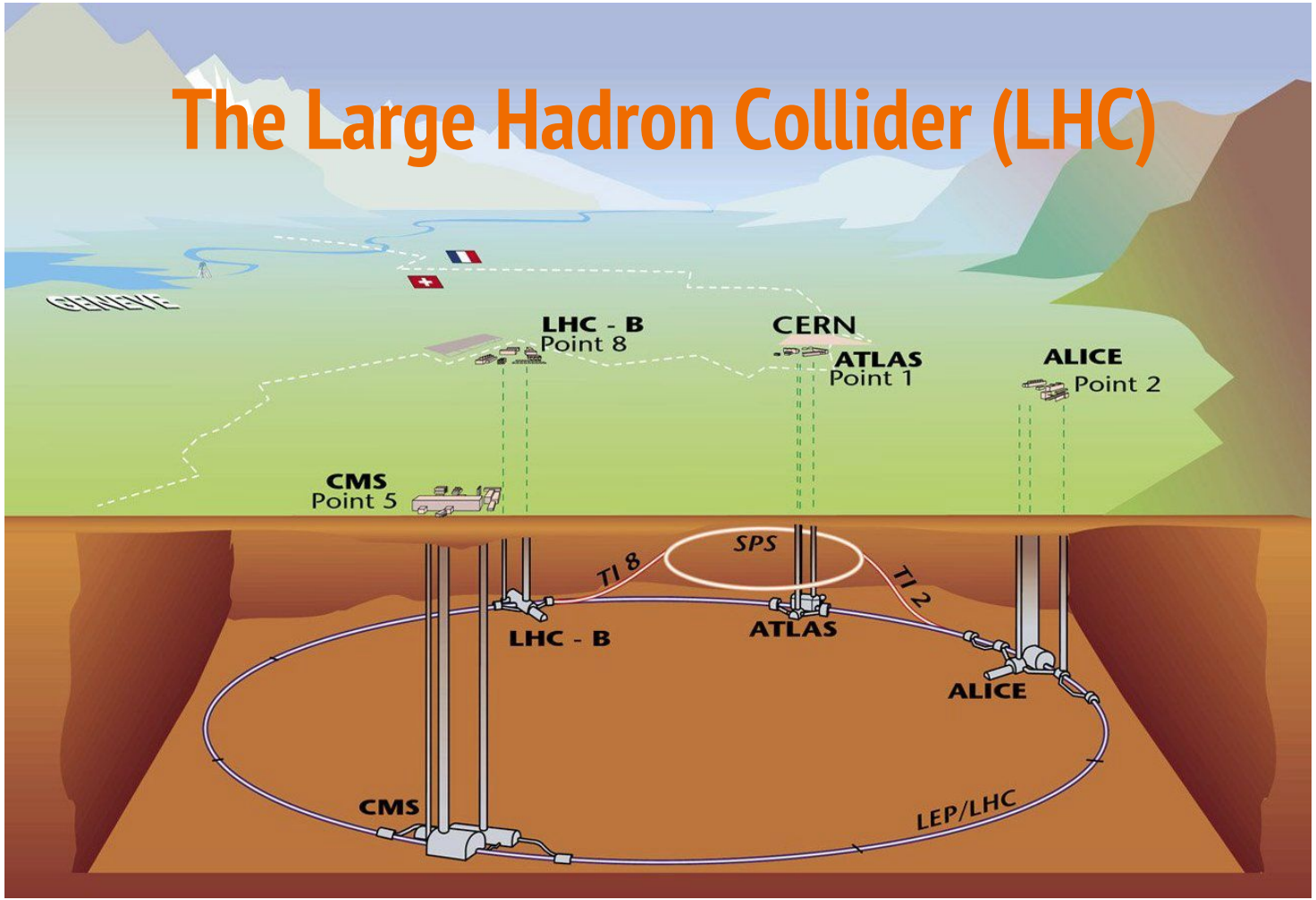
Prologue: Rucio

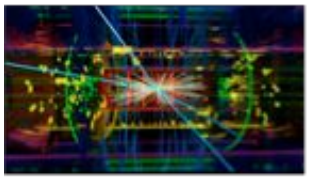
Once upon a time

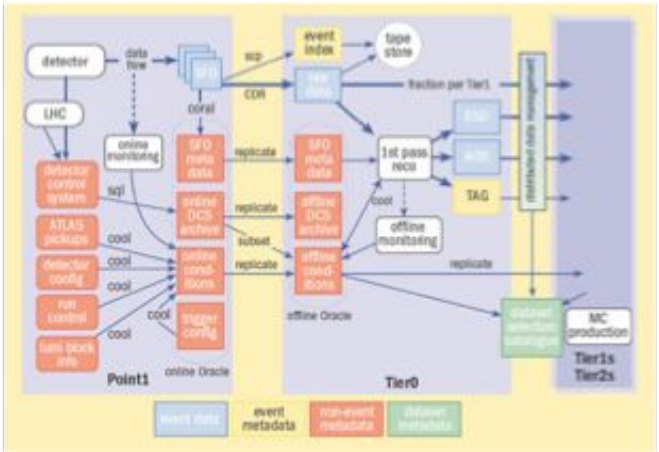
The Large Hadron Collider (LHC)



What is the data for ATLAS?

- C++ objects representing tracks, parts of detector etc, saved in immutable files
- Data is reconstructed and reduced through various formats
  - RAW ⟶ AOD ⟶ NTUP





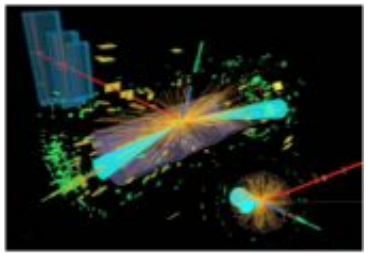


Figure from http://cerncourier.com/cws/article/cnl/34054

Data Management Tools

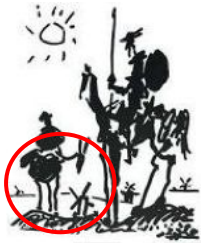
- At time of inception, no global/commercial solution for the distributed computing available for our ‘Big Data’ handling
  - A data intensive instrument which generates unprecedented data volumes
