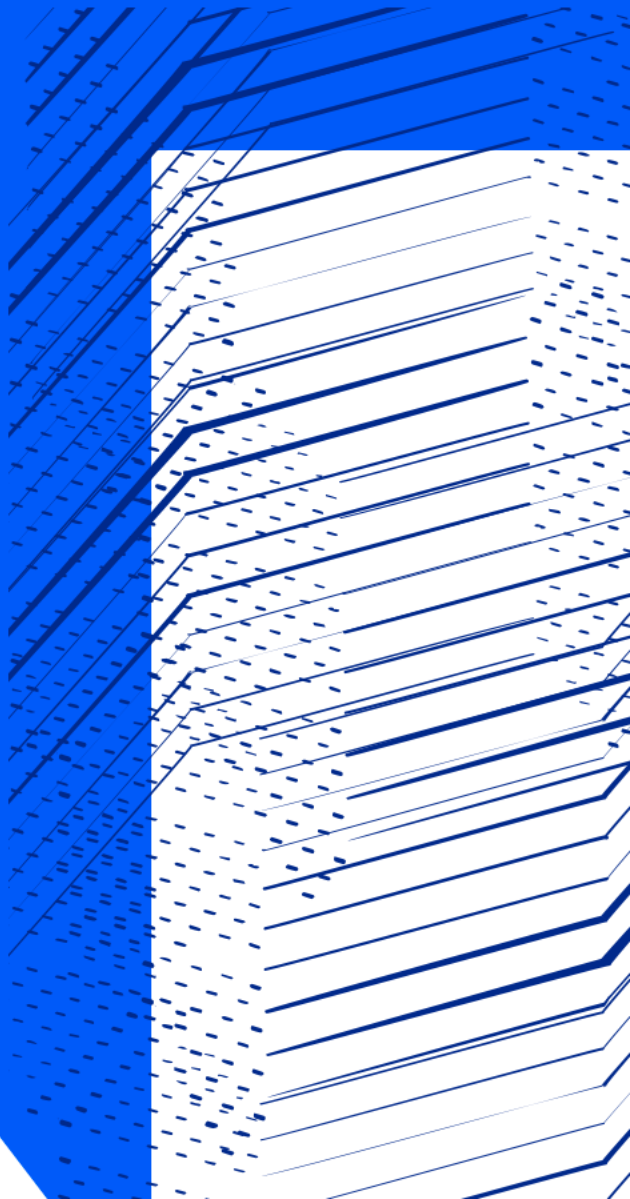
Science and Technology Facilities Council

Globally Accessible Data

Alastair Dewhurst

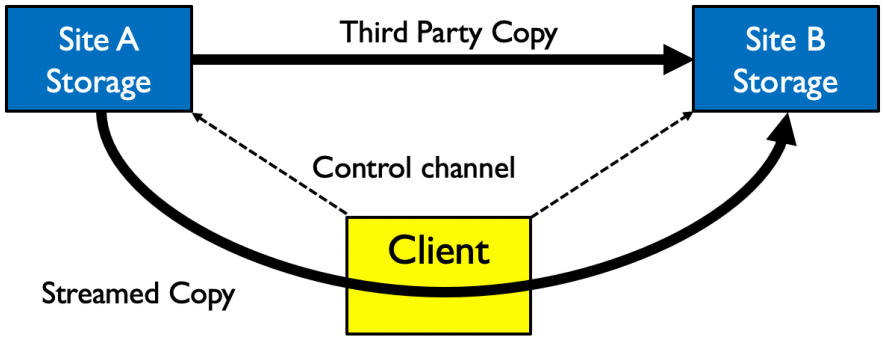


Introduction

- Why can't we just have a globally accessible file system?
  - The semantics of a POSIX file system are not compatible with highly scalable systems.
  - In a research collaboration there is an added complication of needing to rely on multiple heterogenous resources.
- Compromises must be made:
  - Highly scalable → Object Storage, reduces allowed operations (get, put, delete).
  - Read Only → CVMFS has many layers of caching because we don't need to worry about multiple sources of writes.
  - Simultaneous writes → Services like Google Docs or Github.

