SPD: data processing in a distributed environment

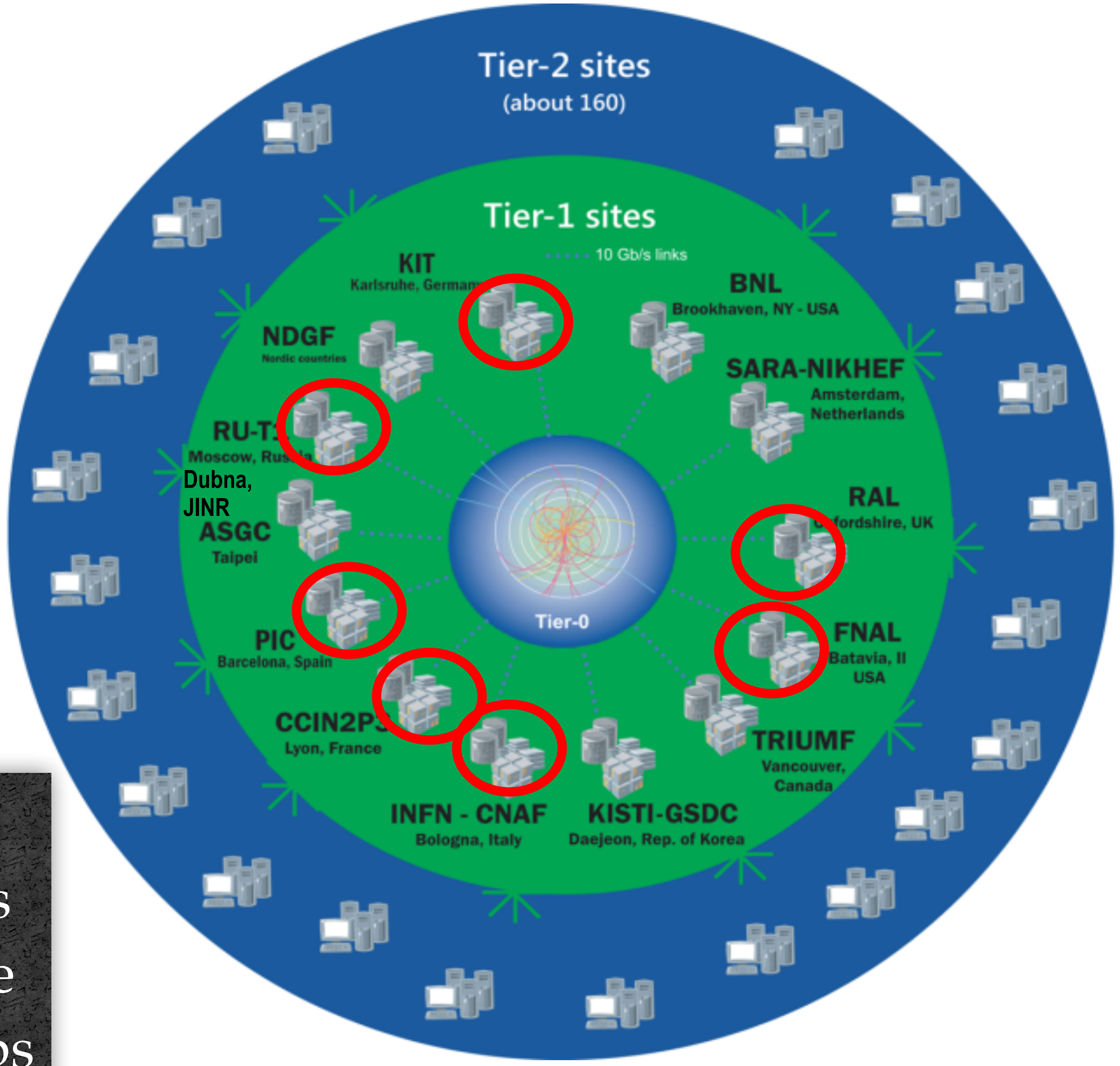
A. Petrosyan, D. Oleynik

Outline

- Computing in LHC era
- Data processing of COMPASS experiment
- SPD simulation use case
- Distributed data processing (distributed computing)
- High Throughput Computing
- Status of distributed computing services at LIT

LHC computing model

The Worldwide LHC Computing Grid (WLCG): integrates computer centres globally to provide computing and storage resources into a single infrastructure accessible by all LHC physicists for data analysis



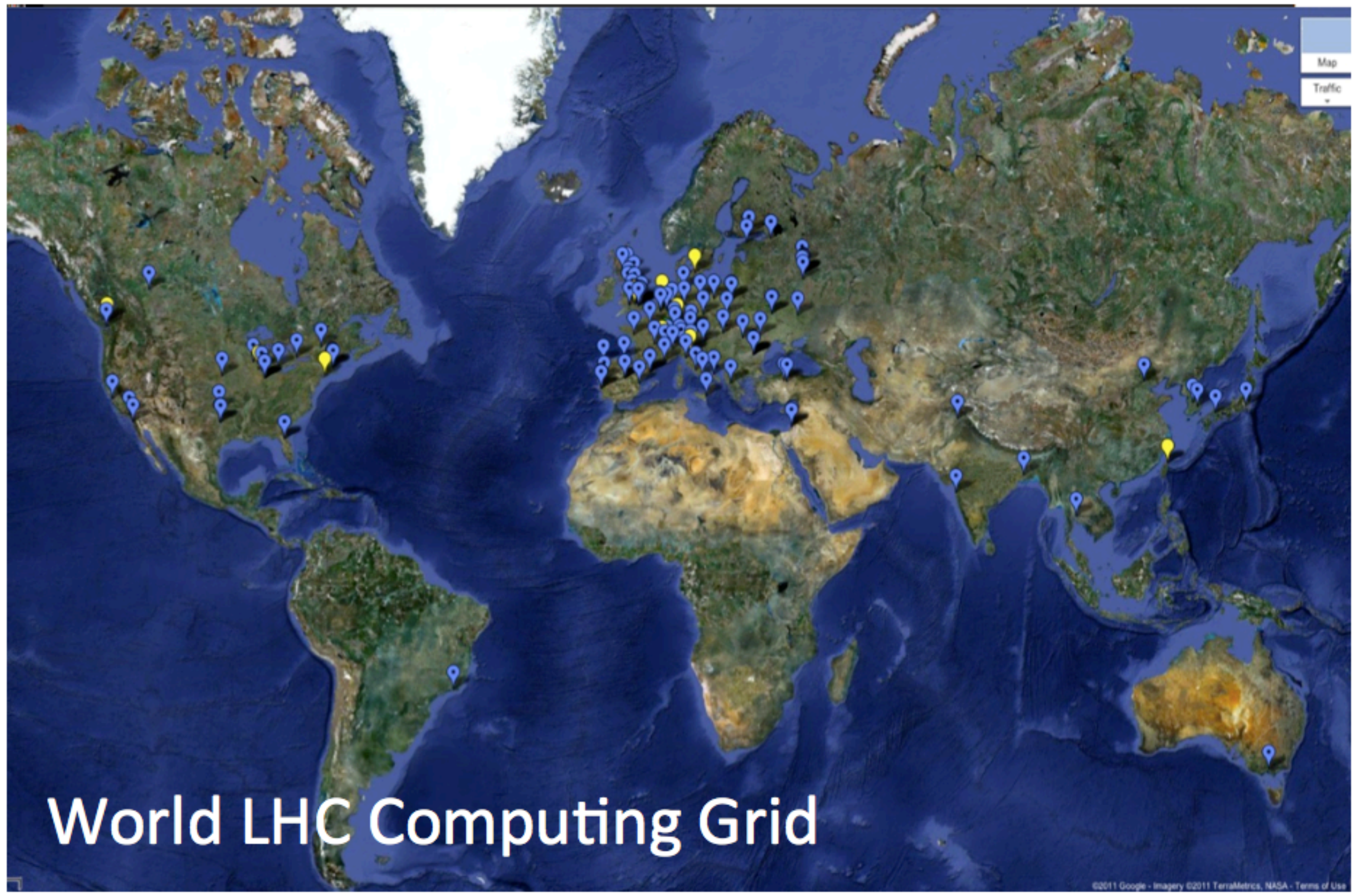
42 countries
~300,000 cores
173 PB storage
> 2 million jobs per day
>10 Gb links