  - Facilities are distributed at multiple locations under different administrative domains
  - Data is produced at many locations where it is neither stored, nor analyzed by researchers nor archived
- ATLAS developed its own tools
  - The first implementation of the data management system was Don Quijote 2 (DQ2)
  - In production from 2006 : Originally designed as a transfer system
  - 2007-2013: Many new features added during LHC Run-1

DQ2 Limitations

- DQ2 would have not scaled for LHC Run-2
- Heavy operational burden
- Difficult to add new features and technologies
- Many lessons learned during Run-1

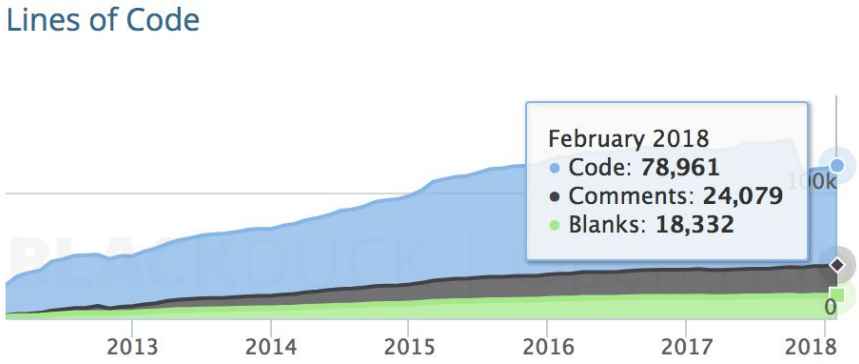
➟ The Rucio software project was developed during Long Shutdown 1 to address the challenges of Run-2 and beyond!

