

# Life, the Universe and Data Lakes

Tigran Mkrthyan for dCache Team  
WLCG Workshop, Naples, 28.03.2018



# Overview

- Who we are
- Where we are
- Where we go
- Summary

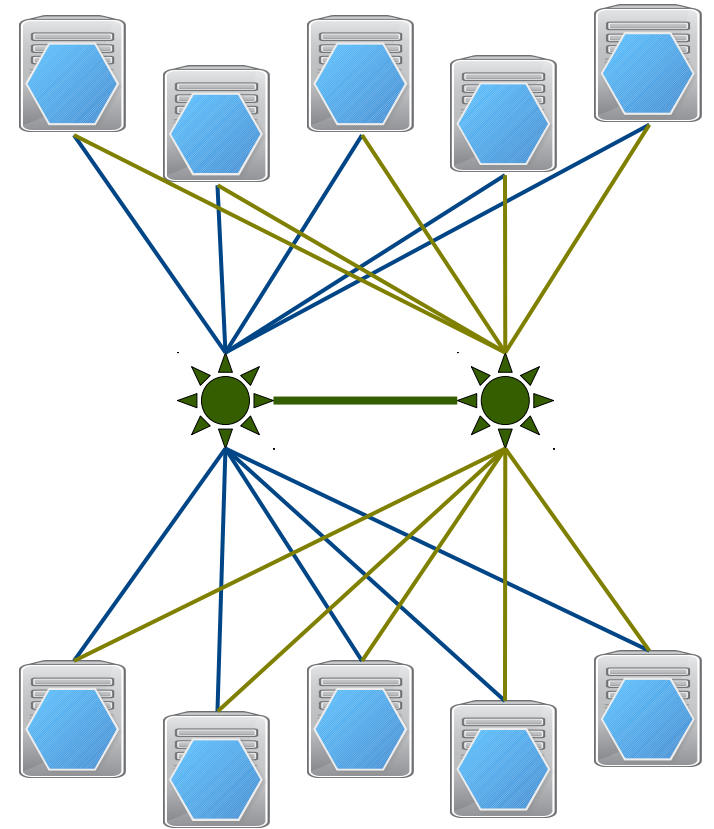
# About dCache

- Joined effort between DESY(2000), FNAL(2001) and NDGF(2006)
- Provides distributed storage for scientific data
- Supports standard and HEP specific access protocols
- Supports standard and HEP specific authentication mechanisms
- Used around the world to store HERA, Tevatron, LHC and others data



# Internals

- Independent components (services)
- Multiple instances of the same component can be started
- Inter-component messages based communication
- Can be updated/restarted independently
- Fault tolerant



# Fault tolerance

- All core services can run multiple instances (replicable)
  - Namespace
  - Pool Manager
  - Space Manager
  - SRM
  - *you name it*
- Door/Pool crashes can be handled by clients
  - NFS
  - dcap
  - xrootd
- Master/slave (multimaster) namespace DB configuration required
  - dCache automatically detects which node runs as **master** when multiple provided

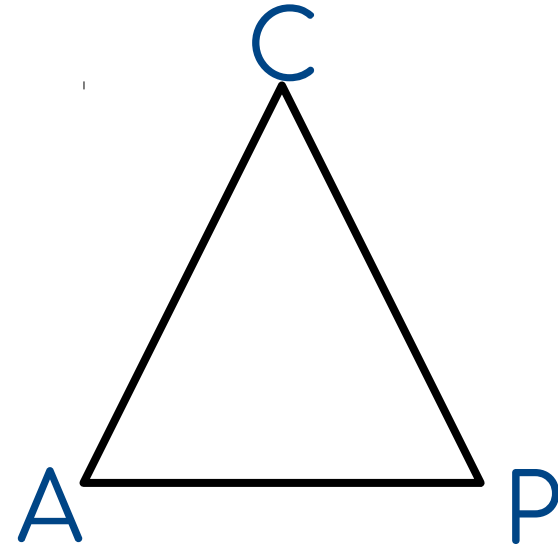
# The CAP Theorem:

