Evolution of the ATLAS Distributed Computing during the LHC long shutdown

Simone Campana CERN-IT-SDC

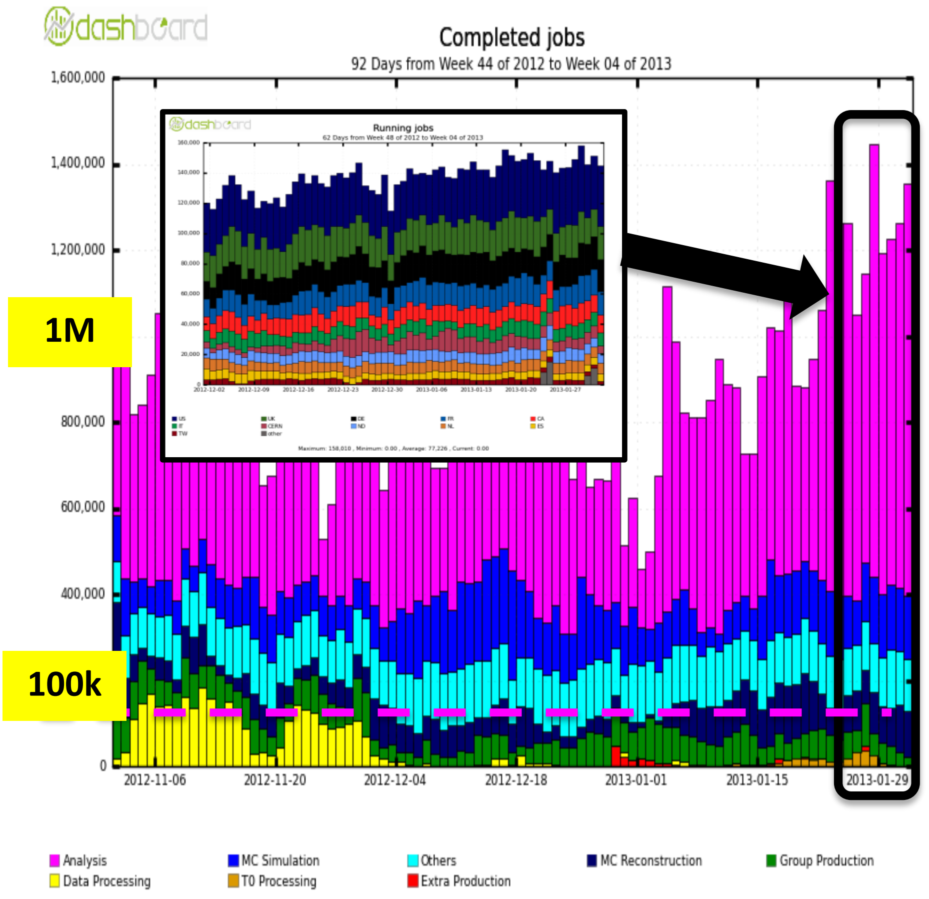
on behalf of the ATLAS collaboration

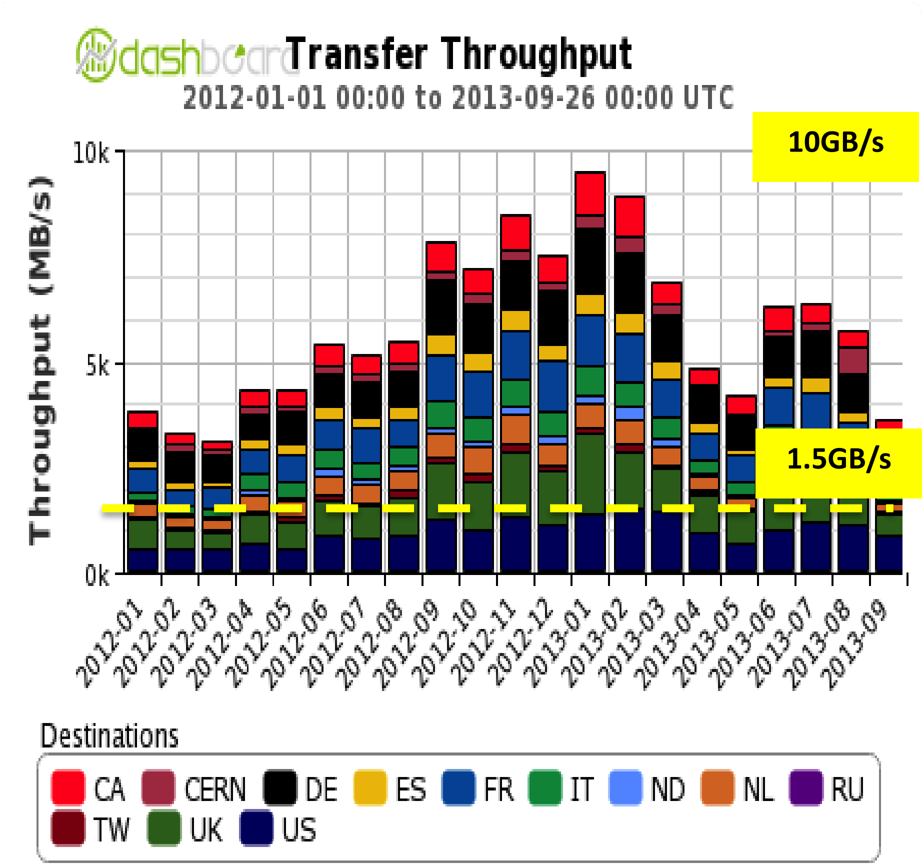
14 October 2013

Run-1: Workload and Data Management

