



ATLAS Computing Model

(I-COOP+ 2016 project: COOPB20247)

Santiago González de la Hoz

Instituto de Física Corpuscular (IFIC)

Centro Mixto Universitat de València - CSIC



Outline

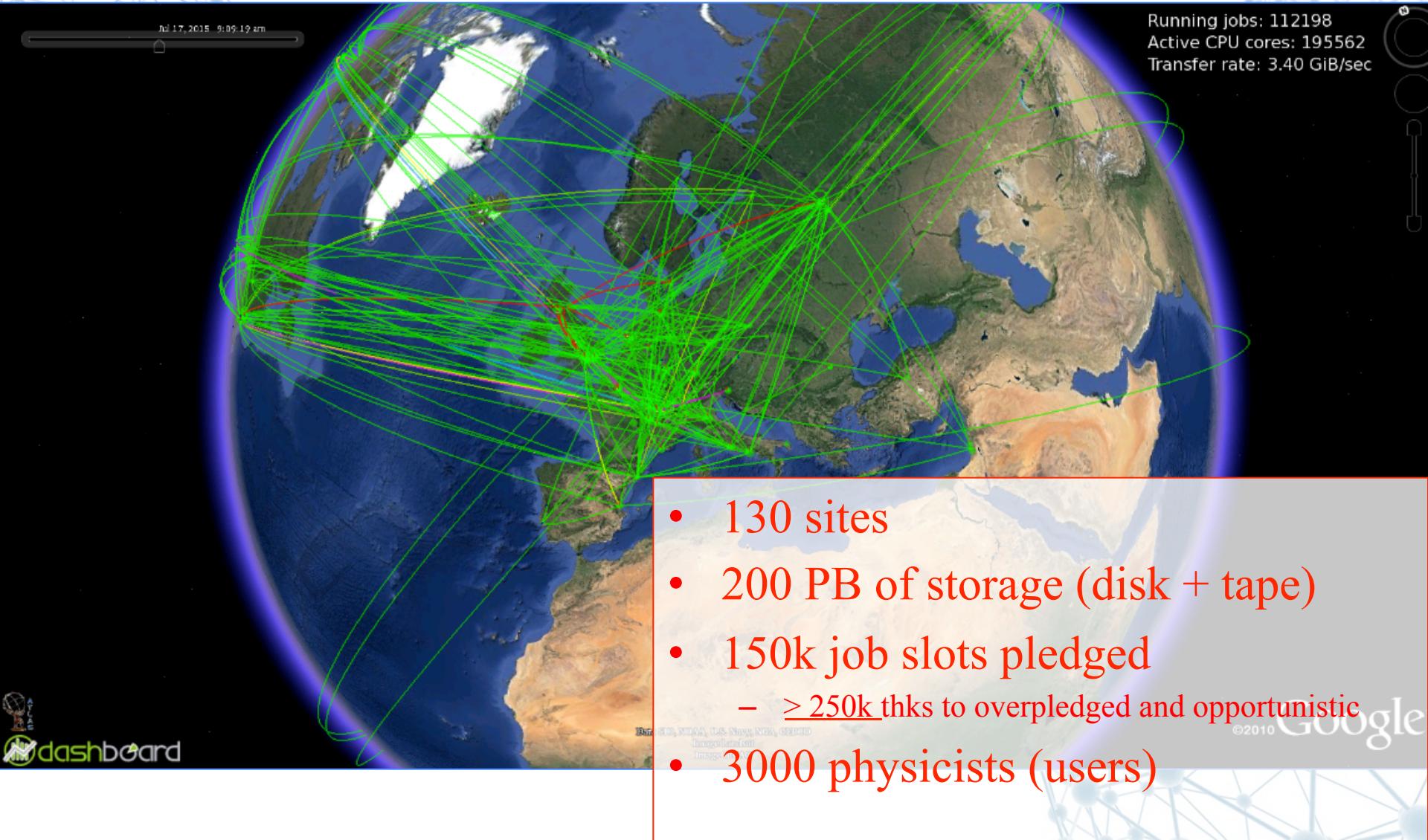
- Introduction to the Grid
- ATLAS Distributed Computing
- The Worldwide LHC Computing Grid (WLCG)
 - ATLAS Clouds
 - Tier roles
 - Networking
- ATLAS Software distribution: CVMFS
- Computing Resources at ATLAS sites
- Rucio
 - Tools for users
 - Web Interface
- Introduction to the ATLAS Event Data Model (EDM)
 - What is the EDM?
 - xAOD: new EDM



Part I: the **GRID**

(General concepts in Jose Salt's slides)

ATLAS Distributed Computing

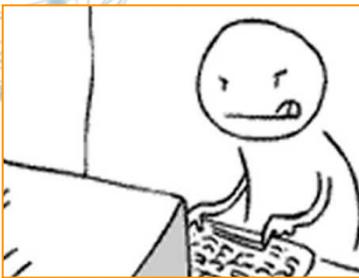


ADC at a glance

- **The experts:** > 200! (not full time)
 - Making things work: developing, integrating, operating
- **The (Central) Services:**
 - Ad-hoc solutions developed by ATLAS and/or WLCG:
 - Panda for Job execution, Rucio for Data Management, FTS for actual file transfers, CVMFS to distribute the SW...
- **The Sites**
 - WLCG: **Worldwide LHC Computing Grid**
 - Grid, off-Grid, HPC, cloud resources, volunteer...
 - Hardware resources for storage and computing
 - ... and expertise!



Basic principles: Users and the Grid



Data lookup:
Which data am I looking for?

- Distributed Data Management: Rucio
 - ATLAS metadata interface
 - EventIndex

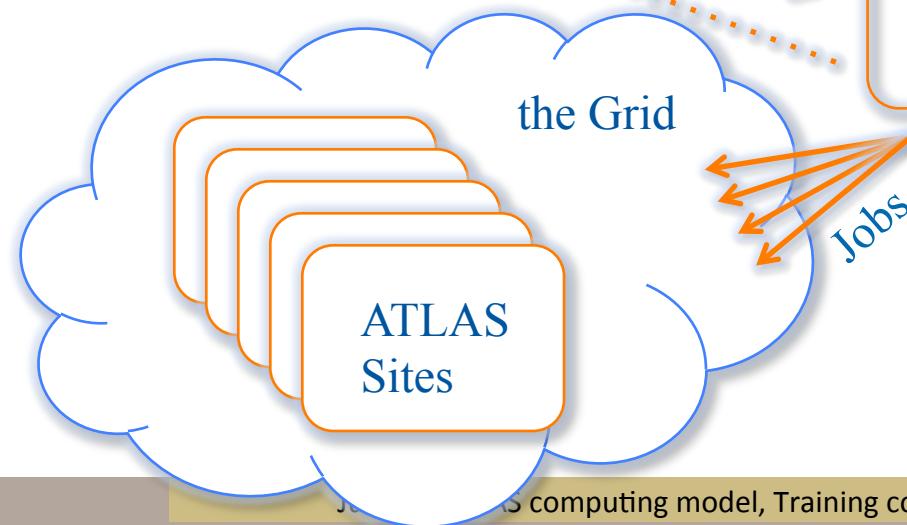
Basic principles: Users and the Grid



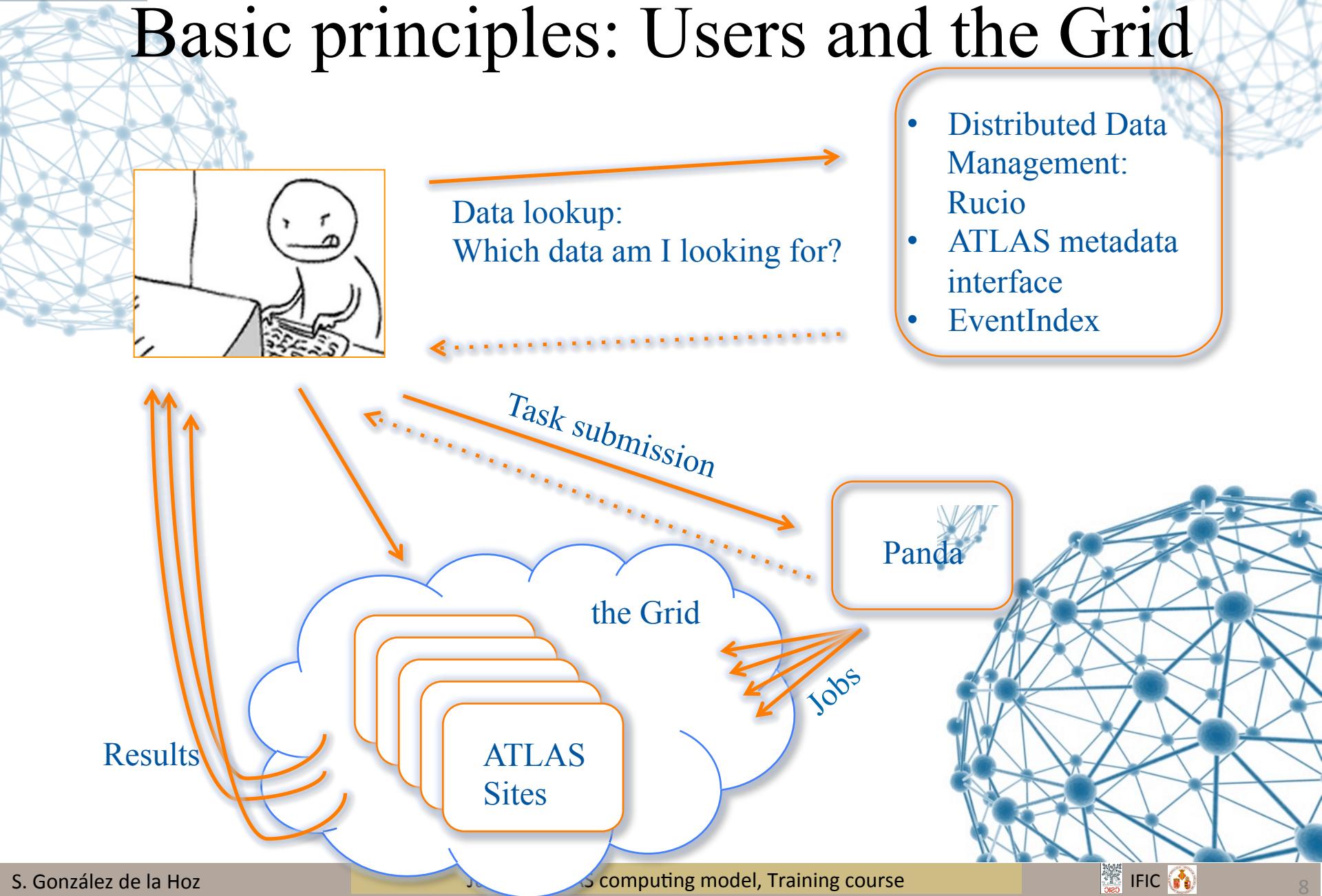
Data lookup:
Which data am I looking for?

