US CMS Data Lake Proposal

OSG All Hands Meeting
September 4 2020

Edgar Fajardo, Frank Wuerthwein, Harvey Newman, Justas Balcas

The Challenge: Prototyping a CMS Data Lake

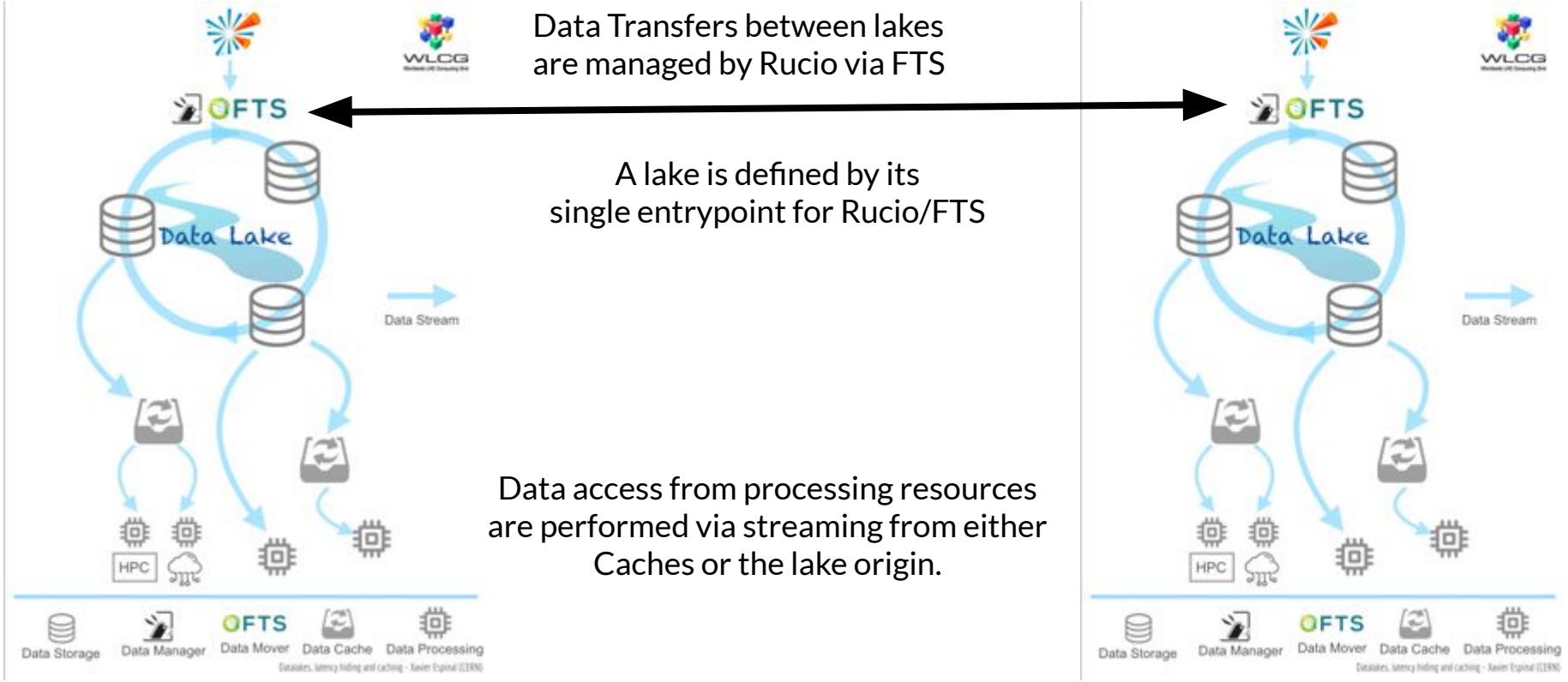
Introduction

- WLCG DOMA Lakes Data Challenge specified a challenge including some milestones: WLCG DOMA Lake Data Challenge: towards a prototype
- It includes both (ATLAS and CMS) and a EU data lake and a US Data Lake
- In these slides we describe our interpretation for a US CMS Data lake proposal.

What is a data lake?

