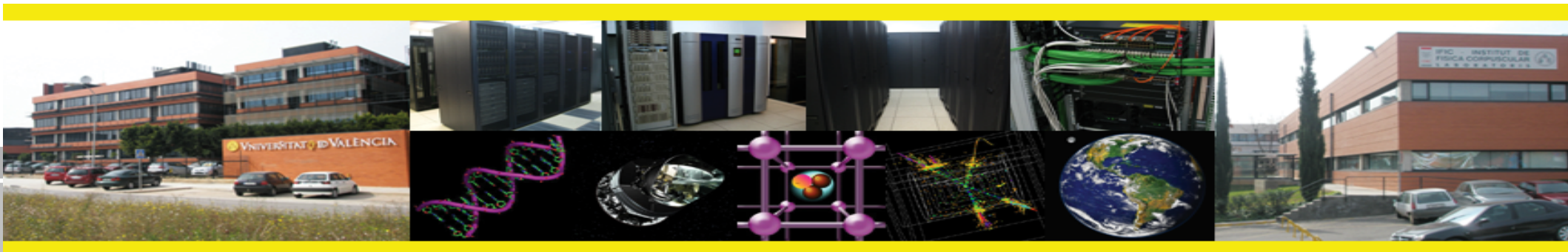
Applications in HEP: ATLAS

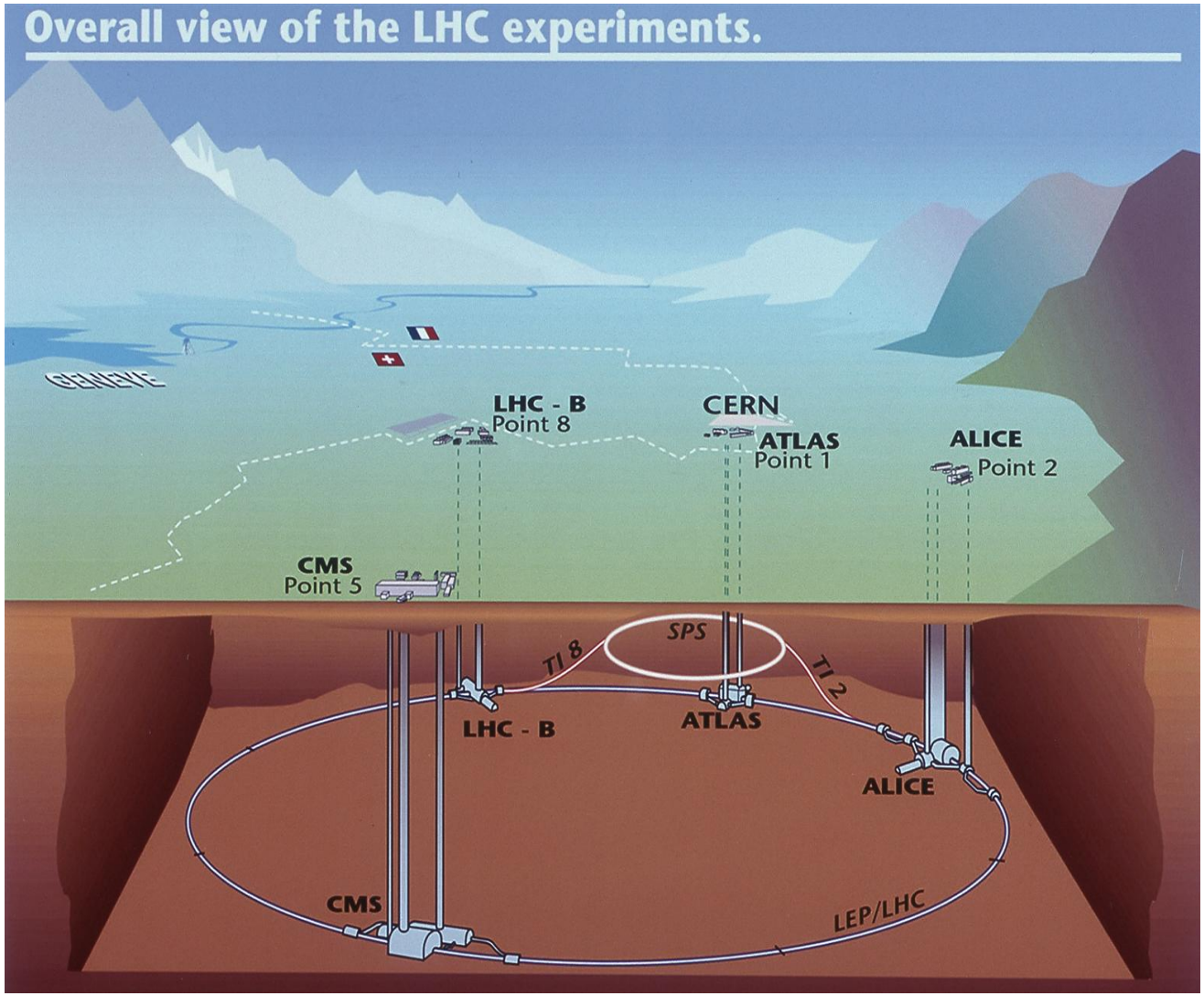
Elena Oliver, Santiago González, Miguel Villaplana



