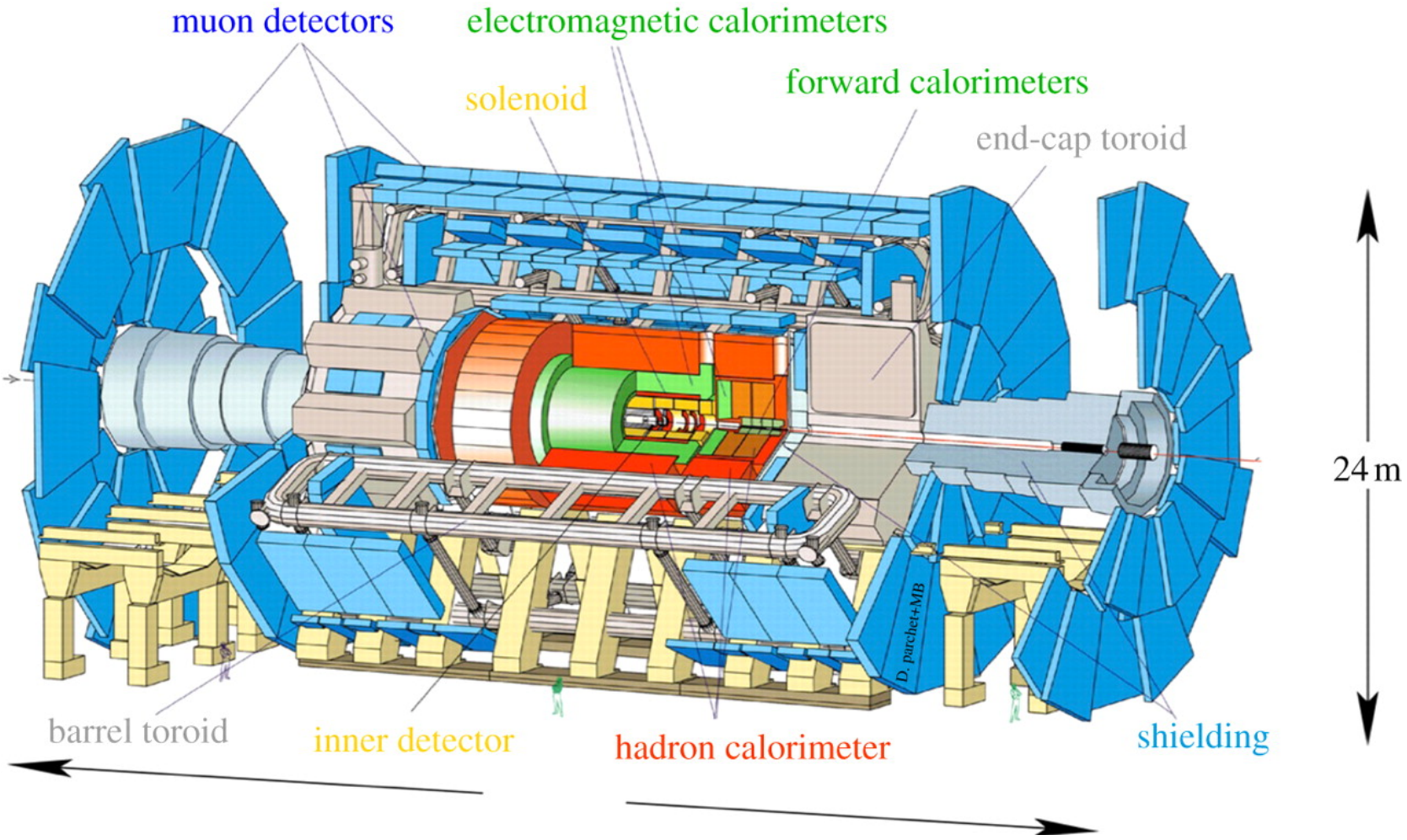
ATLAS production system

F Barreiro, M. Borodin, K. De, D. Golubkov, A. Klimentov, T. Maeno, R. Mashinistov, S.Padolski, T. Wenaus