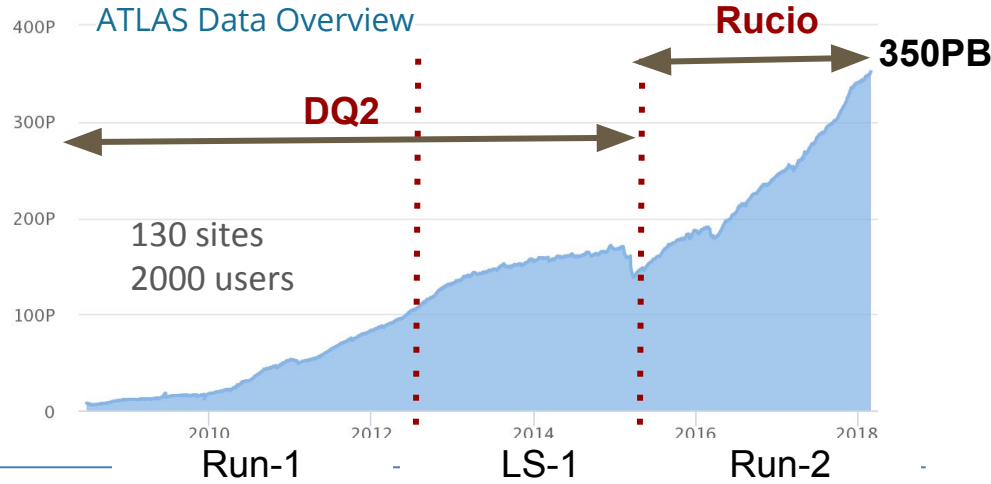
  - 'El rucio' is Sancho Panza's Donkey
  - It’s not really a name more a description which suggests a dappled colour



Rucio Development & Commissioning

- Long initial process:
  - 2012: User surveys, technical studies & design phase ~1 year
  - 2012-2014: Initial development ~2 years
  - 2015: Commissioning & gradual migration from predecessor system DQ2 ~1 year





Rucio Deve

- Long initial p
  - 2012: Use
  - 2012-201
  - 2015: Com

In a Nutshell, Rucio...

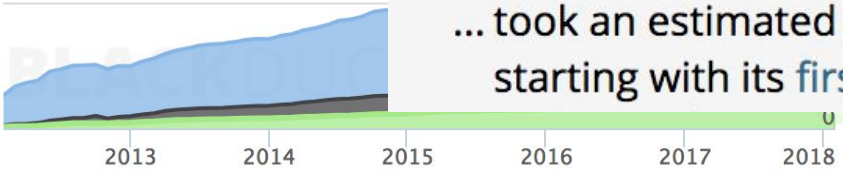
... has had 6,084 commits made by 32 contributors representing 78,961 lines of code

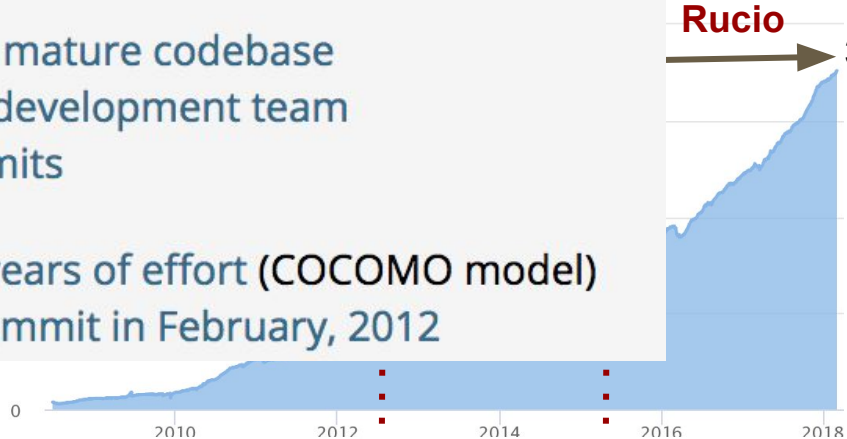
... is mostly written in Python with an average number of source code comments