Index

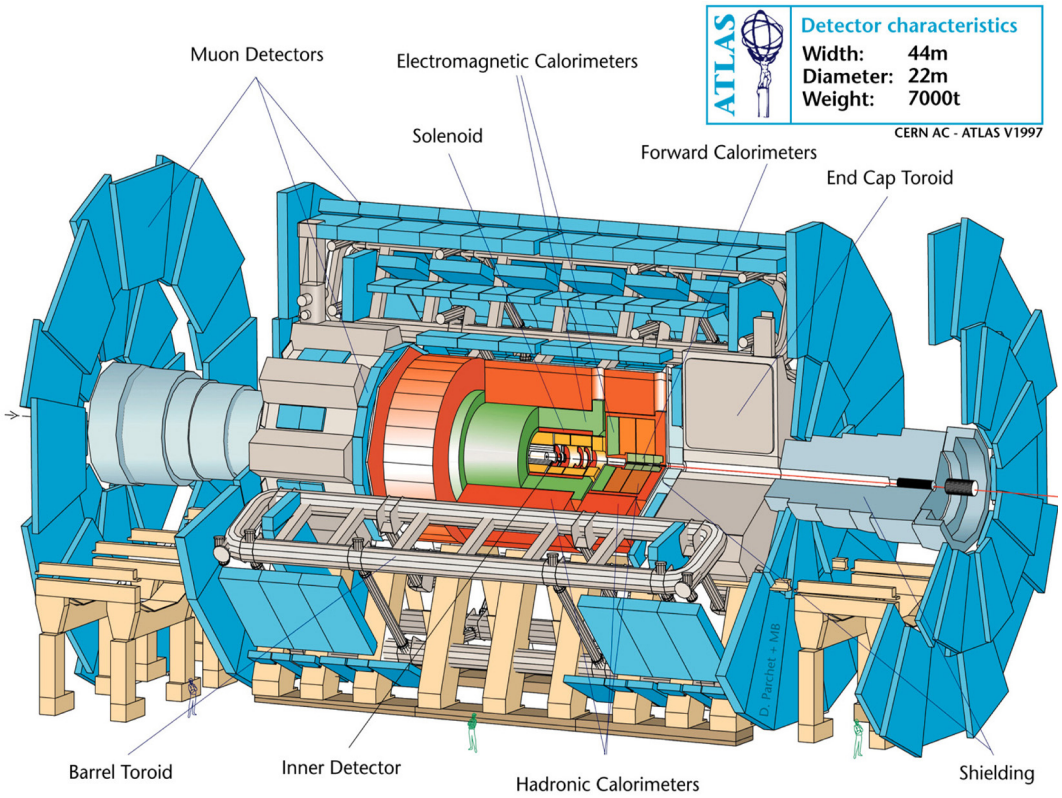
- LHC
- ATLAS
- ATLAS computing challenges
- Data Model
- Computing Model
- ATLAS Monte Carlo Production
- Athena
- ATLAS Space tokens
- Distributed Data Management (DDM)
- Distributed Analysis Tools
- Ganga Job Schema
- ATLAS Distributed Analysis Demo

Large Hadron Collider (LHC)



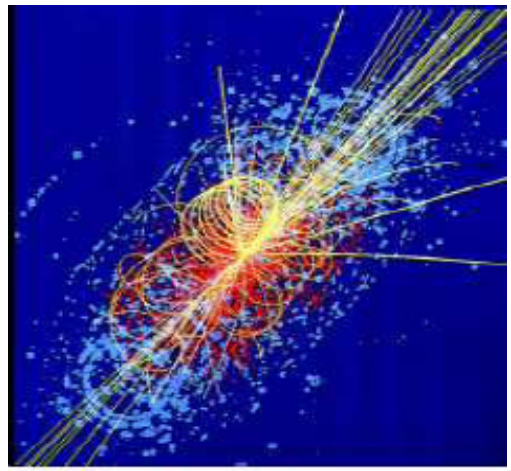
- Particle accelerator
- Between Switzerland and France
- 27 km ring of superconducting magnets
- 100 m underground
- Two beams of protons crashing at 7 TeV (3.5 + 3.5).
- Highest luminosity = 5·1029 cm-2s-1
- Total Collisions (at 7 TeV) = 1550 million
- At -271°C
- Experiments: LHCb, ALICE, CMS and ATLAS

A Toroidal LHC ApparatuS (ATLAS)



- Diameter 25 m
- Lenght 46 m
- 7000 tonnes
- 3000 Km of cables

- Searching:
  - Higgs Boson
  - Dark Matter
  - Extra Dimensions
  - .....
- Collision proton-proton at ATLAS
  - Many particles!! So, we need to reprocess the information.