Tier-0 (CERN):
- Data recording
- Initial data reconstruction
- Data distribution

Tier-1 (11→14 centres):
- Permanent storage
- Re-processing
- Analysis
- Simulation

Tier-2 (>200 centres):
- Simulation
- End-user analysis

LHC, Amazon, Google computing centres



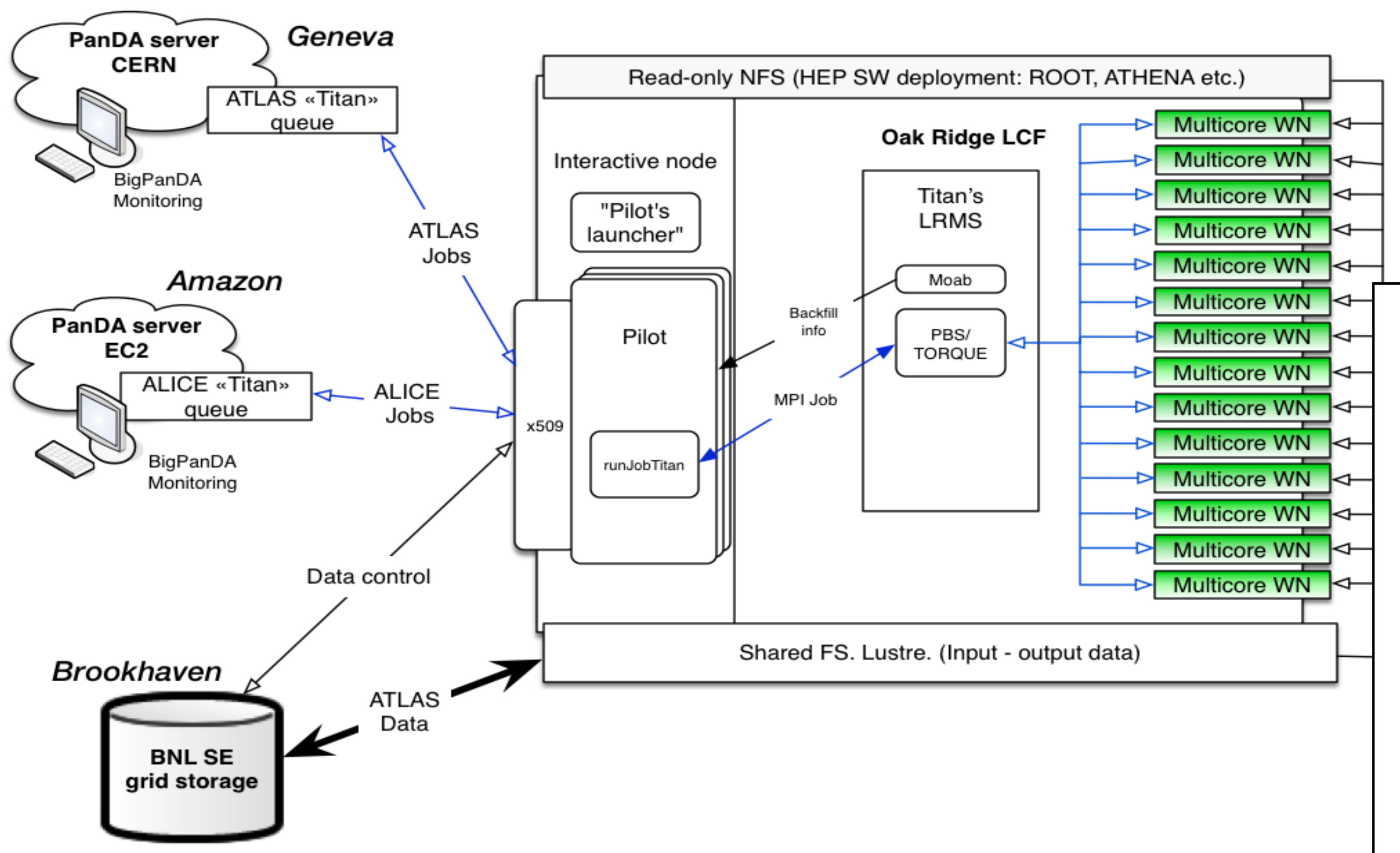
- One Google Data Center is estimated to cost ~$600M
  - An order of magnitude more than the centre at CERN
- Amazon : 9 large sites/zones
  - up to ~2M CPU cores/site, ~4M total
  - 10 x more cores on 1/10 of the sites compared to our Grid
  - 500,000 users
- LHC Computing (WLCG)
  - 200 sites, 42 countries
  - 500+k CPU cores total
  - Disk 350PB, Tape 400+PB
  - ~5000 users

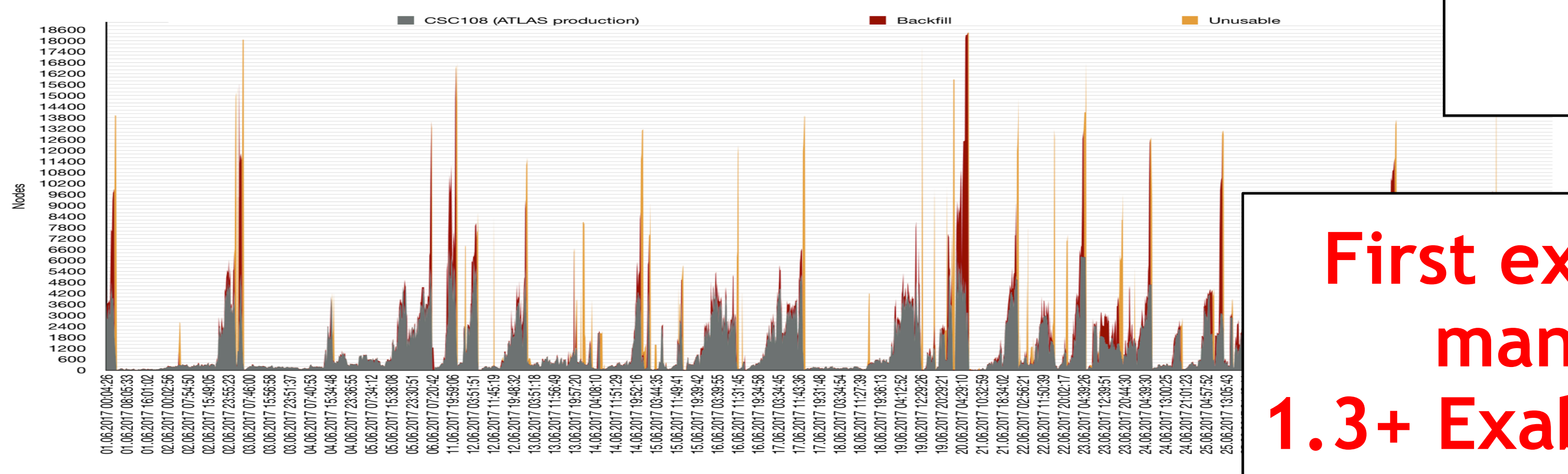
Paradigm shift in HEP computing

Old paradigms	LHC Computing
• Distributed resources are **independent entities**	• Distributed resources are seamlessly **integrated worldwide** through a single submission system • **Hide middleware** while supporting diversity
• Groups of users utilize **specific resources** (whether locally or remotely)	• All users have access to **same resources**
• Fair shares, priorities and policies are managed **locally**, for each resource	• **Global** fair share, priorities and policies allow efficient management of resources
• **Uneven user experience** at different sites, based on local support and experience	• **Automation, error handling, and other features** improve user experience • Central support coordination
• Privileged users have access to **special resources**	• All users have access to **same resources**