GRID'2016

The ATLAS detector



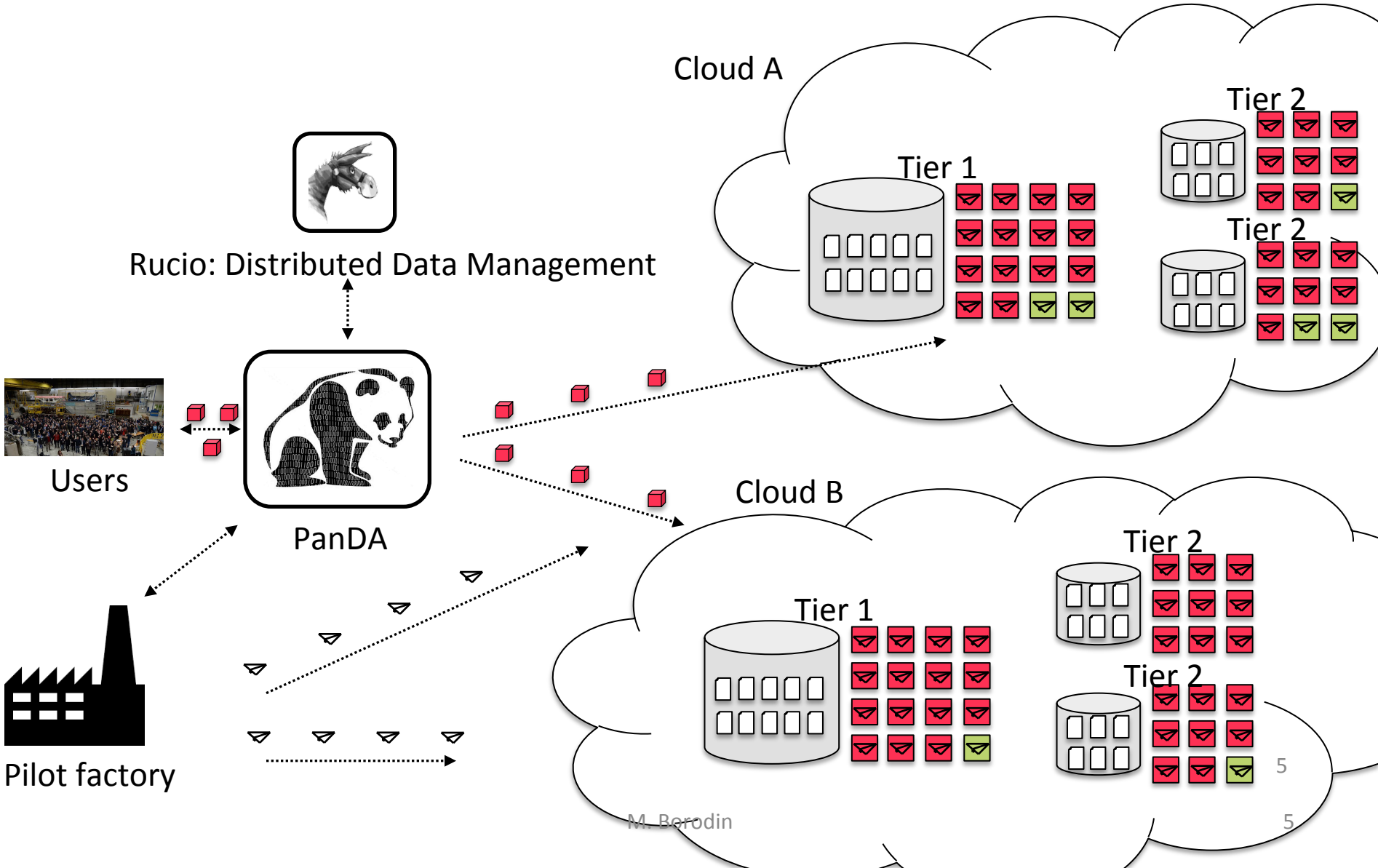

~1/10th of its members

Production system

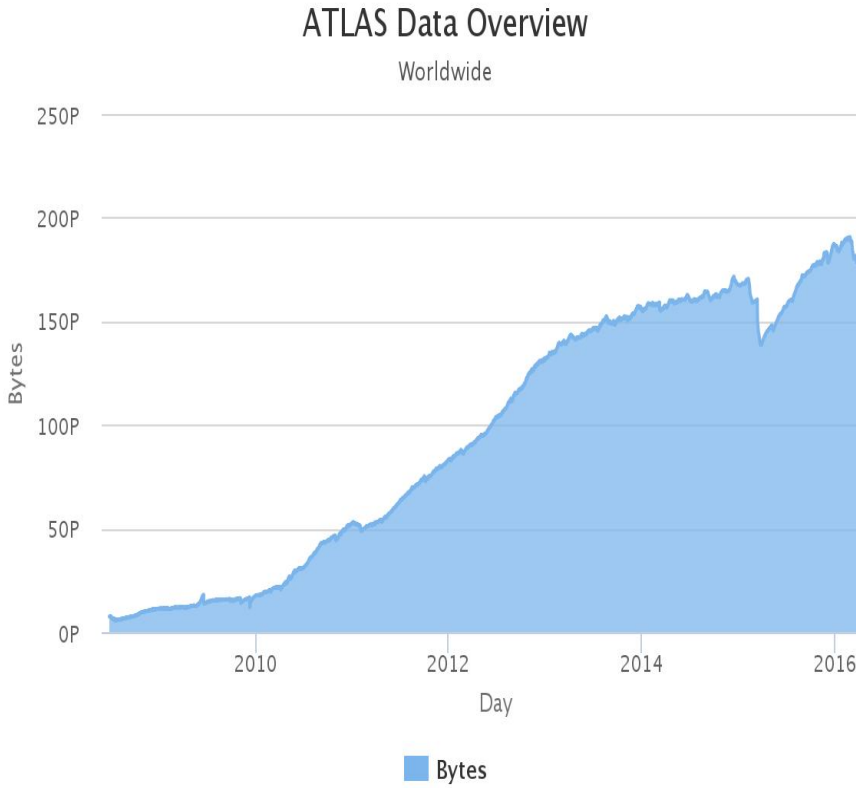
- **P**roduction **a**n**d** **D**istributed **A**nalysis system developed for ATLAS - PanDA
- Now also used by **AMS, ALICE, LSST, COMPASS** and others
- Many international partners: DoE HEP, DoE ASCR, NSF, CERN IT, OSG, ASGC, NorduGrid, European grid projects, Russian grid projects…

