The ATLAS Distributed Data Management System

David Cameron

EPF Seminar

6 June 2007

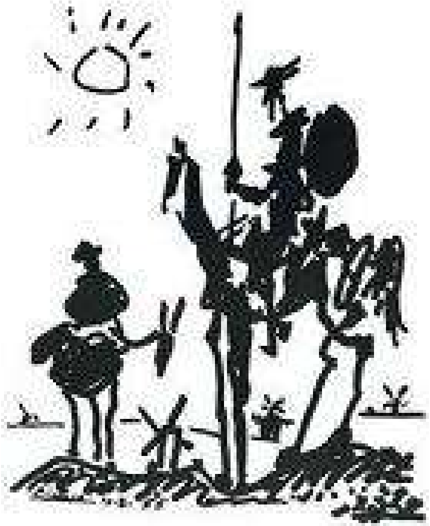
Firstly… about me

- MSci in physics + astronomy (2001, Univ. of Glasgow)
- PhD “Replica Management and Optimisation for Data Grids” (2005, Univ. of Glasgow)
  - Working with the European DataGrid project in data management and Grid simulation
- CERN fellow on ATLAS data management (2005-2007)
  - This talk!
- Developer for NDGF (1st March 2007 -)

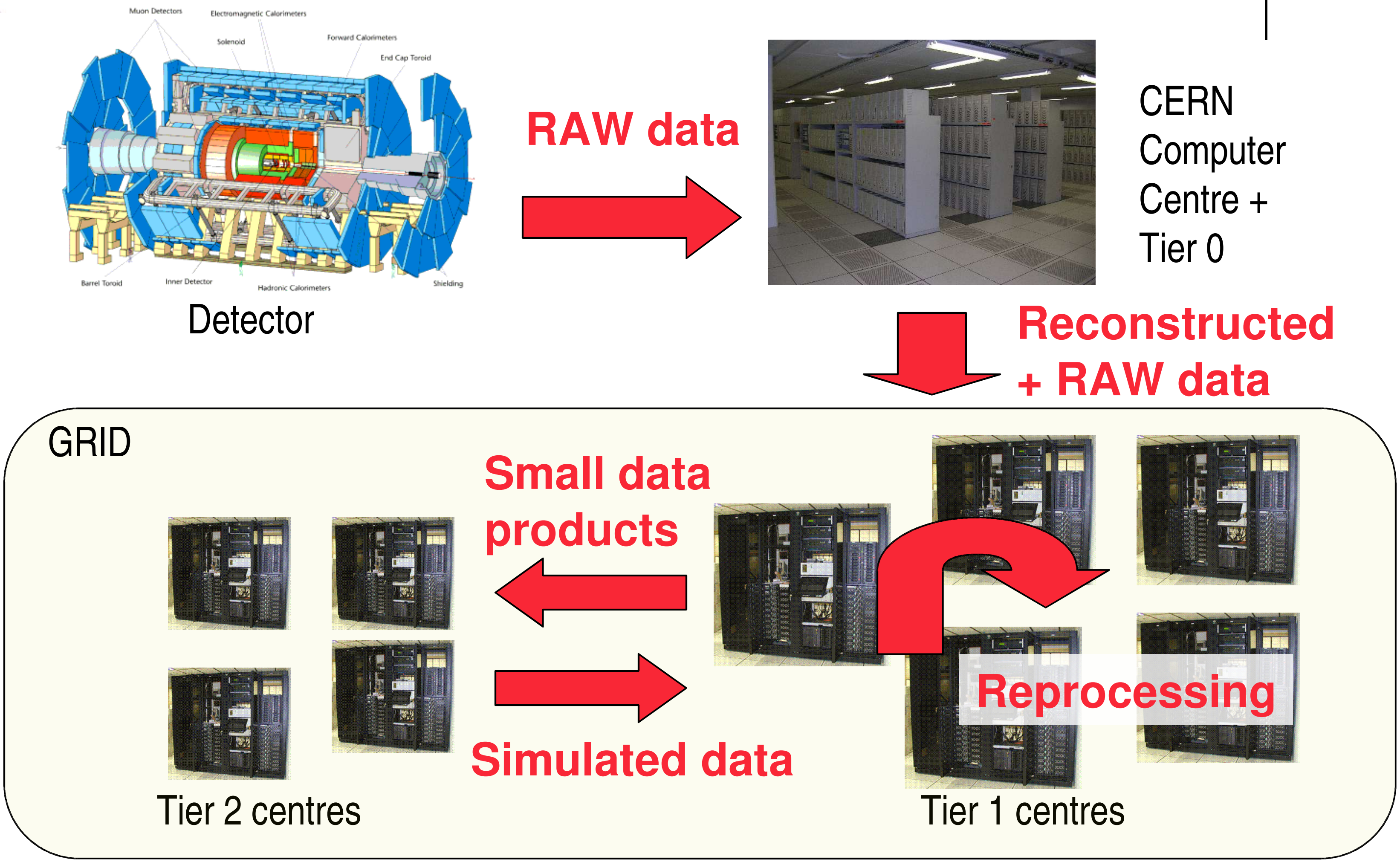
This is not me…

Outline

- The computing model for the ATLAS experiment
- The ATLAS Distributed Data Management system - Don Quijote 2
- Architecture
- External components + NG interaction
- How it is used and some results
- Current and future developments and issues



The ATLAS Experiment Data Flow



Detector

RAW data

CERN Computer Centre + Tier 0

Reconstructed + RAW data

GRID

Small data products

Simulated data

Reprocessing

Tier 2 centres

Tier 1 centres

The ATLAS experiment data flow

- At CERN, first pass processing and distribution of raw and reconstructed data from CERN to the Tier-1s
  - Massive data movement T0 -> 10 T1s (~1 GB/s out of CERN)
- Distribution of AODs (Analysis Object Data) to Tier-2 centres for analysis
  - Data movement 10 T1s -> 50 T2s (~20 MB/s per T1)
- Storage of simulated data (produced by Tier-2s) at Tier-1 centres for further distribution and/or processing
  - Data movement T2 -> T1 (20% of real data)
- Reprocessing of data at Tier-1 centres
  - Data movement T1 -> T1 (10% of T0 data)
- Analysis - jobs go to data
  - But there will always be some data movement requested by physicists