PanDA. Production and Distributed Analysis Workload Management System

https://twiki.cern.ch/twiki/bin/view/PanDA/PanDA





Global ATLAS operations

Up to ~800k concurrent jobs
25-30M jobs/month at >250 sites
~1400 ATLAS users

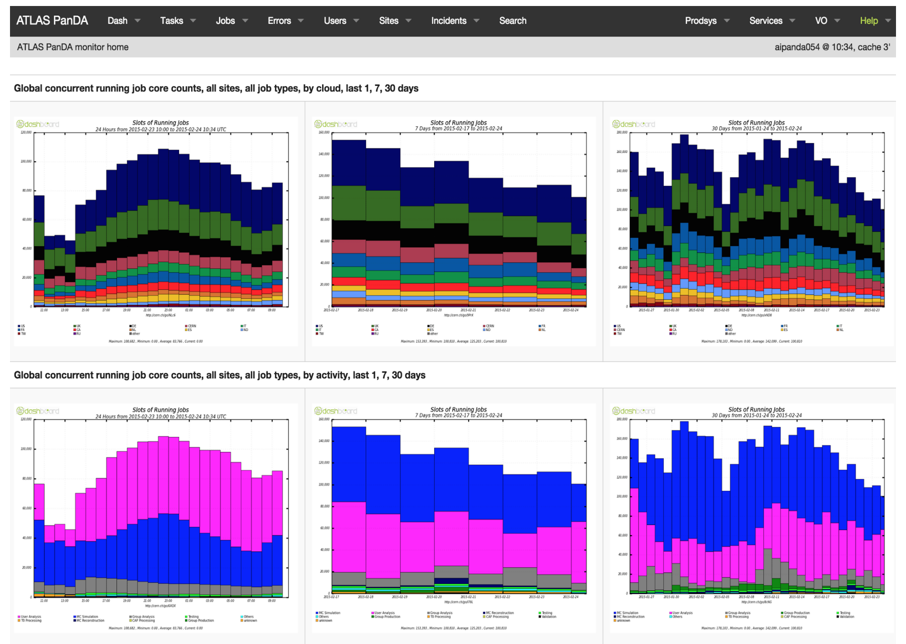
First exascale workload manager in HENP
1.3+ Exabytes processed in 2014 and in 2016-2018
Exascale scientific data processing today

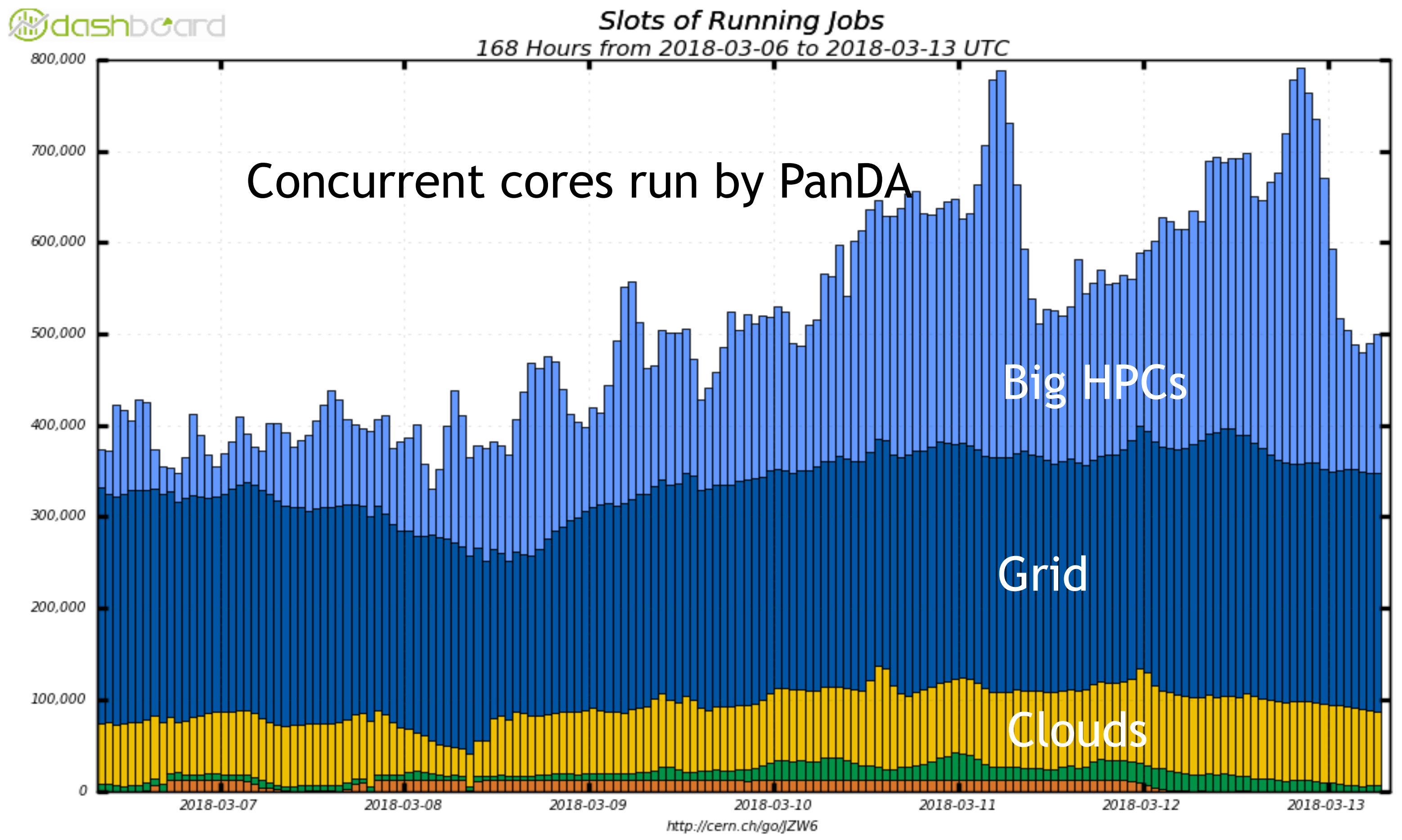
PanDA Brief Story

2005: Initiated for US ATLAS (BNL and UTA)
2006: Support for analysis
2008: Adopted ATLAS-wide
2009: First use beyond ATLAS
2011: Dynamic data caching based on usage and demand
2012: ASCR/HEP BigPanDA project
2014: Network-aware brokerage
2014 : Job Execution and Definition I/F (JEDI) adds complex task management and fine grained dynamic job management
2014: JEDI- based Event Service
2014:megaPanDA project supported by RF Ministry of Science and Education
2015: New ATLAS Production System, based on PanDA/ JEDI
2015 :Manage Heterogeneous Computing Resources
2016: DOE ASCR BigPanDA@Titan project
2016:PanDA for bioinformatics
2017:COMPASS adopted PanDA , NICA (JINR)
PanDA beyond HEP : BlueBrain, IceCube, LQCD