You can have at most two of these properties for any shared-data system

**C**onsistency

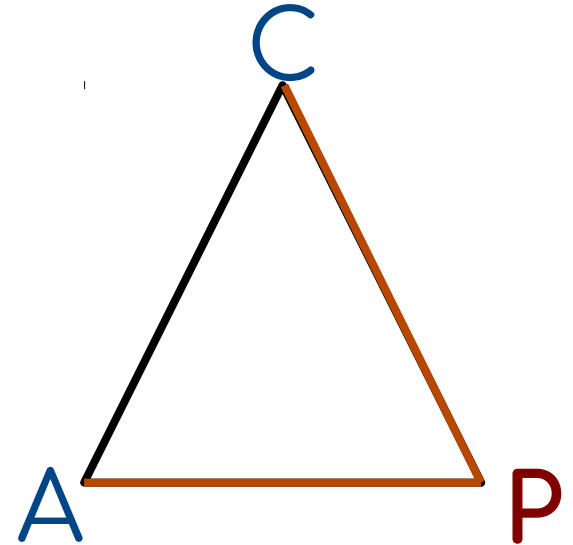
**A**vailability

**P**artition tolerance



# CAP Explained

- No distributed system is safe from network failures.
- In the absence of network failure both availability and consistency can be satisfied.
- The choice is between consistency and availability only when a network partition or failure happens.



# dCache and CAP

- dCache provides consistency over availability.
- All client will see the same data at the same time.
- A timeout or error will be returned, if consistency can't be guaranteed.

## eulake R&D goals

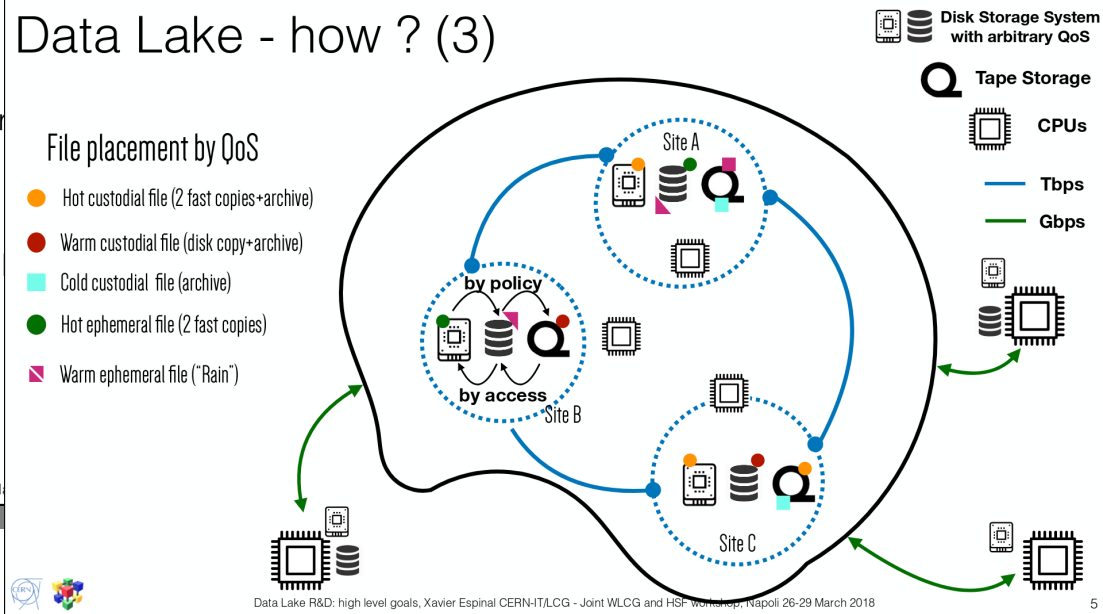
- Our R+D project aims to demonstrate that a dynamically distributed storage system with a common namespace:
  - Has the potential to **lower the cost** of stored data.
  - Has the potential to **ease (local) administration** and work
- The R+D should also demonstrate:
  - That the **efficiencies** in performance, reliability and resili
  - The **compatibility** with HL-LHC **computing models**.