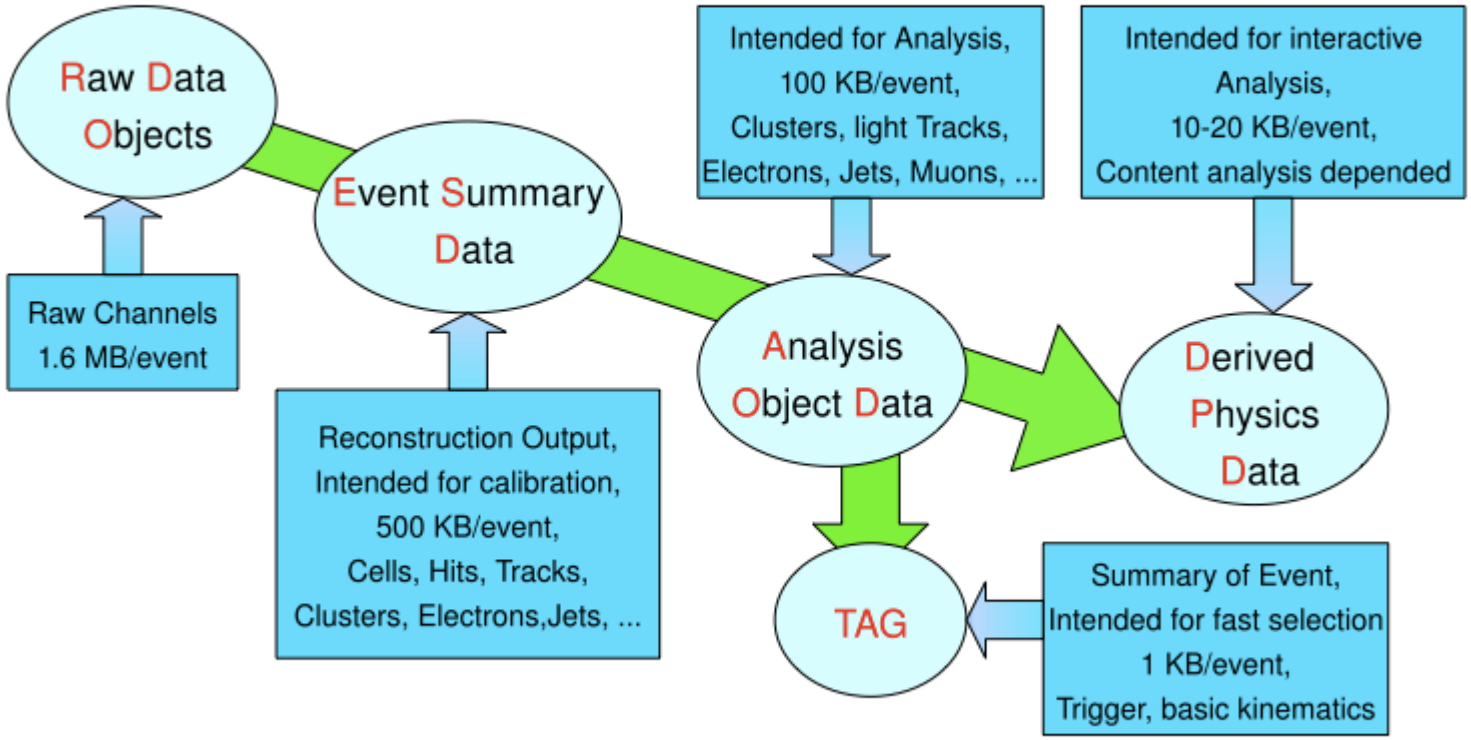
ATLAS computing challenges

- Data access: 3000 scientists around the world have to access to that data.
- Data storage: 3200 TeraBytes per year to be stored = 7Km of CR-ROMs or listening songs for 600 years

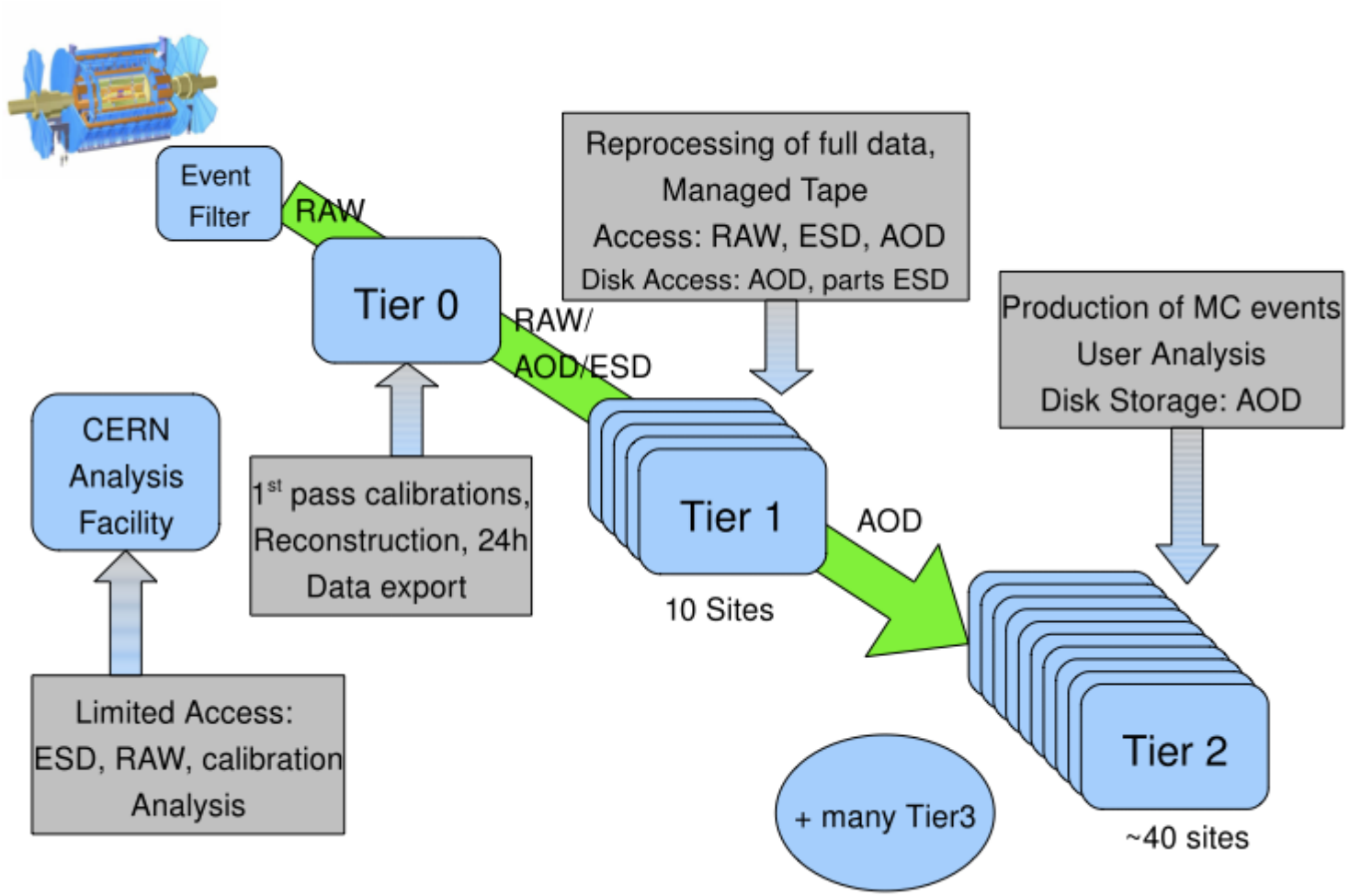
- 5 million lines of code for event analysis
- Data computation: To process that huge quantites of data (3000 online and 100000 offline PCs to processing and analysis).