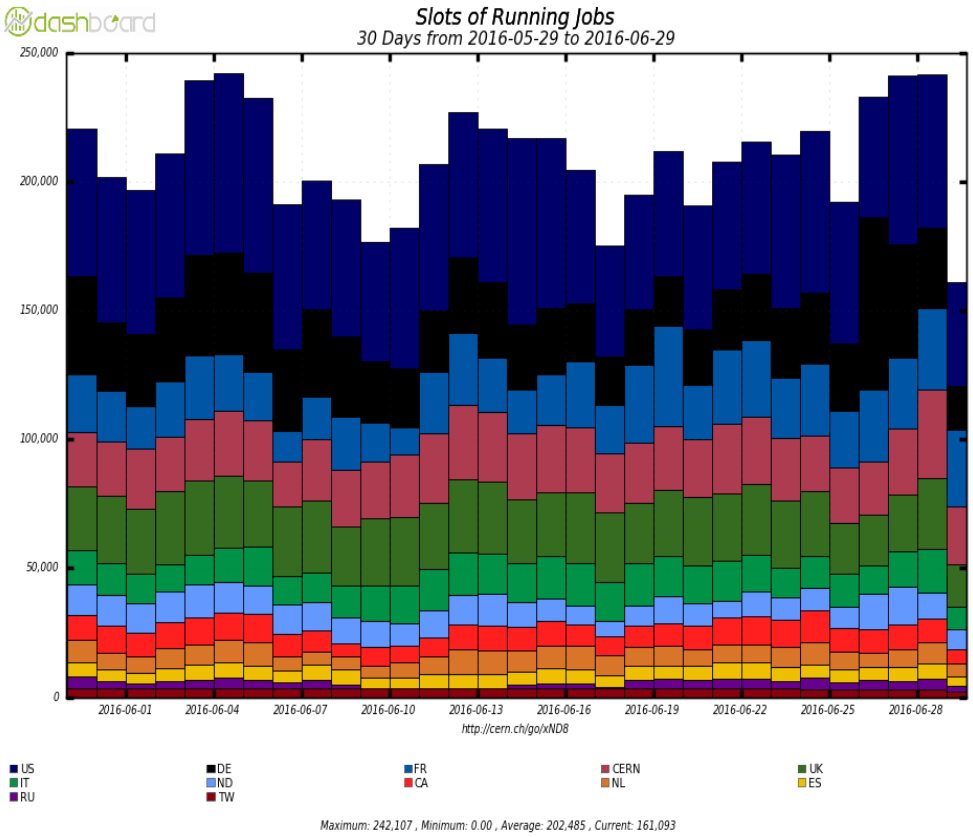
http://news.pandawms.org/

Production system in a glance



Orders of magnitude





200 PB of ATLAS data is stored

More than 200K simultaneous jobs in the system

Core idea in PanDA

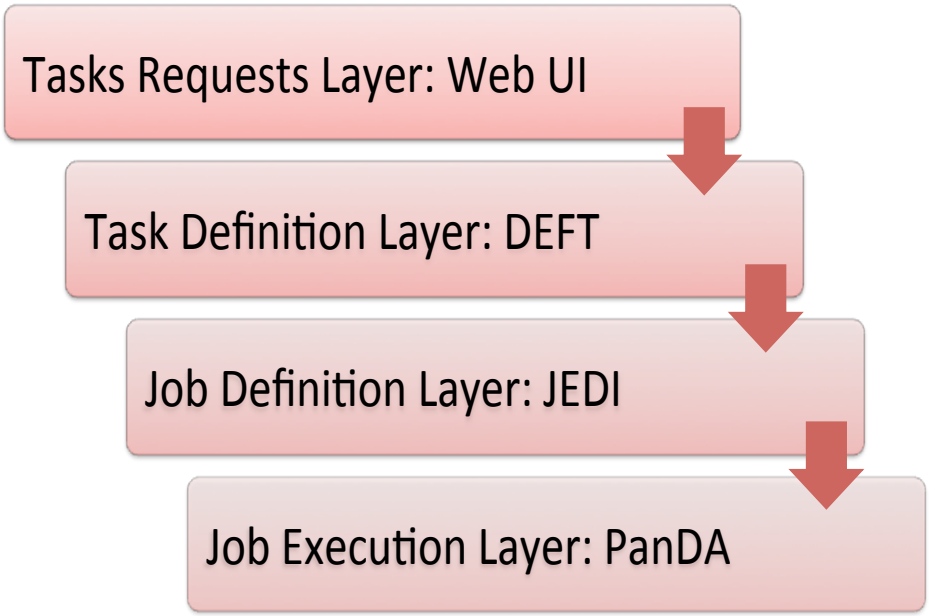
- Single entry point to the WLCG provide a central queue for users – similar to local batch systems
  - Make hundreds of distributed sites appear as local
- Reduce site related errors and reduce latency
  - Build a pilot job system – late transfer of user payloads
  - Crucial for distributed infrastructure maintained by local experts
- Hide middleware while supporting diversity and evolution
  - Atlas production system interacts with middleware – users see high level workflow
- Hide variations in infrastructure
  - Atlas production system presents uniform 'job' slots to user (with minimal sub-types)
  - Easy to integrate grid sites, clouds, HPC sites …
- Production and Analysis users see same system
  - Same set of distributed resources available to all users
  - Highly flexible system, giving full control of priorities to experiment

Key features of ATLAS production system development

- Workflow is maximally asynchronous
- Pilot based job execution system
  - Condor based pilot factory
  - Payload is sent only after execution begins on CE
  - Minimize latency, reduce error rates
- Central job queue
  - Unified treatment of distributed resources
  - SQL DB keeps state - critical component
- Automatic error handling and recovery
- Extensive monitoring
- Modular design
- RESTful communications
- GSI authentication
- Use of Open Source components