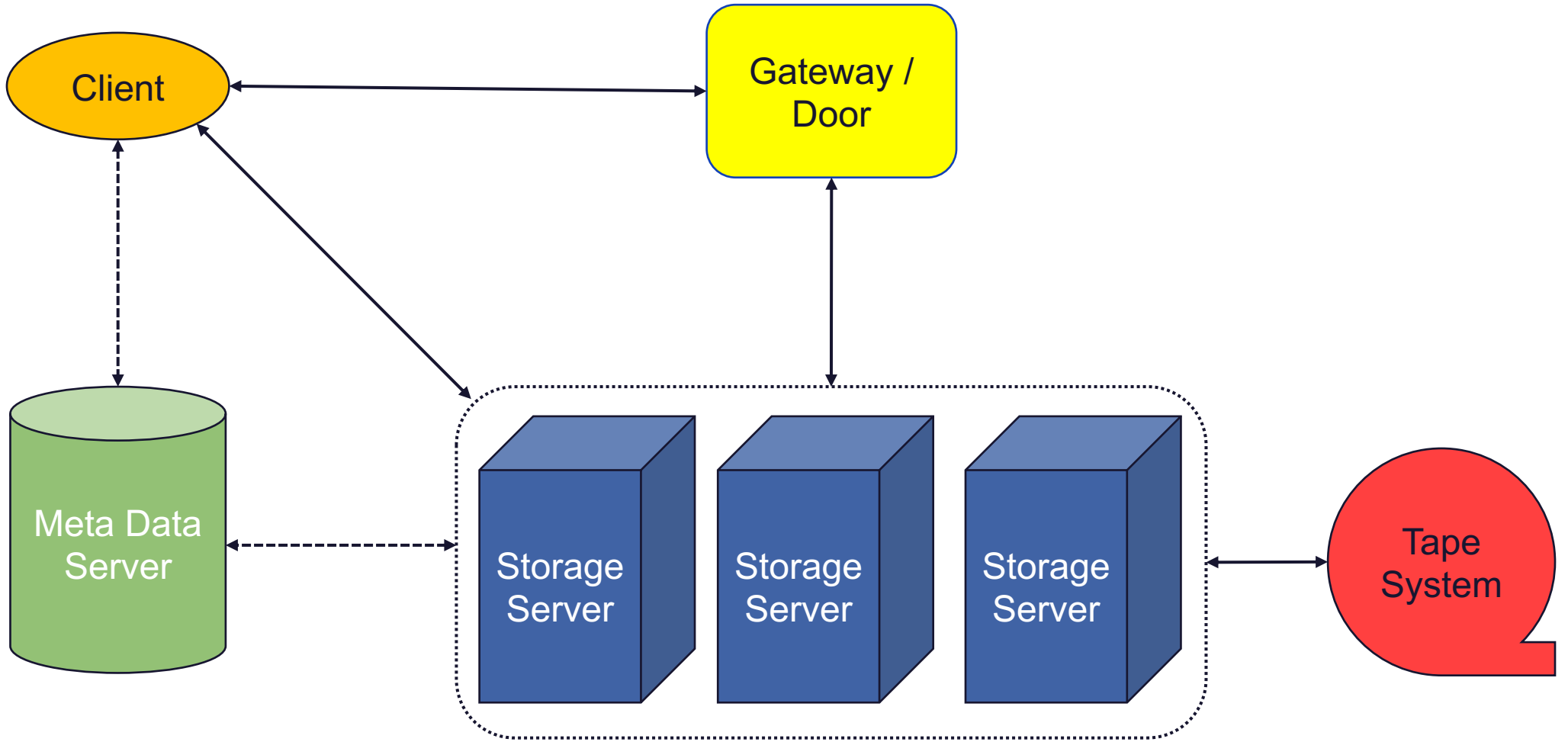
Globally accessible data

- To provide globally accessible data we need to solve two problems:
  1. Store large amounts of data.
     - Ensure data integrity.
     - Provide parallel access to this data.
  2. Make it globally accessible.
     - Provide an authentication and authorization system.
     - Allow third party copies to take place.