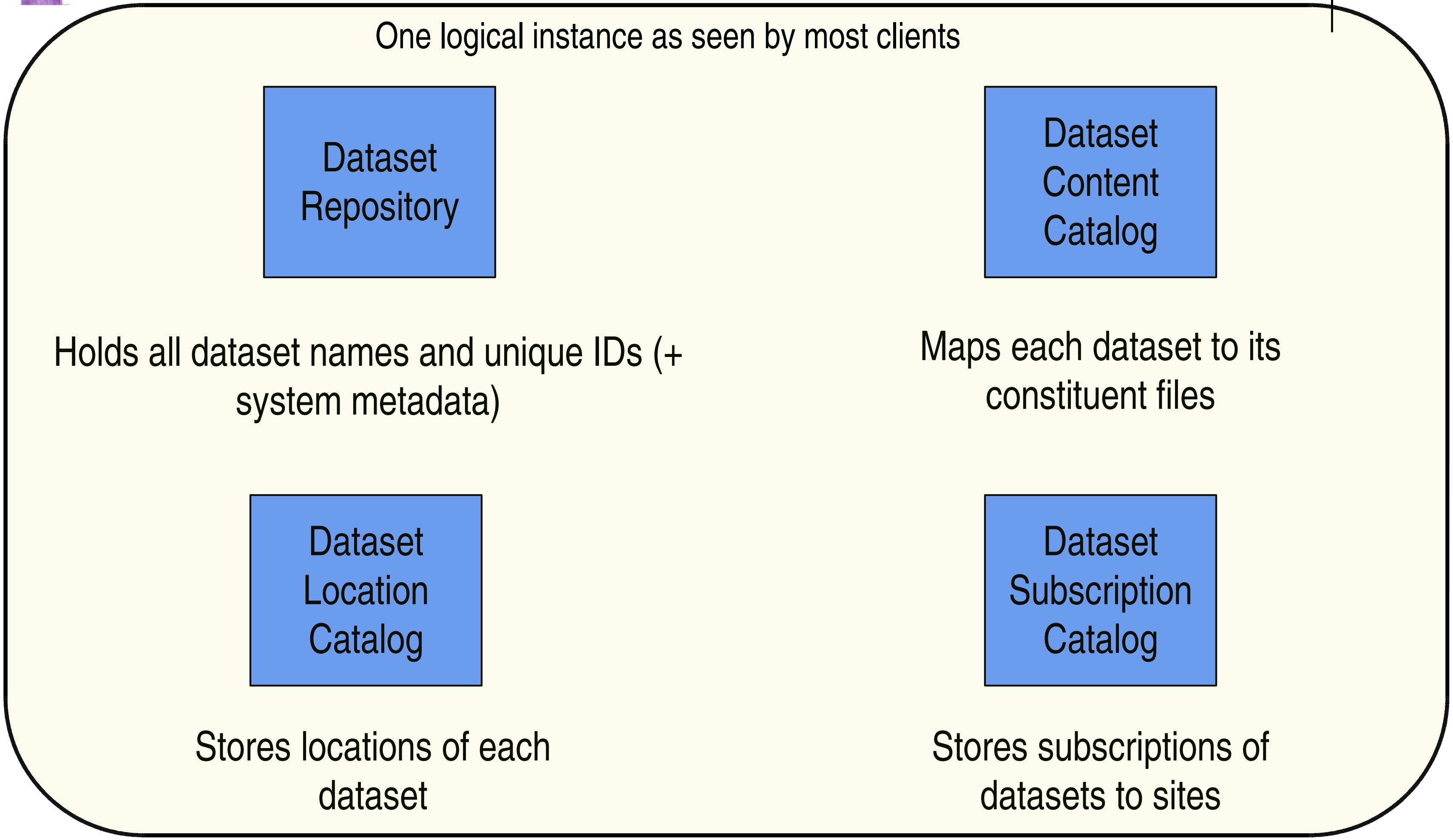
The Need for ATLAS Data Management

- Grids provide a set of tools to manage distributed data
  - These are low-level file cataloging, storage and transfer services
- ATLAS uses three Grids (LCG, OSG, NG), each having their own versions of these services
- Therefore there needs to be an ATLAS specific layer on top of the Grid middleware
  - To bookkeep and present data in a form physicists expect
  - To manage data flow as described in the computing model and provide a single entry point to all distributed ATLAS data

Don Quijote 2

- Our software is called Don Quijote 2 (DQ2)
  - We try to leave as much as we can to Grid middleware
- We base DQ2 on the concept of versioned datasets
  - Defined as a collection of files or other datasets
  - eg RAW data files from a particular detector run
- We have ATLAS central catalogs which define datasets and their locations
  - A dataset is also the unit of data movement
- To enable data movement we have a set of distributed ‘site services’ which use a subscription mechanism to pull data to a site
  - As content is added to a dataset, the site services copy it to subscribed sites
- Tools also exist for users to access this data

Central Catalogs



One logical instance as seen by most clients

Dataset Repository

Holds all dataset names and unique IDs (+ system metadata)

Dataset Content Catalog

Maps each dataset to its constituent files

Dataset Location Catalog

Stores locations of each dataset

Dataset Subscription Catalog

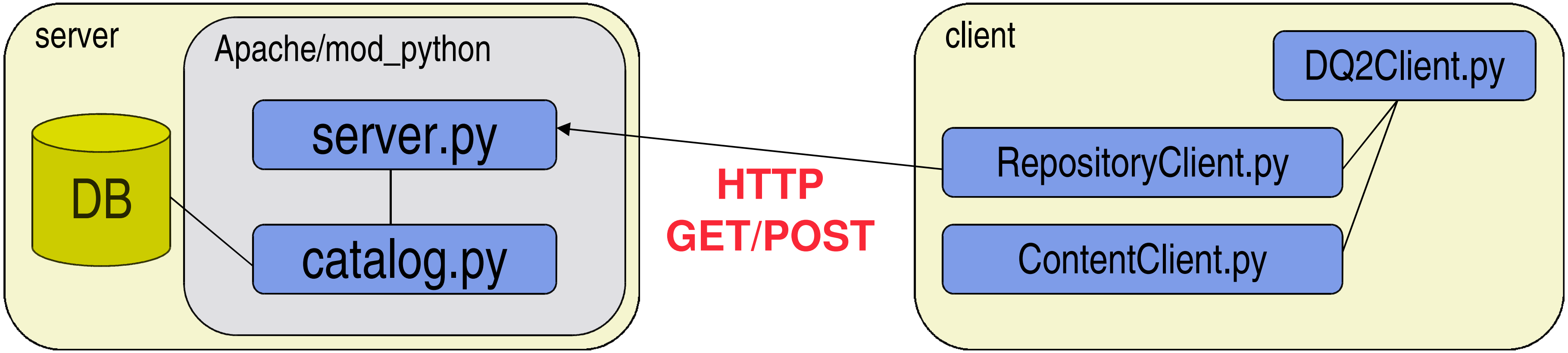
Stores subscriptions of datasets to sites