BigPanDA Monitor

http://bigpanda.cern.ch/

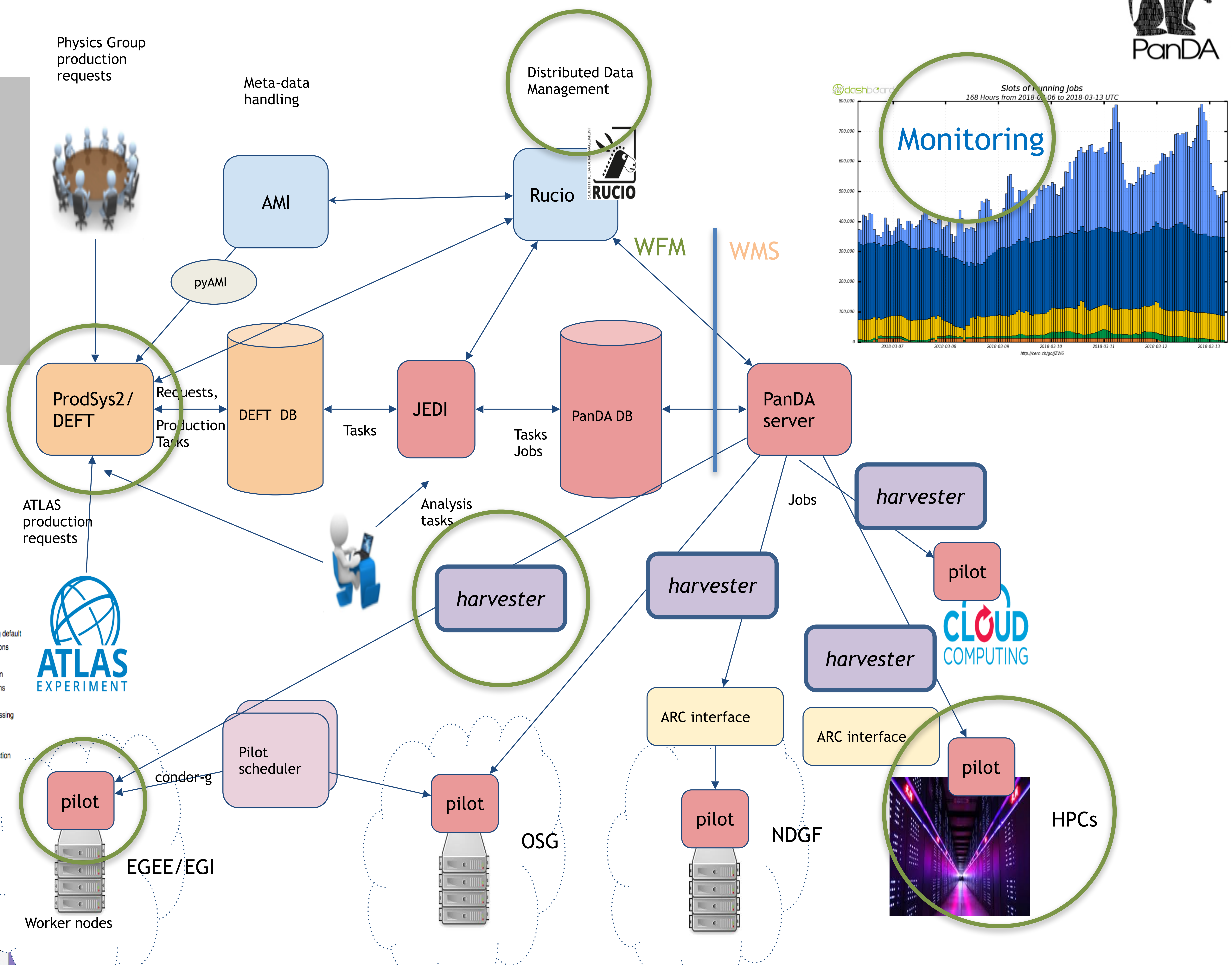




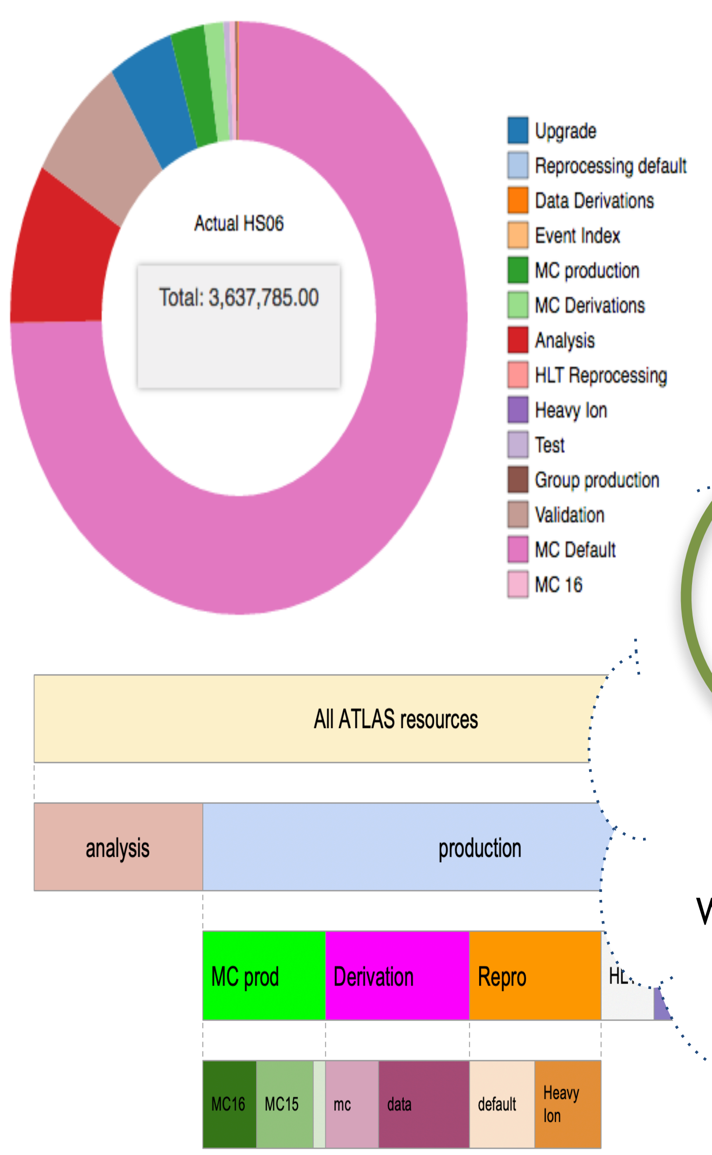
ATLAS Workflow and Workload Management

Orchestrate all ATLAS Workflows :

MC Production
Physics Groups WF
Data reprocessing
T0 spill-over
HLT processing
SW validation
User's analysis
ART



Shares/priorities

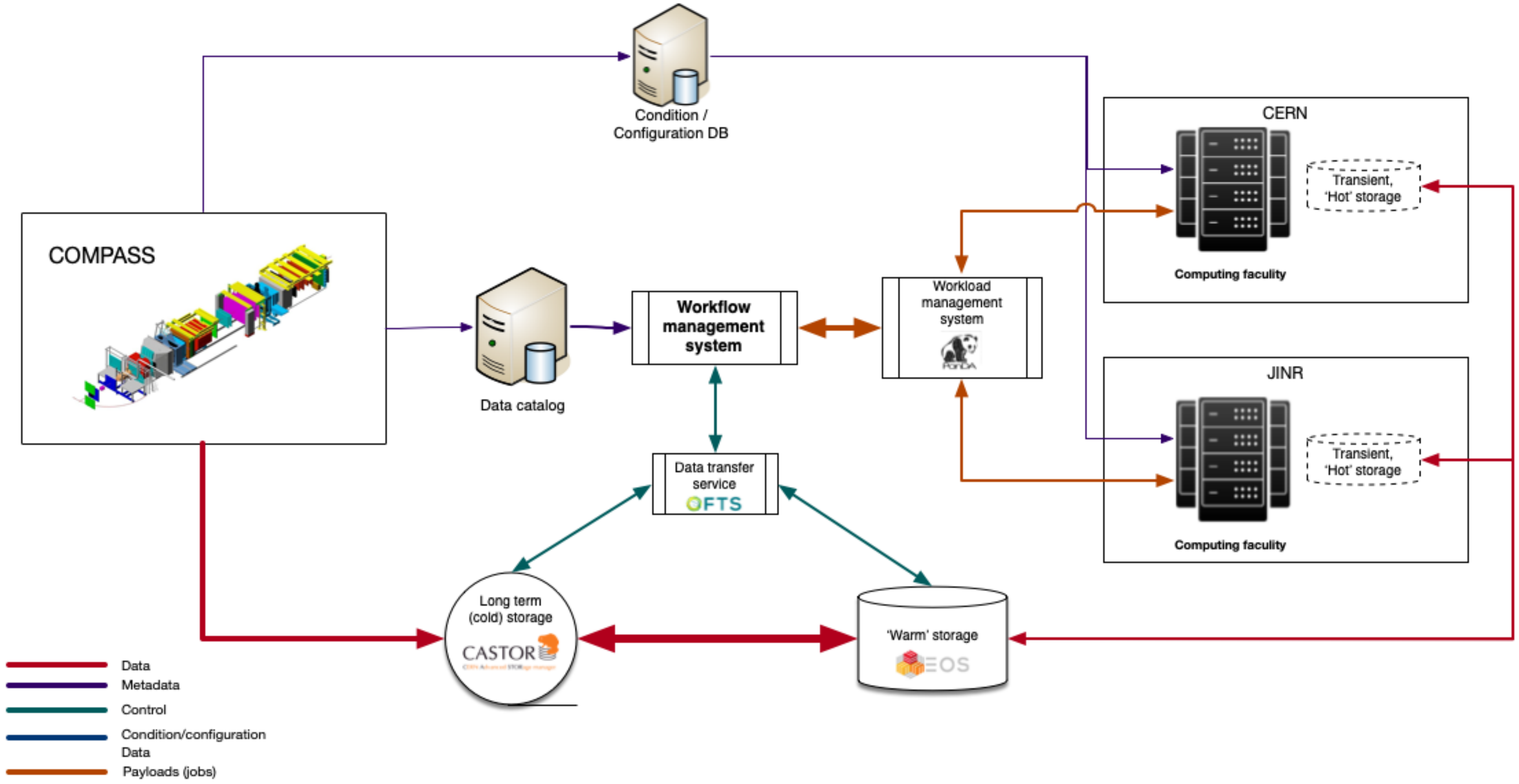


Support ATLAS rich harvest of resources. Integrate WF and data flow

From Systems to Products

- Rucio: ATLAS -> AMS, CMS, BNL, etc.
- AGIS/CRIC: ATLAS -> AMS, CMS, COMPASS, WLCG, etc.
- PanDA: ATLAS -> LSST, nEDM, SciDAC-4, CHARMM, AMS, IceCube, Blue Brain, COMPASS, etc.
- Dirac: LHCb -> Belle II, BES III, IHEP, ILC, Ibergrid, etc.

