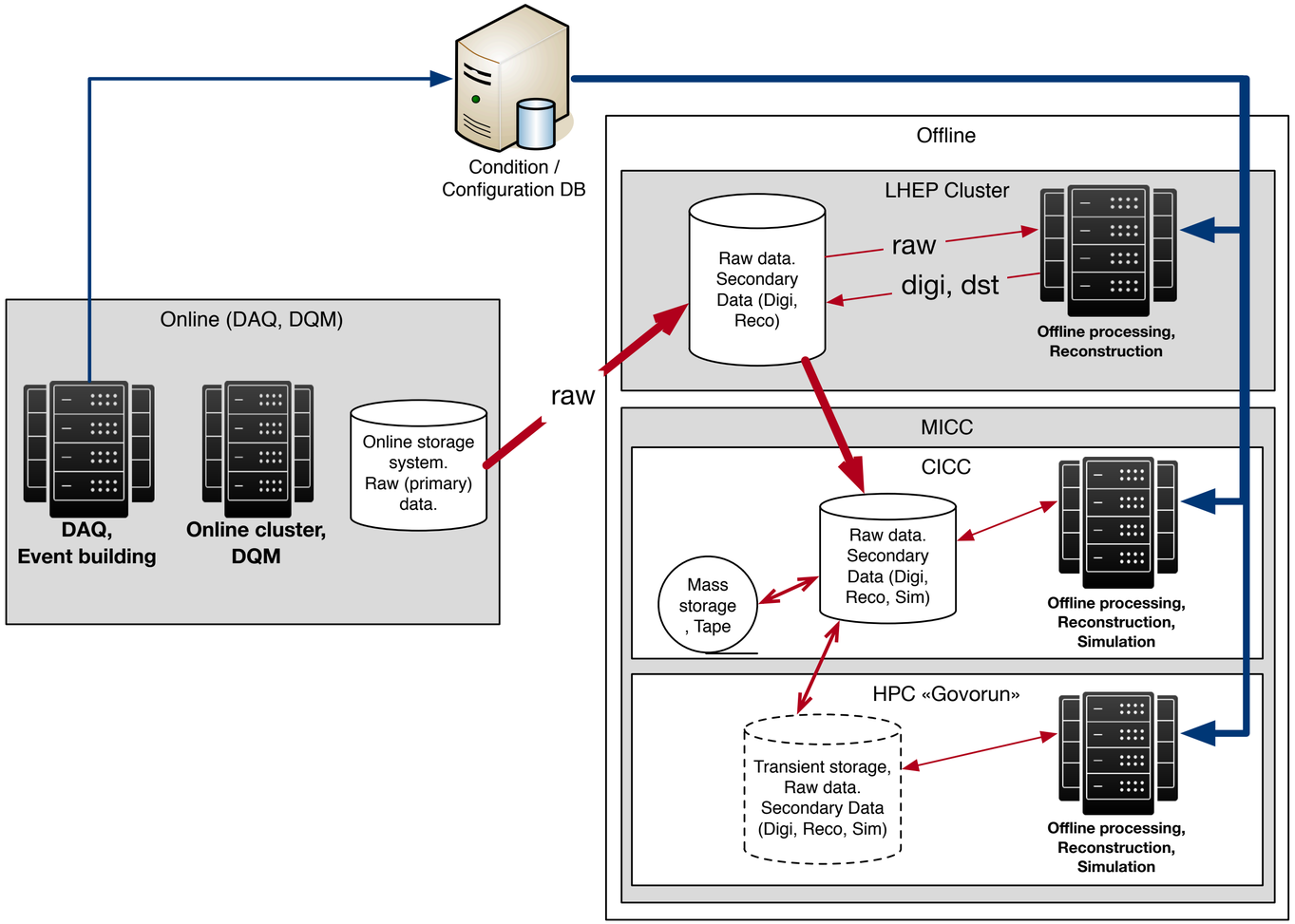
Status of Workflow Implementation for BM@N Distributed Processing