Central Catalogs

- There is no global physical file replica catalog
  - > 100k files and replicas created every day
  - Physical file resolution is done by (Grid specific) catalogs at each site holding only data on that site
- The central catalogs are split (different databases) because we expect different access patterns on each one
  - For example the content catalog will be **very** heavily used
- The catalogs are logically centralised but may be physically separated or partitioned for performance reasons
- A unified client interface ensures consistency between catalogs when multiple catalog operations are performed

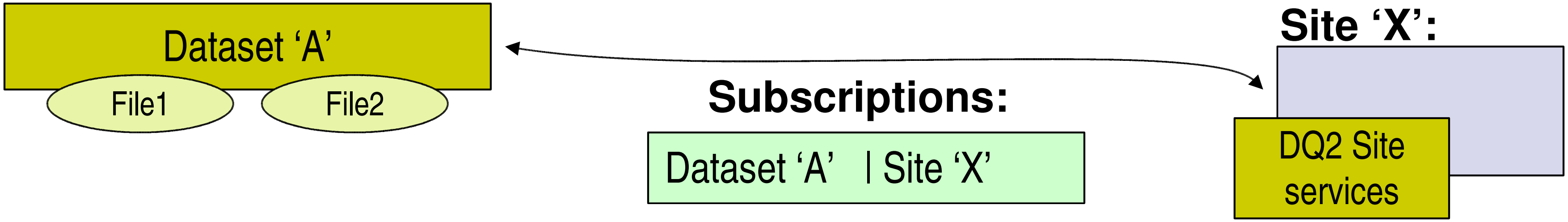
Implementation

- The clients and servers are written in python and communicate using REST-style HTTP calls (no SOAP)
- Servers hosted in Apache using mod_python
- Using mod_gridsite for security and MySQL or Oracle databases as a backend



Site Services

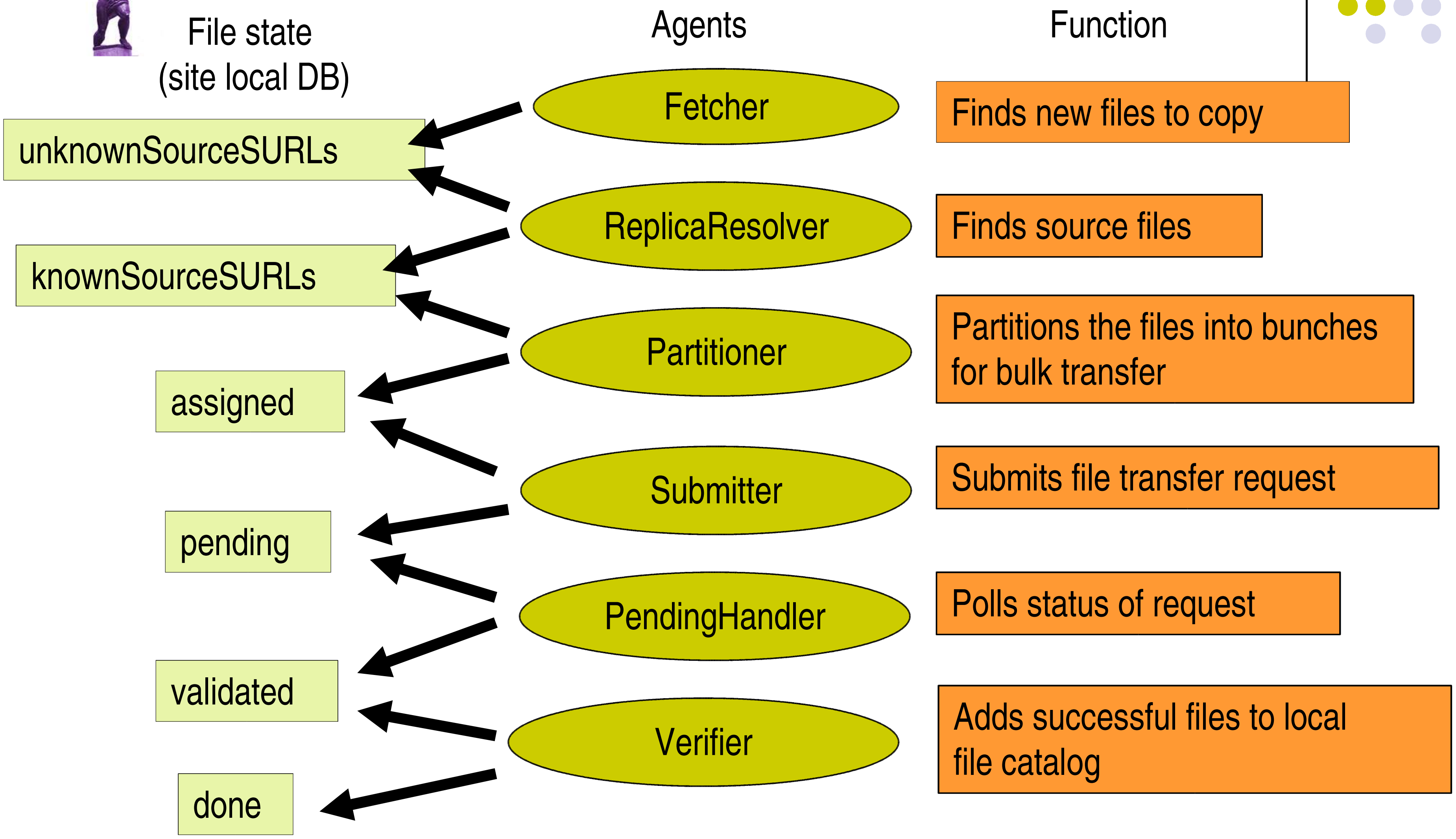
- DQ2 site services are also written in python and pull data to the sites that they serve
- The subscription catalog is queried periodically for any dataset subscriptions to the site
- The site services then copy any new data in the dataset and register it in their site's replica catalog