Data Lake R&D: high level goals, Xavier Espinal CERN-IT/LCG - Joint WLCG and HSF workshop, Napoli

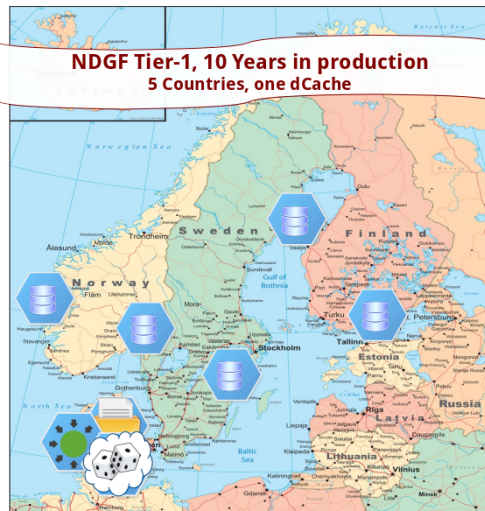
*Shameless stolen from Xavier*

## Data Lake - how ? (3)



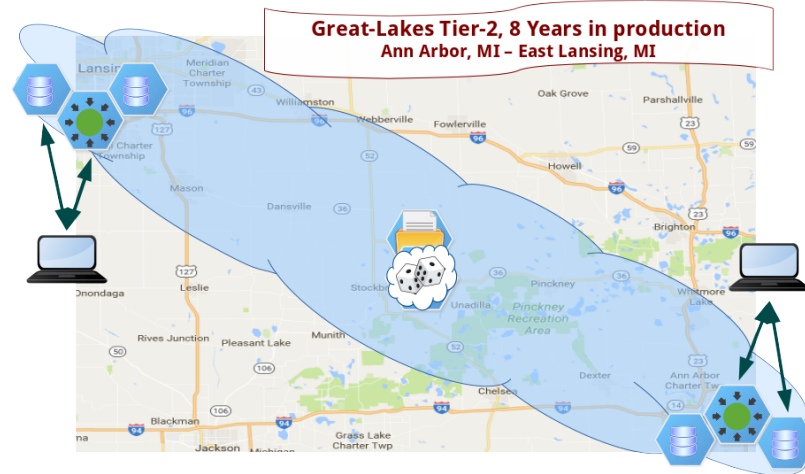
## Multi-Site deployment

- Distribute data over multiple locations
- Multiple administrative domains
- Use available resources