Data Model

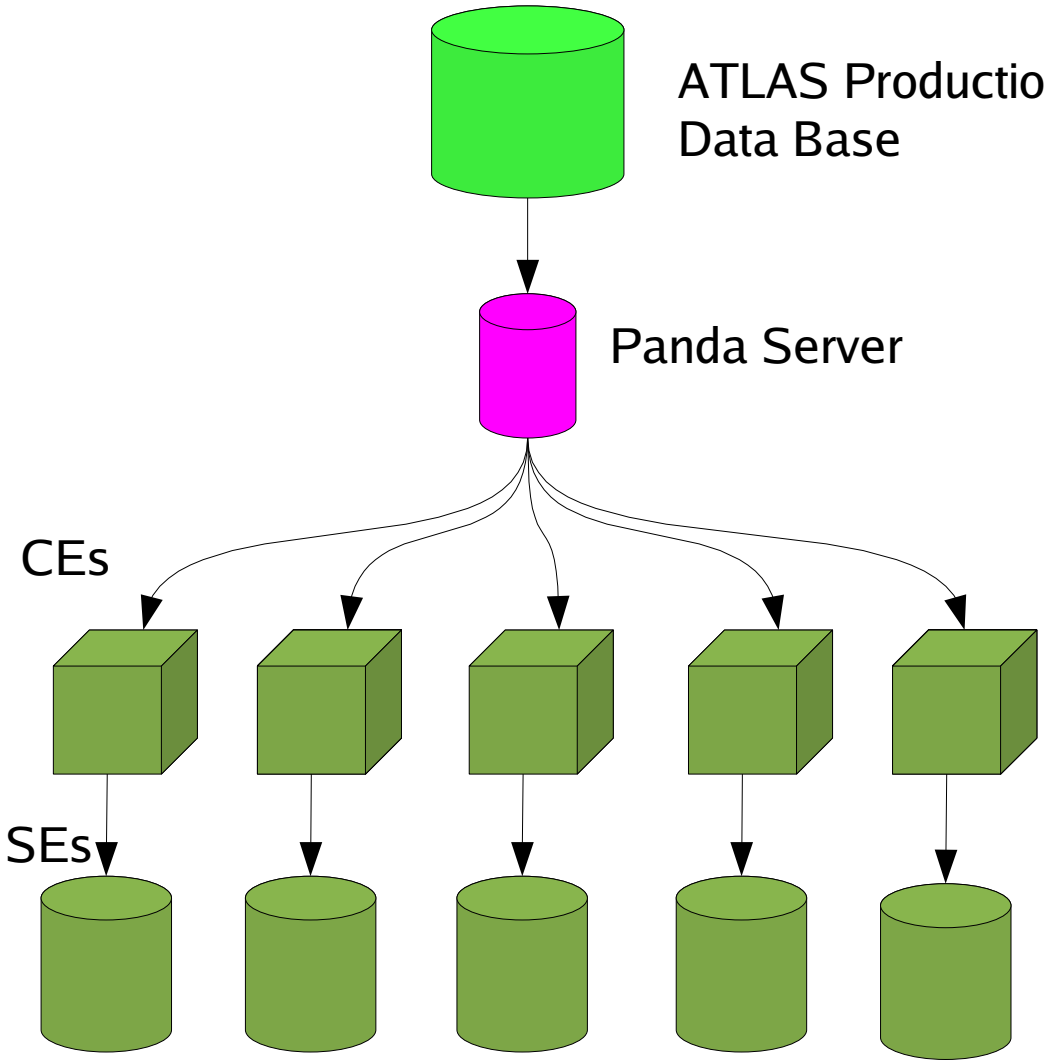
Refining the data by: Add higher level info, Skin, Thin, Slim