Site Services

- Site services are located on so-called VOBOXes
  - On LCG and NG, there is one VOBOX per Tier 1 site and the site services here serve the associated Tier 2 sites
  - On OSG, there is one VOBOX per Tier 1 site and one per Tier 2 site
- The site services work as a state machine
- A set of agents pick up requests and process from one state to the next state
- A local database on the VOBOX stores the files' states
  - With the advantage that this database can be lost and recreated from central and local catalog information

Site Services Workflow



File state (site local DB)

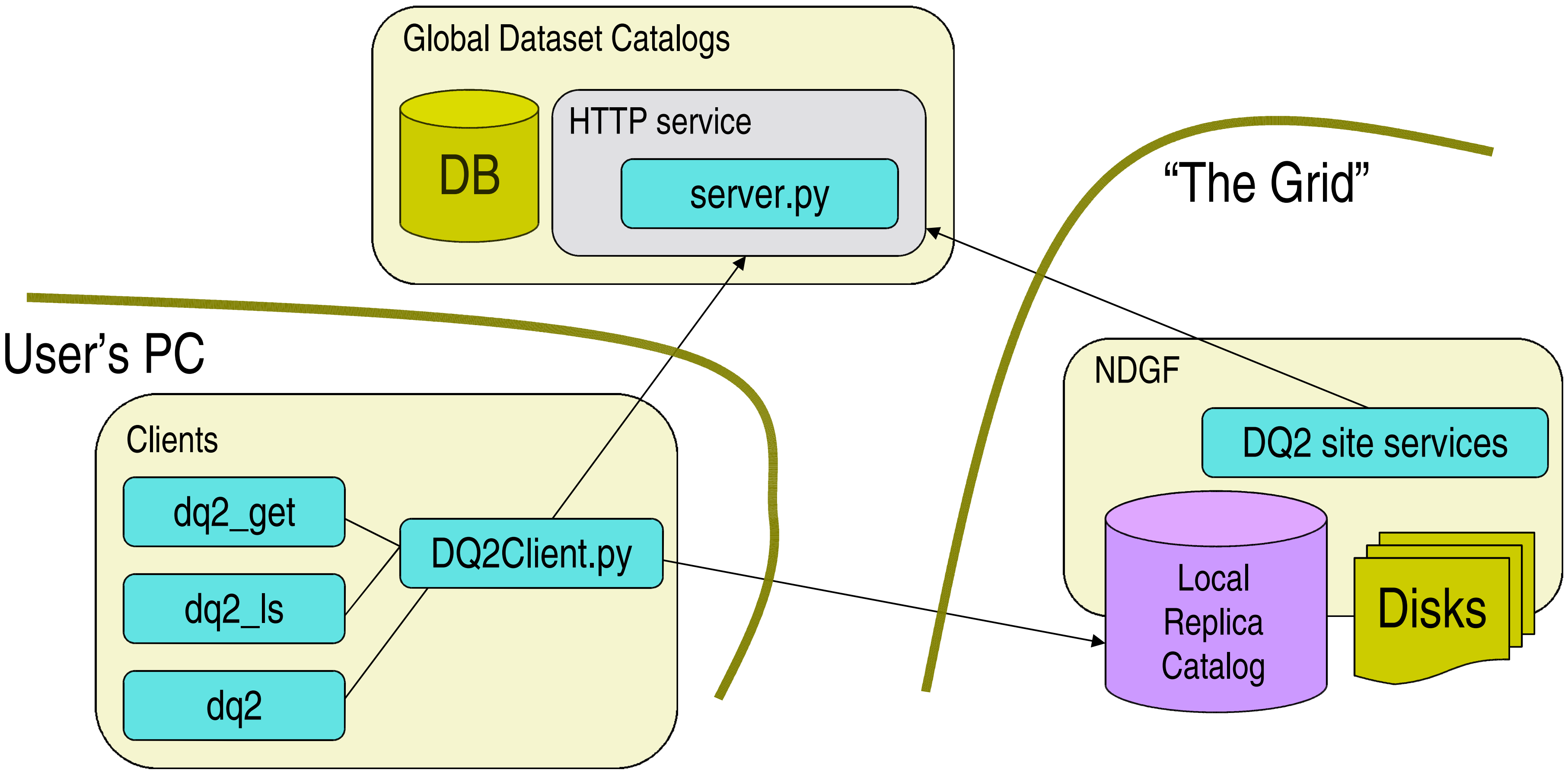
- unknownSourceSURLs
- knownSourceSURLs
- assigned
- pending
- validated
- done

Agents	Function
Fetcher	Finds new files to copy
ReplicaResolver	Finds source files
Partitioner	Partitions the files into bunches for bulk transfer
Submitter	Submits file transfer request
PendingHandler	Polls status of request
Verifier	Adds successful files to local file catalog

External Components
(or where you get lost in acronyms…)

- DQ2 uses several Grid middleware components, some of which are Grid specific
- Replica Catalogs:
  - These map logical file names and GUIDs to physical files
  - LCG has the LFC deployed at each Tier 1 site
  - OSG has the MySQL LRC deployed at all sites
  - NG has a single Globus RLS and LRC (more later..)
- File Transfer:
  - Uses gLite FTS, one server per Tier 1 site
- Storage services:
  - SRM and GridFTP (in NG) services provide Grid access to physical files on disk and tape

DQ2


Global Dataset Catalogs
DB
HTTP service
server.py
"The Grid"
User's PC
NDGF
DQ2 site services
Clients
dq2_get
dq2_ls
dq2
DQ2Client.py
Local Replica Catalog
Disks