- Distributed Data Management: Rucio
- ATLAS metadata interface
- EventIndex

Task submission



Basic principles: Users and the Grid



Grid Computing

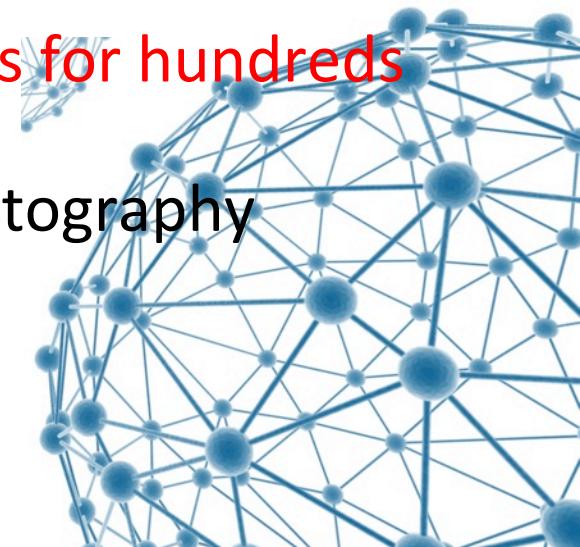


- Idea started in late '90s
- Like the electricity Grid

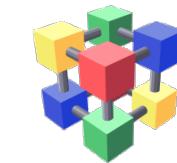
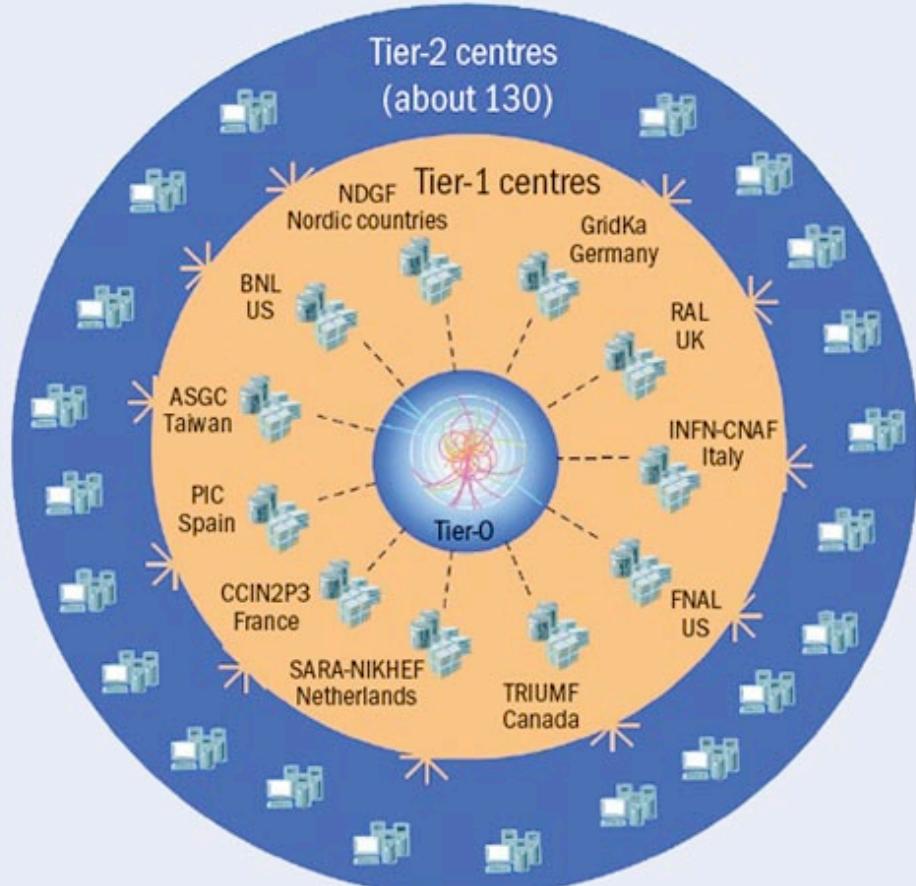
- The Grid: a **technology** that enables optimized and secure access to widely distributed heterogeneous computing and storage facilities of different ownership

The Grid

- The “Grid middleware” exposes heterogeneous resources to the Grid in a uniform interface
 - Computing Elements give access to CPUs
 - Storage Elements give access to data
 - Information systems describe the Grid
- How to allow access to resources?
 - Cannot give usernames and passwords for hundreds of sites to thousands of people!
 - Fundamental basis is X509-based cryptography
- Many Grids
 - WLCG

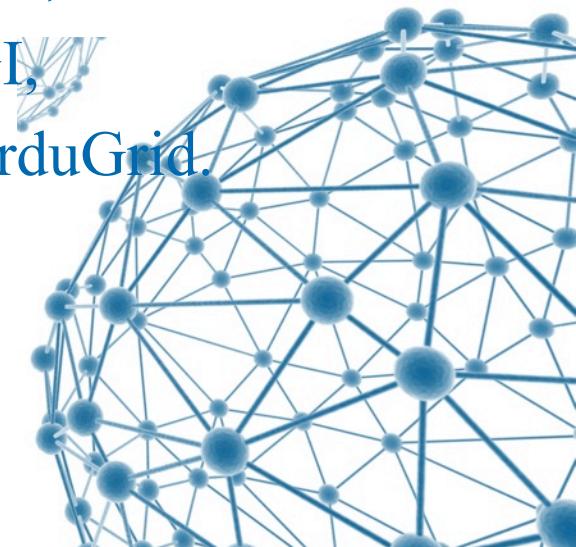


The Worldwide LHC Computing Grid



WLCG
Worldwide LHC Computing Grid

- 3 Grid middleware:
 - OSG,
 - EGI,
 - NorduGrid.



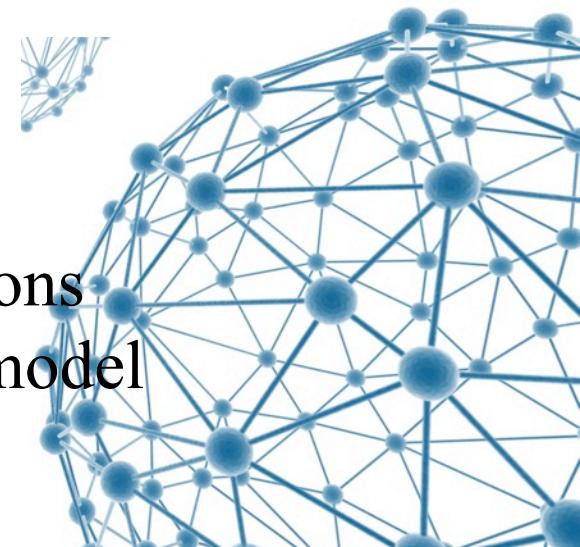
The ATLAS sites

- Site == Computing Centre
- (usually): Storage + Computing power (CPUs)
 - Squid proxy caches for SW and detector conditions distribution
- Tiers:
 - Tier-0 (CERN), Tier-1s (Tape + some “special” services), Tier-2s: sites with signed agreement with ATLAS (MoU)
 - Tier-3 opportunistic resources

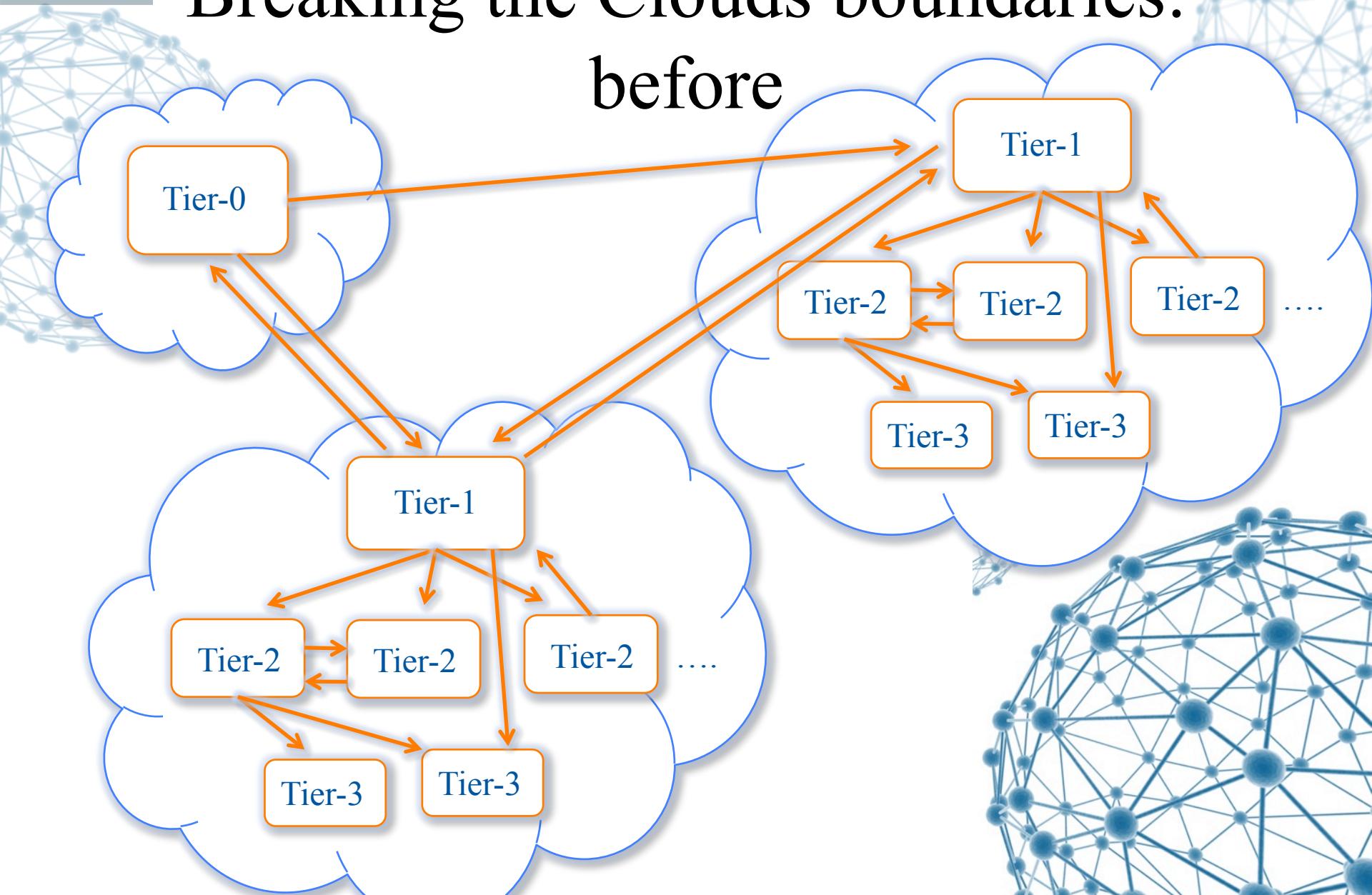


ATLAS Clouds

- ATLAS Clouds **≠** Cloud resources (AWS, Google Compute, Rackspace)
- ATLAS Clouds
 - Logical grouping of sites:
 - one Tier1 plus several Tier2s and Tier3s
 - Mostly belonging to the same country/funding agency
 - Support provided by Cloud Squads
 - close to each site, often same language
- historical concept
 - Useful in the past: networking limitations
 - Still useful especially for the support model