See: WLCG workshop in Manchester

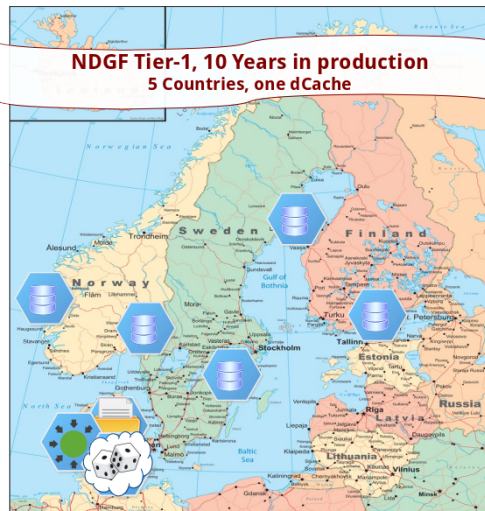
## Multi-Site deployment



# Data-lake/fjord/glacier?

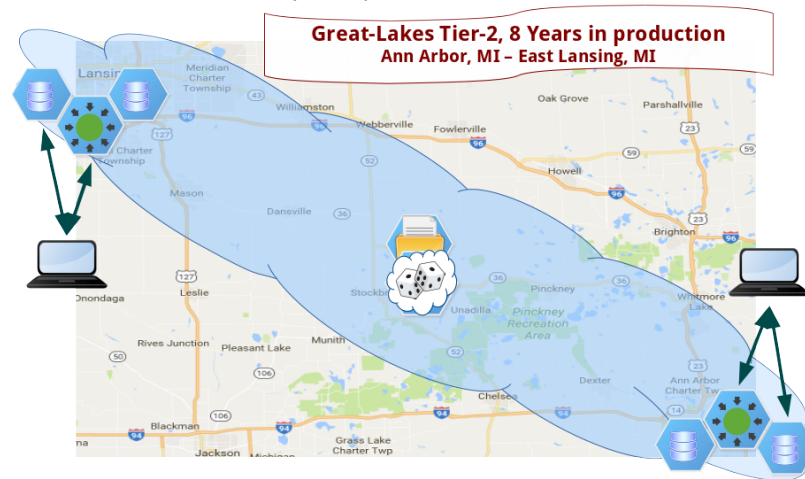
## Multi-Site deployment

- Distribute data over multiple locations
- Multiple administrative domains
- Use available resources



See: WLCG workshop in Manchester

## Multi-Site deployment



Somehow those are not considered as *Data-Lakes* even that one of them has **Lake** in the name. 😊

# Distributed dCache

- Works for all protocols
- Supports HSM connectivity