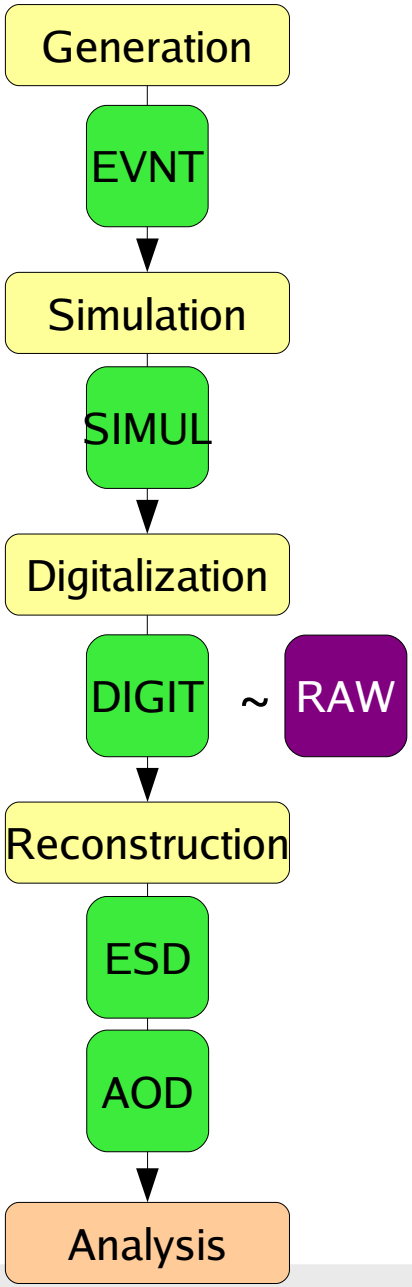


Computing Model



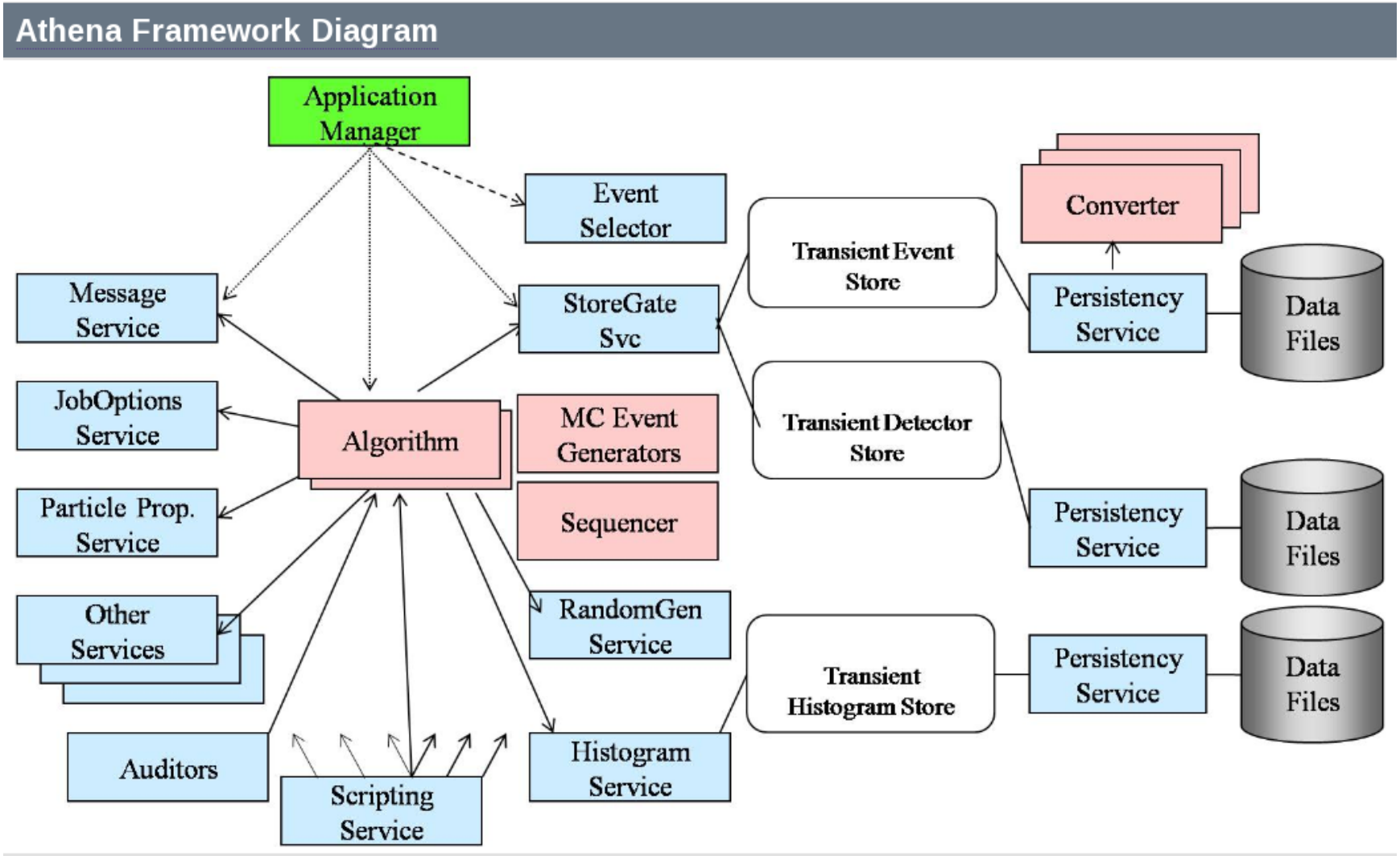
ATLAS Monte Carlo Production





Athena

- Athena is the framework we use to control the execution flow of our applications (HLT, Simulation, Reconstruction, and Analysis).