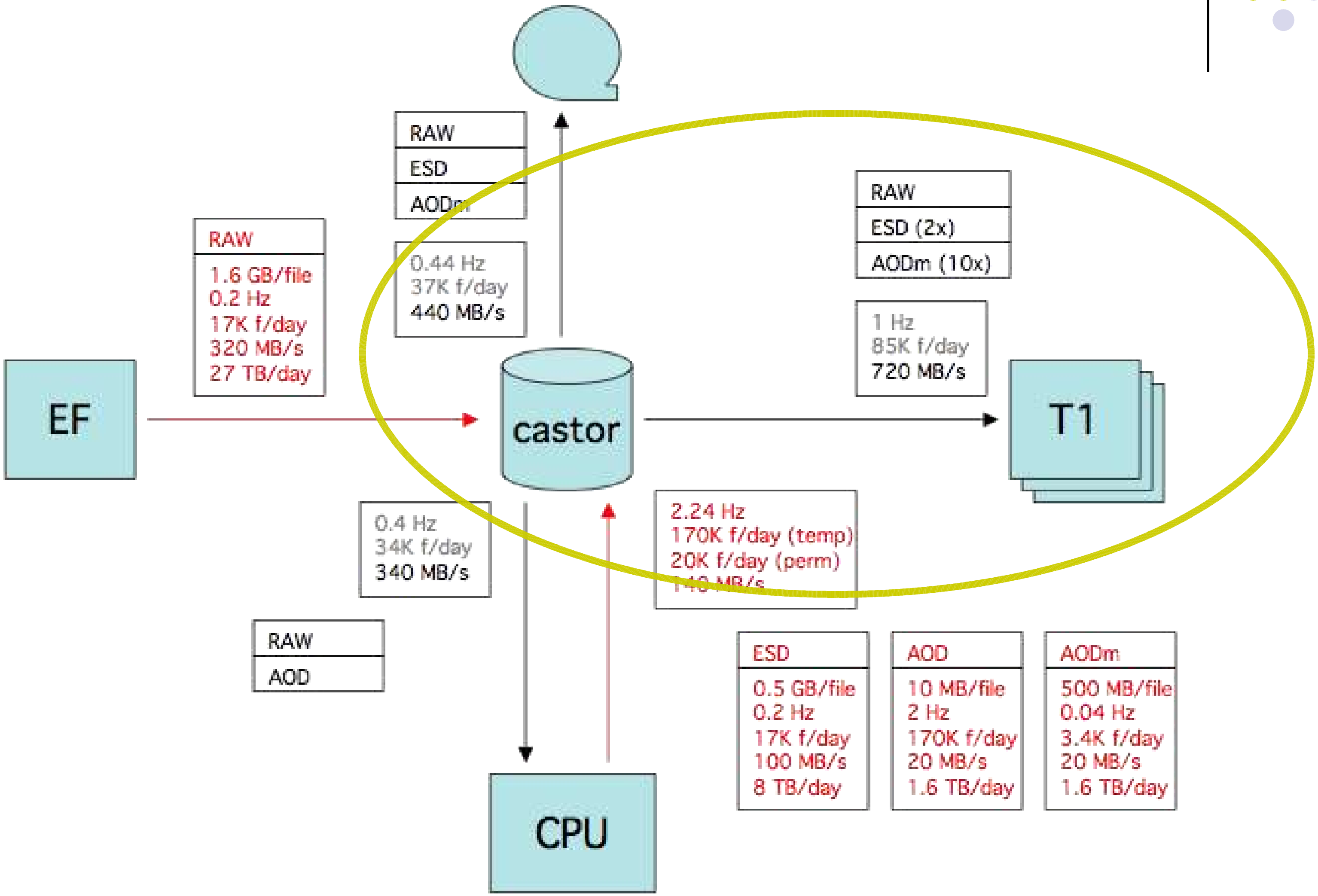
Using DQ2

- DQ2 is the mechanism by which all ATLAS data should move
- Uses cases DQ2 serves
  - Tier 0 data
    - Data from the detector is processed at CERN and shipped out to Tier 1 and Tier 2 sites
  - MC production
    - Simulation of events is done at Tier 1 and Tier 2 sites
    - Output datasets are aggregated at a Tier 1 centre
  - Local access to Grid data for end-users eg for analysis
    - Client tools enable physicists to access data from Grid jobs and to copy datasets from the Grid to local PCs
  - Reprocessing
    - T1 - T1 data movement and data recall from tape (this is the only part not tested fully)

Tier 0 exercise

- The Tier 0 exercise has been the biggest and most important test of DQ2
- This is a scaled down version of the data movement out from CERN when the experiment starts
- Fake events are generated at CERN, reconstructed at CERN and the data is shipped out to Tier 1 centres
  - Some Tier 2 sites also take part in the exercise
- Initially this was run as part of the LCG Service Challenges
  - Now it is constantly running until real data arrives
- The nominal rate for ATLAS data out of CERN is around 1GB/s split (not evenly) between 10 Tier 1 sites
  - And 20MB/s split among each Tier 1 site’s associated Tier 2 sites

Tier 0 data flow (full operational rates)



EF

RAW
1.6 GB/file
0.2 Hz
17K f/day
320 MB/s
27 TB/day

RAW
ESD
AODm

0.44 Hz
37K f/day
440 MB/s

castor

RAW
ESD (2x)
AODm (10x)

1 Hz
85K f/day
720 MB/s

T1

0.4 Hz
34K f/day
340 MB/s

RAW
AOD

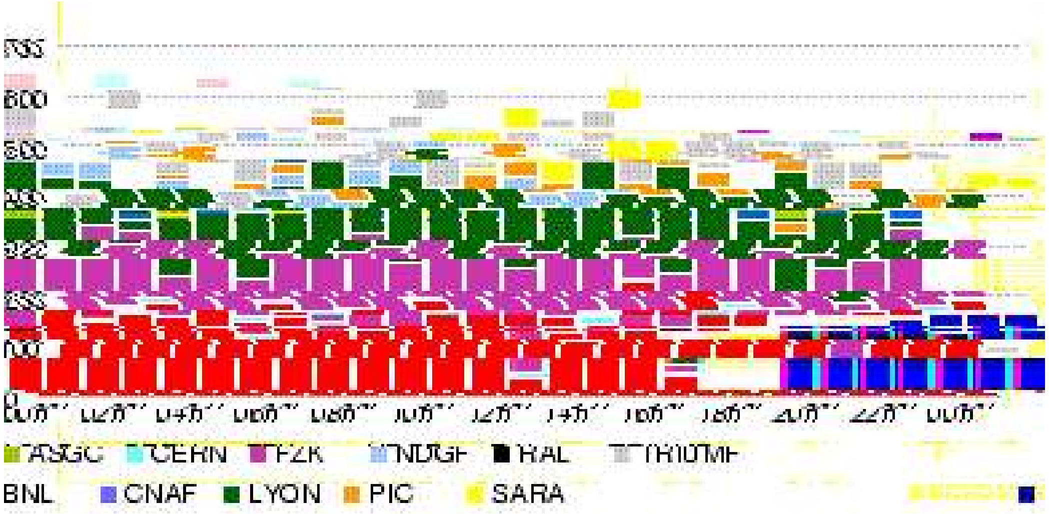
2.24 Hz
170K f/day (temp)
20K f/day (perm)
140 MB/s