COMPASS workflow

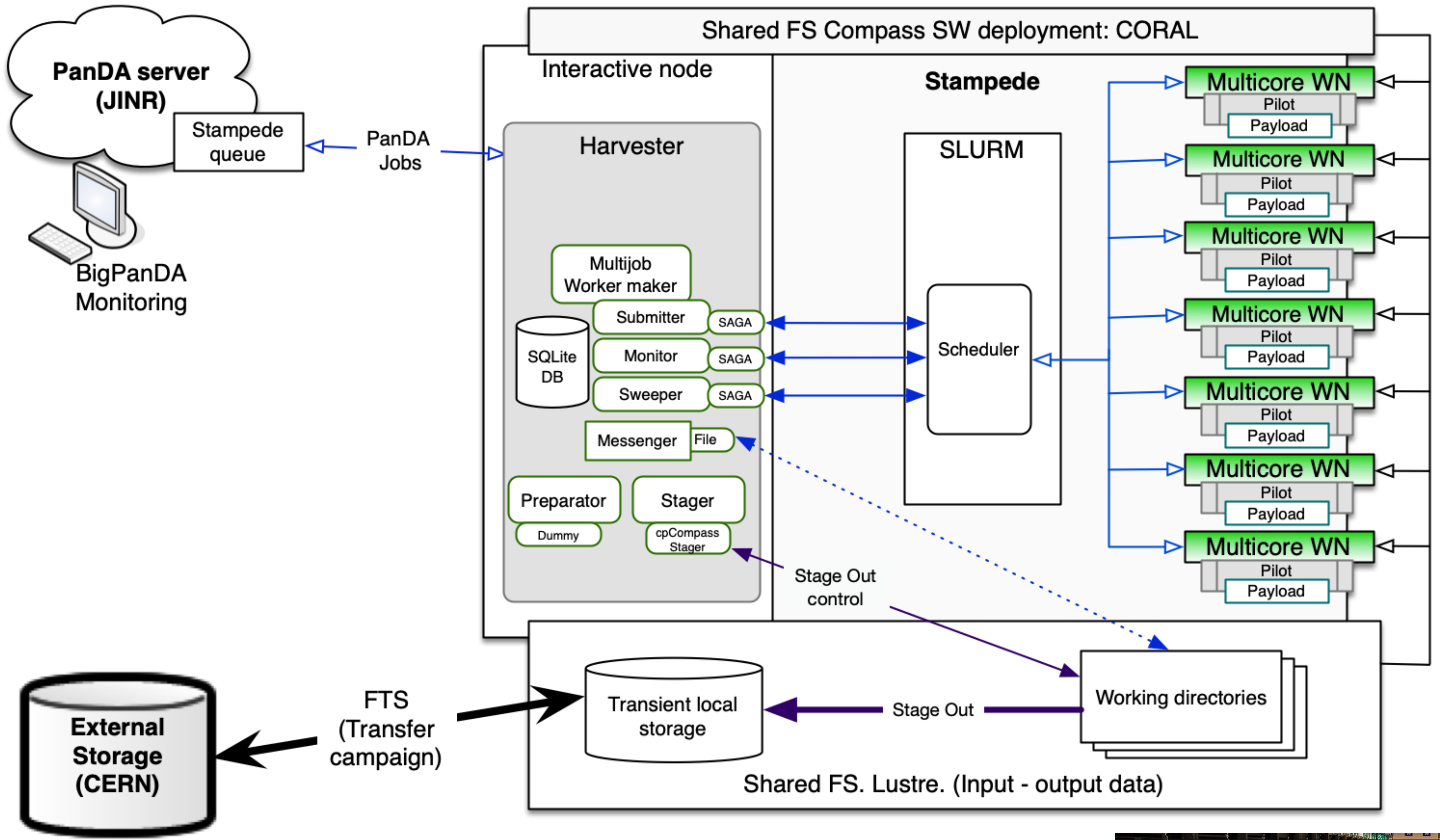


LIT is in charge of development and support of the production management system for COMPASS experiment since 2017.

Supported task types: Monte-Carlo generation and reconstruction, events filtering, real data reconstruction.

ProdSys manages data processing. PanDA provides a central jobs queue. Jobs delivery to the sites is performed by Harvester service using the local HTCondor instance. Final data are being transferred from EOS to Castor via CERN File Transfer Service (FTS).

Processing on TACC’s Stampede2 and Frontera



The latest achievement: integration was done in 2019. We are starting the productio

Edge jobs management service: Harvester.

Data delivery is done manually via FTS, period per campaign.

Stats and performance September, 2019

Since August 2017 the ProdSys manages data processing of the experiment.

~20 standard working queues at CERN and JINR.

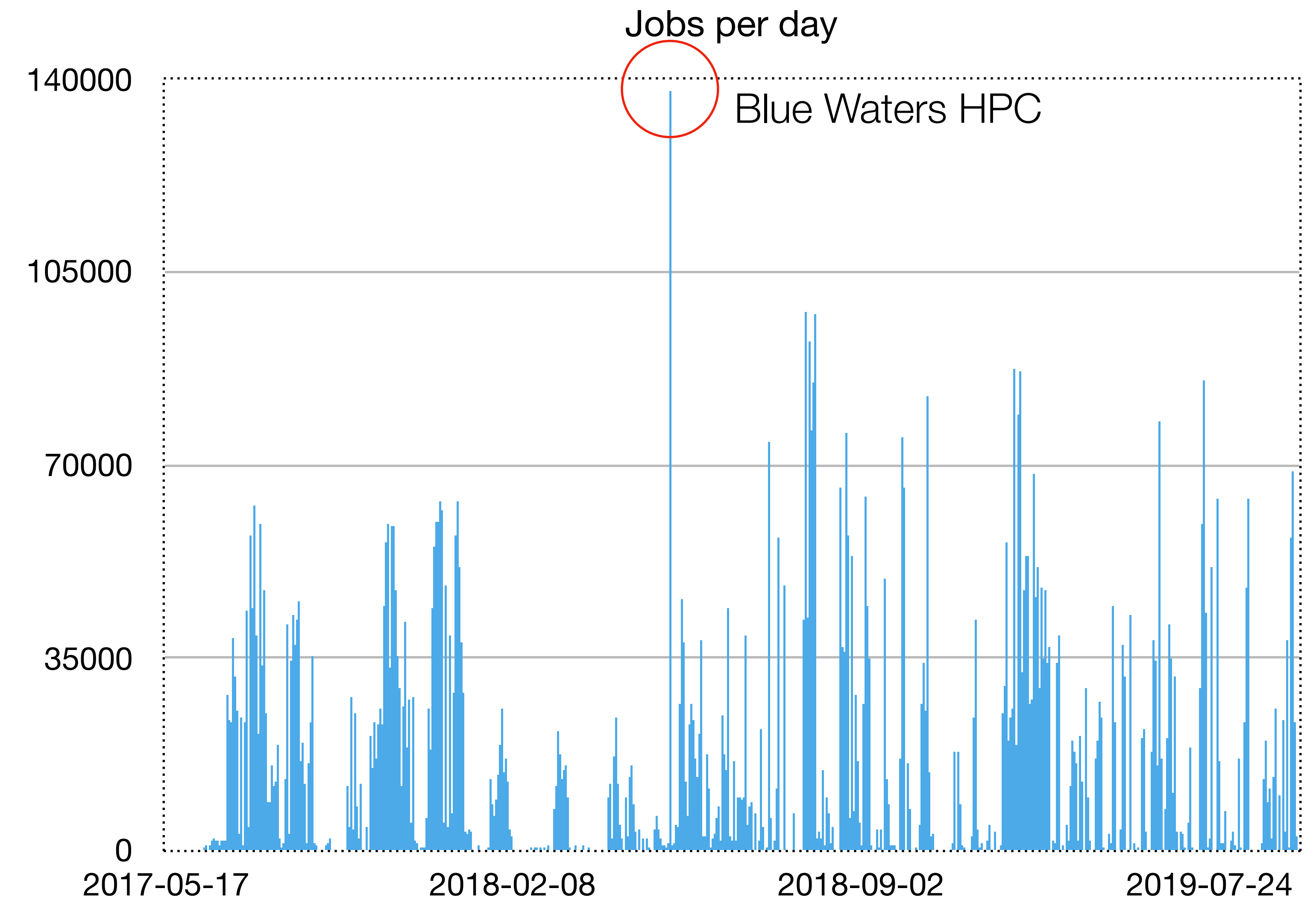
~400 tasks, 5 000 000 raw data files, 100 000 000 events, 9 000 000 jobs were processed.

