installed at each Processing Nodes
  - Optimized HTTP protocol
  - Local cache
  - Files are downloaded and cached only on access

Continuous Deployment

SW files accessed in Run Time, then kept in cache, and released when full



What's next

Now…

How to move on and make it run ?

Euclid SGS Challenges

- The SGS Challenges are kinds of "Proof of Concept" that are deployed at **real scale** based on SGS services or Processing Functions **prototypes**, in an **iterative and incremental** process, **involving** and **motivating** all stakeholders:
  - Clear objectives and directives
  - Each challenge stays active after completion and is the foundation for the next one
  - One challenge ~every 6 months
  - One SDC rotating leadership
  - All SDCs have to play the game
  - Online Dasboard: motivating
  - Either technical or scientific oriented

Euclid SGS Challenges

- Architecture Challenge #1 – 2012-2013 – SDC-DE leadership:
  - Monitoring Network bandwidth btw SDCs (iperf)
- Architecture Challenge #2 – 2013 – SDC-FR leadership:
  - Deployment of a first simulation prototype on any SDC through Jenkins slaves from the Euclid COmmon DEvelopment ENvironment (CODEEN)
- Architecture Challenge #3 – 2013-2014 – SDC-FR leadership:
  - Deployment of an IAL prototype, as a VM, on any SDC
  - Launch simulation prototypes and store outputs locally
  - Store the corresponding metadata into the EAS prototype
- Architecture Challenge #4 – 2014-2015 – SDC-UK/DE leadership:
  - Introduction of DSS, COORS and M&C (Icinga) prototypes
- Scientific Challenge #1 – 2014-2015 – SDC-ES leadership:
  - Simulation of VIS & NISP instruments outputs on ~20 deg^2
- Scientific Challenge #2 – 2015-2016 – leadership Italy:
  - Introduction of 1st level of processing prototypes (VIS, NIR & SIR)
- …

ST Challenge #3

Final goal of challenge : deploying transparently pipelines on all SDCs

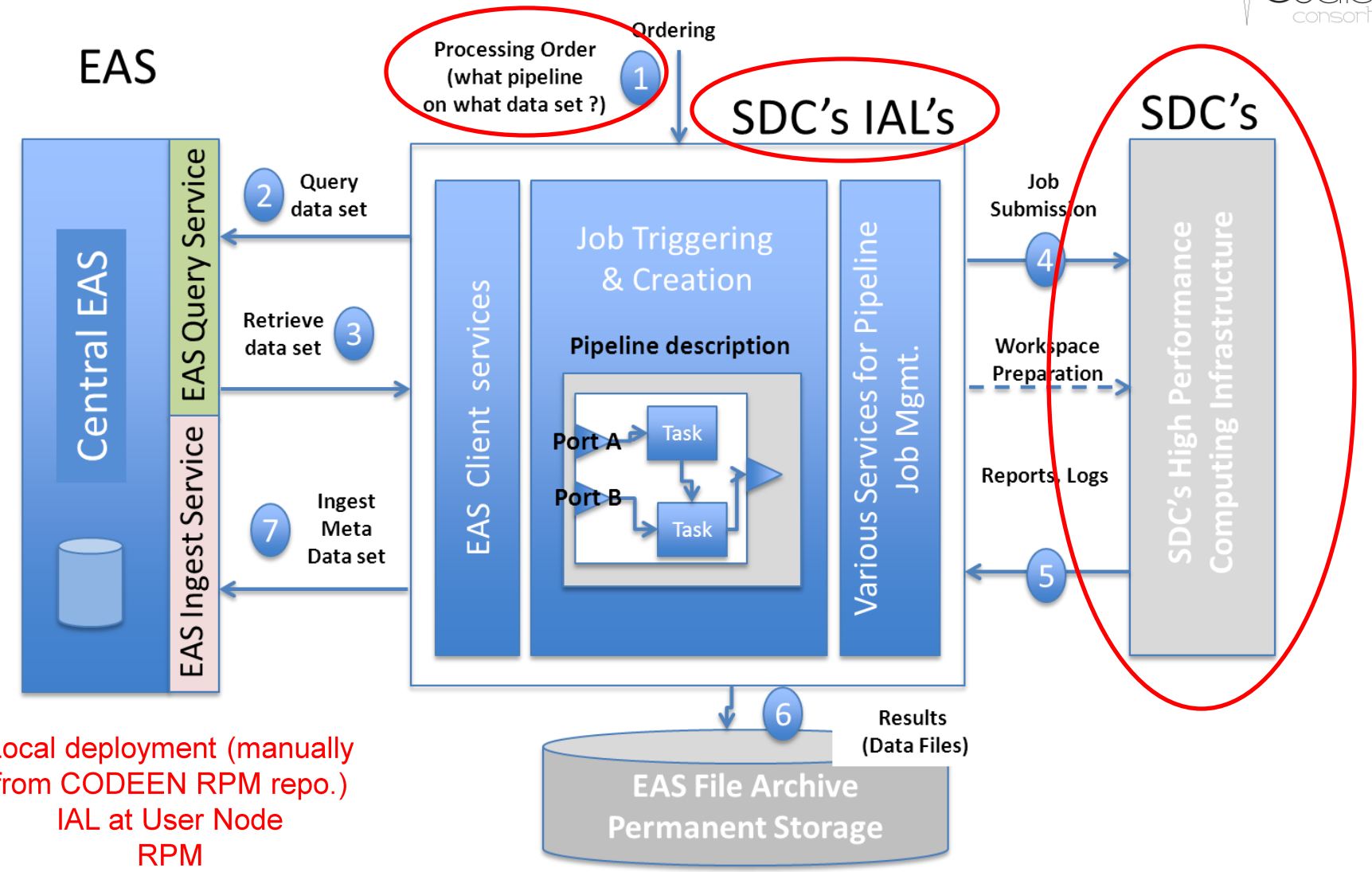
Technical objectives :