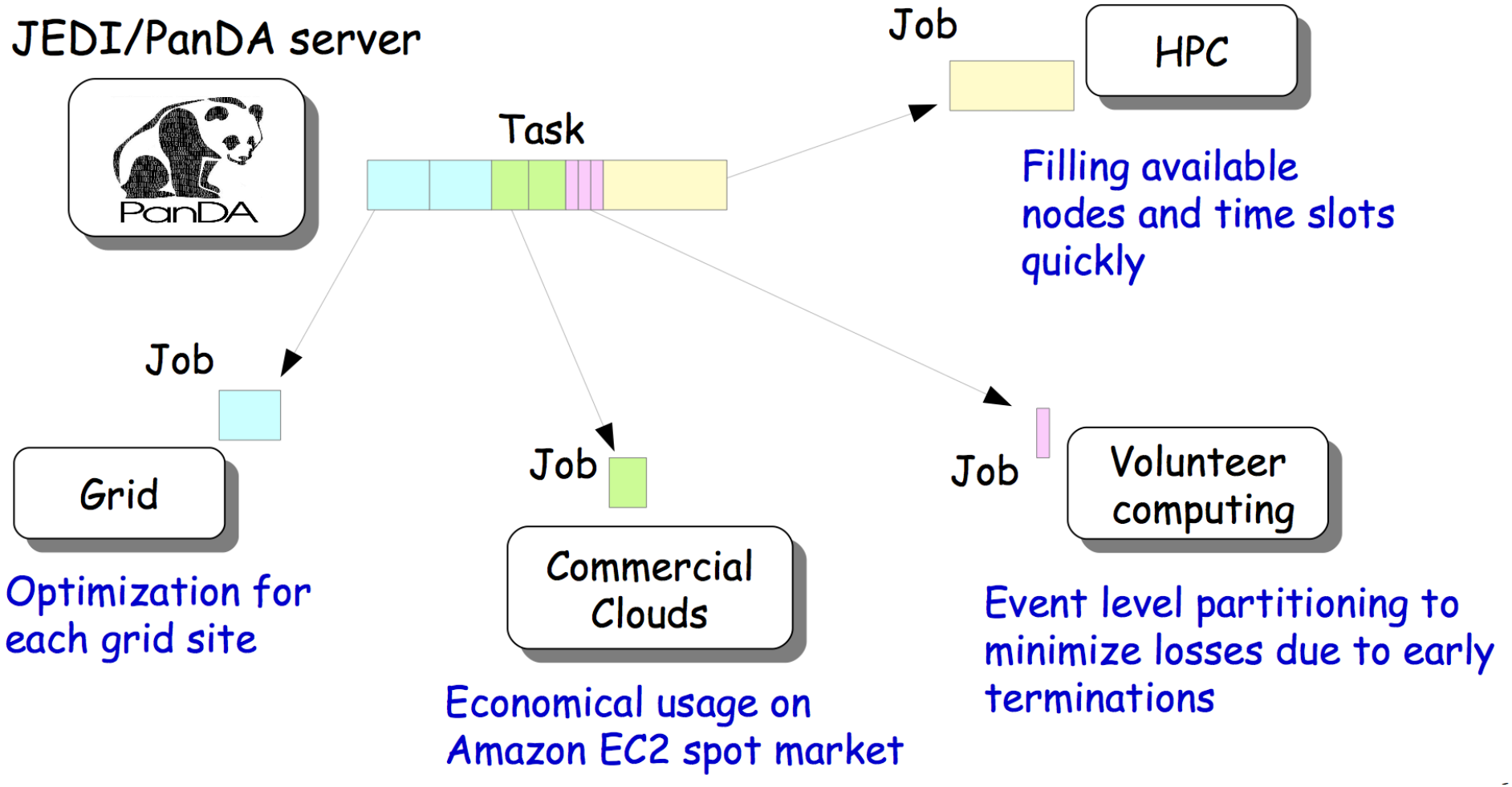
ATLAS production system components

- Web UI for Managers and Users provides the interface for task and production request managing and monitoring at the higher level
- Database Engine for Tasks (DEFT): is responsible for formulating the tasks, **chains** of tasks and also task groups (**production request**), complete with all necessary parameters
  - It also keeps track of the state of production requests, chains and their constituent tasks



- Job Execution and Definition Interface (JEDI): is an intelligent component in the panda server to have capability for **task-level** workload management.
  - Key part of it is '**Dynamic**' job definition, which highly optimizes resources usage compare to 'Static' model used in ProdSys1.
    - Dynamic job definition in JEDI is also crucial for multi-core, HPC's and other new requirements

