

Object-based Storage Integration within the ATLAS DDM System

Wen Guan, Vincent Garonne
on behalf of the ATLAS Collaboration



The ATLAS Distributed Data Management (DDM) System

- The Distributed Data Management project is charged with managing all ATLAS data on the grid
- All for the purpose of helping the collaboration store, manage and process LHC data in a heterogeneous distributed environment, like:
 - Transfer data to/from sites
 - Delete data from sites
- The Integration of new storage types, like object-based storage, is a constant need

Object-based Storage & Use Cases

- The vast majority of cloud storage available in the market leverages an object-based storage architecture.
- Today, we have several object-based storage available for ATLAS:
 - BNL (Amazon, Ceph), Lancaster (Amazon), RAL (Ceph), CERN (Ceph), MTW2(Ceph)
- To start with, we integrated object-based storage for two use cases:
 - Log files management
 - The ATLAS Event service

Integration Within DDM/Rucio

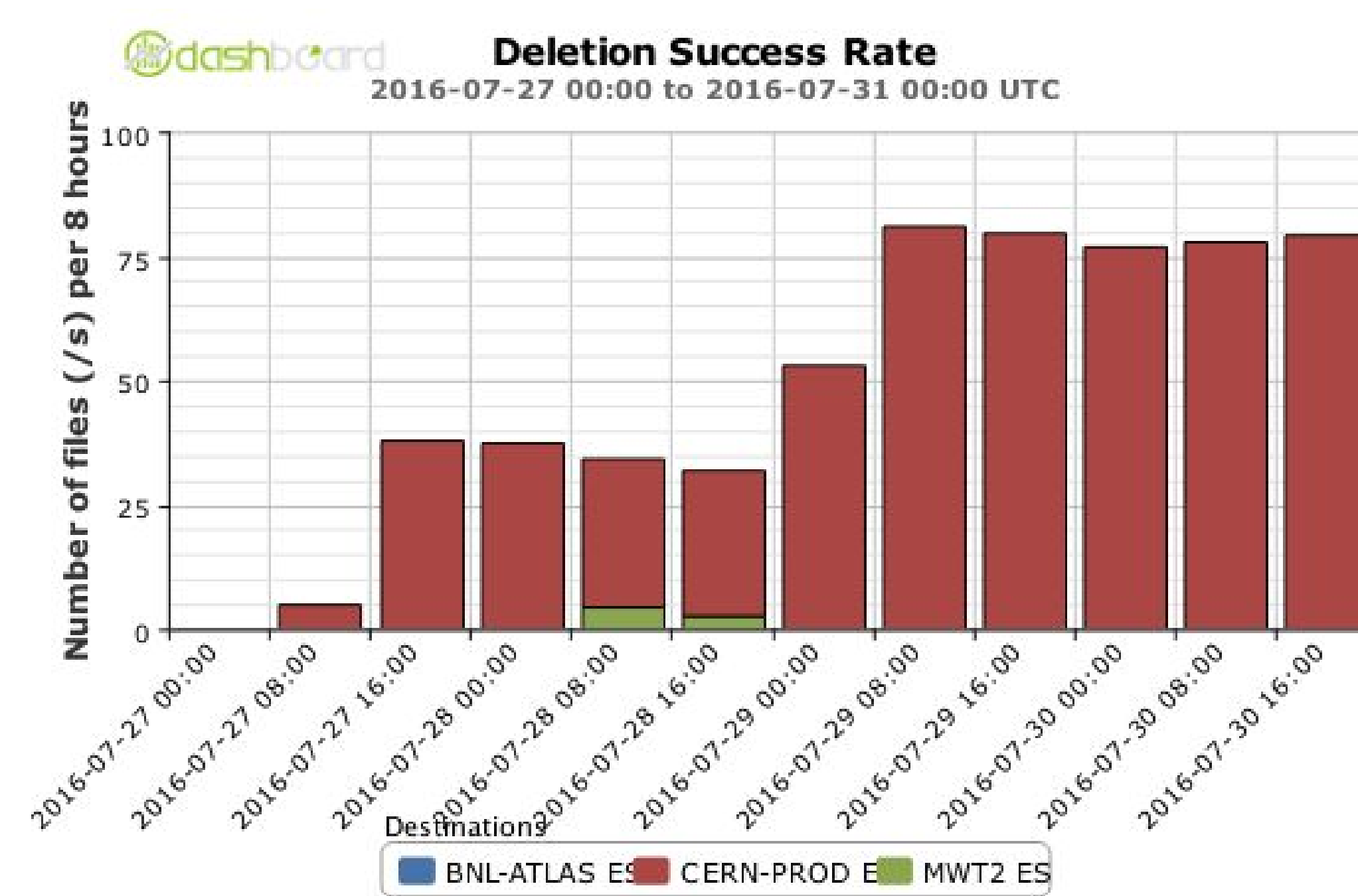
- The main idea was to integrate object-based storage as regular DDM and focus on the Amazon S3 (Simple Storage Service) protocol
- Rucio allows the integration of non posix namespace as `s3://<bucket>[/<prefix>]/<object>`
- We have a flexible mapping protocol-endpoint within DDM/Rucio
- Object-based storage supports other authentication mechanisms than X509
 - Usually based on access and secret Keys