- Demonstrate the capability to deploy IAL VM images into SDCs
- Demonstrate the capability to deploy, in the context of each SDC, the TIPS, NIP and VIS simulators as Euclid pipeline objects
- Demonstrate the capability of IAL, in the context of each SDC, to :
  - *fetch, on the basis of the metadata provided by EAS prototype (in SDC-NL), the pipelines input data in the local SDC storage area*
  - *launch simulators jobs across clusters (when available in SDCs) or dedicated nodes, in accordance with PPOs defined remotely (through Jenkins) or locally (by each SDC leader)*
  - *produce and store output data into the local SDC storage area*
  - *send the appropriate metadata to EAS prototype in SDC-NL*

Schedule:

- Baseline availability for deployment into SDCs : end of December 2013
- By mid-February 2014, all SDCs had successfully fulfilled the challenge

SGS – Mockup (Challenge 3)



Local deployment (manually
from CODEEN RPM repo.)
IAL at User Node
RPM
Basic PO
No DSS

ST Challenge IT #4 - #6

Final goal of challenge : deploying SGS services mockup

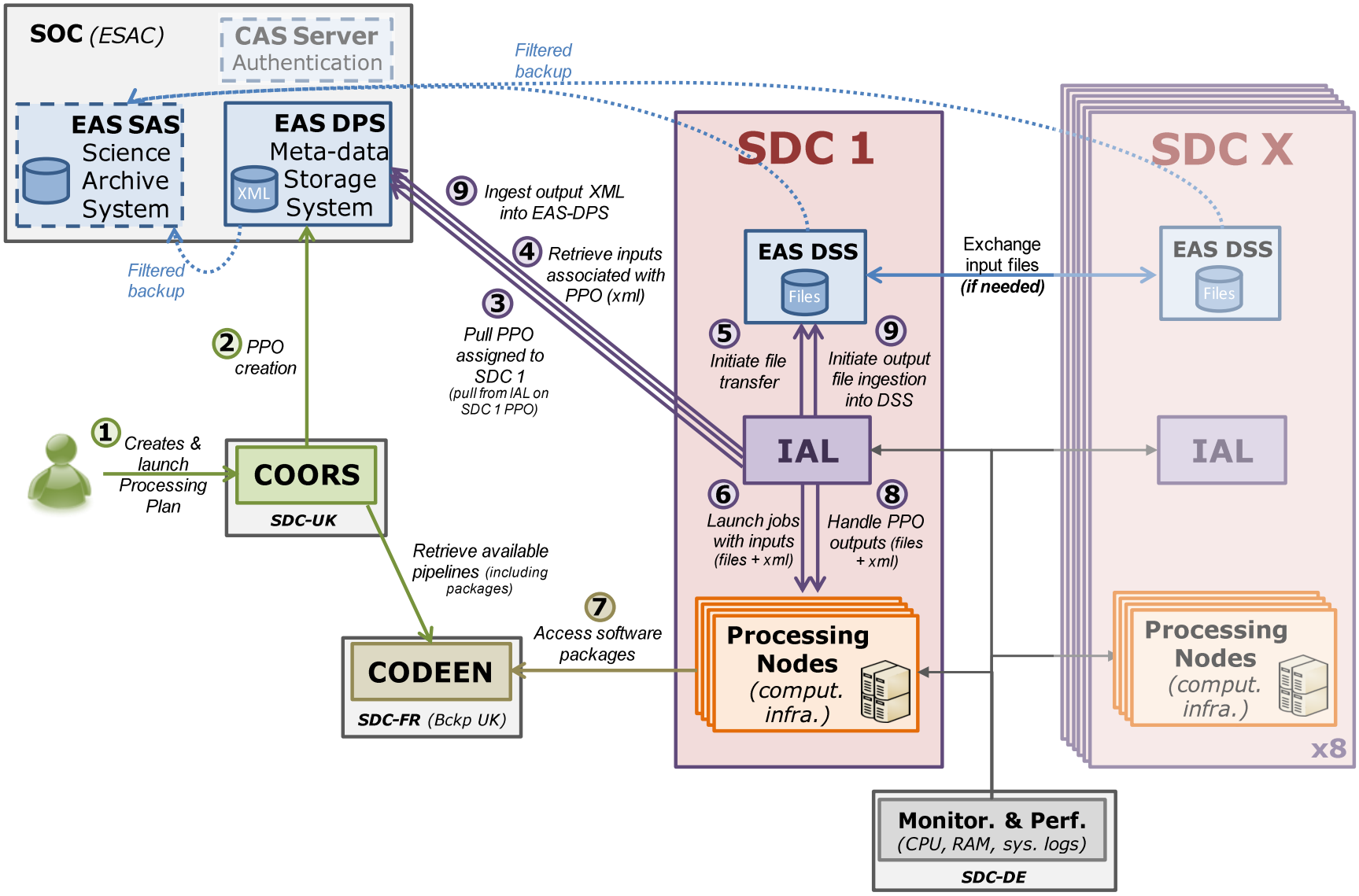
Technical objectives :

- Deploying Distributed Storage Server among SDCs (DSS)
- Deploying Distributed Monitoring mockup
- First Orchestration mockup (COORS)
- Enhanced IAL version (e.g. workflow)
- CernVM-FS deployment testbed
- Docker vs μ CernVM testbeds
- Running instruments (VIS, NISP) simulators prototypes
- First performance tests

Schedule:

- Ongoing 2015-2016

SGS – Dynamic Architecture



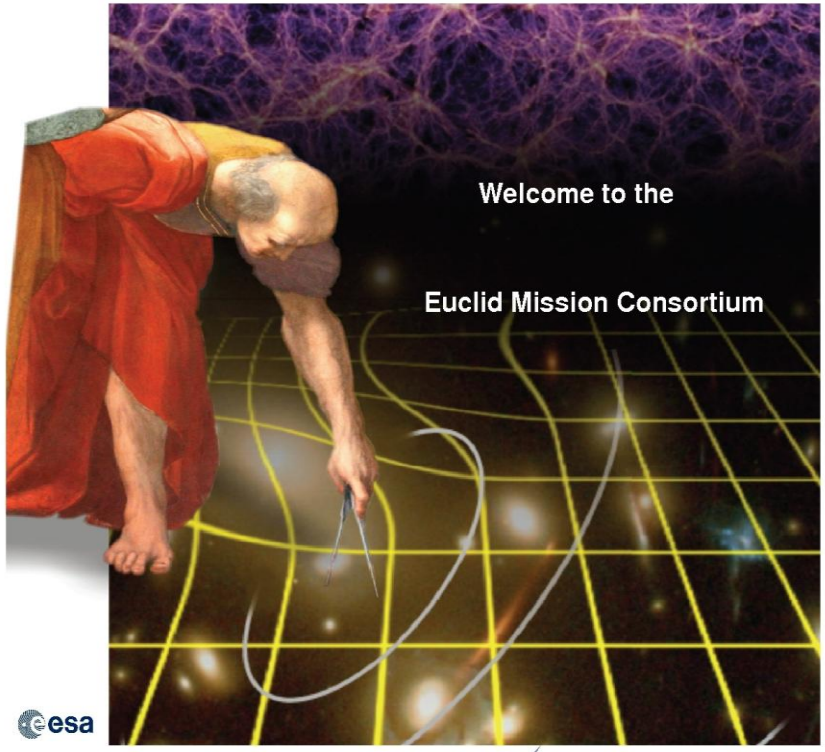
Conclusions

- Big challenge !

- Already active working groups on:
  - Architecture principles
  - POC Mock-up & challenges

- Working prototypes => pillars of the SGS

- Next steps
  - Refine the architecture model according to the scientific processing requirements (granularity, triggering, volumes, …)
  - Identify candidates implementations
  - Interleave scientific & architectural challenges



Thank you for your attention

guillermo.buenadicha@esa.int
Maurice.Poncet@cnes.fr

Acknowledgments: authors are indebted to all the indididuals participating in the Euclid SGS developement inside ESA and EC, too n

Additional Slides

The Euclid Consortium

- The **Euclid Consortium** is in charge of:
  - building and operating the **instruments** (VIS and NISP)
  - developing and running the **data processing** within a unified **Science Ground Segment** (SGS)
  - performing the **science analysis** on the Euclid data products

- The Euclid Consortium is composed of
  - 15 countries
  - 100+ labs
  - 1300+ members

26 PB mean…

1 Tb
1,5 cm
10 w

324 m

…

26 PB =>
390 m
3,5 t
260 Kw

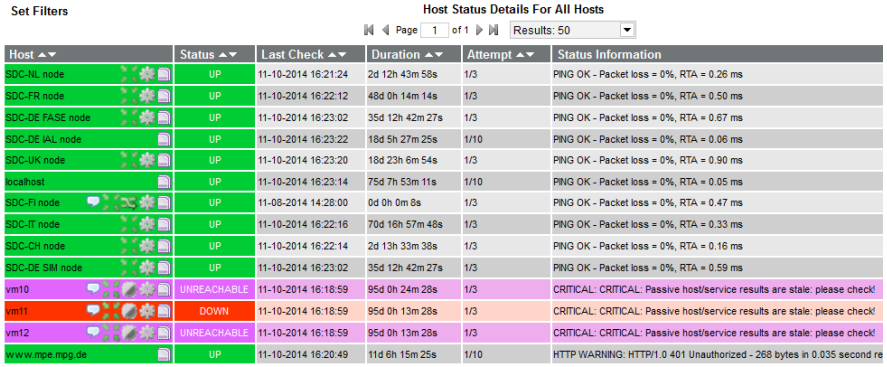
10 Gb/s
13 min/ TB

26 PB =>
240 days!

48 hours !

Euclid – M&C

- M&C aims namely to **monitor** both **bandwidths**, **storage** and **processing** among the SGS
- The current monitoring prototype covers both:
  - Network bandwidht between SDSs (**iperf**)
  - SDCs resource monitoring (**Icinga**)