CPU

ESD	AOD	AODm
0.5 GB/file 0.2 Hz 17K f/day 100 MB/s 8 TB/day	10 MB/file 2 Hz 170K f/day 20 MB/s 1.6 TB/day	500 MB/file 0.04 Hz 3.4K f/day 20 MB/s 1.6 TB/day

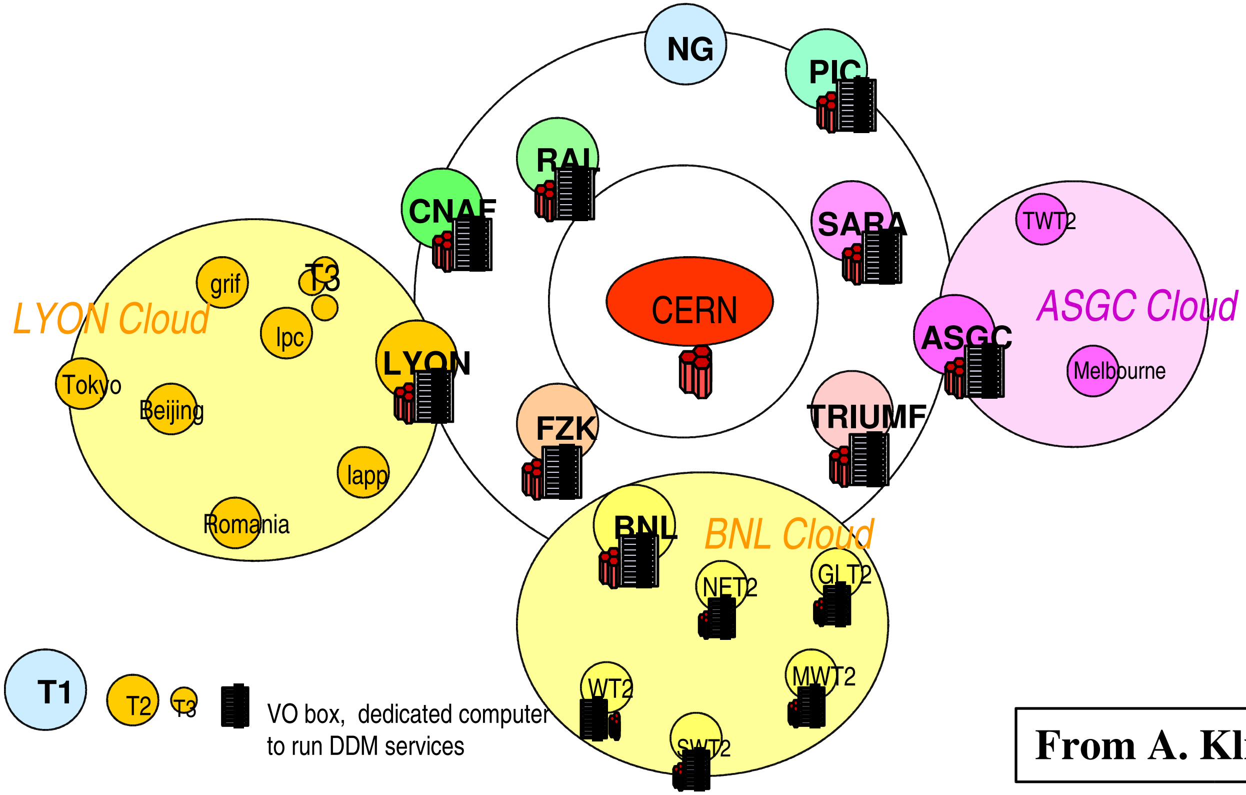
Results from the Tier 0 exercise

- We have reached the nominal rate to most Tier 1 sites (including NDGF T1), but not all of them at the same time
- Running at the full rate to all sites for a sustained period of time has proved difficult to achieve
  - This is mainly due to unreliability of T1 sites storage and limitations of CERN castor
- Throughput on a random good day (25 May):



MC Production and DQ2

- The model for MC production let to the idea of the cloud model



From A. Klimentov

Tiers Of ATLAS - DQ2’s info system

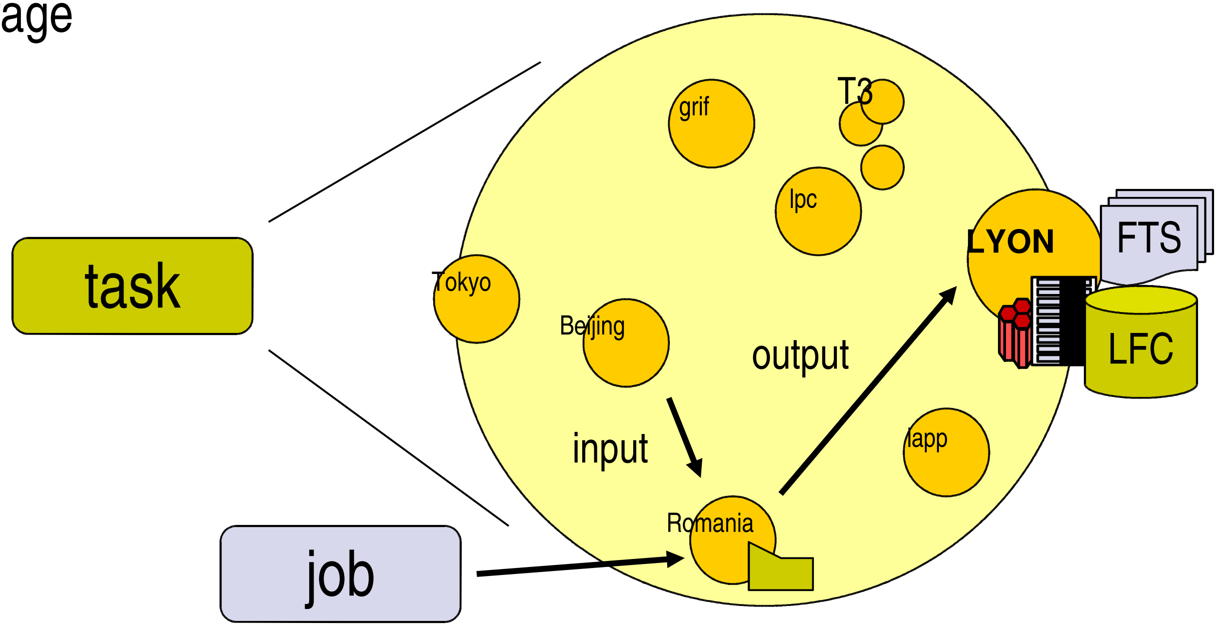
- Tiers of ATLAS is the ATLAS data management information system which defines the ‘clouds’
- It imposes the ATLAS hierarchy of tiers on the Grid(s)
- Idea of disk/tape sites
- It also contains the storage endpoints and replica catalogs