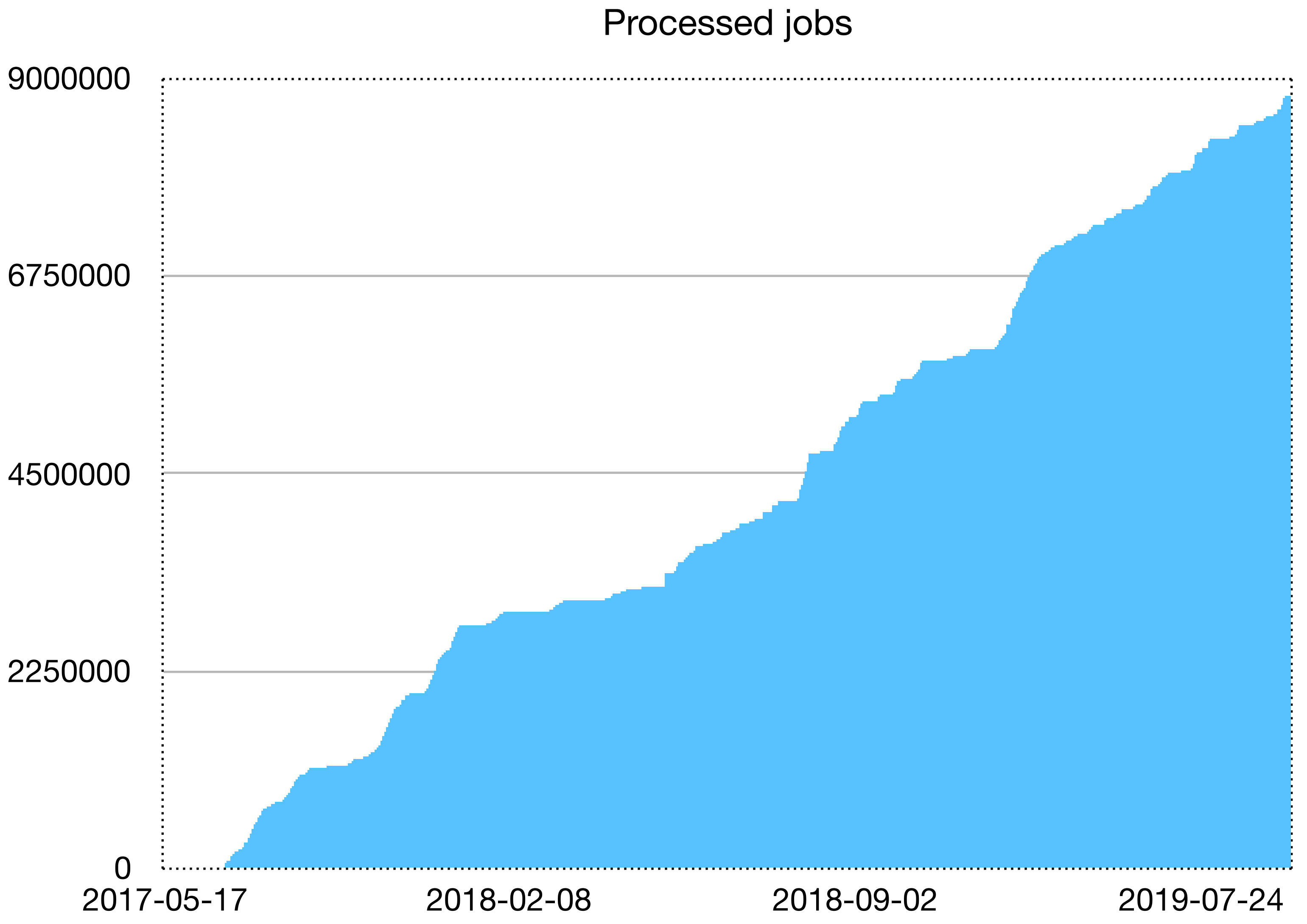
~700TB of final data were written to Castor.

~7.5 hour is an average wall time of the job in the system.

~7500 CPU years were consumed.

PRODUCTION	SITE	TYPE	PRODSLT	PHASTVER	STATUS	JOBS
dy2018P02t3_panda	FRONTERA_COMPASS_MCORE	test production	0	8	jobs ready	all: 108422,
dy2018P02t3_panda	FRONTERA_COMPASS_MCORE	test production	0	8	running	all: 102520,
dy2018P02t3_panda	FRONTERA_COMPASS_MCORE	test production	0	8	running	all: 210942,





Infrastructure overview

2 PanDA instances (for Grid and for HPC), MySQL server, Monitoring, AutoPilotFactory, 2 Harvester instances (for Grid and for HPC), Production management system are deployed in Dubna at JINR cloud service.

Distributed HTCondor CE at CERN and Budapest.

PBS CE at JINR.

Blue Waters HPC at University of Illinois at Urbana Champaign in 2017-2019.

Stampede2 (#19 at Top500) and Frontera (#5 in Top500) HPCs at Texas Advanced Computing Center.

EOS SE at CERN.

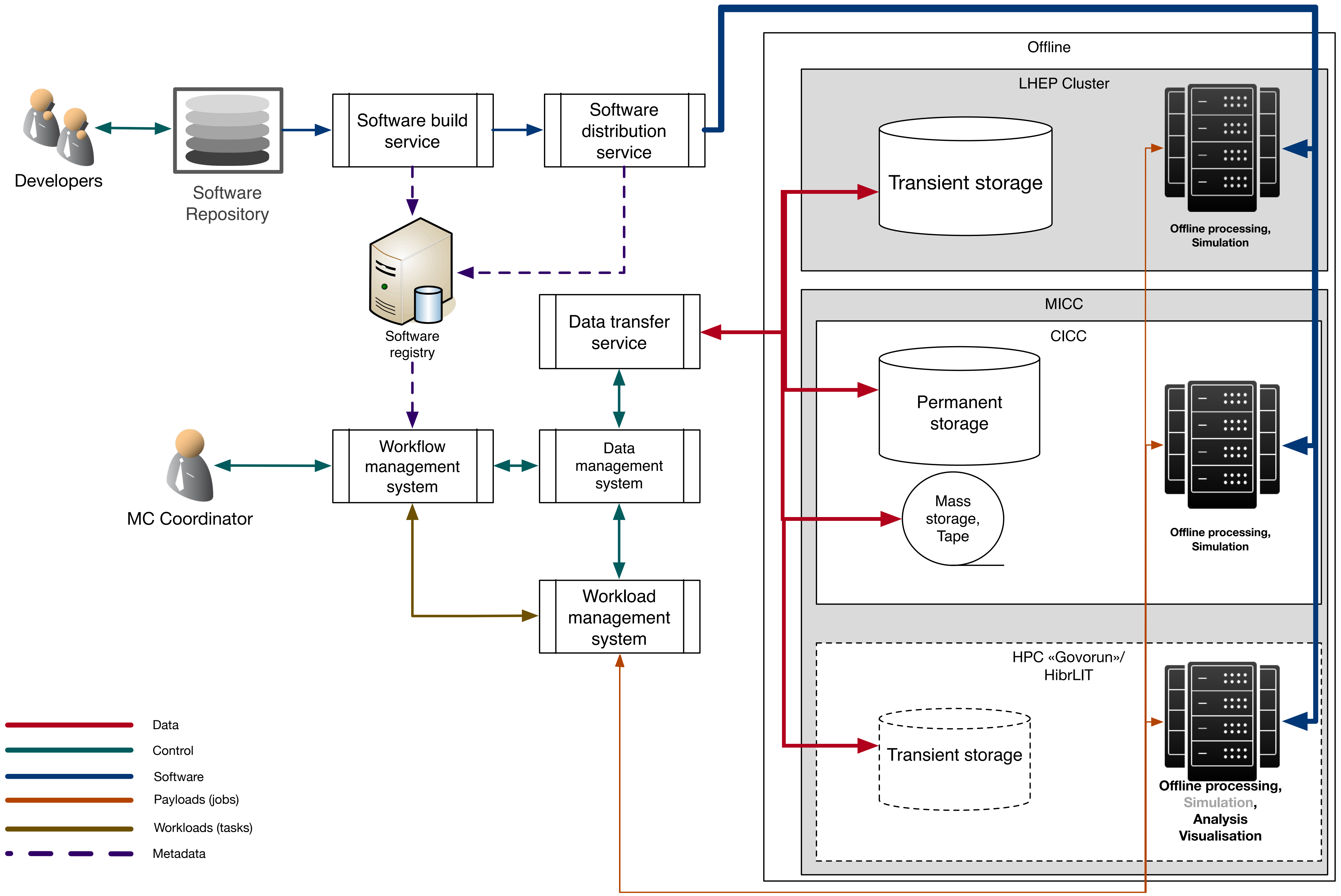
Castor SE at CERN.