Set Filters

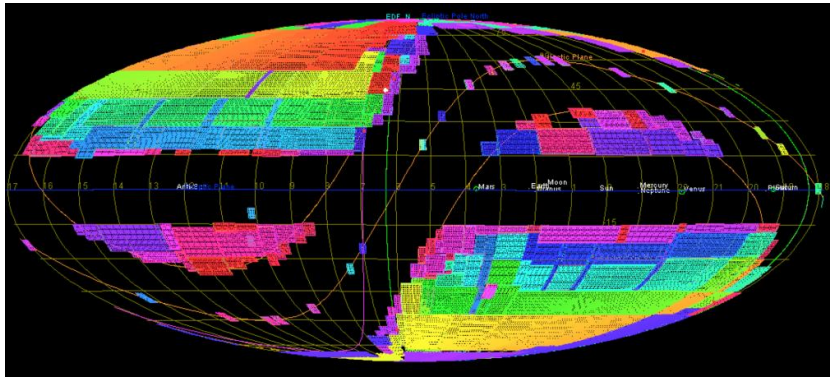
Host Status Details For All Hosts

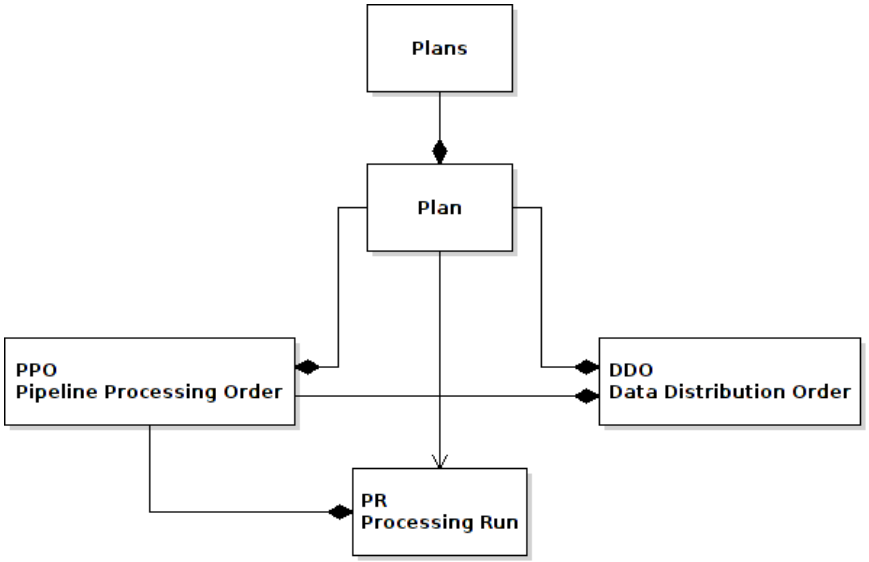
Page 1 of 1 Results: 50

Host	Status	Last Check	Duration	Attempt	Status Information
SDC-NL node	UP	11-10-2014 16:21:24	2d 12h 43m 58s	1/3	PING OK - Packet loss = 0%, RTA = 0.26 ms
SDC-FR node	UP	11-10-2014 16:22:12	48d 0h 14m 14s	1/3	PING OK - Packet loss = 0%, RTA = 0.50 ms
SDC-DE FASE node	UP	11-10-2014 16:23:02	35d 12h 42m 27s	1/3	PING OK - Packet loss = 0%, RTA = 0.67 ms
SDC-DE IAL node	UP	11-10-2014 16:23:22	18d 5h 27m 25s	1/10	PING OK - Packet loss = 0%, RTA = 0.06 ms
SDC-UK node	UP	11-10-2014 16:23:20	18d 23h 6m 54s	1/3	PING OK - Packet loss = 0%, RTA = 0.90 ms
localhost	UP	11-10-2014 16:23:14	75d 7h 53m 11s	1/10	PING OK - Packet loss = 0%, RTA = 0.05 ms
SDC-FI node	UP	11-08-2014 14:28:00	0d 0h 0m 8s	1/3	PING OK - Packet loss = 0%, RTA = 0.47 ms
SDC-IT node	UP	11-10-2014 16:22:16	70d 16h 57m 48s	1/3	PING OK - Packet loss = 0%, RTA = 0.33 ms
SDC-CH node	UP	11-10-2014 16:22:14	2d 13h 33m 38s	1/3	PING OK - Packet loss = 0%, RTA = 0.16 ms
SDC-DE SIM node	UP	11-10-2014 16:23:02	35d 12h 42m 27s	1/3	PING OK - Packet loss = 0%, RTA = 0.59 ms
vm10	UNREACHABLE	11-10-2014 16:18:59	95d 0h 24m 28s	1/3	CRITICAL: CRITICAL: Passive host/service results are stale: please check!
vm11	DOWN	11-10-2014 16:18:59	95d 0h 13m 28s	1/3	CRITICAL: CRITICAL: Passive host/service results are stale: please check!
vm12	UNREACHABLE	11-10-2014 16:18:59	95d 0h 13m 28s	1/3	CRITICAL: CRITICAL: Passive host/service results are stale: please check!
www.mpe.mpg.de	UP	11-10-2014 16:20:49	11d 6h 15m 25s	1/10	HTTP WARNING: HTTP/1.0 401 Unauthorized - 268 bytes in 0.035 second re