... has a well established, mature codebase maintained by a large development team with stable Y-O-Y commits

... took an estimated 20 years of effort (COCOMO model) starting with its first commit in February, 2012

Lines of Code

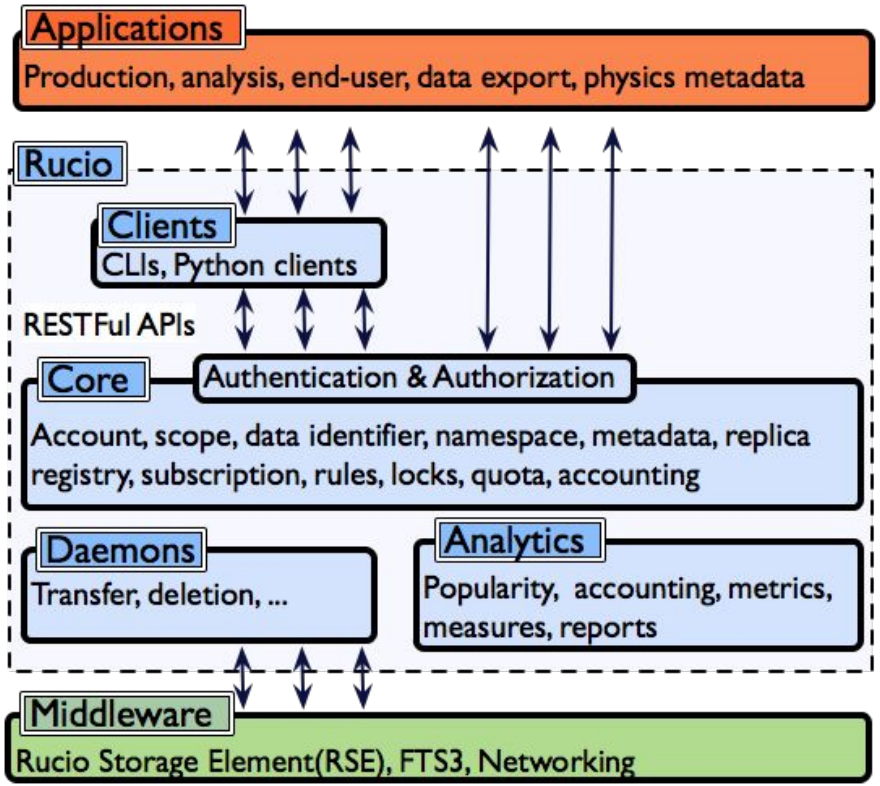




Rucio in a Nutshell

- Rucio provides a complete and generic scientific data management service
  - Designed with more than 10 years of operational experience in large-scale data management!
- Rucio manages multi-location data in a heterogeneous distributed environment
  - Creation, location, transfer, and deletion of replicas of data
  - Orchestration according to both low-level and high-level driven data management policies (usage policies, access control, and data lifetime)
  - Interfaces with workflow management systems
  - Supports a rich set of advanced features, use cases, and requirements
  - Large-scale and repetitive operational tasks can be automated

Quick look into the Architecture



Fully built on open standards and frameworks!
To follow the advances with a flexible design with no dependence on particular implementation

- **Servers**
  - HTTP REST/JSON APIs
  - Token-based authentication
  - Horizontally scalable
- **Daemons**
  - Orchestrates the collaborative work e.g., transfers, deletion, recovery, policy
  - Horizontally scalable
- **Persistence**
  - Object relational mapping
  - Oracle, PostgreSQL, MySQL/MariaDB, SQLite
- **Middleware**
  - Connects to well-established products, E.g., dCache, EOS, S3, ..

Rucio Concepts - Table of Contents

Questions ?

If you have a question but don't get the chance to ask it directly during the session, you can do it here: https://goo.gl/BdSGoC