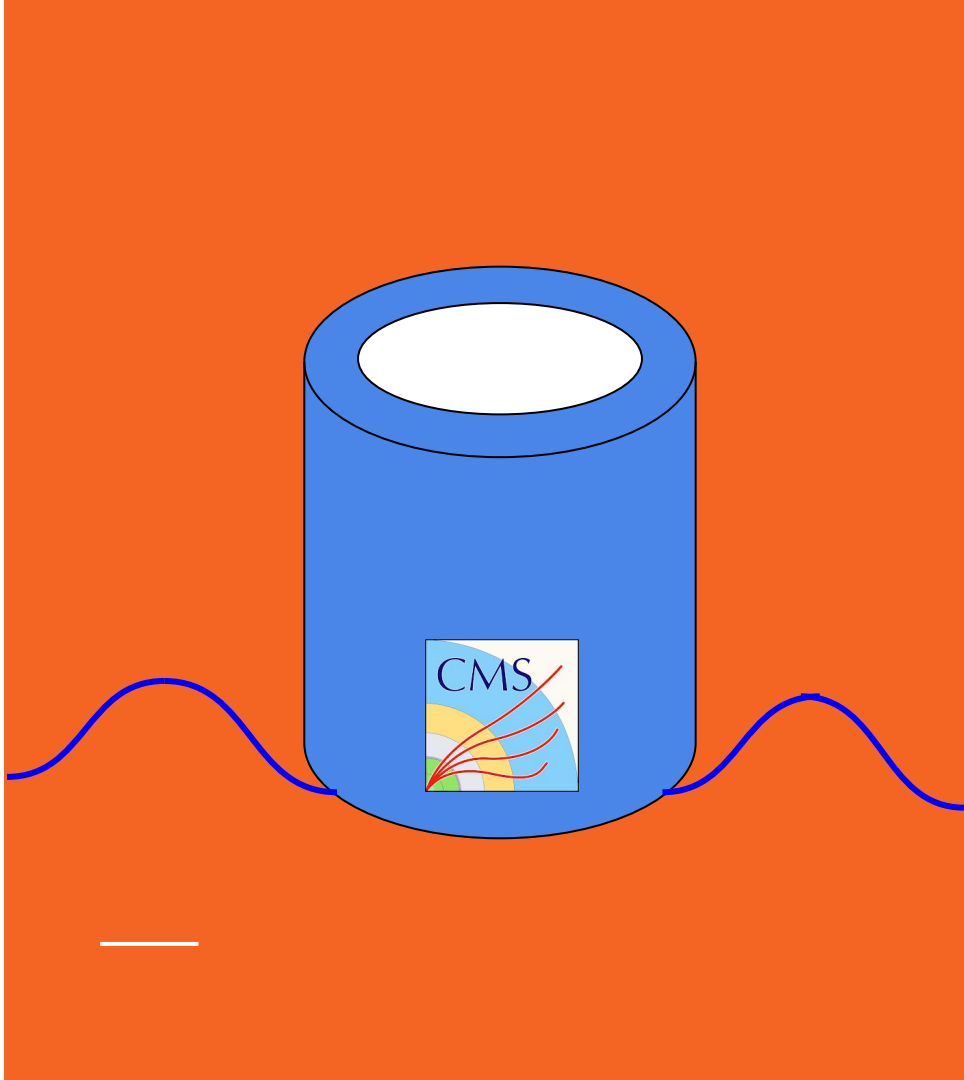
Quick answer: We will follow the WLCG definition.