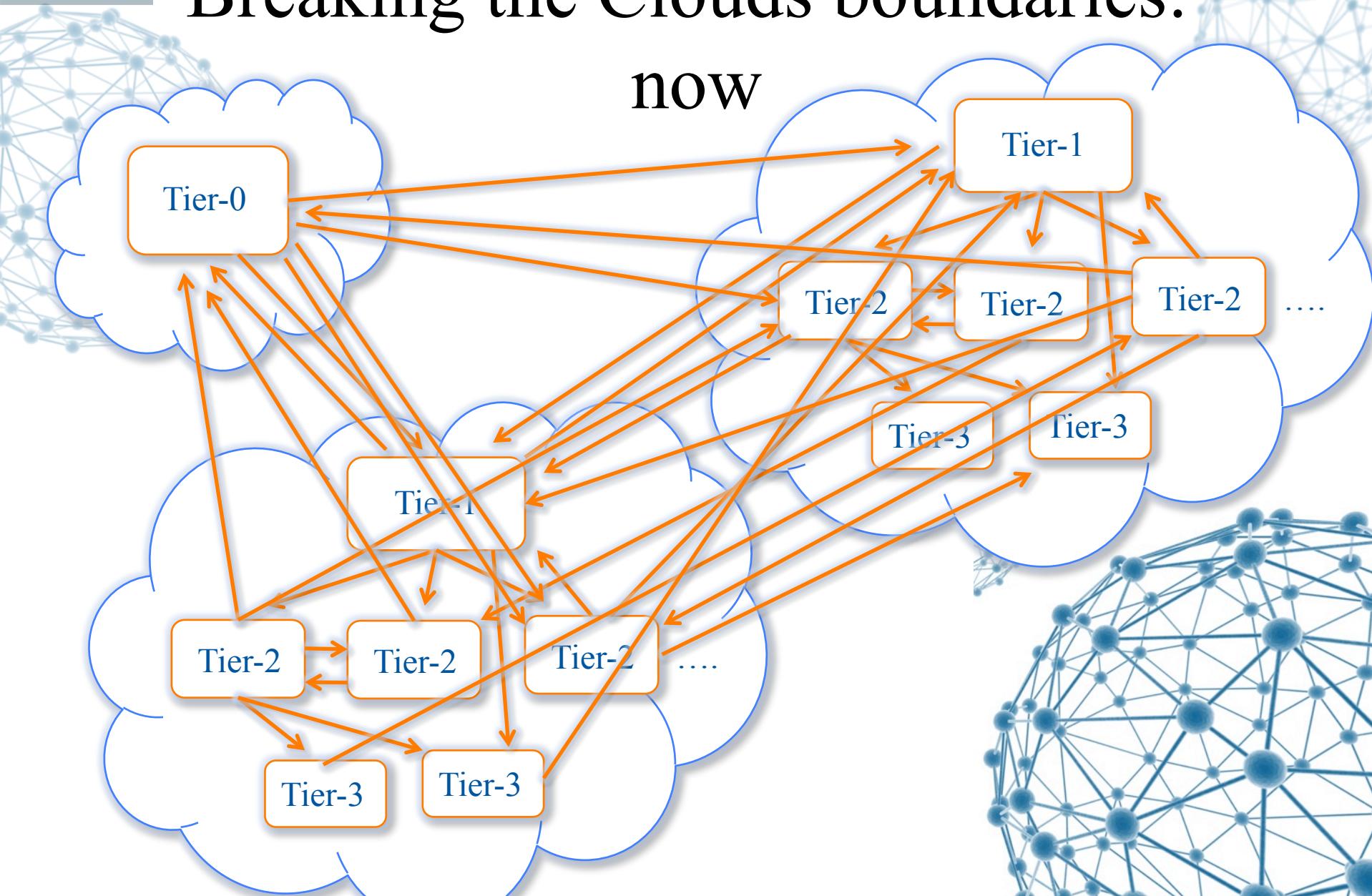


Breaking the Clouds boundaries: before



Breaking the Clouds boundaries:

now

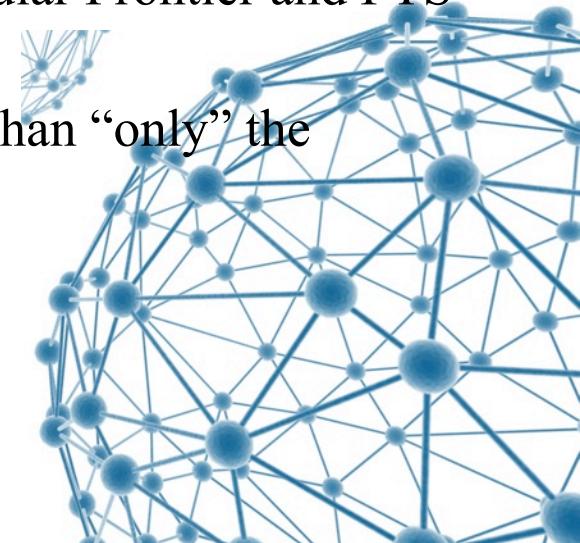


Tier roles: Tier-0

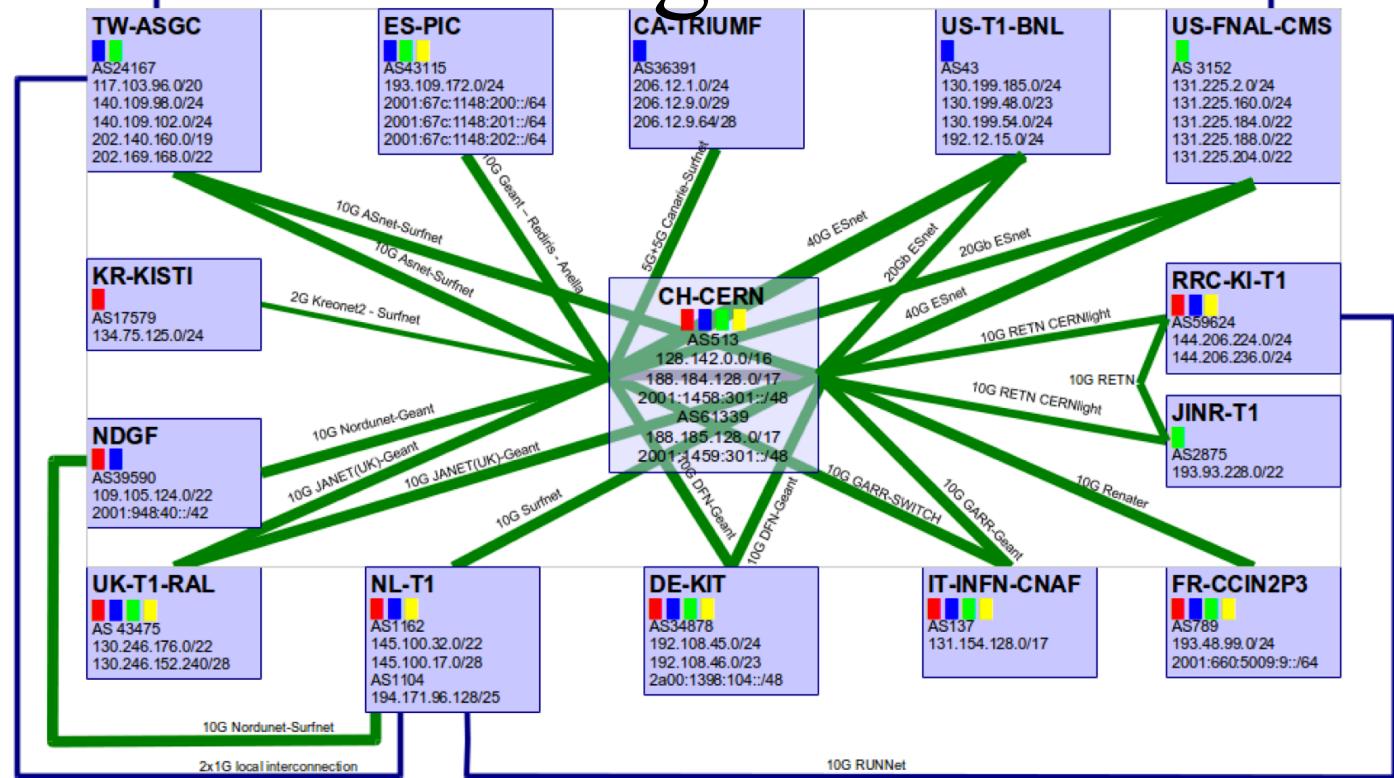
- Tier-0
 - Tape custodial copy of all the RAW data
 - First-step processing of LHC data, from RAW to AOD
 - Most of the ATLAS Central Services are located at CERN (in an highly resilient deployment):
 - Panda, Rucio, DB (Oracle), Stratum-0 (for SW distribution through CVMFS), Frontier (for conditions data), one FTS (File Transfer Service) production instance ...

Tier roles: Tier-1s, Tier-2s and Tier-3s

- Tier-1s and Tier-2s provide pledged storage and computing power
 - Storage for custodial and “primary” data
 - Main difference between Tier-1s and Tier-2s is tape storage for custodial data at Tier-1s and 24x7 (T1s) vs working day (T2s) operational support
 - some Tier1s runs also “central” services, in particular Frontier and FTS to provide redundancy and closeness
 - Tier-1s and Tier-2s in reality provide much more than “only” the pledge
- Tier-3s provide “opportunistic” resources
 - Not unique copy of “important” data



Networking



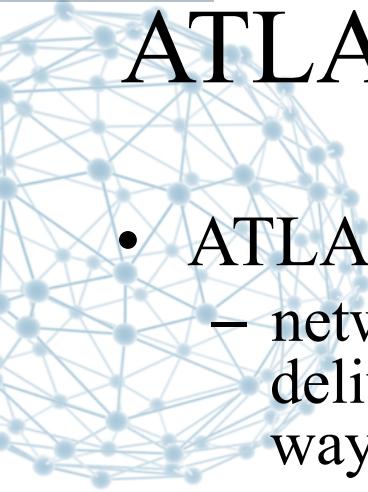
- Tier-0 & Tier-1s: LHCOPN
 - Tier-0, 1 & 2: LHCONE (LHC Open Network Environment)
 - Collaborative effort between Research & Education Network providers
 - flexible usage of bandwidth, sharing costs of expensive links, scalable
 - Layer3 (routed) Virtual Private Network

ATLAS Site: what kind of?