1.4M jobs/day, 150K concurrently running
(2007 gLite WMS acceptance tests: 100K jobs/day)

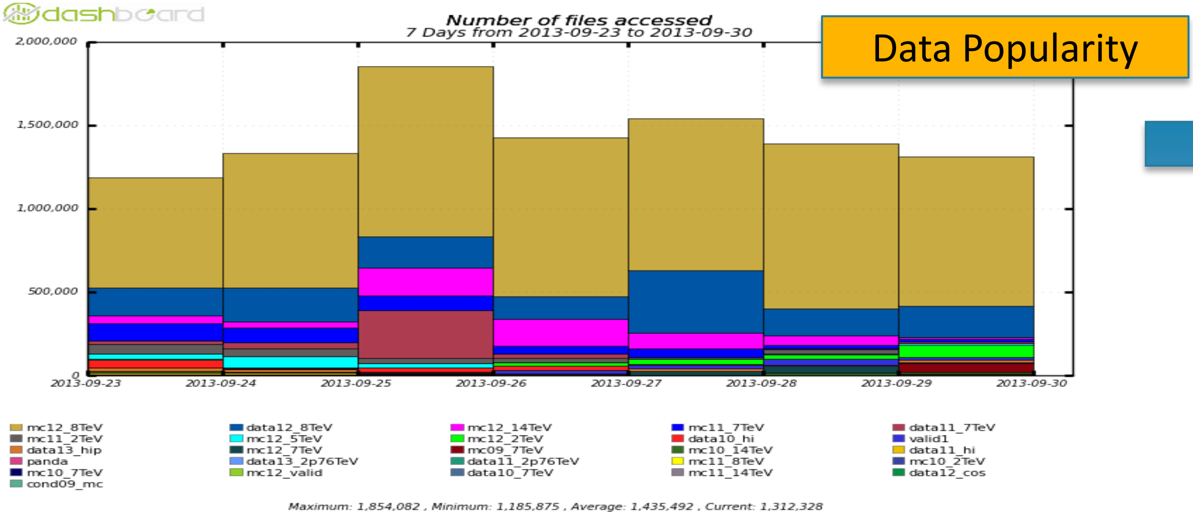
Nearly 10GB/s transfer rate
(STEP09 target: 1.5GB/s)





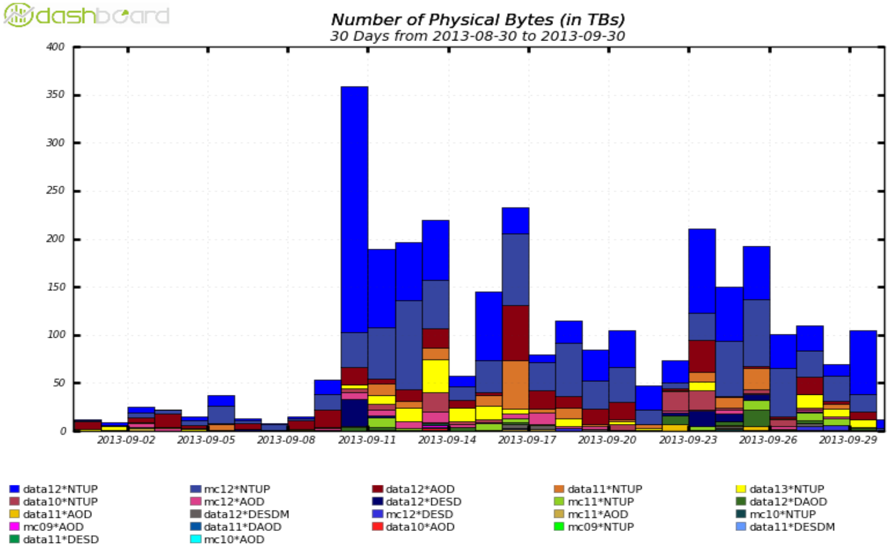
Run-1:Dynamic Data Replication and Reduction