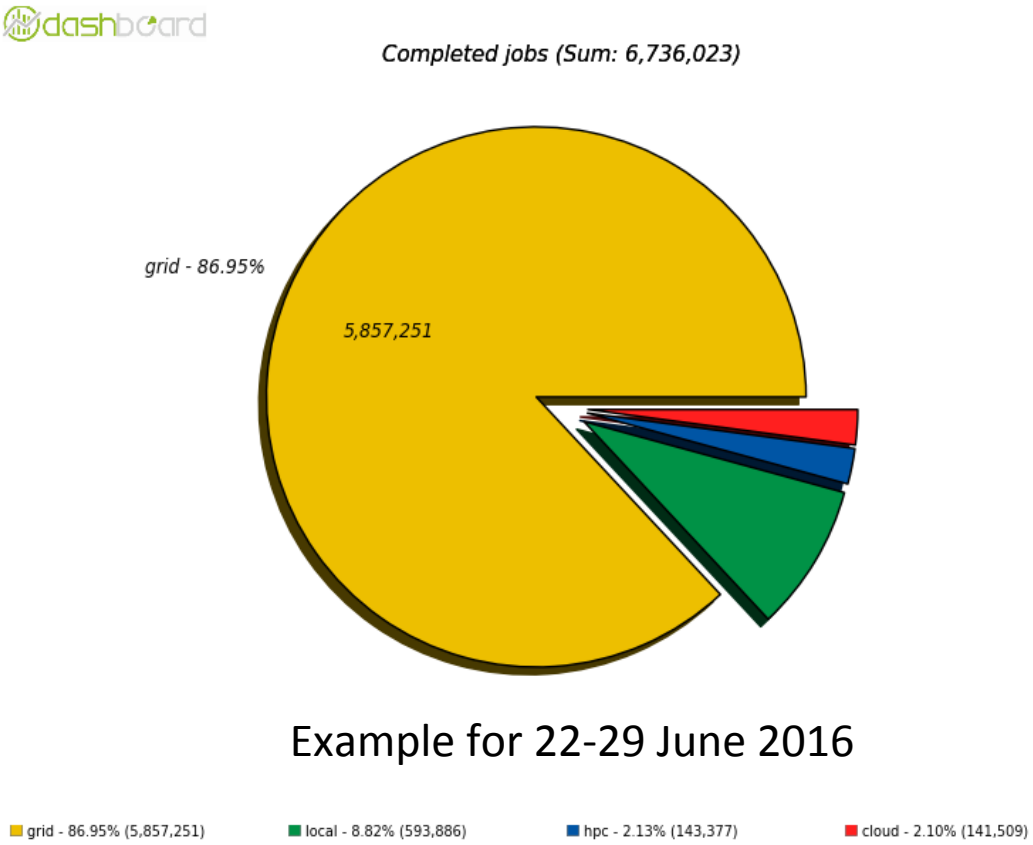
Dynamic Job definition



Dynamic job definition benefits

- Excluding requirements from users of detailed knowledge on computing resources
  - Especially for heterogeneous resources, e.g., many CPU cores, very short walltime limit, etc
- Self-optimization of job parameters
  - Real job metrics are collected using scout jobs
    - A small number (~10) of jobs (= scout jobs) are generated for each task with minimum input chunks
  - Job parameters are optimized using job metrics for the rest of input
- Simplification of client tools and centralization of user functions

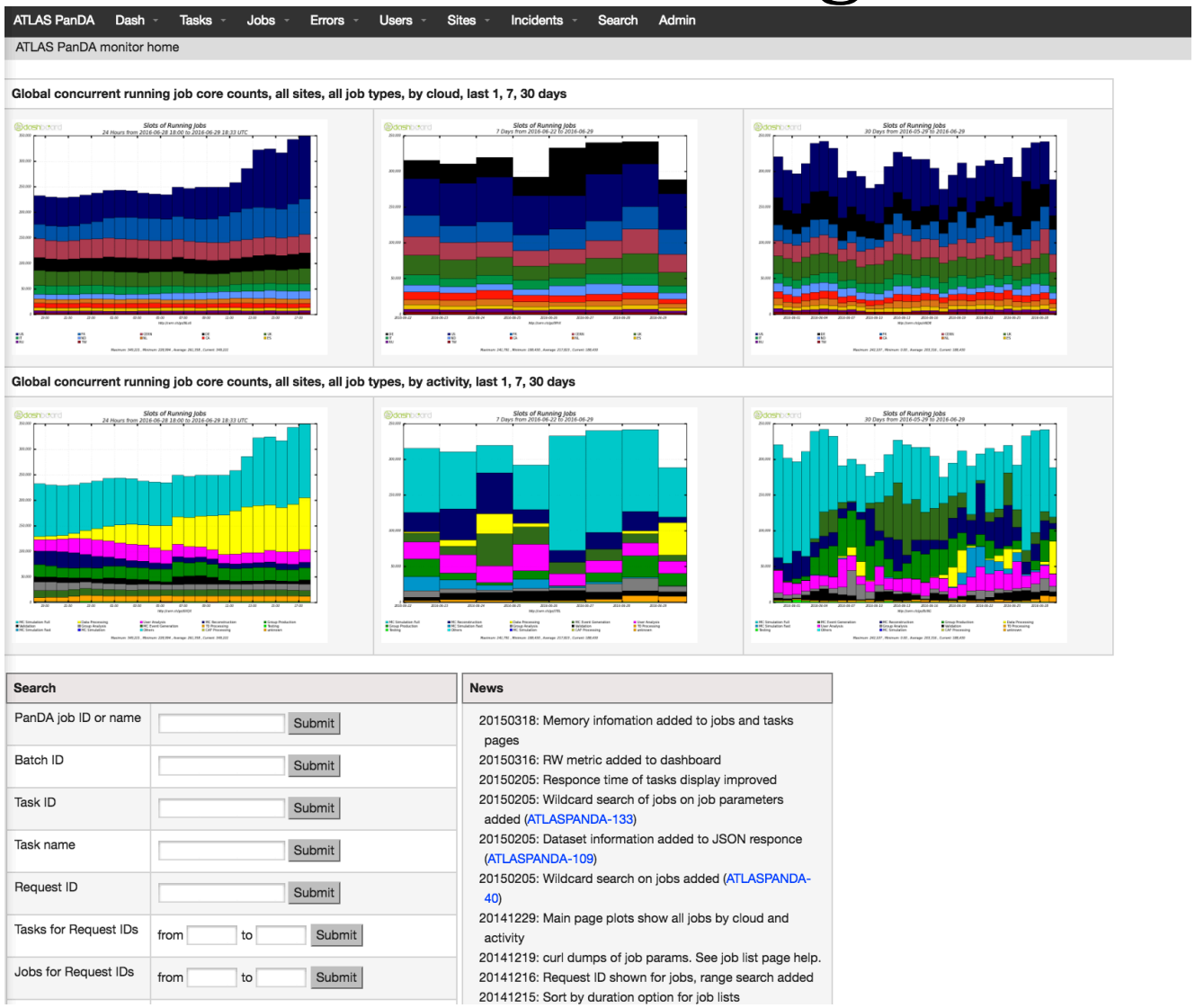
Extending beyond the Grid



Example for 22-29 June 2016

Cloud and HPC resources are steadily gaining territory

Monitoring


ATLAS PanDA
Dash
Tasks
Jobs
Errors
Users
Sites
Incidents
Search
Admin
ATLAS PanDA monitor home
Global concurrent running job core counts, all sites, all job types, by cloud, last 1, 7, 30 days
Global concurrent running job core counts, all sites, all job types, by activity, last 1, 7, 30 days
Search
PanDA job ID or name
Batch ID
Task ID
Task name
Request ID
Tasks for Request IDs
Jobs for Request IDs
Submit
from
to
News
20150318: Memory infomation added to jobs and tasks pages
20150316: RW metric added to dashboard
20150205: Responce time of tasks display improved
20150205: Wildcard search of jobs on job parameters added (ATLASPANDA-133)
20150205: Dataset information added to JSON responce (ATLASPANDA-109)
20150205: Wildcard search on jobs added (ATLASPANDA-40)
20141229: Main page plots show all jobs by cloud and activity
20141219: curl dumps of job params. See job list page help.
20141216: Request ID shown for jobs, range search added
20141215: Sort by duration option for job lists