- Grid sites
 - Storage Element (SE) + Computing Element (CE) to access the HW resources
- Cloud resources
 - Pledged or opportunistic
 - For now only processing resources: outputs on Grid SE
- High-Performance Computing center (**See Esteban's slides**):
 - Only processing, no storage, mostly with edge node to stream outputs into std Grid SE
- Volunteer computing (add running jobs on home PCs):
 - BOINC: only processing, no storage

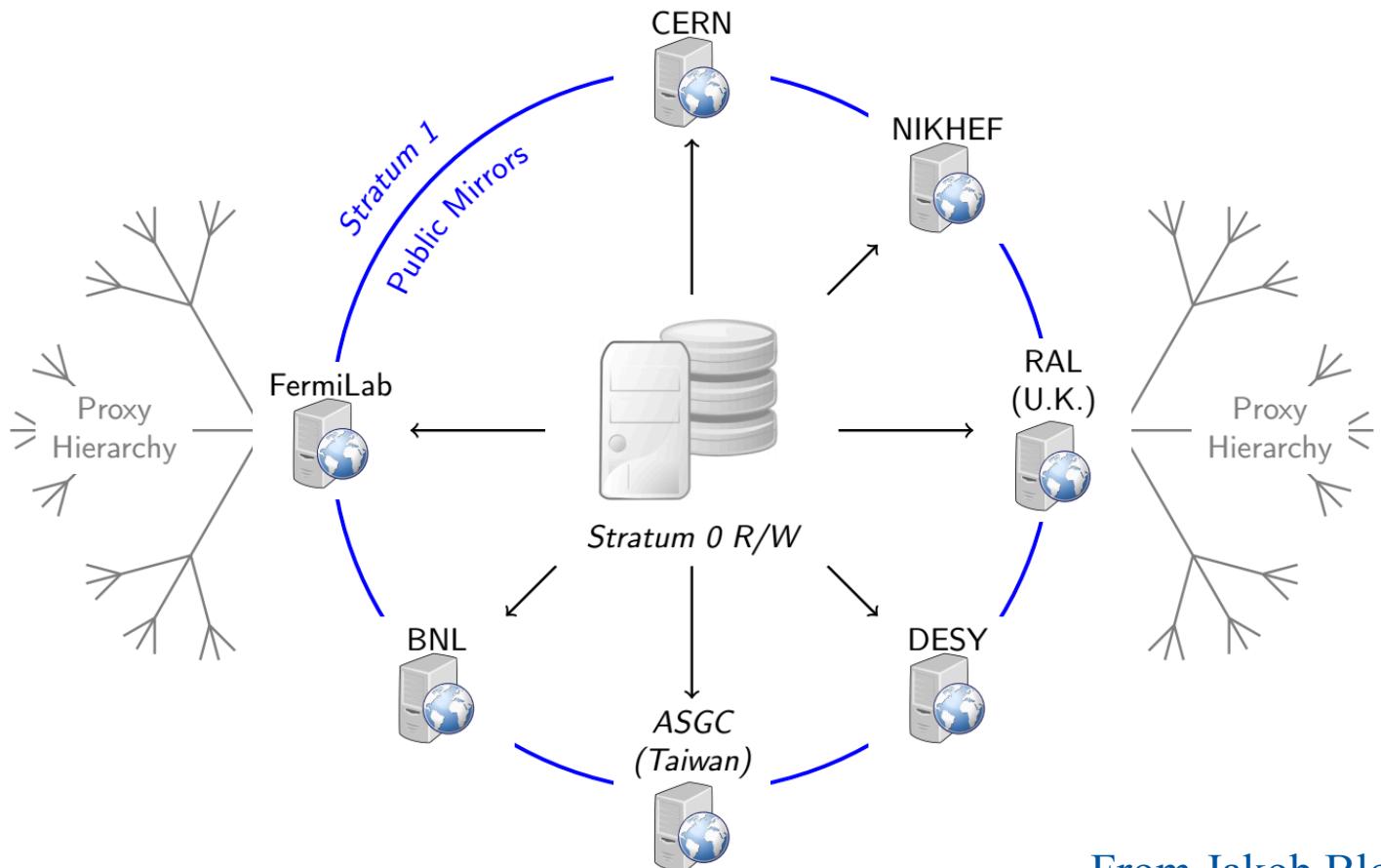
ATLAS Software distribution: CVMFS

- ATLAS relies on CVMFS (CERN VM FileSystem)
 - network file system based on HTTP and optimized to deliver experiment software in a fast, scalable, and reliable way
 - New SW pushed into the system at CERN Stratum-0, replicated to various Stratum-1
 - Massive replication through set of Squids hosted at the sites themselves
 - WN at sites accessing the site's squids
 - Resilient in case of Squid failures, retries going one level up
- All “standard” ATLAS sites use CVMFS
 - Requires connection to the outside world, not suitable for most HPC





CernVM-FS Content Distribution

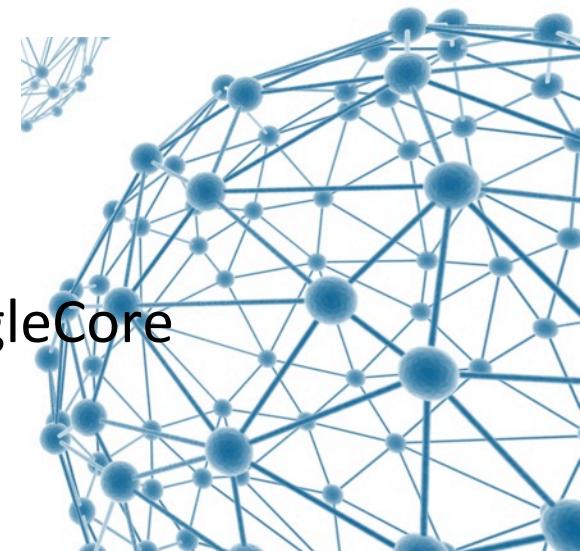


From Jakob Blomer

Computing resources at ATLAS sites

ATLAS needs various types of resources

- Analysis and Production
 - Different proxy roles
- Analysis SingleCore
 - 2GB of RSS per core, order of 10GB of workdir
- Production SingleCore
 - 2GB of RSS per core, order of 10GB of workdir
- Production MultiCore
 - Usually 8 cores, 16GB of RSS, 100 GB of workdir
- (at some sites) Production HighMemory SingleCore
 - >4GB per core



Part II: getting data with RUCIO (see Farida's slides)

Introduction to Rucio

- Rucio is the ATLAS Distributed Data Management framework. It replace the previous system called DQ2 (Don Quijote 2)



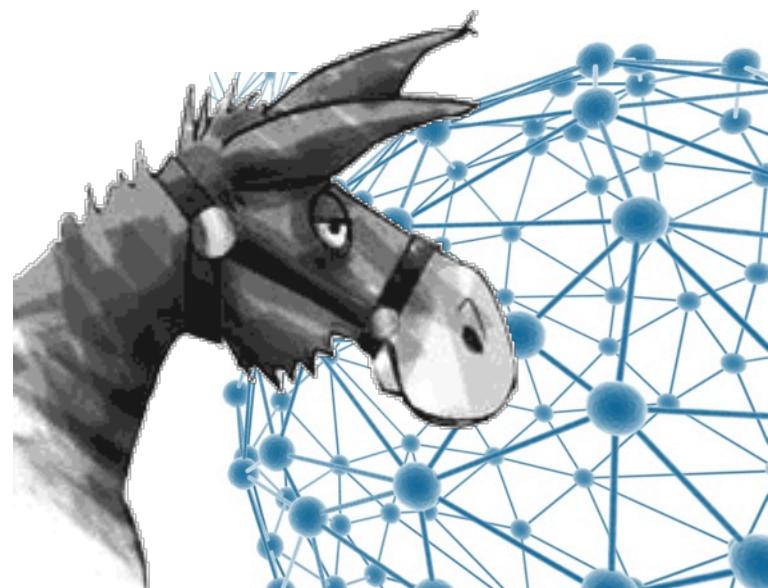
Don Quixote
de la
Mancha

Introduction to Rucio

- Rucio is the ATLAS Distributed Data Management framework. It replace the previous system called DQ2 (Don Quijote 2)



- Rucio catalogs all the ATLAS data and manages the replication and lifecycle of all these data

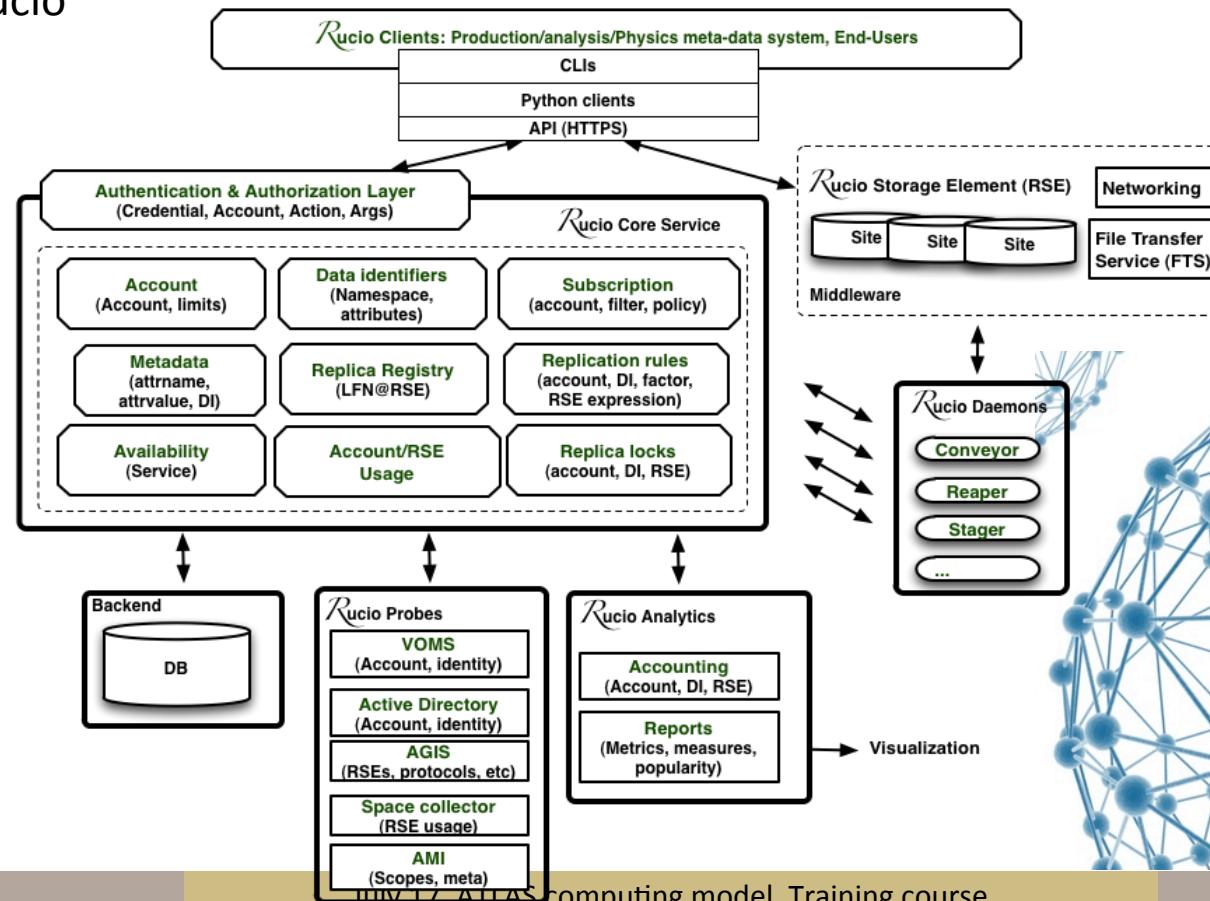


Rucio in numbers

- Data volume stored: 260 PB
- ~900M files
- 130 grid sites
- More than 1000 users
- Up to 2 PB transfers every day (~500 TB to 2M files)
- Up to 14M files deleted/day