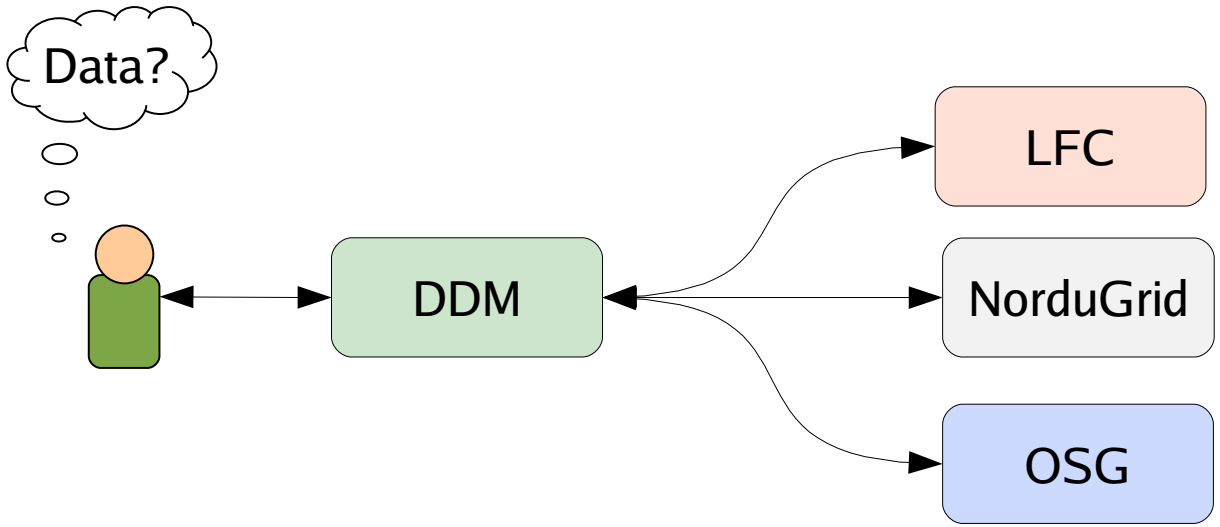
ATLAS space tokens

- Tokens are a mechanism to split storage into virtual areas. The way to distribute the data in the ATLAS Tiers. Table with space tokens at Tier0, 1, 2, 3

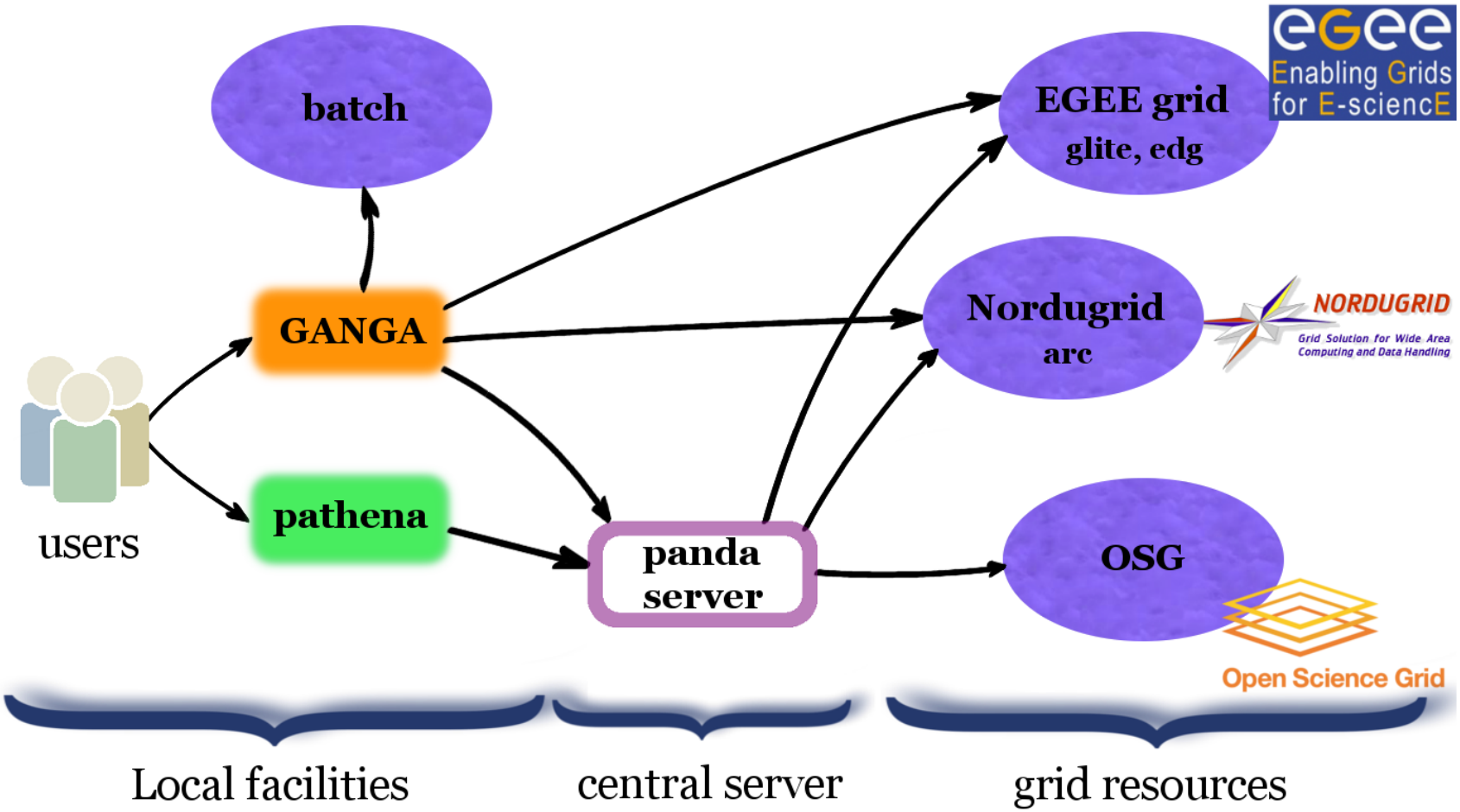
ATLAS Space Tokens for Data Storage DDM aware sites					
Space Token	Used For	@T0	@T1	@T2	@T3
ATLASDATATAPE	RAW from T0, AOD from re-processing	X	X		
ATLASDATADISK	AOD + data on demand	X	X	X	
ATLASMCTAPE	Custodial copy of MC files		X		
ATLASMCDISK	AOD + MC on demand	X	X	X	
ATLASHOTDISK	Files (if needed) replicated within same SE	X	X	X	X
ATLASPRODDISK	buffer for central production			X	X
ATLASGROUPDISK	Data managed by groups	X	X	X	
ATLASUSERDISK	Temp. user Data (only US)	X	X	X	
ATLASSCRATCHDISK	Temp. user Data	X	X	X	X
ATLASLOCALGROUPDISK	Local User Data				X