TiersOfATLASCache.py

```
…
   'NDGFT1DISK':
   {
      'domain': 'srm://srm.ndgf.org.*/pnfs/ndgf.org/data/atlas/disk.*',
      'email': 'adrian.taga@fys.uio.no',
      'toolAssigner': 'lcg',
      'fts': NDGFFTS,
      'srm': 'srm://srm.ndgf.org/pnfs/ndgf.org/data/atlas/disk/',
      'srmsc4': 'srm://srm.ndgf.org/pnfs/ndgf.org/data/atlas/tape/t0test_disk/',
      'ce': [ '' ],
      'alternateName' : [ 'NDGF-T1' ],
   },
   'NDGFT1TAPE':
   {
      'domain': 'srm://srm.ndgf.org.*/pnfs/ndgf.org/data/atlas/tape.*',
      'email': 'adrian.taga@fys.uio.no',
      'toolAssigner': 'lcg',
      'fts': NDGFFTS,
      'srm': 'srm://srm.ndgf.org/pnfs/ndgf.org/data/atlas/tape/',
      'srmsc4': 'srm://srm.ndgf.org/pnfs/ndgf.org/data/atlas/tape/t0test_tape/',
      'ce': [ '' ],
      'alternateName' : [ 'NDGF-T1' ],
   },
….
```

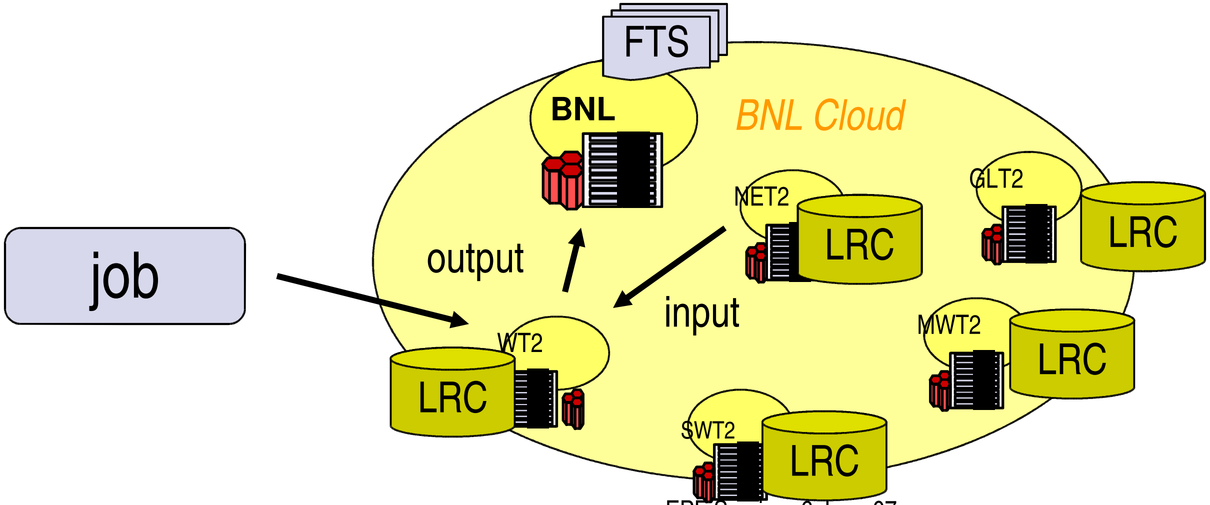
MC Production and DQ2 (LCG)

- A task is assigned to a cloud
  - Jobs run at T1 or T2 sites and copy their output to the local SE (or other SE in the cloud in case of failure)
- A DQ2 subscription is made to gather the dataset at the Tier 1 site for permanent storage



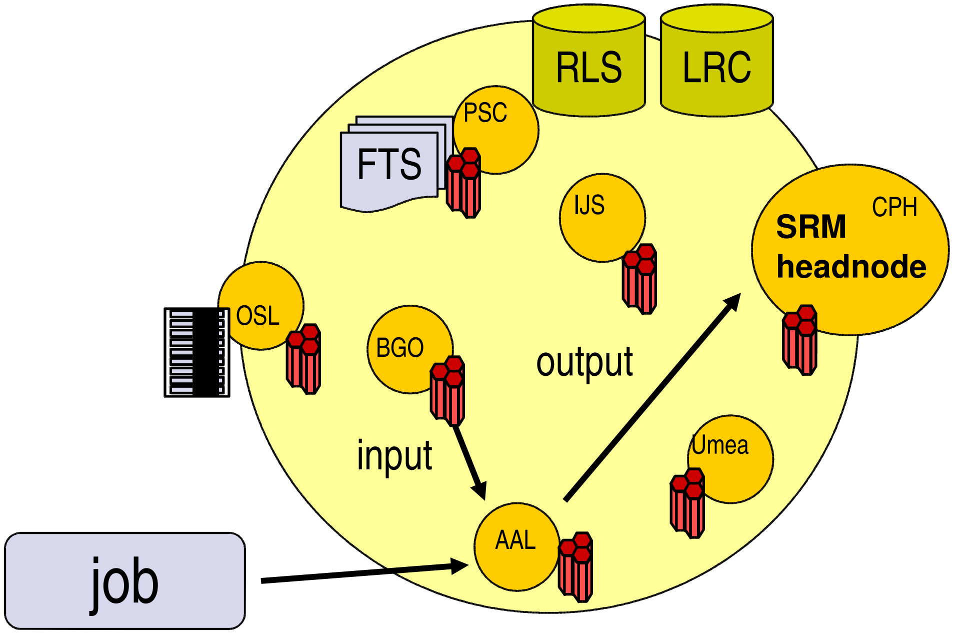
MC Production and DQ2 (OSG)