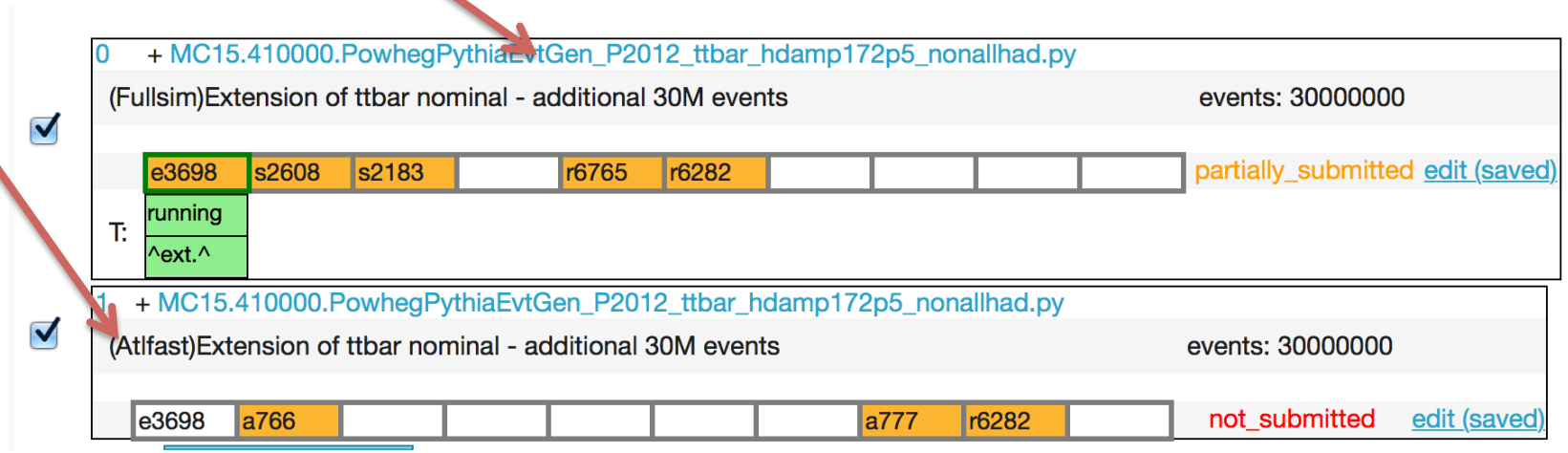
Tasks requests - use-cases

- Production system is a workflow driven system and it's used for dealing with the all variety of ATLAS activities:
  - Simulation (MC production)
  - Data processing ("Tier0" processing and reprocessing)
  - High Level Trigger (HLT) reprocessing
  - Derivation and train production (slimming, skimming…)
  - Event Index
- Typical ATLAS workflow composed of many data transformation steps, e.g. the Monte Carlo simulations workflow is composed of many steps: generate hard-processes, hadronize signal and minimum-bias events, simulate energy deposition in the ATLAS detector, digitize electronics response, simulate triggers, reconstruct data, transform the reconstructed data into reduced forms for physics analysis

MC production request creation

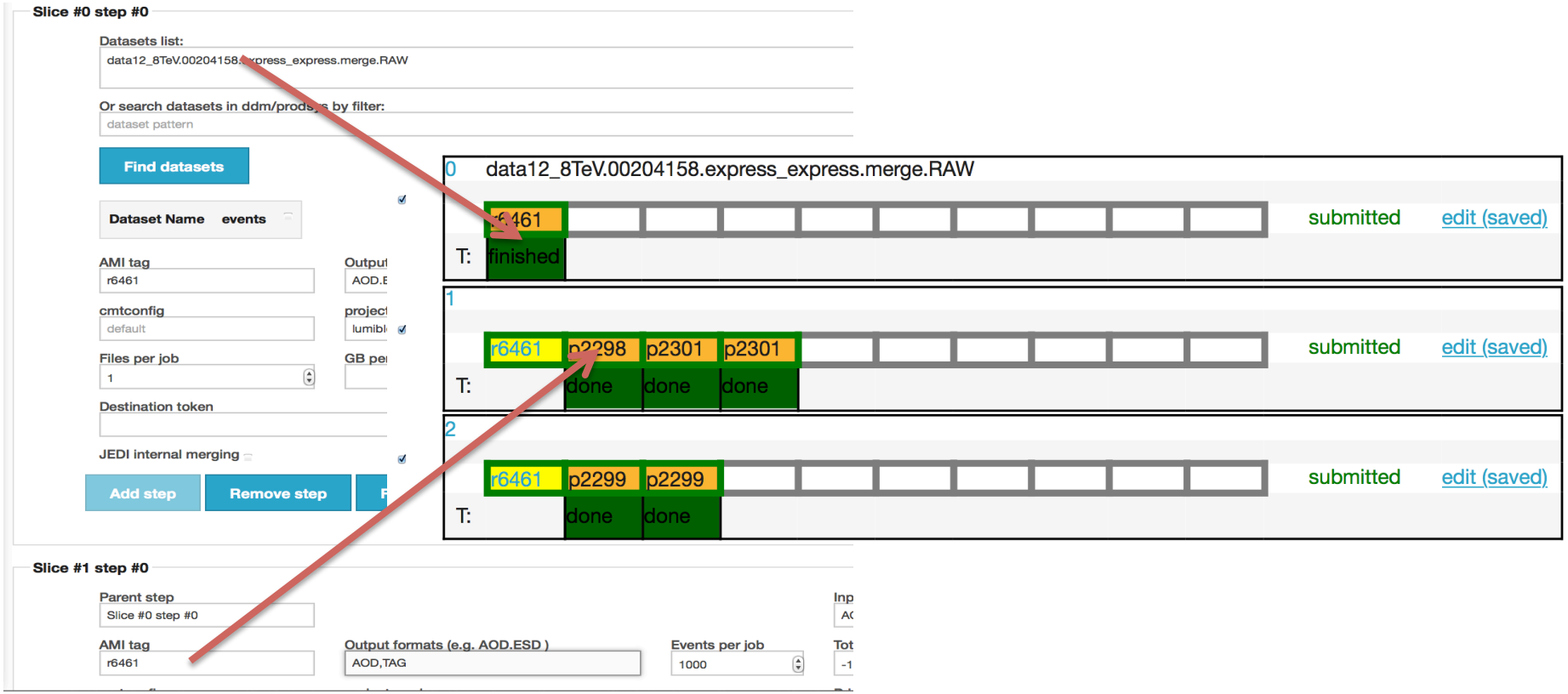
- Creating of production request is one of the examples how different workflows can be integrated to the system
- For MC Google spreadsheet were used by user to provide a data for MC tasks. In ProdSys2 it was adopted so user can submit input data in the same format as they use before.