Distributed Data Management (DDM)

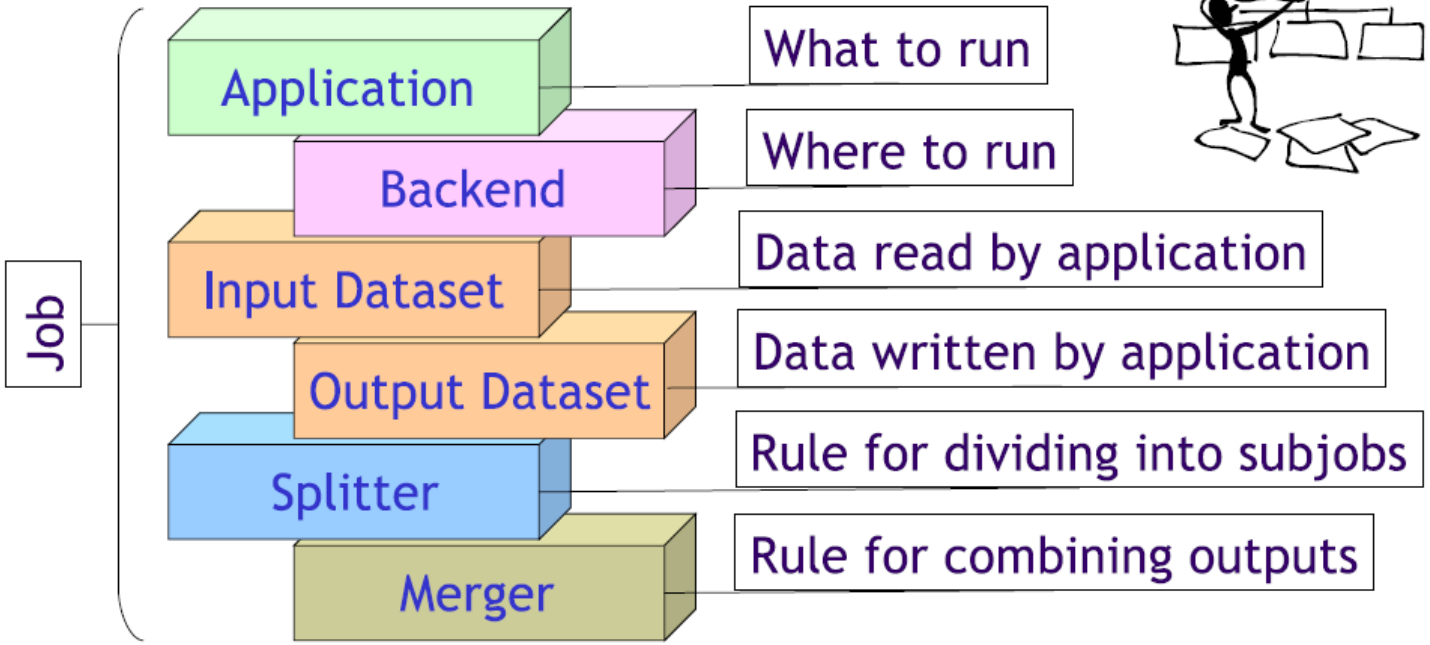
- The **DDM** group is involved in developing a system to manage access to ATLAS data that is distributed at sites **all around the world**.
- The system consists of a **bookkeeping** system (dataset-based) and a set of local site services to handle data transfers, building upon Grid technologies. The software stack is called **DQ2**.
- A **dataset** is a versioned **collection of grid files**, logical entity with a **unique** DSN (Dataset Name), can be **replicated** (create an exact copy) to another grid site.