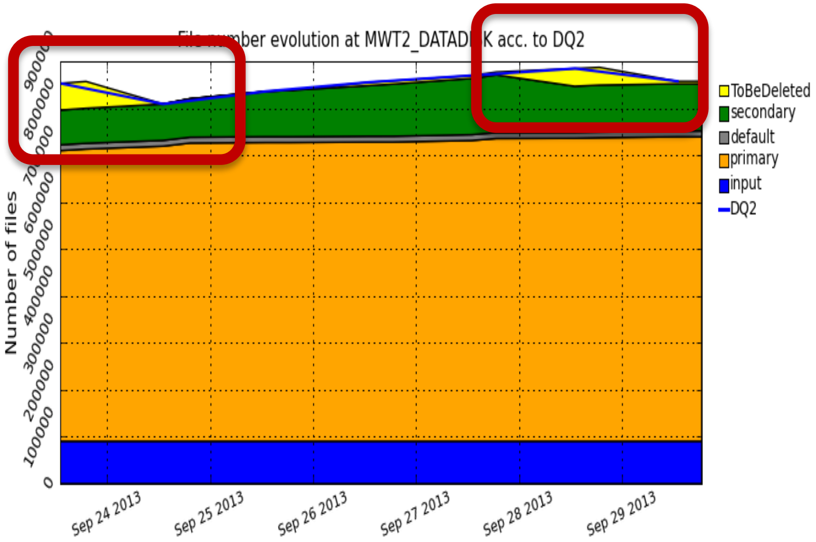
Dynamic Replication



Data Popularity

Dynamic Reduction





Challenges of Run-2

- Trigger rate: from 550Hz to 1kHz
  - Therefore, more events to record and process
- Luminosity increase: event pile-up from 25 to 40
  - so more complexity for processing and +20% event size
- Flat resource budget
  - For storage, CPUs and network (apart for Moore’s law)
  - For operations manpower
- The ATLAS Distributed Computing infrastructure needs to evolve in order to face those challenges

This presentation will …

- … provide an overview of the major evolutions in ATLAS Distributed Computing expected during Long Shutdown 1

- Many items mentioned here will be covered in more detailed presentations and posters during CHEP2013

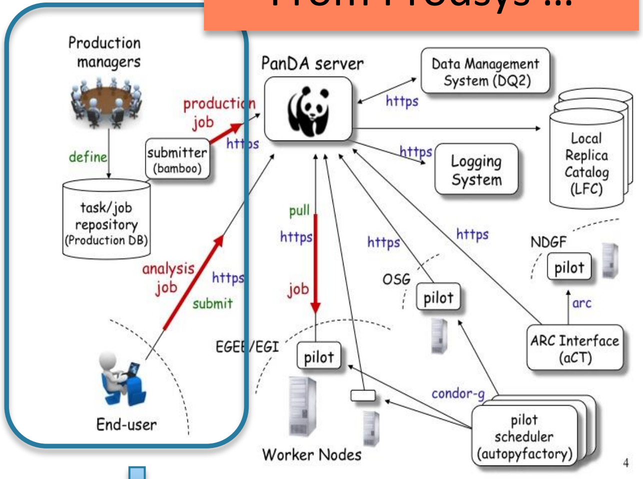
  - This includes items which I decided not to mention here for time reasons, but which are still very important

  - This includes items which I did not mention here because outside the Distributed Computing domain, but still very relevant for Distributed Computing to face the Run-2 challenges