	A	B	C	D	E	F	G	H	I
1	brief	datasetNum	ESD,RDO	Joboptions	Event (FullSim) Config 25ns	Event (AF2) Config 25ns	Priority	Evgen	Sim
2	Powheg+Pythia6 nominal ttbar (hdamp = mtop)	410000		MC15.410000.PowhegPythiaEvtGen_P2012_ttbar_hdamp172p5_nonallhad.py	30000000		0	e3698	s26(
3	Powheg+Pythia6 nominal ttbar (hdamp = mtop)	410000		MC15.410000.PowhegPythiaEvtGen_P2012_ttbar_hdamp172p5_nonallhad.py		30000000	0	e3698	a7(
4									



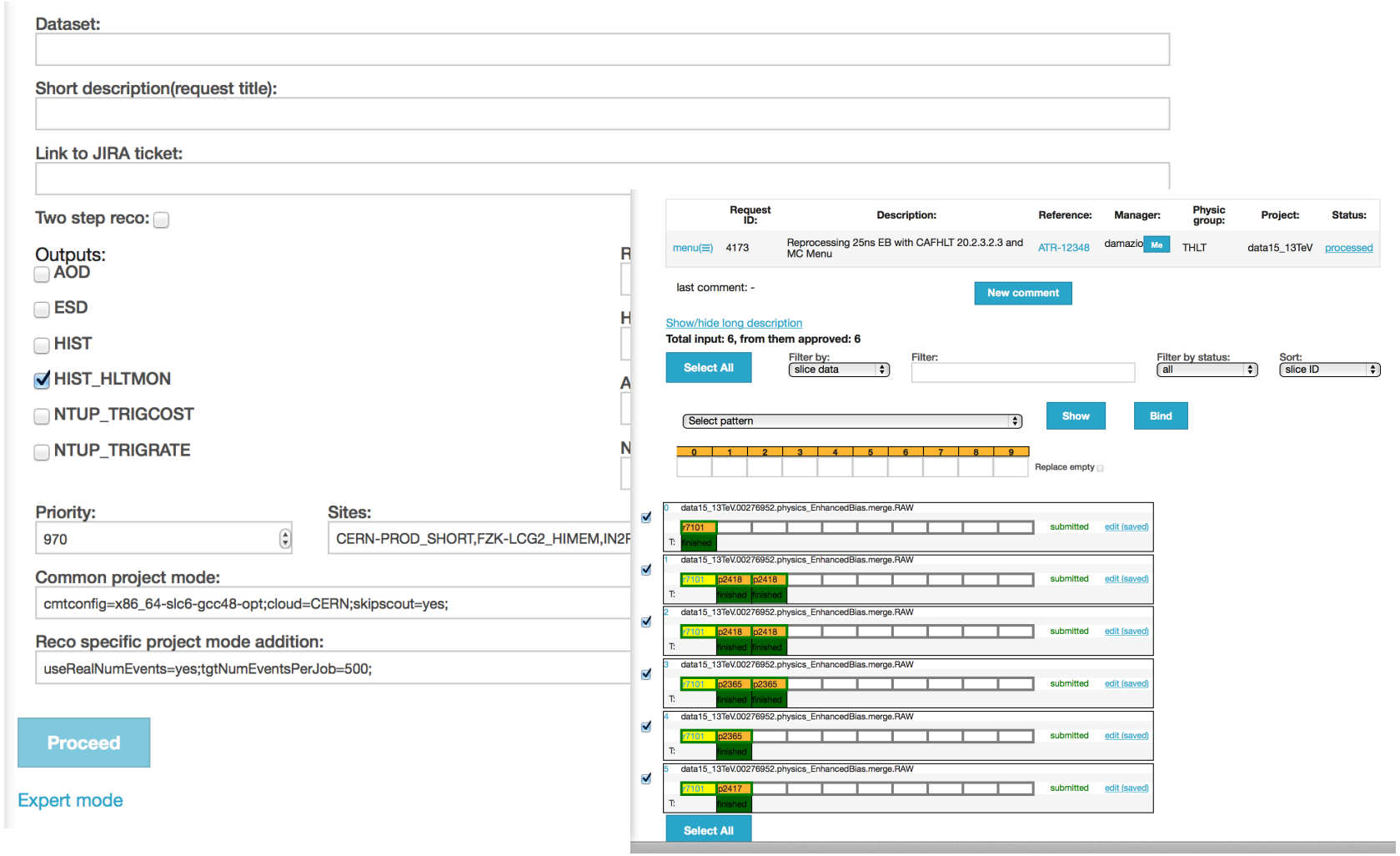
Reprocessing production request creation

- Reprocessing workflow has a tree structure, where output of one task can be an input for several more tasks – interface for creating such structure was developed for it.