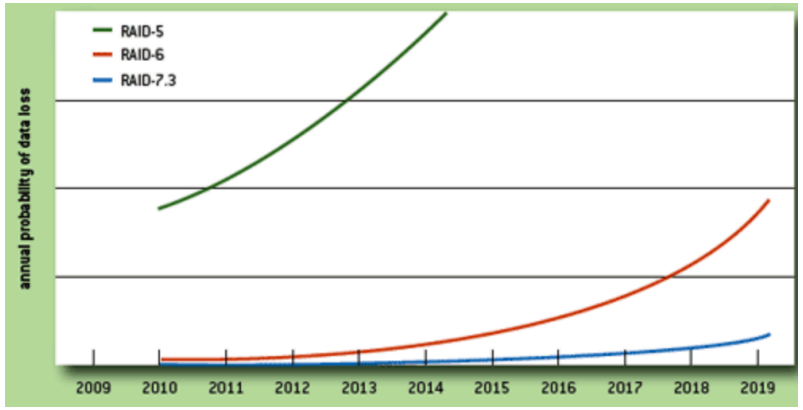
- Some systems try and do both, others are built from multiple different systems.

Generic Storage Architecture



Data Integrity

- In any mass storage system, hardware failures should be considered part of normal operations.
  - At RAL we have over 6000 HDD across ~250 storage nodes in production.
- Solutions:
  - Hardware RAID. RAID6 has provided a cheap and reliable way to ensure data integrity for the last 20 years. It is reaching the end of its useful life. In the last 15 years capacity of HDD has increased by a factor of 30 while performance has remained roughly constant.
  - Replication. Multiple copies is a simple and effective way to ensure data integrity but is expensive.
  - Erasure Coding can be thought of as an extension of RAID and can be done at the software level. The parity chunks can be spread across storage devices avoiding the draw back of hardware RAID.



http://queue.acm.org/detail.cfm?id=1670144

Grid Storage

dCache was originally designed as a system to manage disk caches in front of a tape system. Strong development team has allowed it to evolved to meet many different requirements.

DPM was designed to just manage disk and is simpler to run than dCache. It is still widely used but CERN are slowly phasing out support.

EOS is CERN's current disk based storage service. EOS manages the data on the individual disks. It's excellent if you have a similar use case to CERN.

CTA replaces Castor as CERN's archival tape service. Uses EOS as a frontend to transfer data to other storage endpoints.

StoRM is designed as a frontend for sites running a shared file system as backend storage.

XRootD is both a protocol and a storage service that can provide a Grid Interface. With the phasing out of SRM it has a much more viable option.

Backend Storage

Ceph is a storage service that provides object, block and file level interfaces. Designed without single points of failure.

Lustre, GPFS or other shared filesystems have been providing reliable performant solutions for a long time.

HDFS is designed for streaming data from large datasets and to recover quickly from hardware failure.

If the frontend storage is managing the hardware, ZFS is an improvement over relying on hardware RAID.