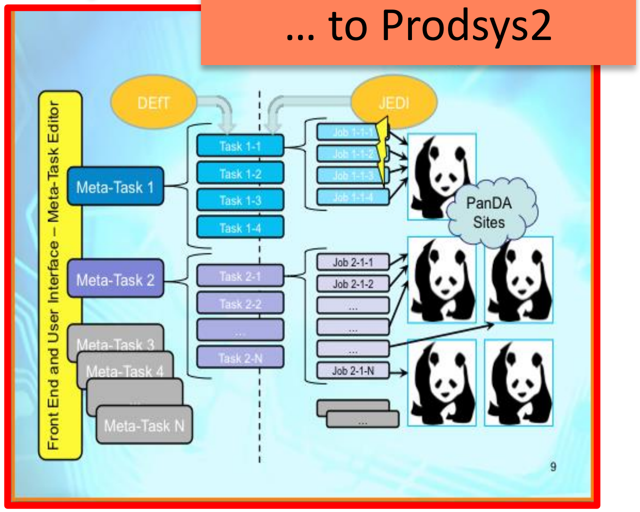
Workload Management in Run-2: Prodsys2

- Prodsys2 core components
  - DEFT: translates user requests into task definitions
  - JEDI: dynamically generates the job definitions
  - PanDA: the job management engine
- Features:
  - Provide a workflow engine for both production and analysis
  - Minimize data traffic (smart merging)
  - Optimized job parameters to available resources

From Prodsys …



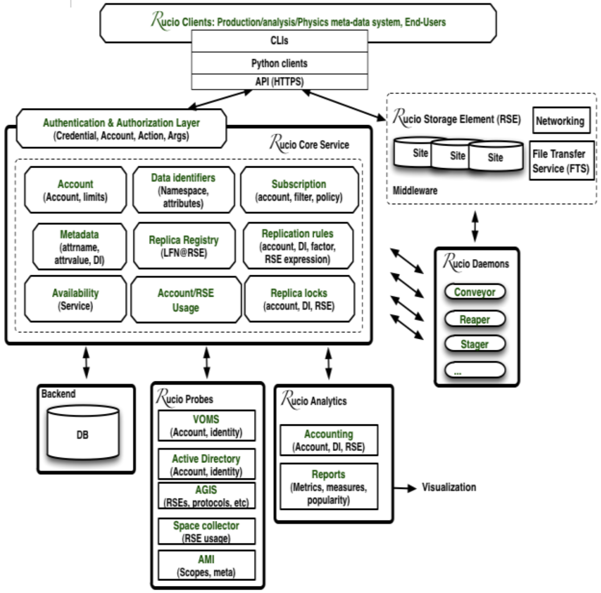
… to Prodsys2



Data Management in Run-2: Rucio

- Implements a highly evolved Data Management model
  - ➢ File (rather than dataset) level granularity
  - ➢ Multiple file ownership per user/group/activity

http://rucio.cern.ch/

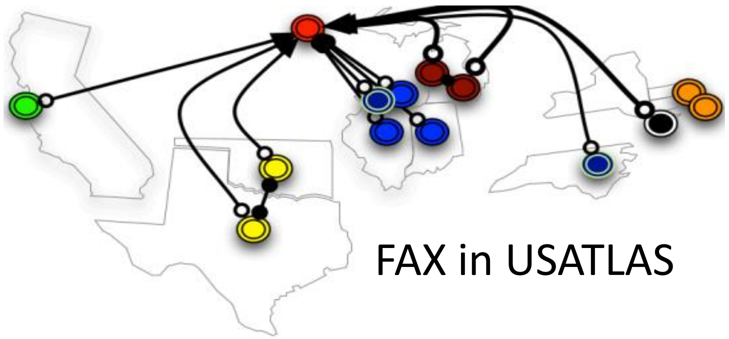


Rucio Clients: Production/analysis/Physics meta-data system, End-Users
CLIs
Python clients
API (HTTPS)
Authentication & Authorization Layer
(Credential, Account, Action, Args)
Rucio Core Service
Account
(Account, limits)
Data identifiers
(Namespace, attributes)
Subscription
(account, filter, policy)
Metadata
(attrname, attrvalue, DI)
Replica Registry
(LFN@RSE)
Replication rules
(account, DI, factor, RSE expression)
Availability
(Service)
Account/RSE Usage
Replica locks
(account, DI, RSE)
Rucio Storage Element (RSE)
Networking
Site
Site
Site
File Transfer Service (FTS)
Middleware
Rucio Daemons
Conveyor
Reaper
Stager
...
Backend
DB
Rucio Probes
VOMS
(Account, identity)
Active Directory
(Account, identity)
AGIS
(RSEs, protocols, etc)
Space collector
(RSE usage)
AMI
(Scopes, meta)
Rucio Analytics
Accounting
(Account, DI, RSE)
Reports
(Metrics, measures, popularity)
Visualization

- Features
  - ➢ Unified dataset/file catalogue with support for metadata
  - ➢ Built-in policy based data replication for space and network optimization
  - ➢ Redesign leveraging new middleware capabilities (FTS/GFAL-2)
  - ➢ Plug-in based architecture supporting multiple protocols (SRM/gridFTP/xrootd/HTTP…)
  - ➢ REST-ful interface