A. Petrosyan[1], D. Oleynik[1], K. Gertsenberger[2], A. Yachmenyov[3], D. Gavrilov[3] ,I. Matveev[3]
[1]LIT JINR, [2]LHEP JINR, [3]Dubna University
6th Collaboration Meeting of the BM@N Experiment at the NICA Facility
VBLHEP, JINR, Dubna, September 26, 2020

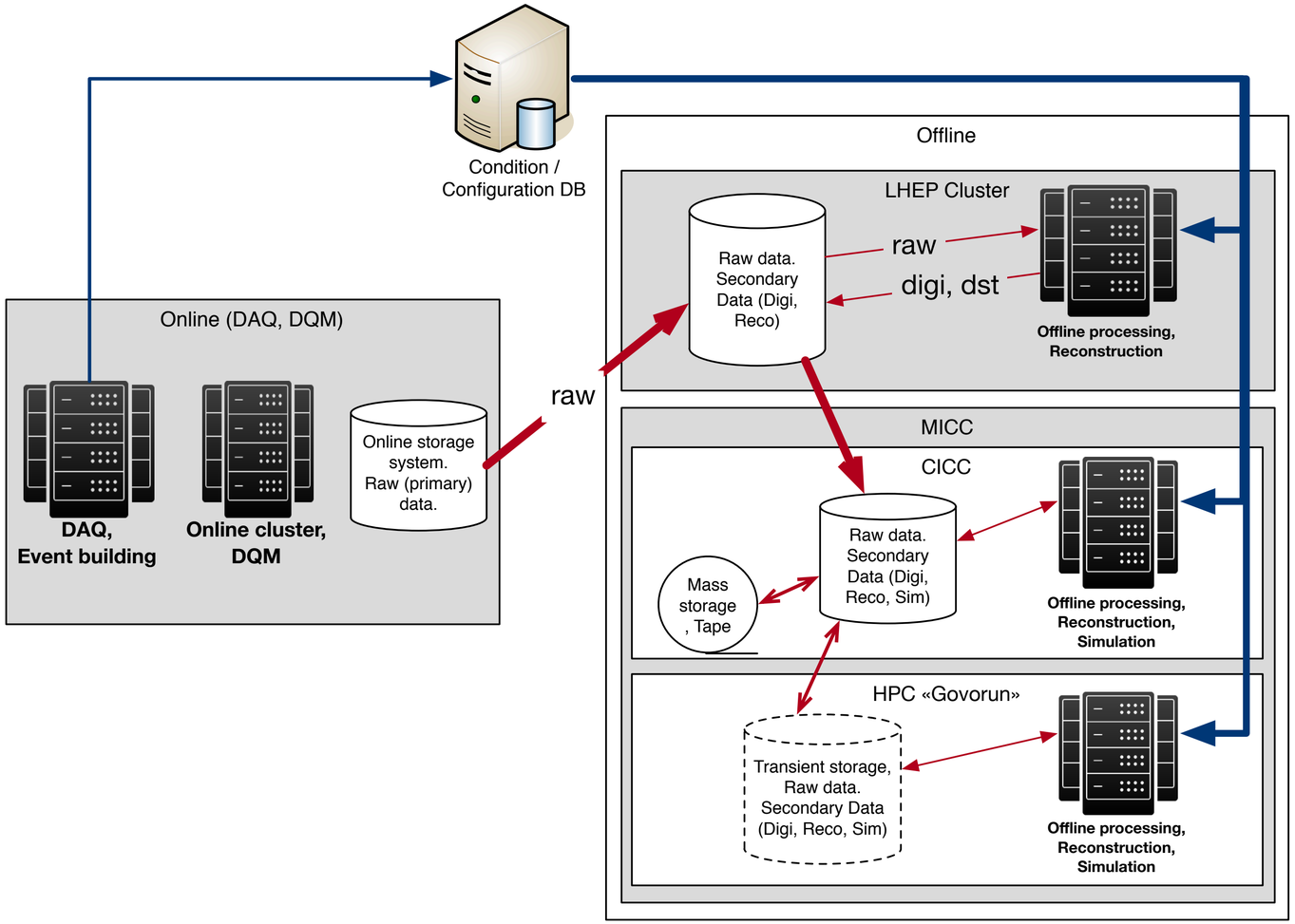
BM@N events reconstruction data flow

- Raw data is produced by DAQ of the detector and stored on the online storage system
  - Initial processing of data (DQM) started on “on-line” resources (dedicated cluster)
- Relevant raw data should migrate to permanent storage and to storages which close to computing facilities
- Data should be processed and results stored for future analysis