WLCG Data Lake Definition



Proposed CMS Data Lake implementation

Disclaimer: The devil is on the details



Some choices we have to make to implement a prototype

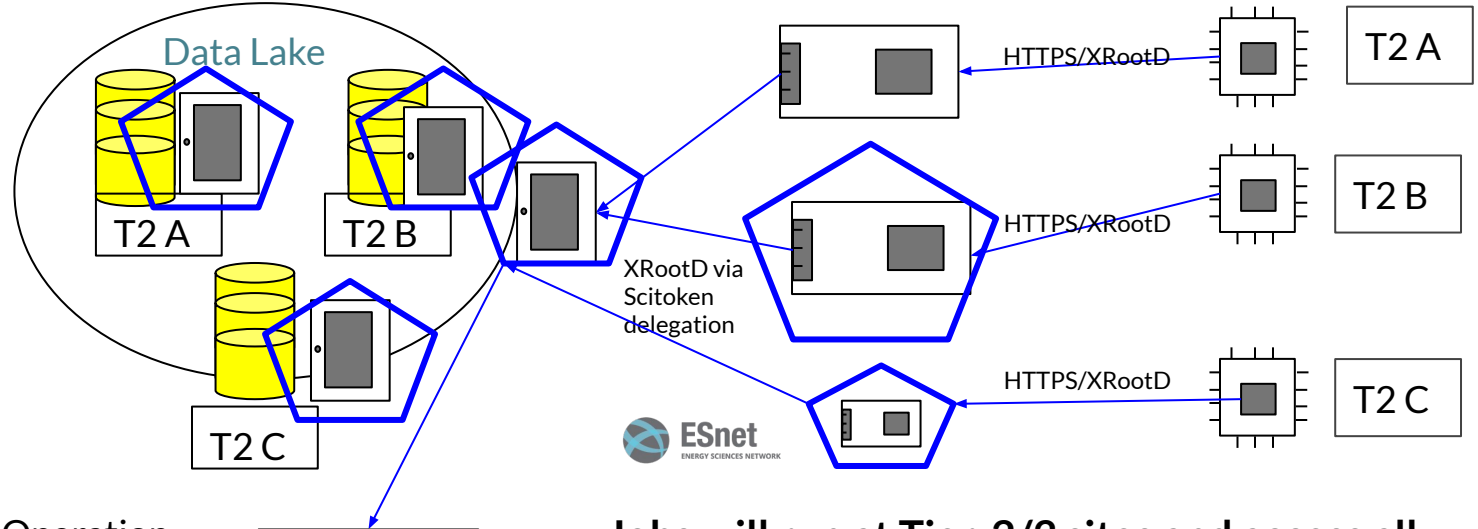
1. How does the data get into the data lake?
2. What keeps track of what is in the data lake?
3. How do applications pick which cache to use?
4. Which datatier(s) go into the lake?
5. How is the lake deployed?
6. Authentication between users and lake?

Our proposal

1. How does the data get into the data lake?
   a. Offer a single HTTPS interface that RUCIO can command to COPY, MOVE, LIST, DELETE
2. What keeps track of what is in the data lake?
   a. Rucio
3. How do applications pick which cache to use?
   a. GeoIP (pick closest geographical cache)
4. Which datatier(s) go into the lake?
   a. NANOAOD/SIM
5. How is the lake deployed
   a. XRootD doors and caches via Kubernetes,
6. Authentication between users and lake?
   a. Scitokens

We propose to ignore tape archival for this prototype.

Accessing data in lake for processing



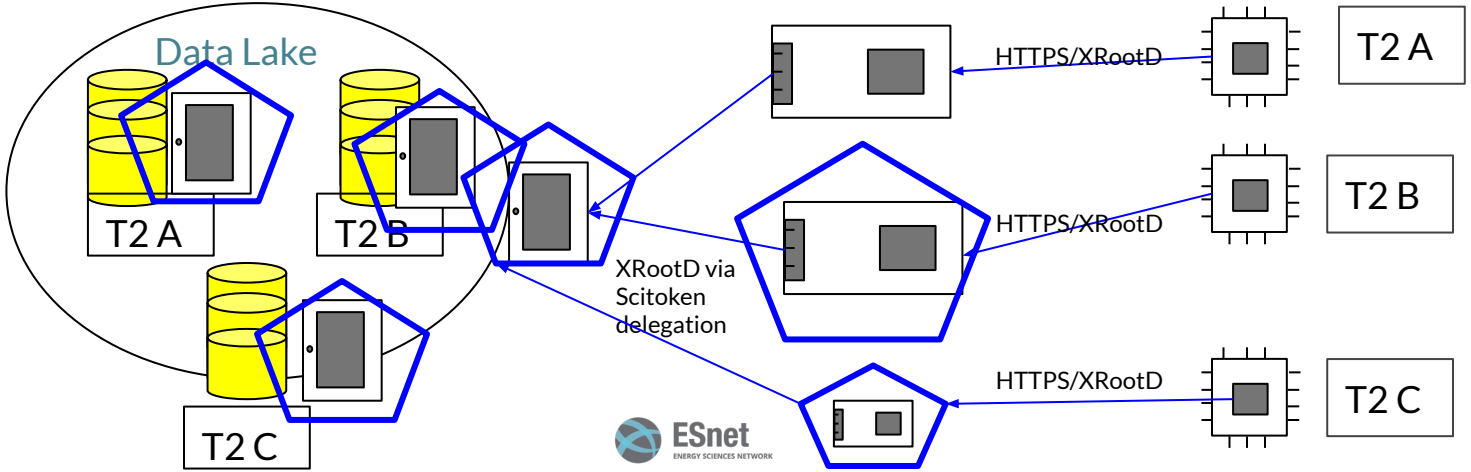
Reading Operation

User Task

Cache

- **Jobs will run at Tier-2/3 sites and access all the data via the closest cache.**
- **The closest cache might not be local to the site.**
- **It might be in the Internet backbone.**
- **Enable redirection to CMS Data federation???**

Accessing data in lake for processing



Reading Operation

User Task

Cache

- **Jobs will run at Tier-2/3 sites and access all the data via the closest cache.**
- **The closest cache might not be local to the site.**
- **It might be in the Internet backbone.**

How does this differ from today?