  - Each pool/site may have it's own tape system
  - Multiple HSM copies are possible
- Pool's may run a different major versions
  - Sites have two years to upgrade
- Preferred write location depending on IP (location) or directory path (if requested)
- Preferred 'local' read access if data locally available
- Replication
  - On Demand, when requested from remote site
  - Permanent, data protection, location adjustment
  - Manual, for data location optimization, maintenance

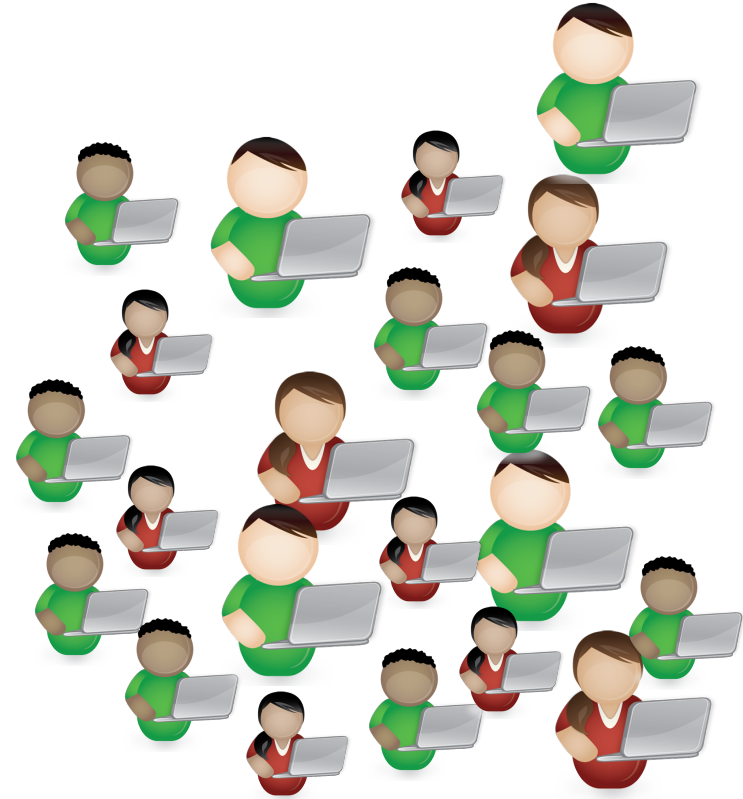
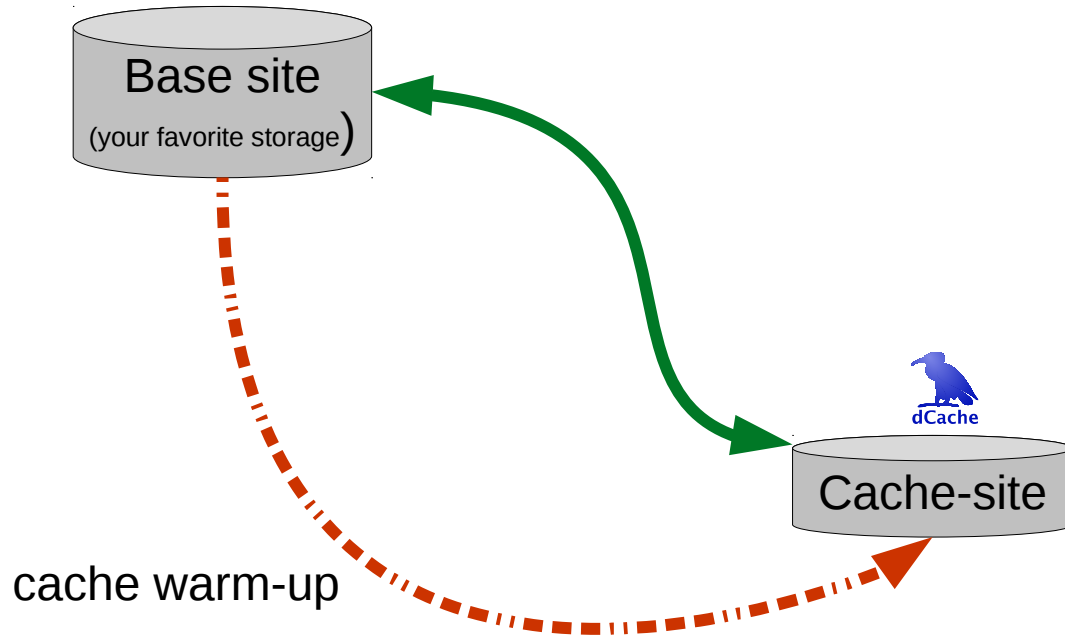
# What is missing