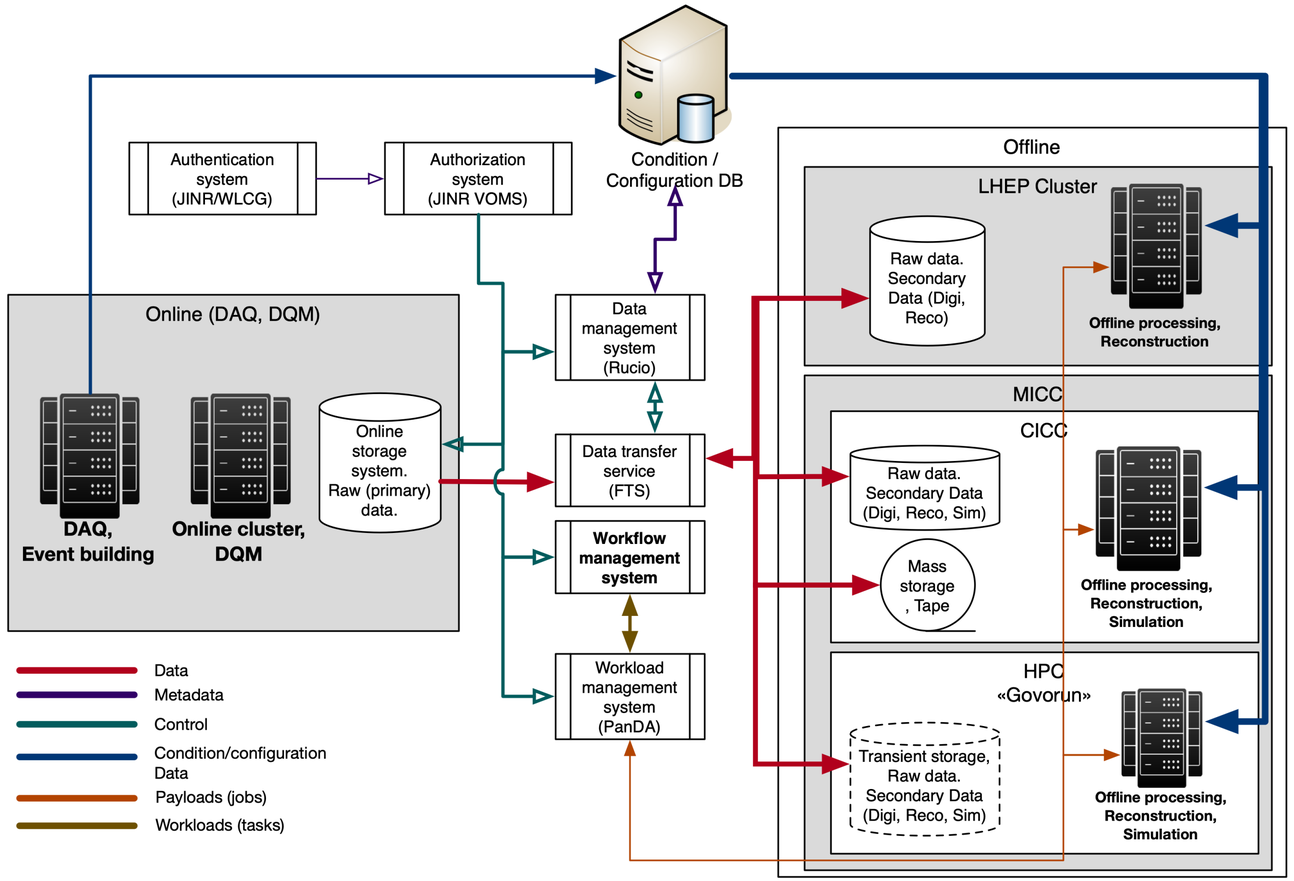
Crucial features of the data flow

- Distributed heterogeneous storage systems: EOS, dCache, tapes
- Distributed heterogeneous computing resources: LHEP cluster, CICC resources, Govorun
- Intensive data processing and transfers