- Today: Each T2 and T1 is its own data lake.
- For HL-LHC: The entire US T1/T2 infrastructure could be one lake

- Advantages:
  - No replication within the US
  - All access relevant replication is via caches
    - Can be minimal in size => trade-off between network use and disk purchases
    - Cache sizes can be chosen to compensate for network bandwidth
  - We could internally perform optimizations that are not visible from the outside.
    - We can deploy data origins of different sizes at different locations., and change this as we see fit without it being visible at all to global CMS.
      - E.g. all origin storage at FNAL, all CPU at T2
        - Saves money as T2s don't pay power nor IDC.
      - Allows FNAL to consider optimizing disk vs tape, or some other cheap media.
    - We could even chose optimizations internally between different types of hardware implementations, e.g. JBOD, erasure encoding, replicated storage.

How Tape Archives fit in the model

There are two extreme ways of fitting tape in:

1) Each QoS, e.g. tape archive including its buffer space, is its own data lake.
   a) This implies that all accesses to/from archive are managed by Rucio/FTS.
   b) Inversely, if accesses to tape are intended to be managed by Rucio/FTS then the archive is its own lake.
2) The entrypoint supports multiple QoS.
   a) This implies that Rucio/FTS decides tape vs HDD vs SSD/NVME as well as custodial vs replica via the QoS flag of the data lake entrypoint.
   b) It allows the archive to be directly accessed from processing centers, e.g. via the dCache buffers in front of the archive also being origins for the XRootd data federation accessed via either XCache or streaming from processing centers.

We propose to ignore tape archival for this prototype.

It is up to US CMS which option we want to offer to global CMS

Benchmarking Goals

1. Exercise deletions and measure missed deletions as a function of:
    a. Scale
    b. Disconnecting an XRootD origin
2. Exercise data input and data removal via FTS
    a. Scale
    b. Success rates
3. Exercise NanoAOD application access.
    a. Recruit students and postdocs with realistic applications
    b. Cpu efficiency as a function of RTT to the closest cache.
    c. Data access pattern (?to be thought about more carefully?)

Proposed Timeline for prototype Deployment

Task	By
All hardware for prototype in Kubernetes cluster	September 2020
Setup the XRootD origins and configure them with a the data lake single entry point.	October 2020
Configure caches to read from the data lake and use Scitokens for authentication.	November 2020
Setup a site (RSE) in RUCIO (UST2DataLake) and have register all NANOAOD to it.	December 2020
Setup submission infrastructure to be cache aware.	January 2021
Data lake testing, benchmarking and DevOps	January 2021 - September 2021