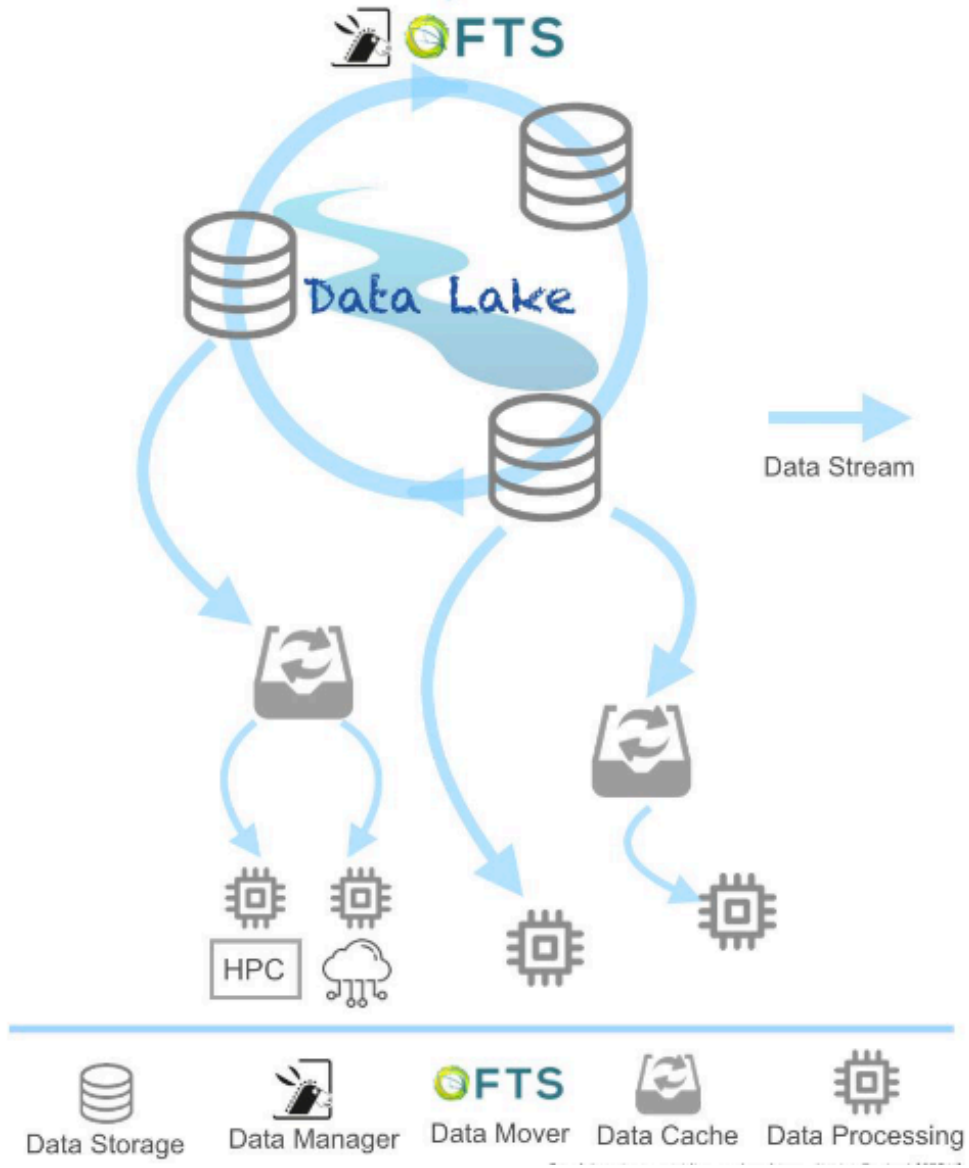
Other Storage Services are available…

Evolution of Grid Storage

- The way data is accessed over the Grid has been constantly evolving.
- In 2006 it was simple:
  - All Storage Elements will have an SRM with a uniform API to use.
  - Authz is provided by X.509.
- SRM provided functionality but not consistent performance.
  - Experiments started talking directly to the storage making use of a variety of protocols.
- X.509 was not user friendly and things like voms were a development cul-de-sac.
- Being able to respond to changes was a key strength of a Grid storage service.

Future direction

- The WLCG is working on a Data Lake concept.
- Storage requirements are tending towards object stores with meta data handled by Rucio.
- Different Quality of Service can be taken into account for different storage types.
- Focus on building the right storage for the data access use case.



UKRI Science and Technology Facilities Council

Questions?

Quality of Service

- For a certain cost Storage can be defined by three properties.
- Capacity: How much data can be stored.
- Throughput: How much data can be transferred per second.
- Latency: How quickly can it respond to queries.