Middleware services

- Middleware services link computing and storage elements to each other, hide the variety and complexity of the IT-infrastructure and provide automation.
- Components:
  - **Workflow management system** — controls the process of data processing on each step of processing. Produces chains of tasks required for processing of certain amount of data, manages execution of tasks.
  - **Workload management system** — processes tasks execution by splitting of the task to the small jobs, where each job process a small amount of data. Manages the distribution of jobs across the set of computing resources. Takes care about generation of a proper number of jobs till task will not be completed (or failed).
  - **Data management system** — responsible for distribution of all data across computing facilities, data management (storage, replication, deletion, etc.)
  - **Data transfer service** — takes care about major data transfers. Allow asynchronous bulk data transfers.
  - **Information system** — stores information about all services, storage systems and computing resources, protocols, etc.



FTS
File Transfer Service

- Distributes majority of LHC data across WLCG infrastructure
- 7 WLCG and 14 non-WLCG instances
- ~28 Virtual Organisations
  - ATLAS, CMS, LHCb, AMS, NA62, Compass, ILC, Magic, Belle ll, Mice, Xenon, Snoplus, GridPP, DUNE, LZ, Solidexperiment.org, SKA, Ligo, Icecube, Elixir, NP02 (part of DUNE), CAST, ESCAPE, Eiscat.se, Virgo, BES III, JUNO, Pierre Auger Observatory, CEPC
- Integrated with experiment frameworks: Rucio, PhEDEx, DIRAC
- Transferred in 2019 more than 800 million files and 0.95 Exabyte of data

Community


Advanced European Network of E-infrastructures
for Astronomy with the SKA


LCLS


Belle II


Science & Technology
Facilities Council


COMPASS


CERN

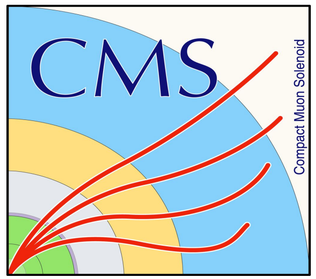

ARC


ATLAS
EXPERIMENT


ESCAPE
European Science Cluster of Astronomy &
Particle physics ESFRI research Infrastructures