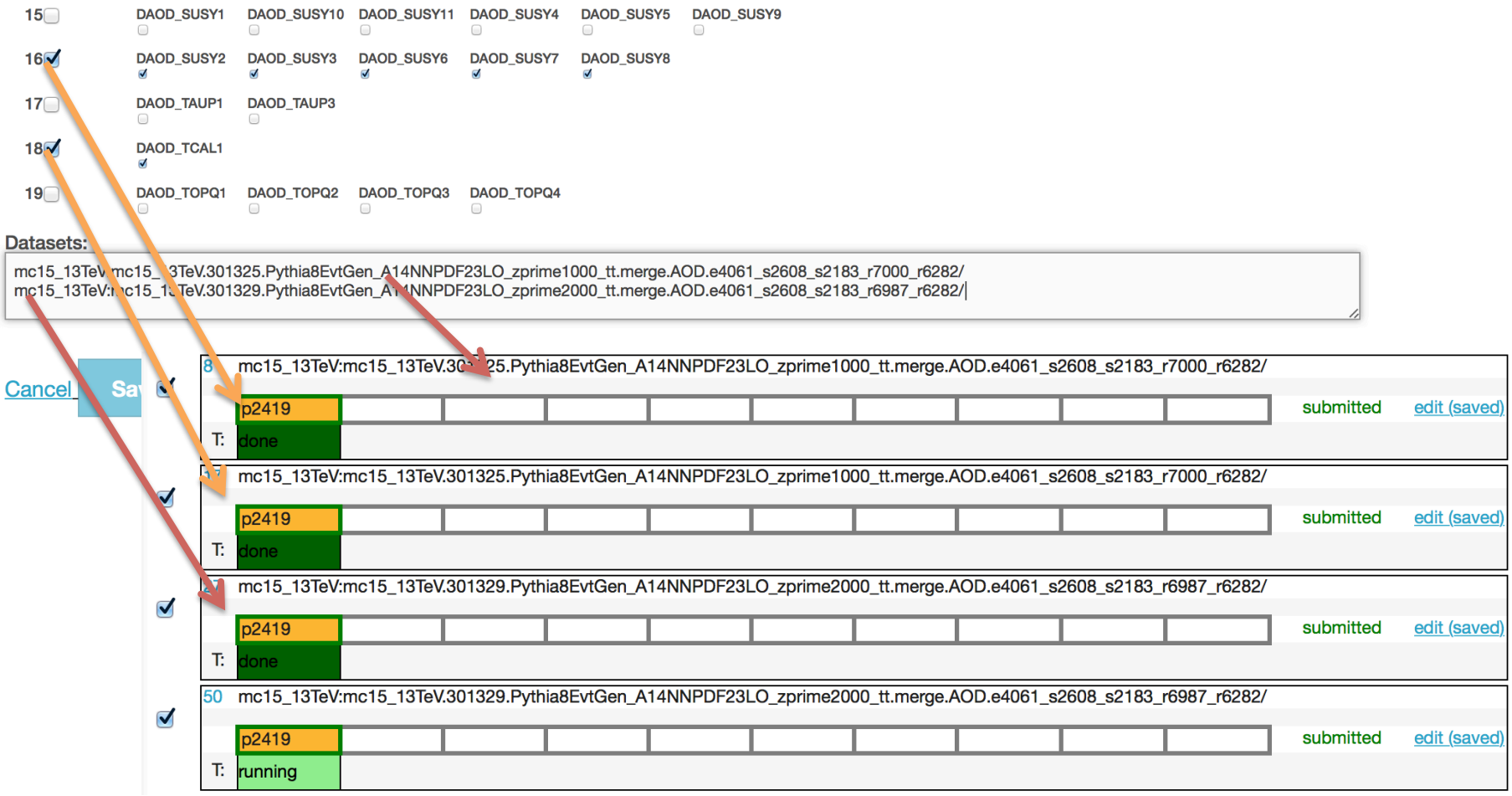
HLT production request creation

- HLT reprocessing has a well defined workflow, so interface for HLT production request creation includes only a few fields.



Derivation production request creation

- Derivation is using so called "train" model, there each input runs on some of many predefined outputs. To manage with complexity, pattern request is created first, and system is using this pattern to generate possible options for the specific derivation production request creation interface.



Work with production request

- Tasks requests Web U/I provides many general and experiment specific features:
  - **Bookkeeping**.
  - **Approve management**. E.g. MC production request required several levels of approval.
  - **Monitoring**. User can easily follow progress of a running tasks.
  - **Error Handling**. Task could failed because of many permanent (e.g. bug in software) and temporal (storage is down) reasons. To be able quickly understand root of the problem and fix it by redefining the task is one of the major feature of the production system.
  - **Chaining** one production to the other. E.g. derivation production could be chained to MC or reprocessing task, that significantly speed up them.
  - **Automation** task submission. User can defined a pattern and when new data appears tasks are started automatically.
  - …

Summary

- Production system has performed well for ATLAS including the LHC Run 1 data taking period
- New components and features have been delivered to ATLAS before LHC Run 2
- Many developments and challenges to come
  - New resources as Clouds and HPC
  - Full integration of network as a resource in workload management