VOMS at CERN.

PerfSonar instance at JINR cloud network segment to monitor network between JINR and CERN.

Use case: SPD simulation

- Simulation - another huge consumer of computing resources.
- Can be (should be) started before facility will be ready to collect data
- Accompanied by intensive software development
- Key components:
  - WFMS
  - WMS
  - DMS & DTS
  - Software build service - required for automation of building of new releases of SW
  - Software distribution service - service which allow automatic deployment of new versions of SW in heterogeneous environment



Data processing in a distributed computing infrastructure

- Distributed computing infrastructure is a computing system whose components are located on different remote computers and components interact with one another to achieve a common goal.
  - One well-known example: WLCG (Worldwide LHC Computing Grid)
  - JINR already have a set of facilities which can (should) be integrated into the distributed computing infrastructure
- Advantages of using distributed computing systems:
  - **High fault tolerance**: failure of a single computing facility is not a blocker for data processing chain
  - **Flexibility**: wide range of computing resources can be in integrated into a common infrastructure
  - **Balanced support expenses**: no needs to upgrade all computing facilities at the same time (etc.) Support expenses mostly on the facilities provider side

HTC - High Throughput Computing

High-throughput computing *(**HTC**) a computing paradigm that focuses on the efficient execution of a large number of loosely-coupled tasks.*

European Grid Infrastructure (EGI)

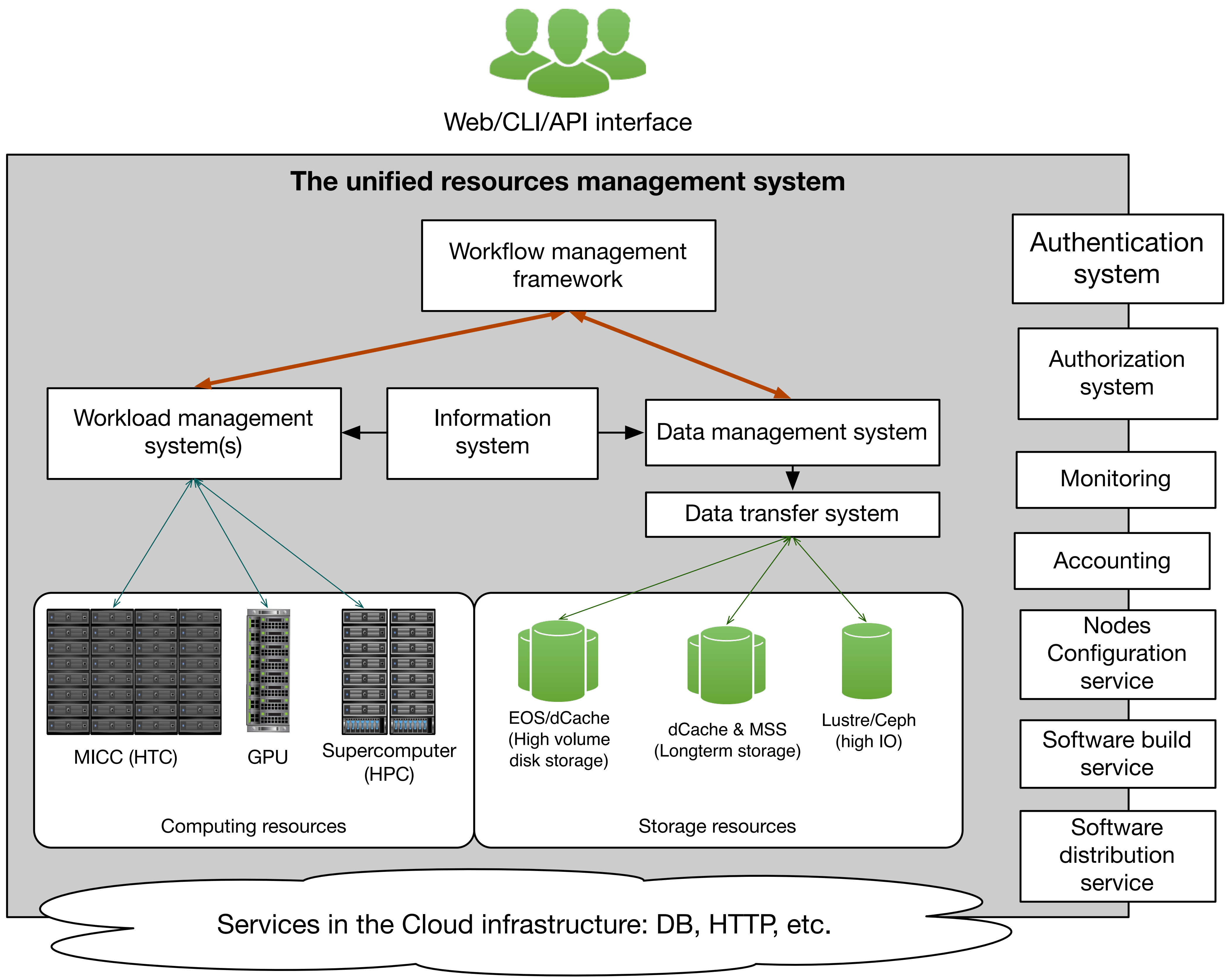
- Why we should agree on HTC paradigm in case of massive data processing in distributed computing systems?
  - Heterogeneity of computing facilities joined in common infrastructure (architectures, performance, capability)
  - Connection through WAN between facilities
  - High volume of data, which should be processed - data processing speed is more important than maximum computing performance
  - High flexibility