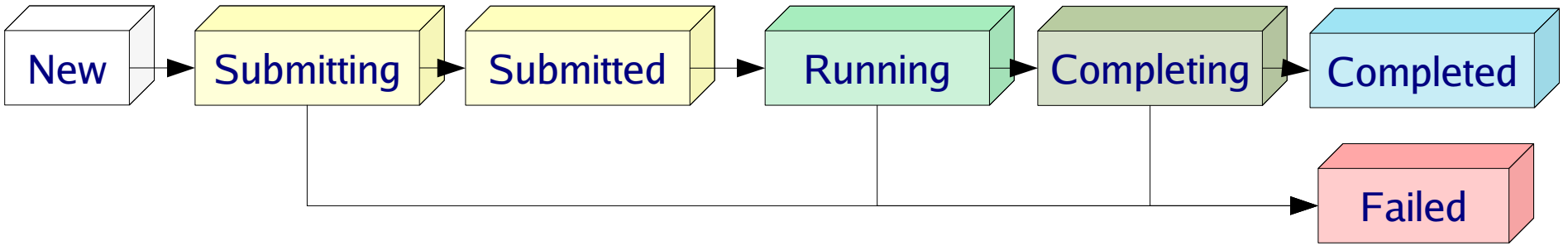


Distributed Analysis Tools



Ganga Job Schema





ATLAS Distributed Analysis Demo

DDM: DQ2 examples

- ATLAS Data Management
- Searching dataset

  dq2-ls 'data10_7TeV*merge*AOD*'
- See their files

  dq2-ls -f 'data10_7TeV.00153159.physics_L1CaloEM.merge.AOD.r1297_p161_tid143468_00'
- See the sites

  dq2-ls -r 'data10_7TeV.00153159.physics_L1CaloEM.merge.AOD.r1297_p161_tid143468_00'
- Get a file from the dataset

  dq2-get -n 1 'data10_7TeV.00153159.physics_L1CaloEM.merge.AOD.r1297_p161_tid143468_00'

Grid job example

- Submit jobs with Ganga
- The simplest GRID job, only 3 lines:

```
j=Job()
j.backend=LCG()
j.submit()
```

- Copy the job, see other GRID destinations:

```
j2=j.copy()
j2.submit()
```

- Web page to see all GRID sites via googlemaps:

  BDII Locations

Analysis job example

- Python script for Athena Analysis on GRID:

```
j = Job()
j.application = Athena()
j.application.option_file = 'AnalysisSkeleton_topOptions.py'
j.application.max_events = -1
j.application.athena_compile = False
j.application.prepare()
j.splitter = DQ2JobSplitter()
j.splitter.numfiles=1
j.inputdata = DQ2Dataset()
j.inputdata.dataset="mc08.105200.T1_McAtNlo_Jimmy.merge.AOD.e357_s462_d150_r642_t53_tid
    105572_00"
j.outputdata=DQ2OutputDataset()
j.outputdata.outputdata=['AnalysisSkeleton.aan.root']
j.backend=LCG()
j.backend.requirements=AtlasLCGRequirements()
j.backend.requirements.cloud = 'ALL'
j.submit()
```

Get Analysis Output Files

- See output dataset name:

  jobs(19).outputdata.datasetname

  !dq2-ls -r $jobs(19).outputdata.datasetname

- Get the output:

  !cd /tmp/oliverg

  !dq2-get $jobs(19).outputdata.datasetname

Analysis job example (II)

- Python script for Athena Analysis to GRID with Panda backend:

```
j = Job()
j.application = Athena()
j.application.option_file = 'AnalysisSkeleton_topOptions.py'
j.application.max_events = -1
j.application.athena_compile = False
j.application.prepare()
j.splitter = DQ2JobSplitter()
j.splitter.numfiles=1
j.inputdata = DQ2Dataset()
j.inputdata.dataset="mc08.105200.T1_McAtNlo_Jimmy.merge.AOD.e357_s462_d150_r642_t53_tid
    105572_00"
j.outputdata=DQ2OutputDataset()
j.backend=Panda()
j.backend.requirements= PandaRequirements()
j.backend.requirements.cloud='US'
j.submit()
```

- Web Panda monitor to see job status:

  My Panda job

References

- LHC:
  - http://public.web.cern.ch/public/en/LHC/LHC-en.html
- ATLAS:
  - http://atlas.ch/
- Monte Carlo Production:
  - https://twiki.cern.ch/twiki/bin/view/Atlas/WorkBookFullChain
- Spacetokens:
  - https://twiki.cern.ch/twiki/bin/view/Atlas/StorageSetUp
- Athena:
  - https://twiki.cern.ch/twiki/bin/view/Atlas/CoreSoftware
- DDM:
  - https://twiki.cern.ch/twiki/bin/view/Atlas/DistributedDataManagement
- GANGA:
  - http://ganga.web.cern.ch/ganga/