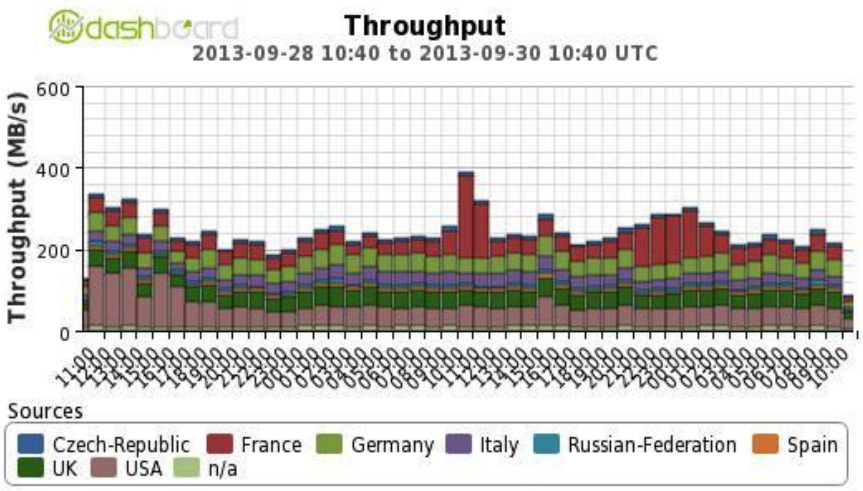
Data Management in Run-2: FAX

- ATLAS is deploying a federated storage infrastructure based on xrootd
  - ➢ Complementary to Rucio and leveraging its new features
  - ➢ Offers transparent access to “nearest” available replica
  - ➢ The protocol enables remote (WAN) direct data access to the storage
  - ➢ Could utilize different protocols (e.g. HTTP) in future



FAX in USATLAS

- Scenarios (increasing complexity)
  - ➢ Jobs failover to FAX in case of data access failure
    - If the job can not access the file locally, it then tries through FAX
  - ➢ Loosening the job-to-data locality in brokering
    - From “jobs-go-to-data” to “jobs-go-as-close-as-possible-to-data”
  - ➢ Dynamic data caching based on access
    - File or even event level



Opportunistic Resources: Clouds

- A “Cloud” infrastructure allows to demand resources through an established interface
  - ➢ (If it can) it gives you back a (virtual) machine for you to use
  - ➢ You become the “administrator” of your cluster

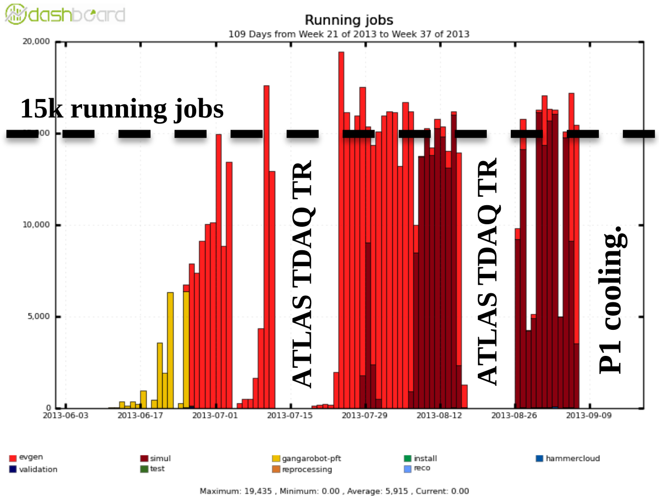
- “Free” opportunistic cloud resources
  - ➢ The ATLAS HLT farm is accessible through cloud interface during the Long Shutdown
  - ➢ Academic facilities offering access to their infrastructure through a cloud interface

- “Cheap” opportunistic cloud resources
  - ➢ Commercial Infrastructures (Amazon EC2, Google, …) offering good deals under restrictive conditions

- Work done in ATLAS Distributed Computing
  - ➢ Define a model for accessing and utilizing cloud resources effectively in ATLAS
  - ➢ Develop necessary components for integration with cloud resources and automation of the workflows

ATLAS HLT farm

WCT Efficiency
CERN Grid: 93.6%
HLT: 91.1%



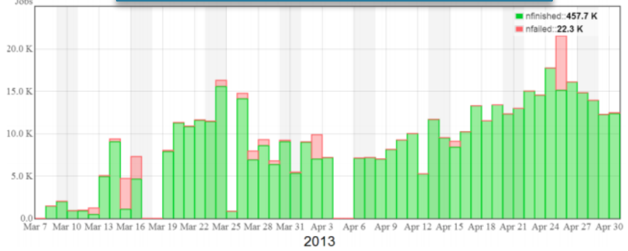
15k running jobs

ATLAS TDAQ TR

ATLAS TDAQ TR

P1 cooling.