Basic requirements for using HTC paradigm

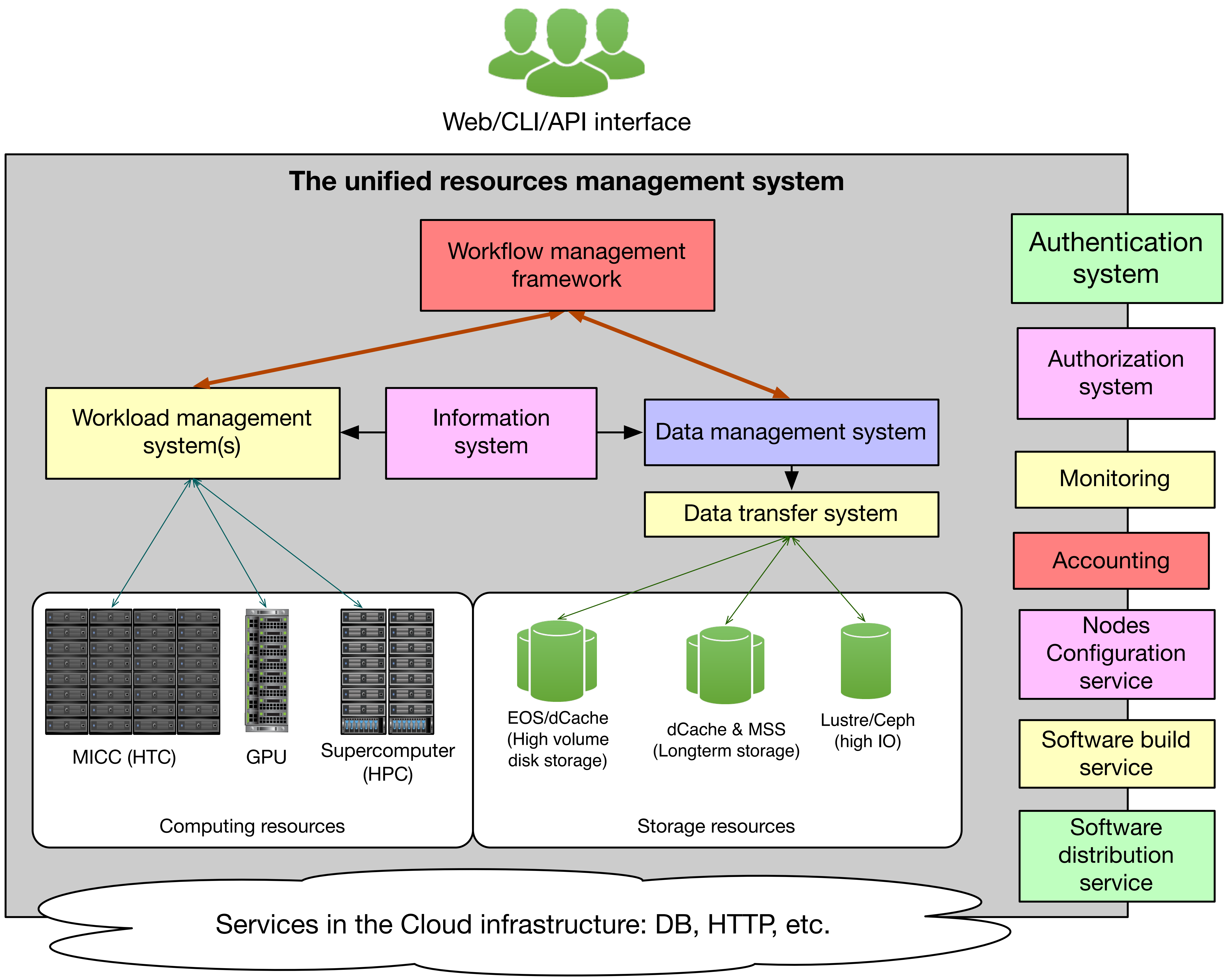
- A reasonable size of processed data chunks, not too small to avoid an extremely high number of tasks (jobs), not too big to avoid the long processing time
- Common authentication and authorization services across infrastructure
- A service which controls task (jobs) execution: workload management system
- A service which takes care of proper data catalog and data distribution
- Low-depends between elementary computing tasks (jobs)

Unified Resource Management System

- The Unified Resource Management System is a IT ecosystem composed from the set of subsystem and services which should:
  - Unify of access to the data and compute resources in a heterogeneous distributed environment
  - Automate most of the operations related to massive data processing
  - Avoid duplication of basic functionality, through sharing of systems across different users (if it possible)
  - As a result - reduce operational cost, increase the efficiency of usage of resources,
  - Transparent accounting of usage of resources

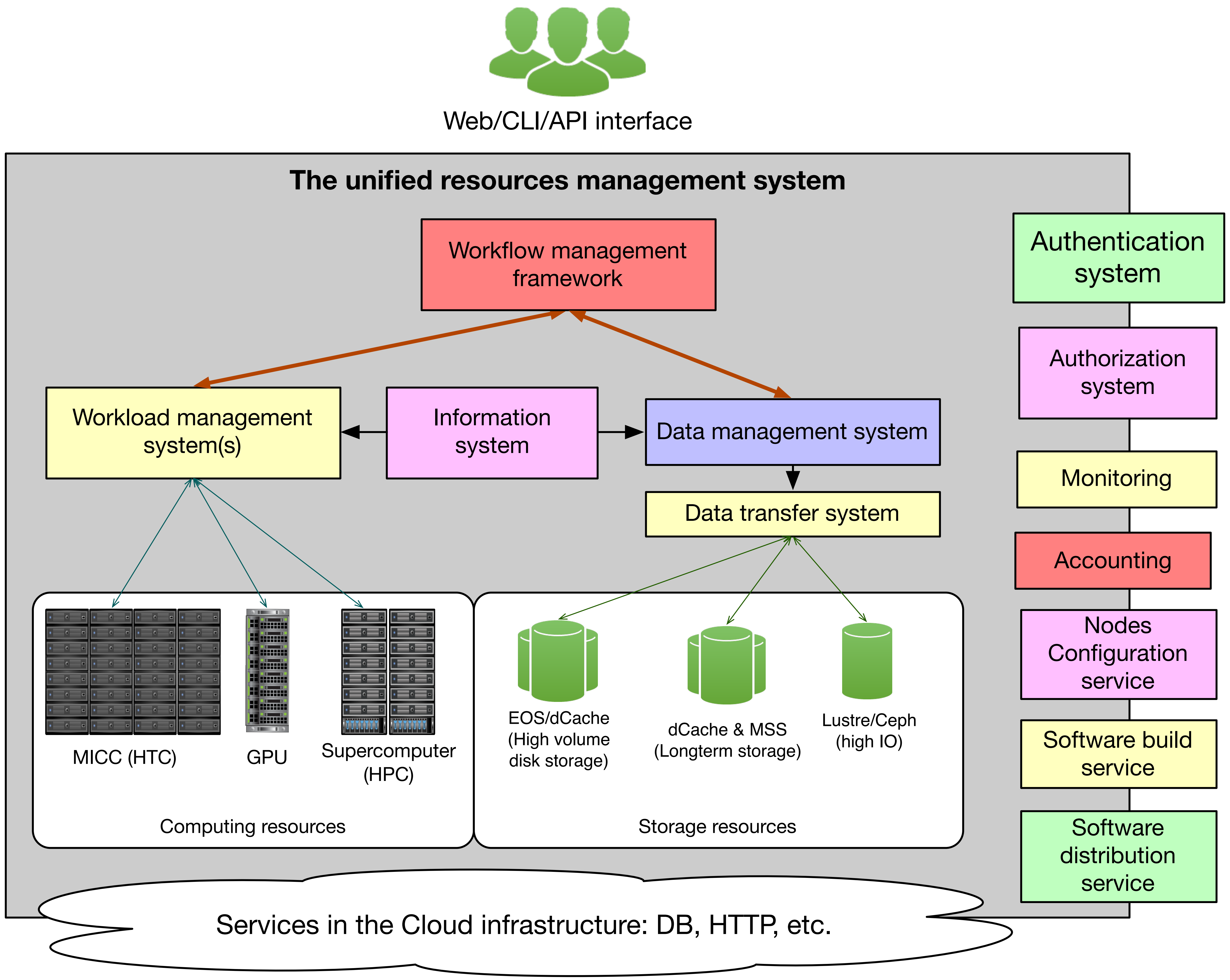


URMS: first steps



- Some core subsystem already exist in JINR
  - Authentication system (Kerberos based, with SSO supporting for Web applications)
  - CVMFS as Software distribution service
- In progress:
  - deployment of FTS as the core of Data transfer system
  - We already have some infrastructure monitoring
  - A lot of research in WFMS and WMS fields, we may declare a list of requirements:
    - We should avoid limitations by scale as much as possible.
    - Advanced monitoring system
    - WMS with MultiVO support
    - Priority and share management
    - Task-based job management
  - Looks like that Rucio will be natural choice as cross experiment Data Management System
  - Software build service prototype already exists in the Cloud infrastructure

URMS: next steps



- Common Authorization System which will be used to manage user access to resources. The closest candidate is VOMS - but, we need to be coherent with Authentication System
- Accounting is required to understand system behaviour and analysing of bottlenecks.
- Nodes configuration - should be automated as much as possible
- Information system store and provide a description of computing and storage resources, including availability (shutdowns) of resources.

Thank you for attention!