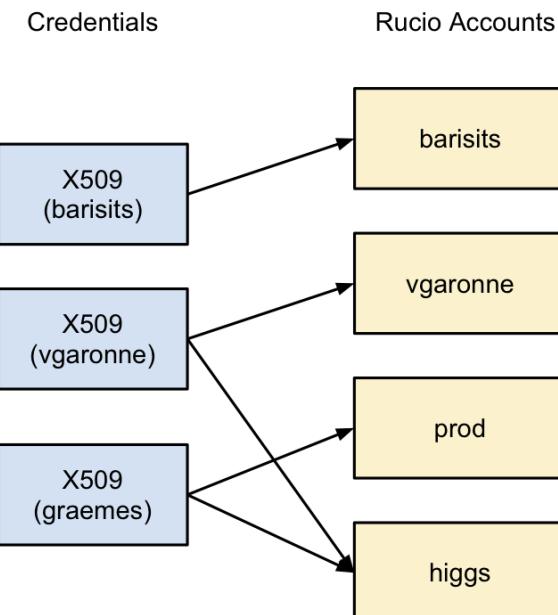
3 main components

- Catalog of the datasets, content, location
- Set of daemons to move the data from one site to another, manage the data deletion (all datasets in ATLAS have a lifetime)
- Clients tools (CLI, web interface) that allow end-users or other service to interact with Rucio



Rucio concepts (I)

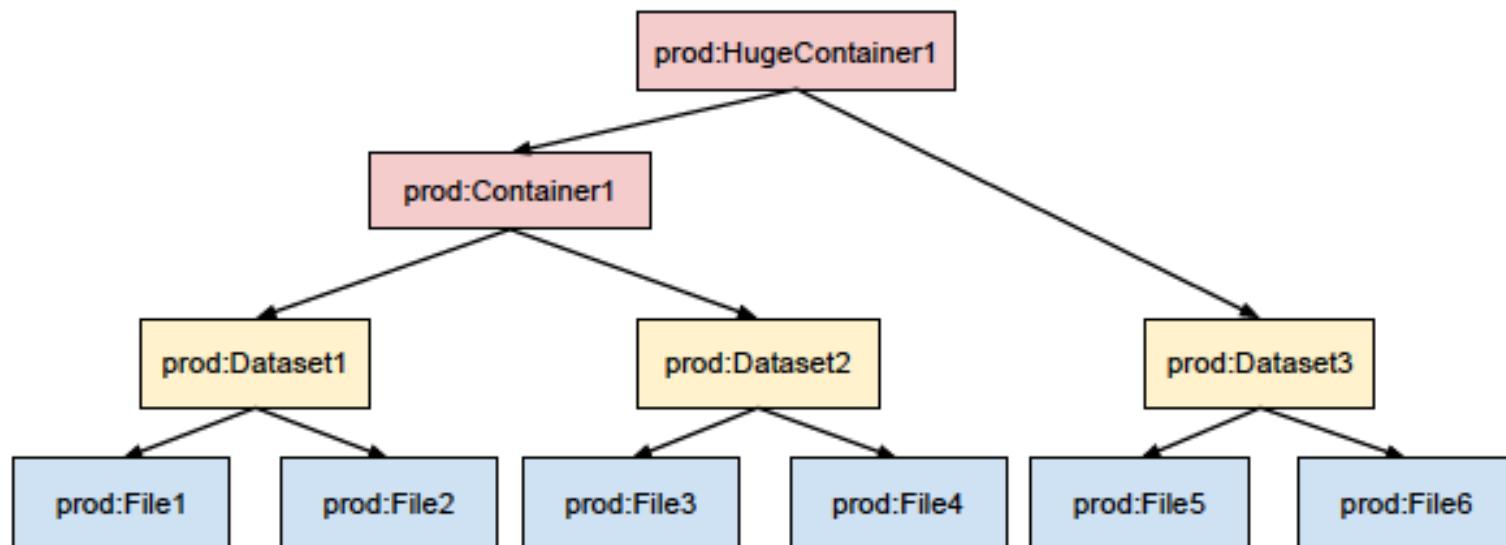
- Rucio account:
 - It can represent users (e.g sgonzale), groups (higgs), activities (tier0).
 - Quota, permissions tunable and associated to one account.
 - One can connect to a Rucio account using x509 certificate/proxy,, kerberos, userpass.
 - One credential can be used to map to different accounts.
 - For normal users the Rucio account is the same as you CERN login name



Rucio concept (2)

- Rucio namespace:

- 3 types of Data Identifiers (DIDs): File, Datasets, Containers. Allows multiple hierarchy level for containers.
- All Data Identifier are identified by a scope and a name. A name is unique within a scope but can be used in other scopes (e.g. user.sgonzale:mytest.root vs user.serfon:mytest.root).



Rucio concepts (3)

- Rucio Storage Elements (RSE):
 - Abstraction for storage end-point.
 - Can be grouped in various ways with tags (e.g. tier=1, cloud=ES).
- Replication rules:
 - Describe how a Data Identifier must be replicated on a list of Rucio Storage Elements.
 - E.g. Make 2 replicas of dataset data16_13TeV:mydatasetname on tier=1&disk=1.
 - Rucio will create the minimum number of replicas to optimise storage space, minimise the number of transfers and automate data distribution.
- Subscriptions:
 - Replication policies based on Data Identifiers metadata, for Data Identifiers that will be produced in the future.
 - E.g.: Make 2 replicas of datasets with scope=data16_13TeV and datatype=AOD on tier=1&disk=1.

Rucio Concepts (4)

- Permission/quotas:
 - 2 different types of RSEs open to the end-users to store output datasets:
 - SCRATCHDISK area: Default area for job outputs. Default lifetime: 2 weeks
 - LOCALGROUPDISK area;: For long term storage of user data. No default lifetime.
 - Every user has a quota on all SCRATCHDISK RSEs.
 - Everyone gets a quota by default on LOCALGROUPDISK RSEs in their country.
 - Admins of the country can modify the quota, approve replication requests, delete old data.

Tools for users

- Rucio CLI (rucio commands)
 - To find datasets, get their locations, download them, upload data on the grid.
 - To check quota/space usage...
- R2D2 (web interface)
 - Request transfer of dataset by making rules

RUCIO CLI (1)

- Setup ATLAS and Rucio Client:

```
$ setupATLAS  
$ localSetupRucioClients  
$ voms-proxy-init -voms atlas
```

if setupATLAS is undefined, then

```
$ export ATLAS_LOCAL_ROOT_BASE=/cvmfs/atlas.cern.ch(repo/ATLASLocalRootBase  
$ alias setupATLAS='source ${ATLAS_LOCAL_ROOT_BASE}/user/atlasLocalSetup.sh'
```

- The main command line interface is rucio
- Type rucio without arguments to get a list of possible sub-commands.

RUCIO CLI (2)

- For instance list replica locations of a dataset
 - Rucio list-dataset-replicas:

```
$> rucio list-dataset-replicas mc12_14TeV:mc12_14TeV.119996.Pythia8_A2MSTW2008LO_minbias_inelastic_high.merge.HITS.e1133_s2079_s1964_tid04640
```

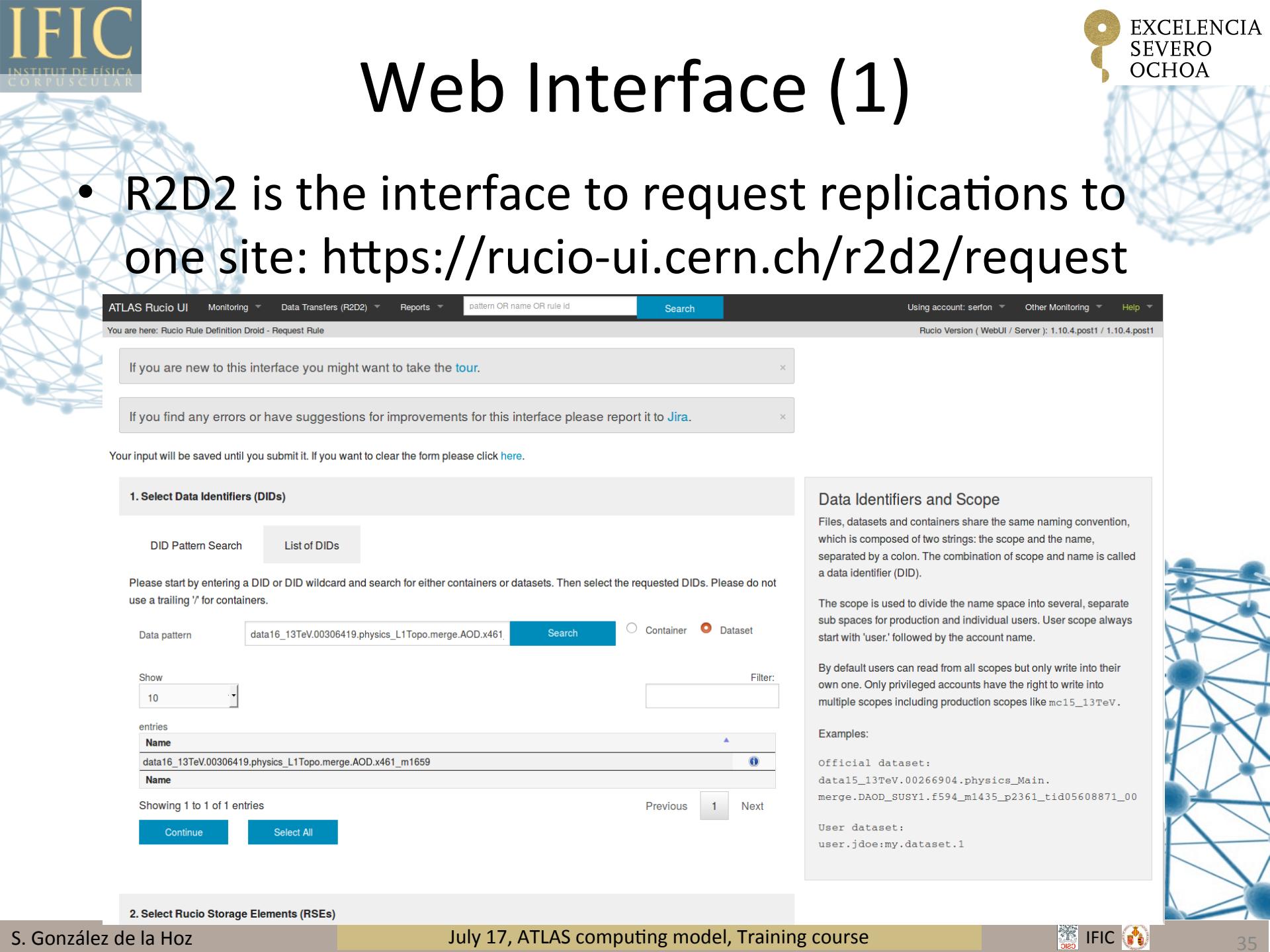
```
DATASET: mc12_14TeV:mc12_14TeV.119996.Pythia8_A2MSTW2008LO_minbias_inelastic_high.merge.HITS.e1133_s2079_s1964_tid04640638_00
```

RSE	FOUND	TOTAL
BNL-OSG2_DATADISK	1000	1000
RAL-LCG2_DATADISK	1000	1000
SARA-MATRIX_MCTAPE	1000	1000

- More info:
 - <https://twiki.cern.ch/twiki/bin/view/AtlasComputing/RucioClientsHowTo>

Web Interface (1)

- R2D2 is the interface to request replications to one site: <https://rucio-ui.cern.ch/r2d2/request>



The screenshot shows the R2D2 web interface for requesting replication. At the top, there's a navigation bar with links for ATLAS Rucio UI, Monitoring, Data Transfers (R2D2), Reports, and Help. A search bar contains the placeholder "pattern OR name OR rule id" with a "Search" button. Below the navigation, a message box says "If you are new to this interface you might want to take the [tour](#)". Another message box encourages reporting errors to [Jira](#). A note states that input will be saved until submission. The main area is titled "1. Select Data Identifiers (DIDs)". It includes a "DID Pattern Search" field containing "data16_13TeV.00306419.physics_L1Topo.merge.AOD.x461", a "Search" button, and radio buttons for "Container" and "Dataset" (which is selected). A dropdown "Show" menu is set to 10 entries. A "Filter:" input field is present. A table lists two entries: "data16_13TeV.00306419.physics_L1Topo.merge.AOD.x461_m1659" and "Name". Below the table, it says "Showing 1 to 1 of 1 entries" and has "Previous" and "Next" buttons. At the bottom are "Continue" and "Select All" buttons. A second section, "2. Select Rucio Storage Elements (RSEs)", is partially visible at the bottom.