Google cloud



- Most of the job failures occurred during start up and scale up phase – as expected
- Average error rate ~6%
- Reached throughput of 15k jobs per day
- Finished 457k jobs, generated 214M events

Opportunistic Resources: HPCs

- HPC offers important and necessary opportunities for HEP
  - Possibility to parasitically utilize empty cycles

- Bad news: very wide spectrum of site policies
  - No External connectivity
  - Small Disk size
  - No pre-installed Grid clients
  - One solution unlikely to fit all

- Good news: from code perspective, anything seriously tried so far did work
  - Geant4, ROOT, generators

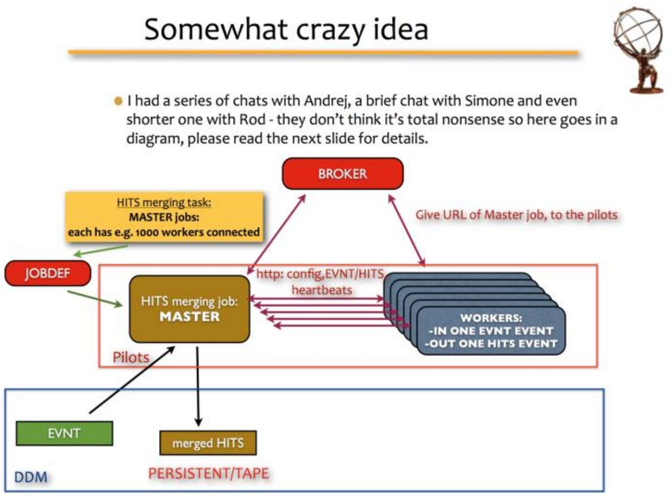
- Short jobs preferable for backfilling

Oak Ridge Titan System	
Architecture:	Cray XK7
Cabinets:	200
Total cores:	299,008 Opteron Cores
Memory/core:	2GB
Speed:	20+ PF
Square Footage	4,352 sq feet