References / Contact

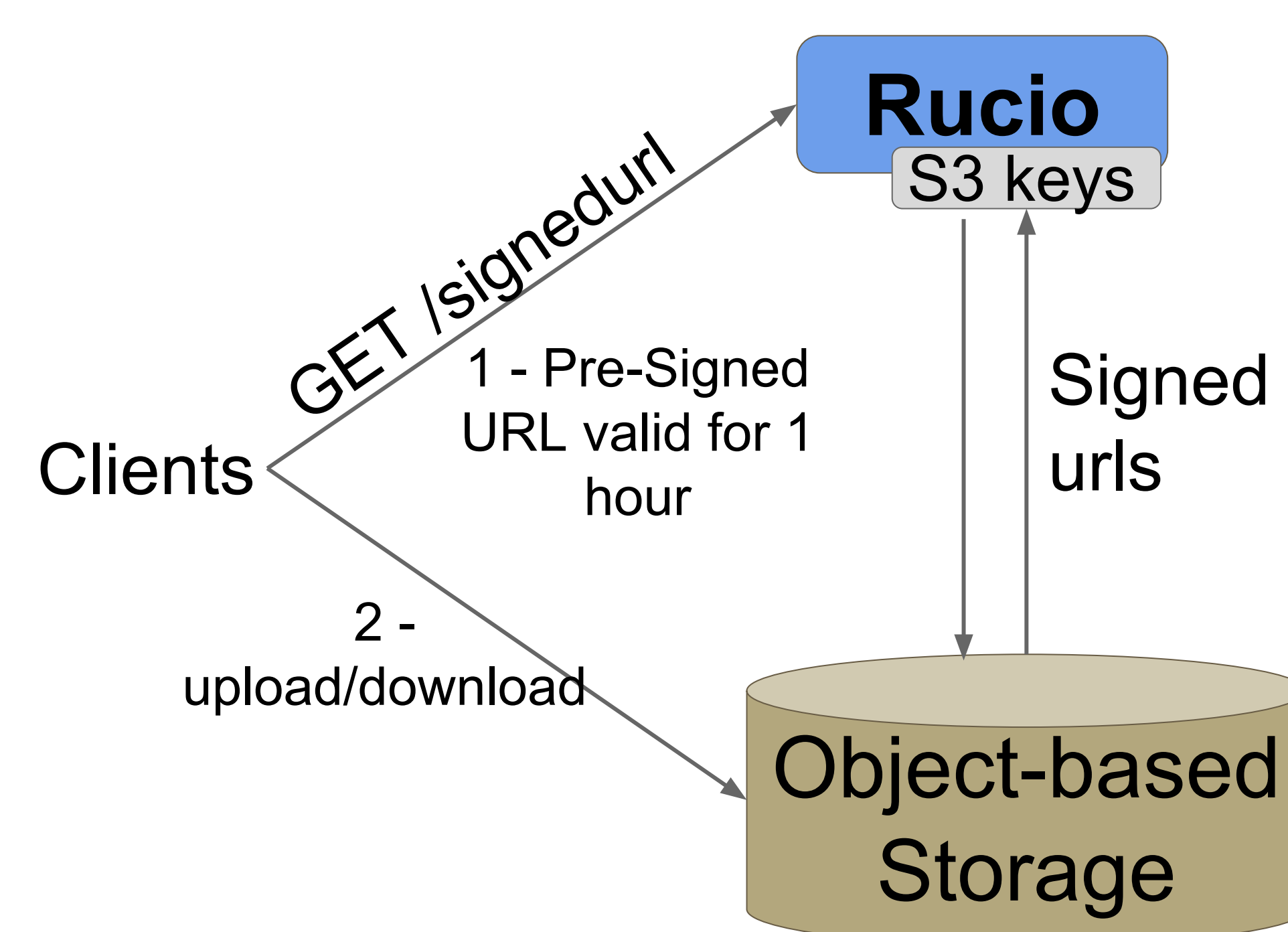
<https://rucio.cern.ch>
rucio-dev@cern.ch

Implementation

- Two protocols have been implemented:
 - `s3://` and `s3+rucio://`
- `s3://` has been implemented with the boto python library and requires to have the Access and Secret keys available
- It is the preferred way on central machines where we have access to them in a secure way, e.g., central deletion



- `s3+rucio` has been implemented with pure http python library and relies on Pre-Signed URLs
- A pre-signed URL gives you (temporary read/write) access to the object



1. is a secure operation and requires to be authenticated with Rucio with no need to distribute the S3 keys
2. is a simple http PUT/GET operation in the pre-signed url