Data Identifiers and Scope

Files, datasets and containers share the same naming convention, which is composed of two strings: the scope and the name, separated by a colon. The combination of scope and name is called a data identifier (DID).

The scope is used to divide the name space into several, separate sub spaces for production and individual users. User scope always start with 'user.' followed by the account name.

By default users can read from all scopes but only write into their own one. Only privileged accounts have the right to write into multiple scopes including production scopes like mc15_13TeV.

Examples:

Official dataset:
data15_13TeV.00266904.physics_Main.
merge.DAOD_SUSY1.f594_m1435_p2361_tid05608871_00

User dataset:
user.jdoe:my.dataset.1

Web Interface (2)

- View to monitor the replication of the rules:
<https://rucio-ui.cern.ch/r2d2>

ATLAS Rucio UI Monitoring Data Transfers (R2D2) Reports pattern OR name OR rule id Search Using account: serfon Other Monitoring Help

You are here: Rucio Rule Definition Droid - List Rules Rucio Version (WebUI / Server): 1.10.4.post1 / 1.10.4.post1

Rules

New request

Account	RSE	State	Activity	Interval	Apply
ddmadmin	RSE		Data Rebalancing	14 days	<input checked="" type="checkbox"/>

Show 100 entries Search:

Name	Account	RSE Expression	Creation Date	State	Locks OK	Locks Replicating	Locks Stuck
mc12_8TeV:mc12_8TeV.129662. Pythia8_MRST2004QED_ggToMumu_60M200.merge. NTUP_COMMON. e2154_s1581_s1586_r4485_p4540_p1575_tid01355337_00	ddmadmin	SLACXRD_DATADISK	Tue, 04 Apr 2017 07:45:41 UTC	REPLICATING	0	100	0
data12_8TeV:data12_8TeV.0020967.physics_Egamma.merge. NTUP_COMMON.r4644_p1517_p1675_tid01452813_00	ddmadmin	SLACXRD_DATADISK	Tue, 04 Apr 2017 07:45:05 UTC	REPLICATING	0	73	0
data12_8TeV:data12_8TeV.00208126.physics_Muons.merge. NTUP_COMMON.r4065_p1278_p1675_tid01452987_00	ddmadmin	SLACXRD_DATADISK	Tue, 04 Apr 2017 07:45:02 UTC	REPLICATING	0	27	0
data12_8TeV:data12_8TeV.00202660.physics_Egamma.merge. NTUP_COMMON.r4065_p1278_p1675_tid01451029_00	ddmadmin	SLACXRD_DATADISK	Tue, 04 Apr 2017 07:45:01 UTC	REPLICATING	0	3	0
data12_8TeV:data12_8TeV.00206717.physics_JetTauEtmiss. merge.NTUP_COMMON.r4065_p1278_p1575_tid01403449_00	ddmadmin	SLACXRD_DATADISK	Tue, 04 Apr 2017 07:45:01 UTC	REPLICATING	0	1	0
data12_8TeV:data12_8TeV.00209994.physics_HadDelayed. merge.NTUP_COMMON._4487_p1476_p1575_tid01323572_00	ddmadmin	SLACXRD_DATADISK	Tue, 04 Apr 2017 07:45:01 UTC	REPLICATING	0	36	0
data12_8TeV:data12_8TeV.00211902.physics_ZeroBiasOverlay. merge.NTUP_COMMON.r4065_p1278_p1680_tid01486550_00	ddmadmin	SLACXRD_DATADISK	Tue, 04 Apr 2017 07:44:59 UTC	REPLICATING	0	13	0
data12_8TeV:data12_8TeV.00206971.debugrec_hltacc.merge. NTUP_COMMON.g1_r4065_p1278_p1575_tid01463167_00	ddmadmin	SLACXRD_DATADISK	Tue, 04 Apr 2017 07:44:58 UTC	REPLICATING	0	1	0
mc12_8TeV:mc12_8TeV.188602. ProtoPythia_AUET2BMSTW2008LO_sVLQB1000GeV01.merge. NTUP_COMMON. e2738_a220_a205_r4540_p1575_tid01482496_00	ddmadmin	SLACXRD_DATADISK	Tue, 04 Apr 2017 07:44:58 UTC	REPLICATING	0	20	0
mc12_8TeV:mc12_8TeV.110502. ProtoPythia_P2011C_st_tchan_lept_VG_1p165_0_0_p018. merge.NTUP_COMMON. e3609_a220_a263_a264_r4540_p1575_tid05136324_00	ddmadmin	SLACXRD_DATADISK	Tue, 04 Apr 2017 07:44:57 UTC	REPLICATING	170	195	0

Getting help

- General HOWTO, FAQ:
[https://twiki.cern.ch/twiki/bin/view/AtlasComputing/
RucioClientsHow](https://twiki.cern.ch/twiki/bin/view/AtlasComputing/RucioClientsHow)
- 1st level support provided by DAST:
 - Can solve most of the problems
 - If cannot solve, they scale to 2nd level support (DDM operation team)
- In case of bugs in code/web UI (not transfer errors) or feature requests: <https://its.cern.ch/jira/browse/RUCIO>
- Rucio is an open source project. If you want to contribute, you are welcome. Just contact ruico-dev@cern.h (they have qualification tasks)

Part III: the ATLAS Event Data Model

What is an Event Data Model?

- Event Data Model (EDM):
 - The collection of classes – interface and concrete objects – and their relationship, which, together, provide a representation of an event detected by ATLAS and ease its manipulation by the analysis developer



What is an Event Data Model?

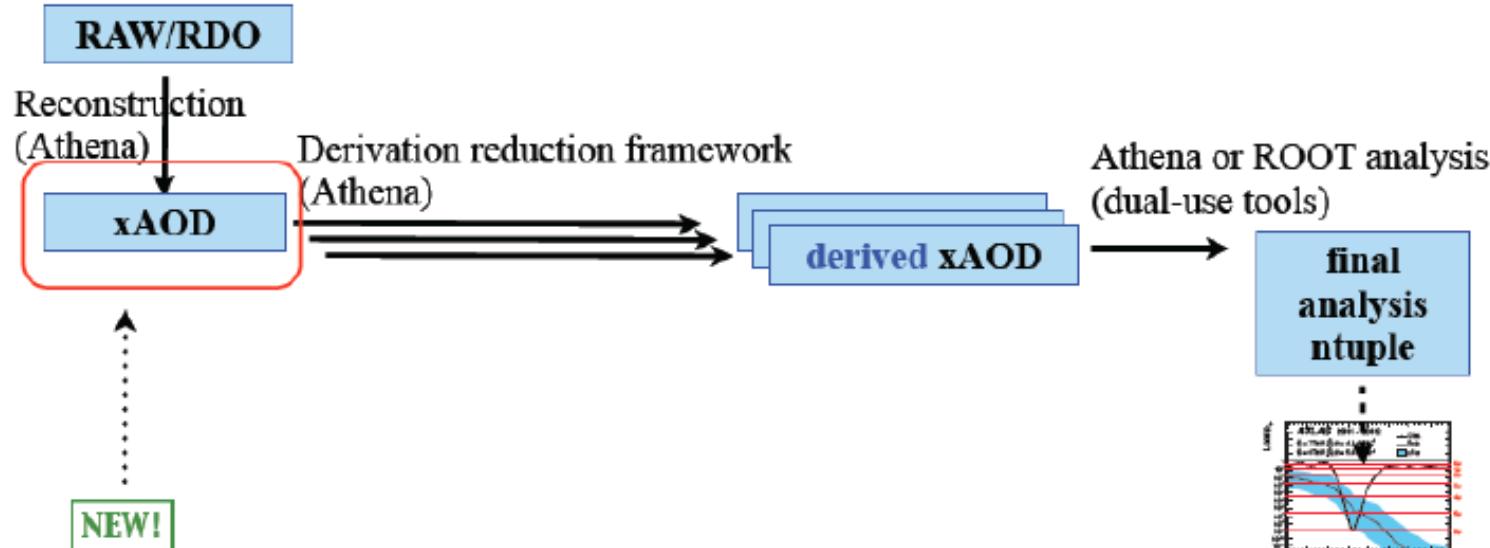
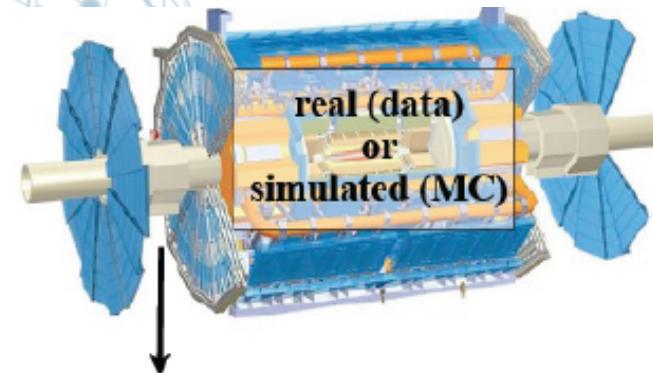
- Event Data Model (EDM):
 - The collection of classes – interface and concrete objects – and their relationship, which, together, provide a representation of an event detected by ATLAS and ease its manipulation by the analysis developer
 - In plain word:
 - How “electrons”, “jets”, “muons”, etc. are stored and how to use them



The New EDM: xAOD

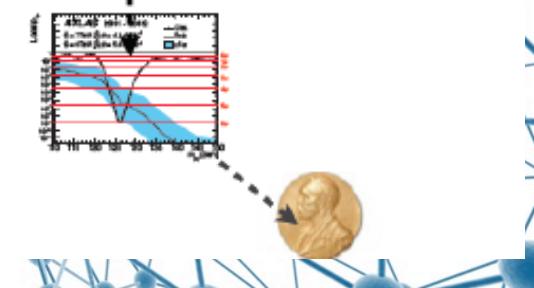
- Design considerations:
 - Usable from Athena and ROOT
 - Easy to use for analysis
 - Fast and flexible