HPC exploitation is now a coordinated ATLAS activity

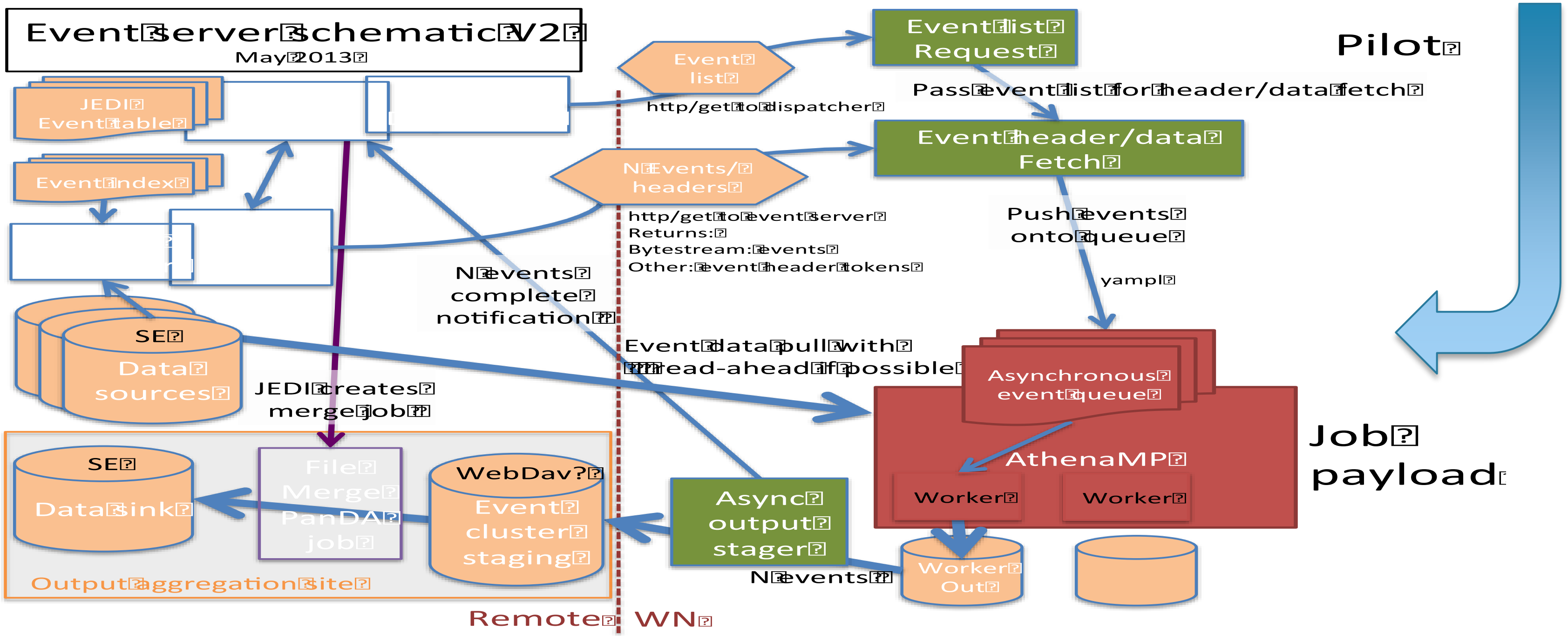
Event Service

- A collaborative effort within ATLAS SW&C
- Reduces the job granularity from a collection of events to a single event
- Would rely on existing ATLAS components



Somewhat crazy idea

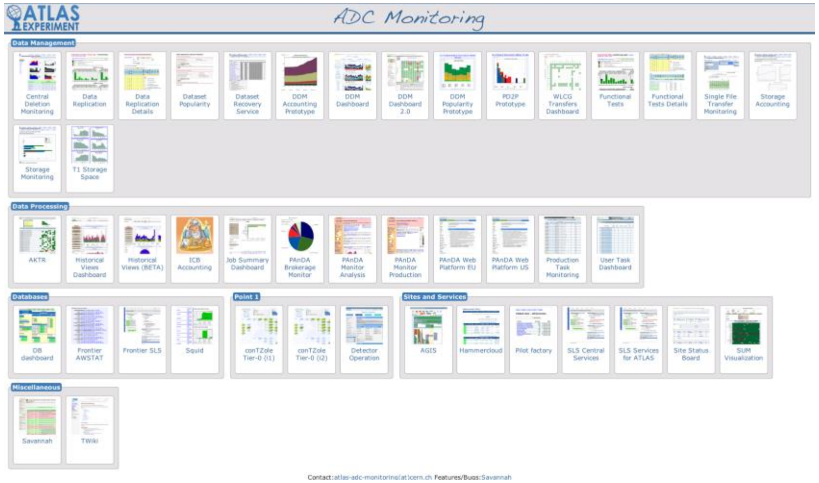
- I had a series of chats with Andrej, a brief chat with Simone and even shorter one with Rod - they don't think it's total nonsense so here goes in a diagram, please read the next slide for details.



Monitoring

http://adc-monitoring.cern.ch/

- Excellent progress in last 2 years
  - we really have most of what we need
    - Still, monitoring is never enough
  - Oriented toward many communities
    - Shifters and Experts
    - Users
    - Management and Funding Agencies
  - High quality for presentation and rendering



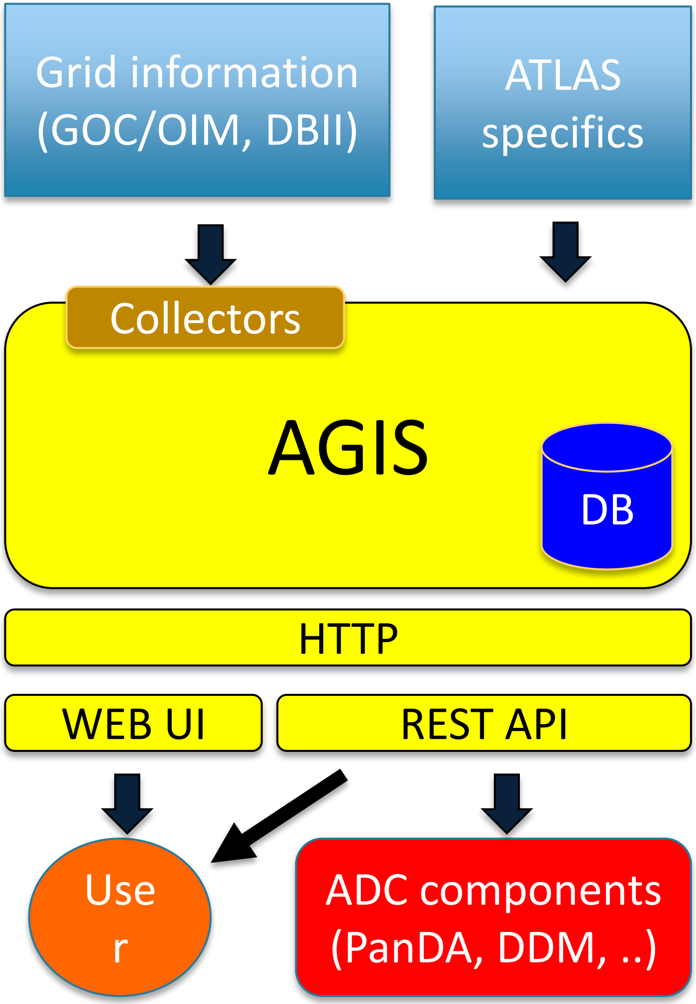
Contact:atlas-adc-monitoring(at)cern.ch Features/Bugs:Savannah