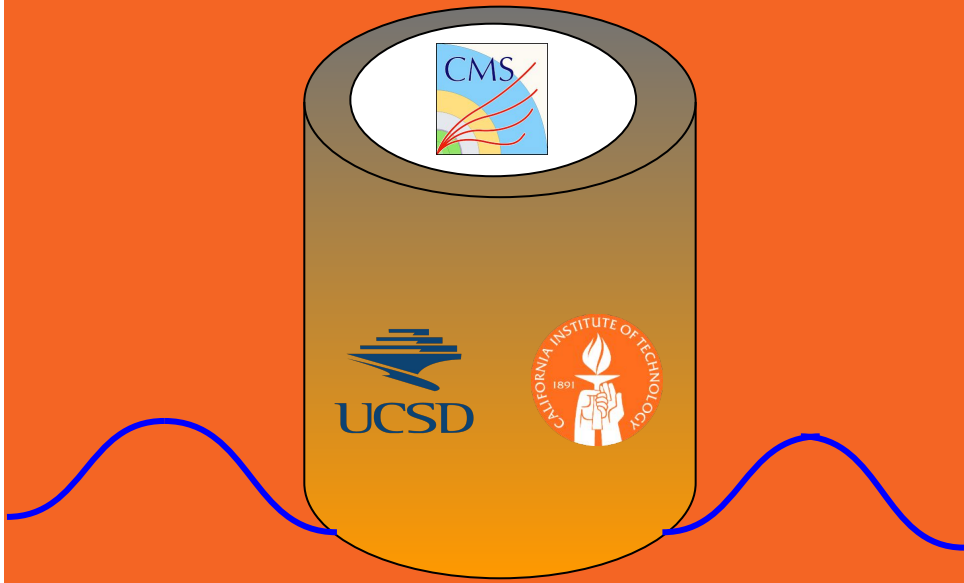
Concerns

1. Consistency: Dirty deletes. If FTS asks for a deletion of a file and the XRootD server at that site was down the data lake can end up with files in server and Rucio does not know about them.
2. We will give users a built in env variable (XCACHE_CLOSEST) to point to the nearests cache but there is some sociological aspect to get users to transition to it.
3. Our implementation does not have data movement inside the lake. So initial placement based on available space (cms.schedd) will be crucial to work.
4. Metrics of success are fuzzy.
    a. More thought required.

Assuming Prototype is successful

Merge Caltech and UCSD CMS Namespaces into one

(i.e. turn prototype into production system in SoCal to gain experience during Run3)



SoCal Data Lake

Objectives

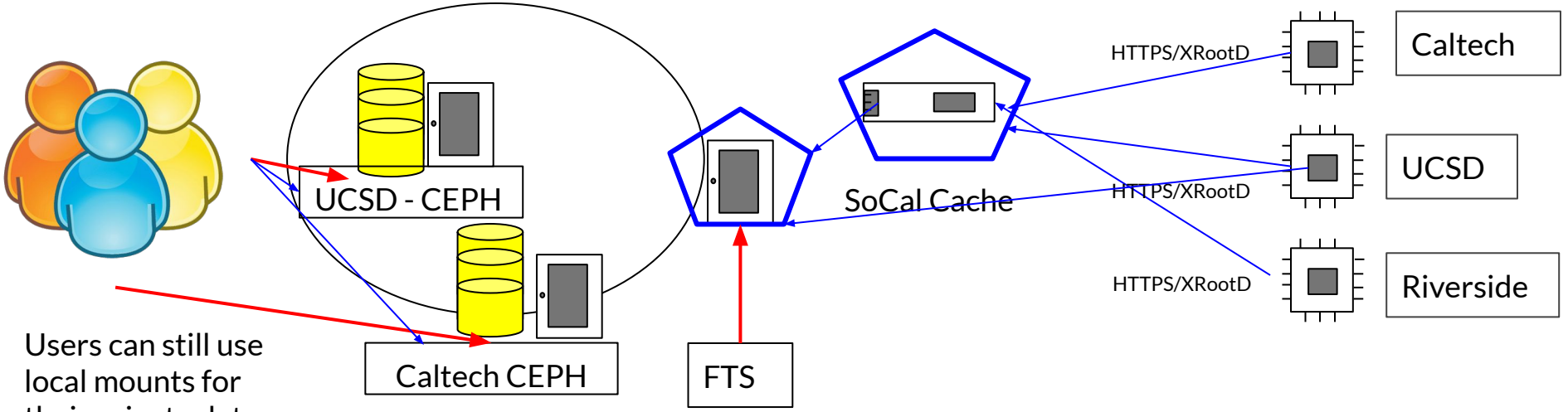
1. Merge the CMS namespace of Caltech and UCSD (/store)
2. Transform the lessons learned from the data lake prototype into a production data lake for Run 3.
3. Free up space (for the datasets that are hosted both at UCSD and Caltech, for example: MinBias)

Assumptions

1. Rucio will be managing all CMS namespace at sites
2. The US T2 data lake prototype is successful
    a. Goals of pilot is to measure scalability necessary to be a SoCal production system.
3. We can route all global pools jobs to UCSD or Caltech if the dataset is at the site SoCal Data Lake.
4. What happens to user space?
    a. Requires more thought

How would the SoCal lake look like?

SoCAL data lake proposed architecture



Users can still use local mounts for their private data: /store/user/foo

Questions?

Supporting Slides

Hardware requirements

1. **Caches**: Ideally three.
    a. We would like to deploy caches at Nebraska, Purdue or FNAL, and UCSD or Caltech.
    b. 20 TB per cache
    c. At least 10 Gbps connection to the WAN.
2. **Origins:** Two or more sites to set up (50 TB/site) of hardware in one or more machines and connect them to a Kubernetes cluster. (NANOAOD total is about ~100TB).
    a. Looking for volunteers (We will help connecting to k8s cluster).
    b. 10 Gbps connection to the WAN.