The Run 2 Analysis model



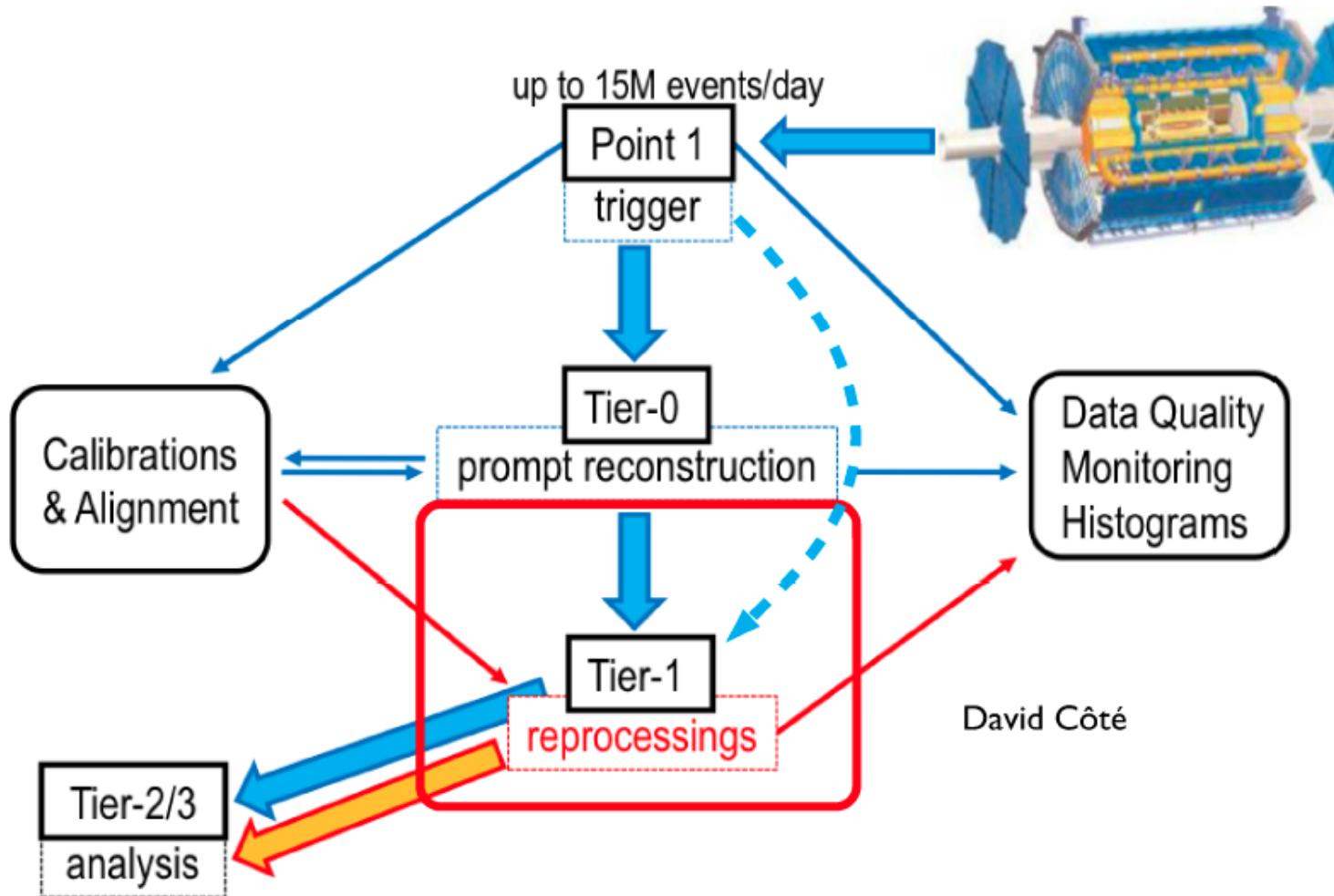
The **xAOD** format:

- the format is readable by both Athena and ROOT
- it is fast and compact
- it is uniform across all reconstructed object types (electrons, muons, jets, etc.)
- more ‘object’ oriented



Reconstruction & Reprocessing

Reprocessing at Tier-1s



Streaming Data

- The output of the trigger, RAW data, is organised into inclusive Streams.
 - “Inclusive” means that events can end up in one or more streams, depending on which triggers they pass.
- Each trigger chain (HLT output) belongs to one stream.
- One primary physics streams: physics Main ~1 KHz of data
 - There are also other special purpose physics streams: e.g. physics_Late.
- Many special streams for operations e.g calibration, cosmic, debug...
 - Calibration streams only have partial event information (e.g. only tracking).
- ...and an Express stream: express_express
 - Used for quick Data Quality and prompt calibration
- Can find streams produced in any given run (and many other useful things) using RunQuery:
 - <https://atlas-runquery.cern.ch>

Dataset Nomenclature

Data:

project tag: 2012 pp data 8 TeV Run number stream merged files Data type: AOD

data12_8TeV.00209980.physics_Egamma.merge.AOD.f476_m1223

AMI tag describes configuration of each step (Tier-0 bulk reconstruction **t**, file merging **m**)

Simulation:

project tag: "mc12" setup 8 TeV	MC DSID unique # for process	"human-readable" description of MC sample	merged files	Data type: SM W/Z ntuples
mc12_8TeV.119353.MadGraphPythia_AUET2BCTEQ6L1_ttbarW.merge.NTUP_SMWZ.e1352_s1499_s1504_r3658_r3549_p1328/				

AMI tag describes config

each step (evt generation **e**, full simulation **s**, reconstruction **r**, D3PD creation **p**)

`/` is a “container” (points to other datasets)

- Datasets are collections of files, form fundamental unit of ATLAS data and MC.
 - Each processing step changes the data type, and adds the AMI tag used.
 - More about AMI tomorrow!



xAOD::EventInfo

- It contains information about the given event, e.g.:
 - What was the pile-up for this event?
 - What is the current run number, event number, luminosity block number?
- There is one and only one object of this type for a given event

```
uint32_t runNumber () const
The current event's run number.

void setRunNumber (uint32_t value)
Set the current event's run number.

unsigned long long eventNumber () const
The current event's event number.

void setEventNumber (unsigned long long value)
Set the current event's event number.

uint32_t lumiBlock () const
The current event's luminosity block number.

void setLumiBlock (uint32_t value)
Set the current event's luminosity block number.

uint32_t timeStamp () const
POSIX time in seconds from 1970. January 1st.

void setTimeStamp (uint32_t value)
Set the POSIX time of the event.

uint32_t timeStampNSOffset () const
Nanosecond time offset wrt. the time stamp.

void setTimeStampNSOffset (uint32_t value)
Set the nanosecond offset wrt. the time stamp.

uint32_t bcid () const
The bunch crossing ID of the event.

void setBCID (uint32_t value)
Set the bunch crossing ID of the event.
```

Doxxygen documentation:

<http://atlas-computing.web.cern.ch/atlas-computing/links/nightlyDocDirectory/xAODEventInfo/html/index.html>

xAOD: How does it look like?

ROOT Object Browser

File Edit View Options Tools Help

Files |

Draw Option: []

ROOT Files

AOD.01572118._003538.pool.root.6

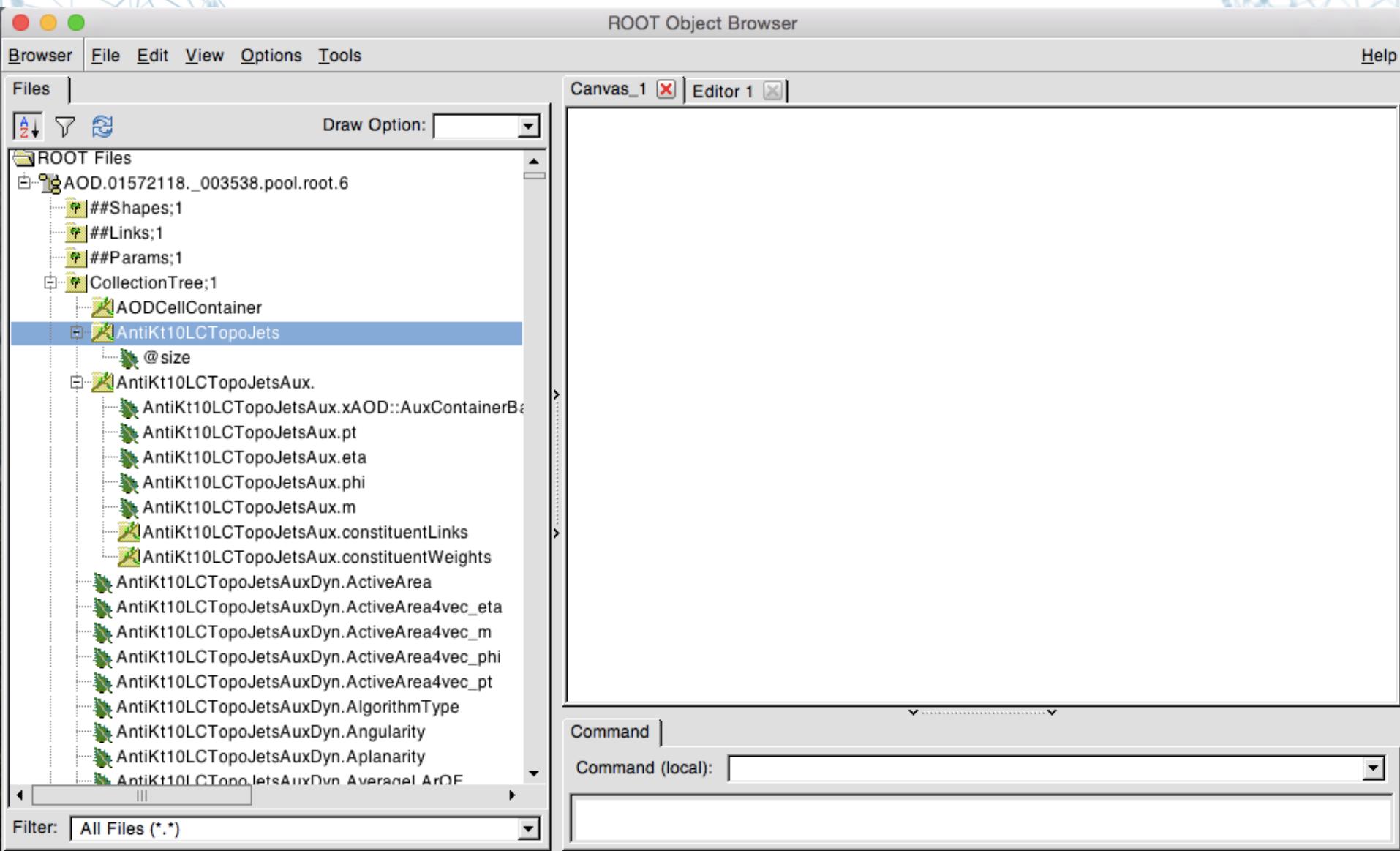
- ##Shapes;1
- ##Links;1
- ##Params;1
- CollectionTree;1
 - AODCellContainer
 - AntiKt10LCTopoJets**
- AntiKt10LCTopoJetsAux.
 - @size
 - AntiKt10LCTopoJetsAux.xAOD::AuxContainerBase
 - AntiKt10LCTopoJetsAux.pt
 - AntiKt10LCTopoJetsAux.eta
 - AntiKt10LCTopoJetsAux.phi
 - AntiKt10LCTopoJetsAux.m
 - AntiKt10LCTopoJetsAux.constituentLinks
 - AntiKt10LCTopoJetsAux.constituentWeights
 - AntiKt10LCTopoJetsAuxDyn.ActiveArea
 - AntiKt10LCTopoJetsAuxDyn.ActiveArea4vec_eta
 - AntiKt10LCTopoJetsAuxDyn.ActiveArea4vec_m
 - AntiKt10LCTopoJetsAuxDyn.ActiveArea4vec_phi
 - AntiKt10LCTopoJetsAuxDyn.ActiveArea4vec_pt
 - AntiKt10LCTopoJetsAuxDyn.AlgorithmType
 - AntiKt10LCTopoJetsAuxDyn.Angularity
 - AntiKt10LCTopoJetsAuxDyn.Aplanarity
 - AntiKt10LCTopoJetsAuxDyn.AverageDeltaRof