- OSG runs a different production system called PANDA
- It uses a model of 'pilot jobs'
- Pilot jobs subscribe input data to the site using DQ2
  - When complete DQ2 sends a callback to the central PANDA server to release the job
- Output is written locally then subscribed to BNL



MC Production and DQ2 (NG)

- On NG workload and data management is controlled by the ARC middleware
  - No data aware scheduling
- Data is read from and written directly to the T1 (distributed) SRM

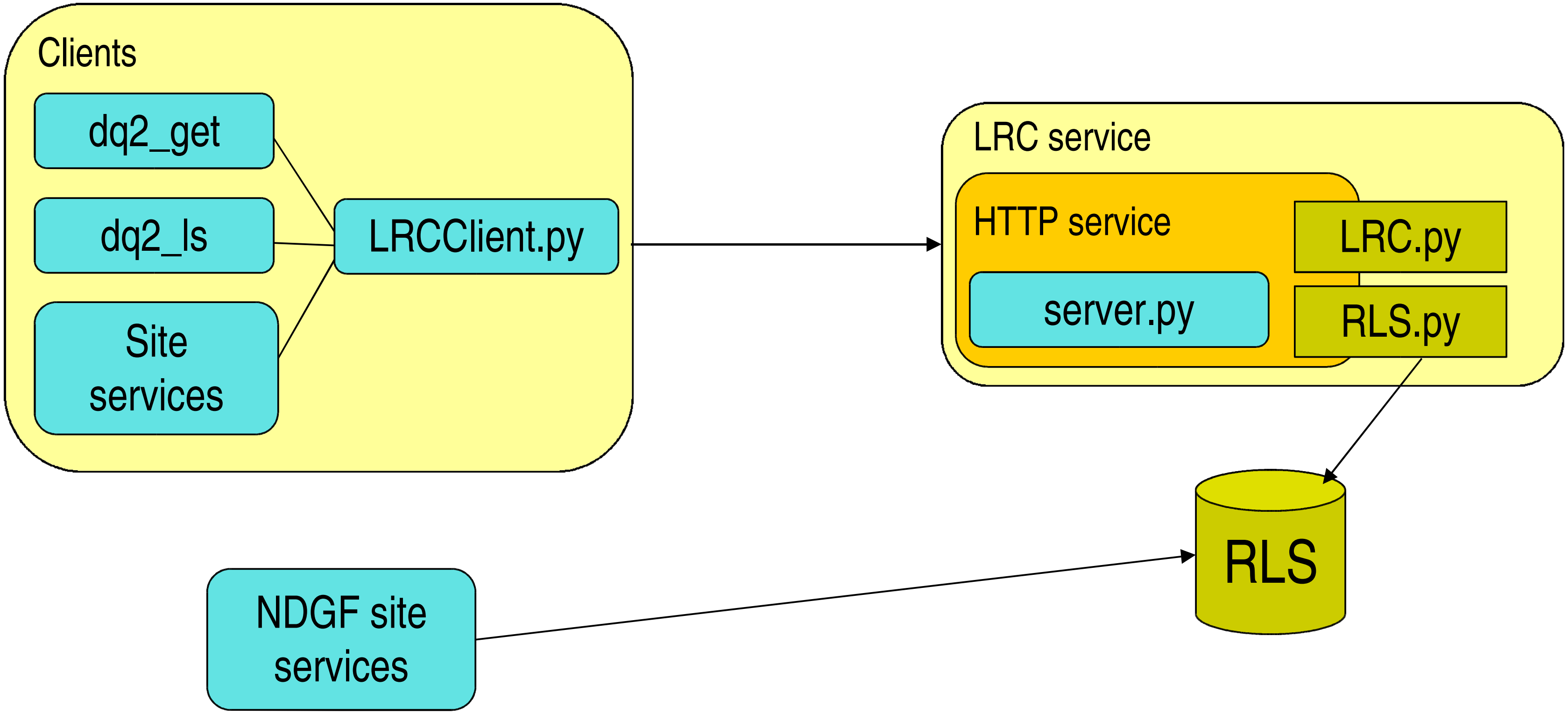


DQ2, RLS and LRC

- The current production version of DQ2 cannot read RLS
- As a temporary measure an LRC was set up for DQ2 to use
  - This is kept sychronised with the RLS
- It is more desirable for DQ2 to use RLS
  - Without RLS client dependencies
- Therefore we should create a front end service to RLS
- The DQ2 team are already writing a webservice for the LRC
  - We can use this and plug in an RLS backend
  - This service does not use GSI security and is for **query** only
    - The only dependency to read RLS is curl
    - RLS is only modified by NDGF site services which can use direct RLS access
  - For clients it is transparent whether the catalog behind the service is LRC or RLS

DQ2, RLS and LRC

All clients can query via the web service
The NDGF site services use direct access



Conclusions and issues to address

- The DQ2 architecture (datasets, central catalogs, site services) makes the data flow of the ATLAS computing model manageable
- There is still some way to go before we can move data around at the nominal rate for long periods of time
  - In general these are not problems with DQ2
- But we believe the system can handle the requirements of the model
- Issues to solve
  - Consistency between the layers (DQ2, file catalogs, storage elements)
  - The Tiers of ATLAS information system is not integrated with any Grid information system
  - User/group quotas and integration to SRM level
  - Deleting data - difficult when files can cross datasets
  - NG specific:
    - Scheduling jobs to data - how do we know where the data is?
    - Tier 2s and Tier 3s
    - Production system, SRM and storage service
  - Many more… see the DQ2 savannah pages tasks and bugs:
    - https://savannah.cern.ch/projects/atlas-ddm/

Links and monitoring

- DDM wiki page
  - https://twiki.cern.ch/twiki/bin/view/Atlas/DistributedDataManagement
- Development Savannah page
  - https://savannah.cern.ch/projects/atlas-ddm/
- The ATLAS dashboard receives its data from callbacks from the DQ2 site services
  - http://dashb-atlas-data.cern.ch/dashboard/request.py/site
- Dashboard for v0.3 (running T0 tests)
  - http://dashb-atlas-data-test.cern.ch/dashboard/request.py/site
- GridVIEW is LCG’s throughput monitoring system
  - http://gridview.cern.ch/GRIDVIEW/
- Internal NDGF monitoring of the T1
  - http://wiki.ndgf.org/index.php/Operation:Monitoring
- …

Nå skal vi spise kake!!