Euclid – COORS principles

- COmmon ORchestration System (**COORS**) principles:
  - Manages **data and processing distribution** among the SDCs through the IAL and the DSS
  - Behaves according to **Processing plans**
  - **Data distribution policy** should be static and coherent with mission planning and sky areas, incl. data redundancy



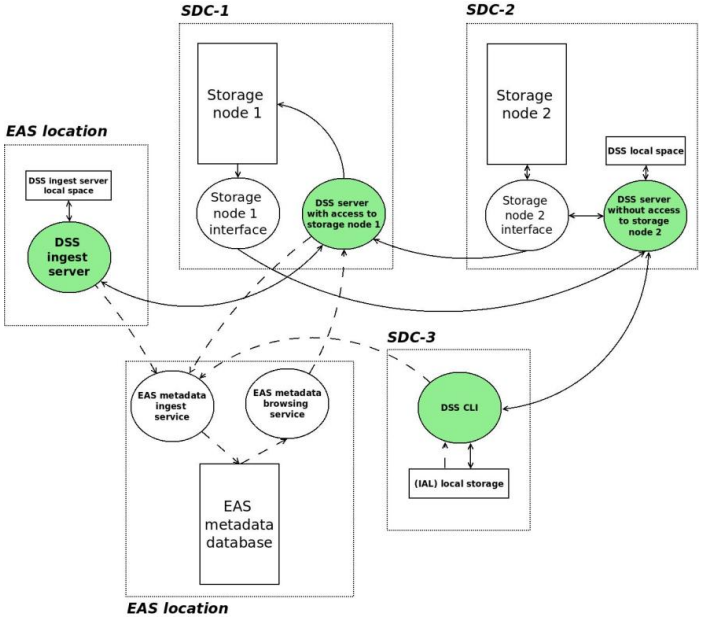


Euclid – EAS principles

- Euclid Archive System (**EAS**) :
  - Data **Inventory**
  - Data **Localization**
  - **Metadata** Repository (generic header + specific metadata, incl. data quality and lineage)
  - Metadata **CRUD** access
  - **Query** Interface (DBMS agnostic)
  - **Publication** (data access for the science community)
  - Version control
  - Data access **rights** management

Euclid – DSS principles

- Distributed Storage System (**DSS**):
  - Data are **stored** at SDC level
  - Data are **distributed** among SDCs
  - Data are **replicated** : at least a primary storage and a secondary storage
  - Data **distribution policy** should minimize the data transfers
    - By data processing level
    - By sky area
    - …
  - DSS relies on **SDC existing storage**
  - DSS provides:
    - Retrieve, Store, Copy, Delete **operations**



Euclid – IAL Principles

- **Infrastructure Abstraction Layer** (IAL) isolates the pipeline from the underlying infrastructure :
  - **Pre processing** step :
    - Queries and retrieves the input data
    - Takes care of the resources needed by the pipeline
    - Creates the working context for the pipeline
  - **Running step**
    - The pipeline runs in a "sandbox" and knows only about it (no external access)
    - IAL manages pipeline control and data flow
    - A Set of minimal and basic pipeline interfaces : inputs/outputs, M&C, parameterization
  - **Post processing** step :
    - inventory and storage of outputs (metadata + data),
    - notification

Development Environment

- DEVOPS principles: continuous integration & deployment
- CODEEN (Collaborative Development Environment)
  - ePlatform based on Jenkins engine allowing **continuous integration and deployment** approach
  - Steps: Build, Doc., Tests, Quality Check, Packaging, Deployment
- LODEEN (Local Development Environment)
  - Ready to use VM dedicated to S/W **development** on local machine
- Standards (EDEN)
  - O/S : SL6, => **CentOS 7**
  - Languages: **C++ / Python**
  - Restricted set of supported **libs**
  - **Coding standards**

Development Environment - EDEN