- Converged on an “ADC monitoring architecture”
  - Standard de facto

- Challenges for the Long Shutdown
  - Rationalization of our monitoring system
  - Porting monitoring to the newly developed components (not coming for free)
    - Prodsys2 and Rucio in primis

The ATLAS Grid Information System

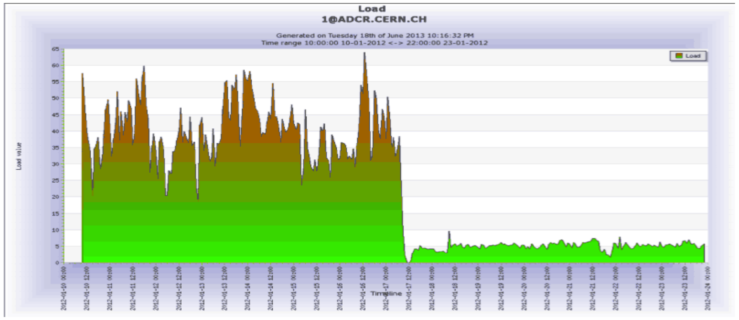
- We successfully deployed AGIS in production
  - Source repository of information for PanDA and DDM
  - More a configuration service than an information system
- The effort was not only in software development
  - Information was spread over many places and not always consistent
  - Rationalization was a big challenge
- Challenges in LS1
  - AGIS will have to evolve to cover the requirements of the newly developed systems
  - Some already existing requirements in the TODO list

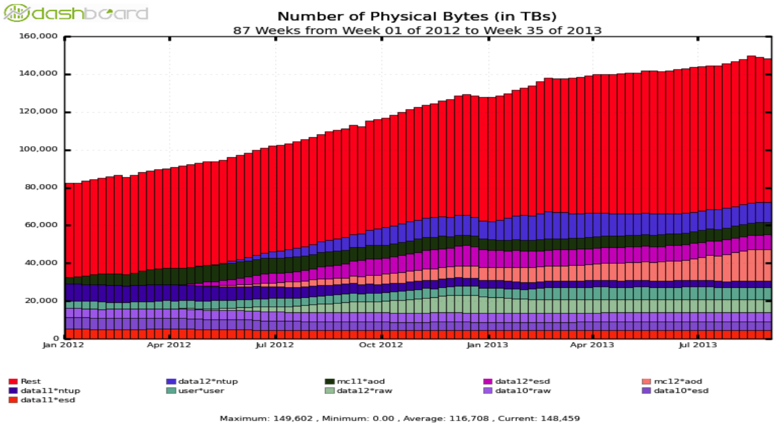

Grid information (GOC/OIM, DBII)
ATLAS specifics
Collectors
AGIS
DB
HTTP
WEB UI
REST API
Use r
ADC components (PanDA, DDM, ..)

Databases

- Relational databases (mostly Oracle) are currently working well
  - At today’s scale

- Big improvement after the 11g migration
  - Better hardware (always helps)
  - More redundant setup from IT-DB (standby/failover/..)
  - Lots of work from ATLAS DBAs and ADC devs to improve the applications





- Many use cases might be more suitable for NoSQL solution
  - WLCG converged on Hadoop as mainstream (big ATLAS contribution)
  - Hadoop already used in production in DDM (accounting)
  - Under consideration as main technology for an Event Index service

- Frontier/Squid fully functional for all remote database access at all sites

Summary

- ADC development is driven by operations
  - Quickly react to operational issues
- Nevertheless we took on board many R&D projects
  - With the aim to quickly converge on possible usability in production
  - All our R&Ds made it to production (NoSQL, FAX, Cloud Computing)
- Core components (Prodsys2 and Rucio) seem well on schedule
  - Other activities started at good pace
- Our model of incremental development steps and commissioning has been a key component for the success of Run-1