Alpha Magnetic Spectrometer
AMS-02
Europe • Asia
North America


Xe
XENON
Dark Matter Project


CMS
Compact Muon Solenoid


DUNE
DEEP UNDERGROUND
NEUTRINO EXPERIMENT


egi


3D
EISCAT


cta
cherenkov telescope array


LSST
Large Synoptic Survey Telescope


IceCube
SOUTH POLE NEUTRINO OBSERVATORY


FTS


VIRGO


Fermilab


PAWSEY
supercomputing centre


SKA
SQUARE KILOMETRE ARRAY


LIGO


Open Science Grid

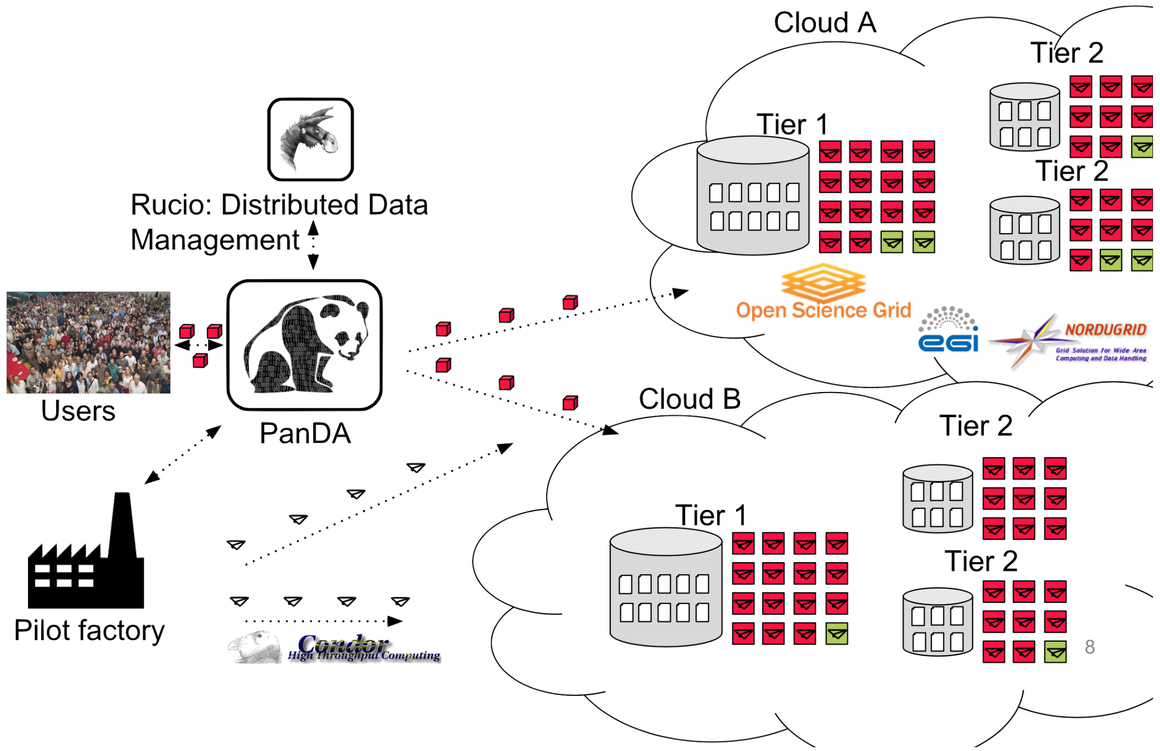

eXtreme DataCloud


BROOKHAVEN
NATIONAL LABORATORY


EUXDAT

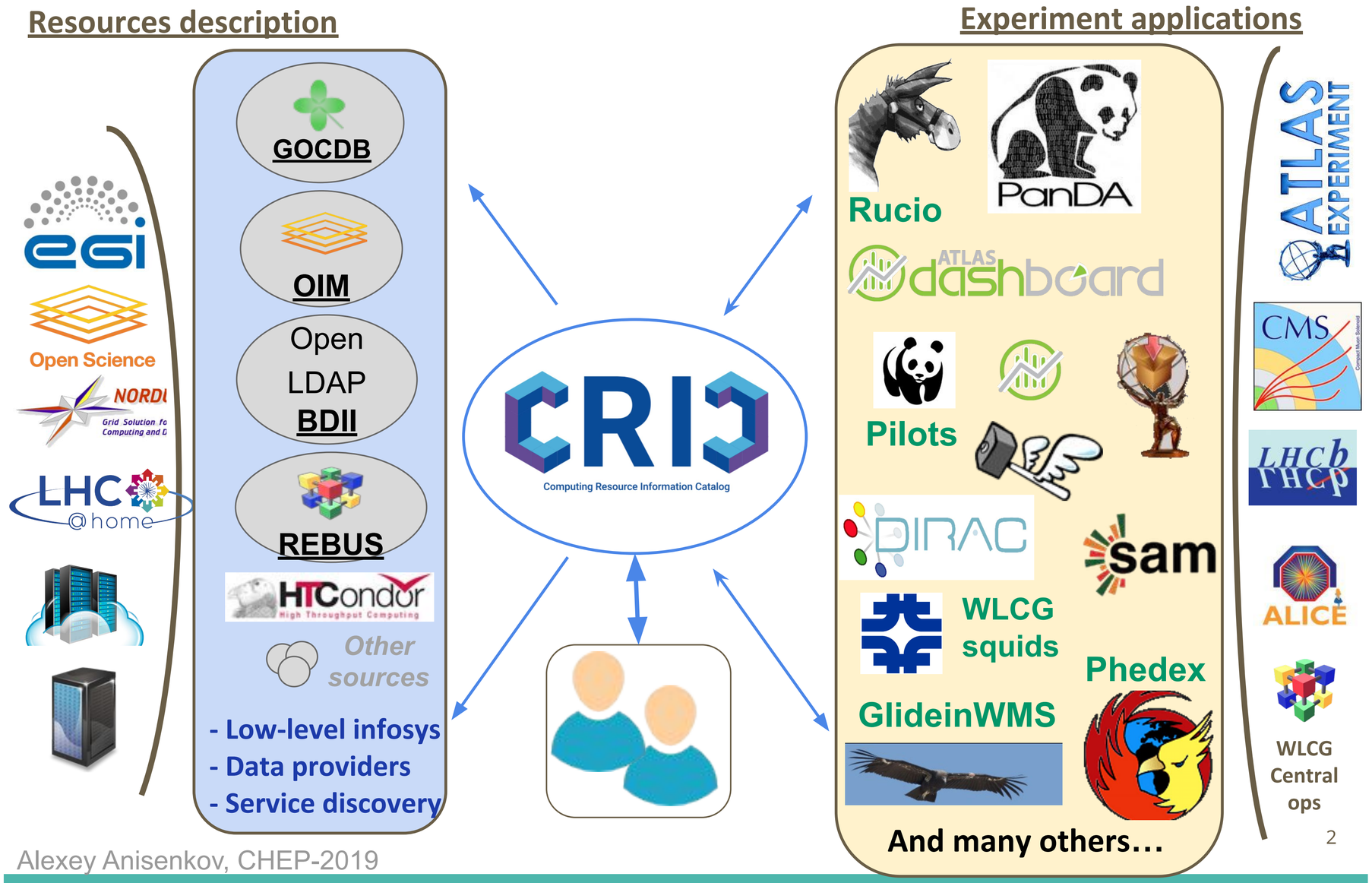
Workload management system

- Initially developed for ATLAS experiment at CERN, PanDA now widely used by many other experiments, in HEP and beyond
- PanDA WMS allows to organise highly scalable data processing on almost any type of resources: grid sites, academic or industry clouds, high performance computers



CRIC: a high-level information middleware

to configure computing environment and describes resources as VOs need



Alexey Anisenkov, CHEP-2019

2

Workflow management

- Key component of the infrastructure — workflow management system
- All other systems and services, such as FTS, Rucio, PanDA, CRIC — can be used literally out of the box, while workflow management system has to be developed basing on the needs of the particular experiment
- Each pipeline — Monte Carlo simulation, reconstruction, events filtering — must be declared in the workflow management system
- Based on Apache Airflow framework

Scalable

Airflow has a modular architecture and uses a message queue to orchestrate an arbitrary number of workers. Airflow is ready to scale to infinity.

Dynamic

Airflow pipelines are defined in Python, allowing for dynamic pipeline generation. This allows for writing code that instantiates pipelines dynamically.

Extensible

Easily define your own operators and extend libraries to fit the level of abstraction that suits your environment.

Elegant

Airflow pipelines are lean and explicit. Parametrization is built into its core using the powerful Jinja templating engine.

Status

- Information system — deployed, integration with JINR SSO — in progress, is being filled by data
- File transfer service — deployed, testing
- Distributed data management service — deployed, testing
- Workload management service — deployed, tested, ready to receive tasks
- Workflow management service — deployed, integration with JINR SSO — done, integration with the workload management service — done, pipelines are being designed and implemented
- VOMS service — deployed, production
- Overall status: all services are ready to receive tasks
- Ongoing work: workflow pipelines implementation and conducting a task between the components of the infrastructure