Canvas_1 Editor 1

Command |

Command (local): []

Filter: All Files (*.*)



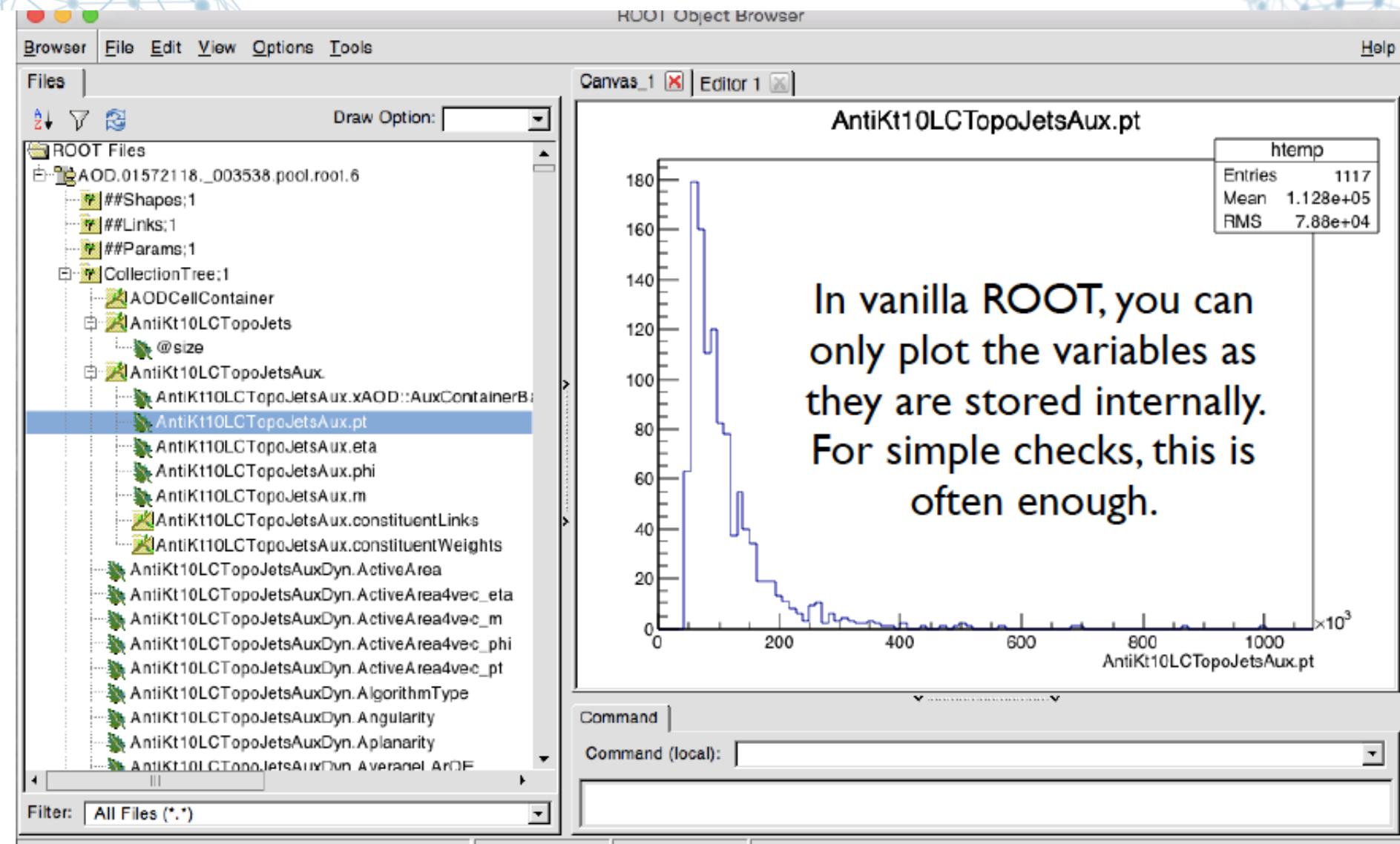
xAOD: How does it look like?

The screenshot shows the ROOT Object Browser interface. The left pane displays a tree view of files and objects. The current file is `AOD.01572118._003538.pool.root.6`. The tree structure includes `#Shapes;1`, `#Links;1`, `#Params;1`, `CollectionTree;1` (which is selected), and `AntiKt10LCTopoJets`. The `AntiKt10LCTopoJets` node has a child node `@size`. Below `AntiKt10LCTopoJets` is a list of auxiliary store objects: `AntiKt10LCTopoJetsAux.AOD::AuxContainerB`, `AntiKt10LCTopoJetsAux.pt`, `AntiKt10LCTopoJetsAux.eta`, `AntiKt10LCTopoJetsAux.phi`, `AntiKt10LCTopoJetsAux.m`, `AntiKt10LCTopoJetsAux.constituentLinks`, and `AntiKt10LCTopoJetsAux.constituentWeights`. To the right of the browser is a text area containing bullet points explaining the structure:

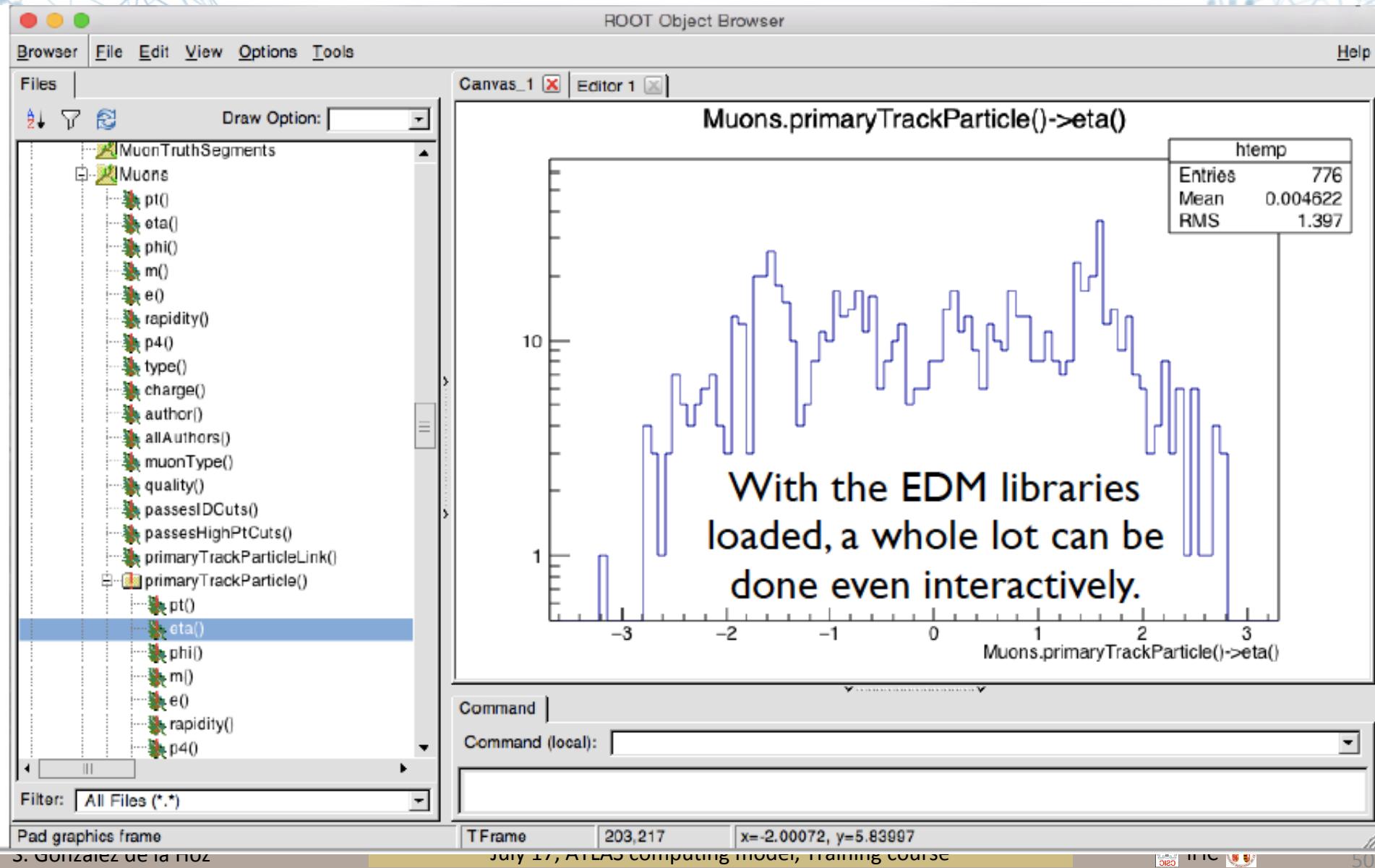
- Event data is always in a `TTree` called “`CollectionTree`”
- The interface container (`DataVector<xAOD::Jet>` in this case) doesn't hold any payload
- All containers of name “`Bla`” are accompanied by an auxiliary store with name “`BlaAux.`”
- The auxiliary store can hold variables that were:
 - already defined at compile time
 - were only created at run time (dynamic)

Arrows from the text points to the `CollectionTree` node, the `AntiKt10LCTopoJets` node, the `AntiKt10LCTopoJetsAux` node, and the list of auxiliary store objects.

xAOD: How does it look like?



xAOD: How does it look like?



Conclusion

Part I: GRID

- The **grid** is a complex beast
 - Which we (try to) keep under control!



Conclusion

- Part II: Rucio (ATLAS Distributed Data Management system)
 - For the purpose of helping the collaboration store, manage and process LHC data in a heterogeneous distributed environment
 - Responsible for moving and cataloguing all ATLAS data on the Grid

Conclusion

- Part III: EDM
 - We have a new, common Event Data Model
 - Designed for analysis users, flexibility, speed, ease to use
 - Readable in both Athena and ROOT
 - All particle objects share a common interface:
xAOD
 - The user should interact with the interface and NOT with the underlying auxiliary store technology



ATLAS
EXPERIMENT

Enjoy the rest of the tutorial!

Run: 282712

Event: 474587238

2015-10-21 06:26:57 CEST