- All data servers must be dCache pools
- No data locality preference for internal communication
- No central/dCache way to update whole setup
- No operation, if consistency can't be guaranteed (See: CAP slides)

# The future

- Maturing NDGF and AGLT2 -like deployments
- Addressing some of missing functionality
  - use of non dCache components as data servers
- Locality aware internal communication
- Pure-caching deployments
- Non dCache components

# Pure-caching deployments

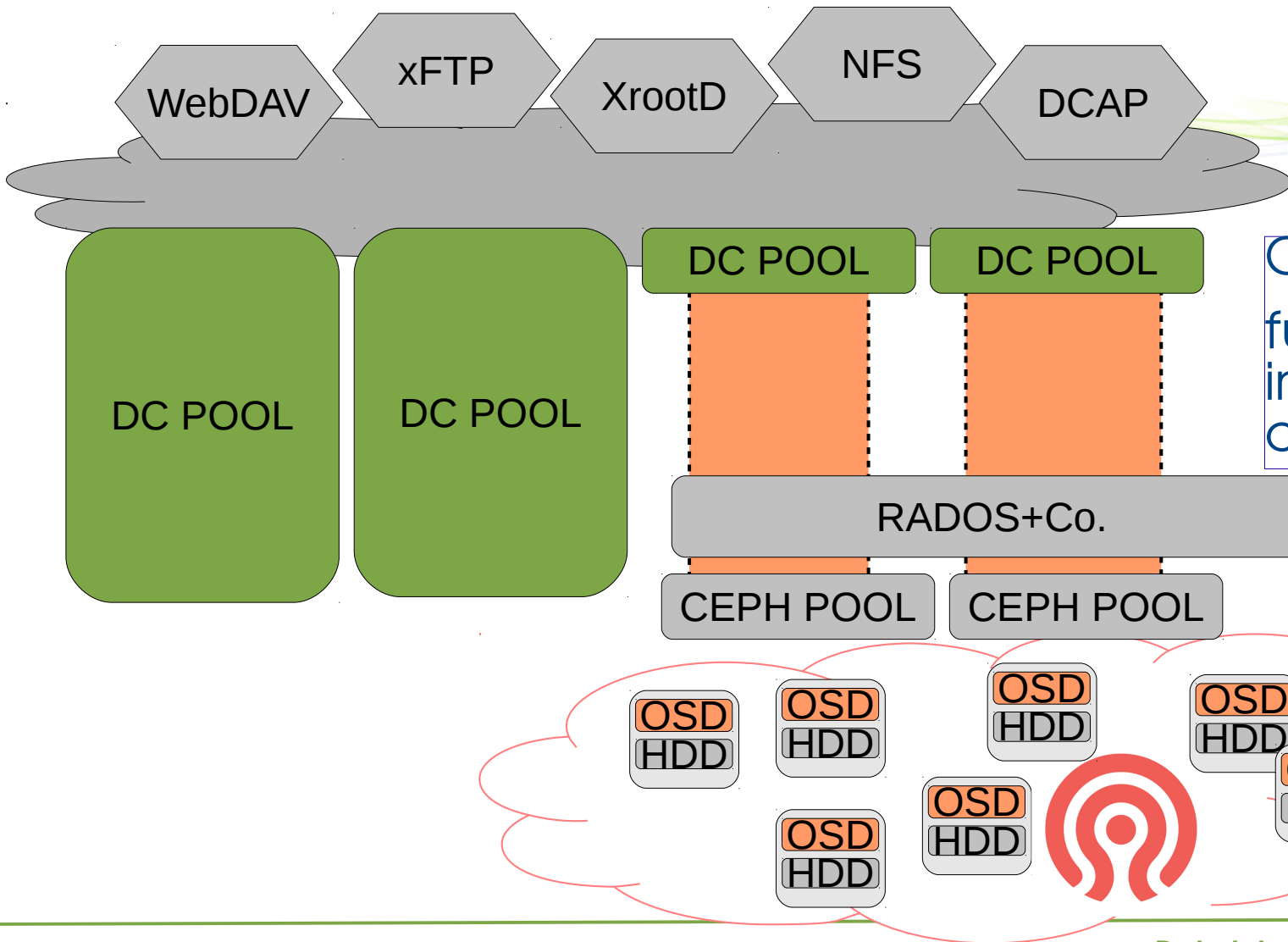


# Pure caching site

- Read-through/write-(back/through) semantic
- Local cache of data
- Local cache of metadata
- Cache warm-up
- Independent operation when base site unavailable
  - Breaks consistency
- Ability to turn any site into cache and back

# Non dCache storage blocks

- Use of standard data blocks
  - Object Stores (S3, SWIFT)
  - WebDAV
- Make use of locally installed storage
  - EOS
  - DPM
  - XrootD
  - Preserve namespace?



CEPH pool support full functionality, including HSM connectivity

**Whatever we do MUST not  
compromise stability of any site.**

**No discussions!**

# Summary and Conclusions

- dCache has a long tradition in providing distributed storage for WLCG (even if it's not called a *Data-Lake*)
- The configuration flexibility allows to control data placement and replication
- Distributed deployments add new challenges developers
- Fault-tolerant setup is recommended for a distributed deployment
- We have ongoing development to harden and mainstream distributed deployments
- We solve technical issues, sites have to coordinate distributed operation

# Thank You!

<https://www.dcache.org>



This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 777367



12-th international dCache  
workshop: May 28-29,  
Hamburg

# Backup...

# Project funding

- Sites, running and developing dCache
  - DESY, 5 FTE
  - FermiLab, 2 FTE
  - NDGF, 1 FTE
- Projects
  - XDC, 2 FTE

- **C**onsistency
  - Every read receives the most recent write or an error.
- **A**vailability
  - Every request receives a (non-error) response – without guarantee that it contains the most recent write.
- **P**artition tolerance
  - The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes.