The DataLake model for the scientific computing of the next decade

As discussed at the DOMA kickoff meeting at CERN with the XDC perspective

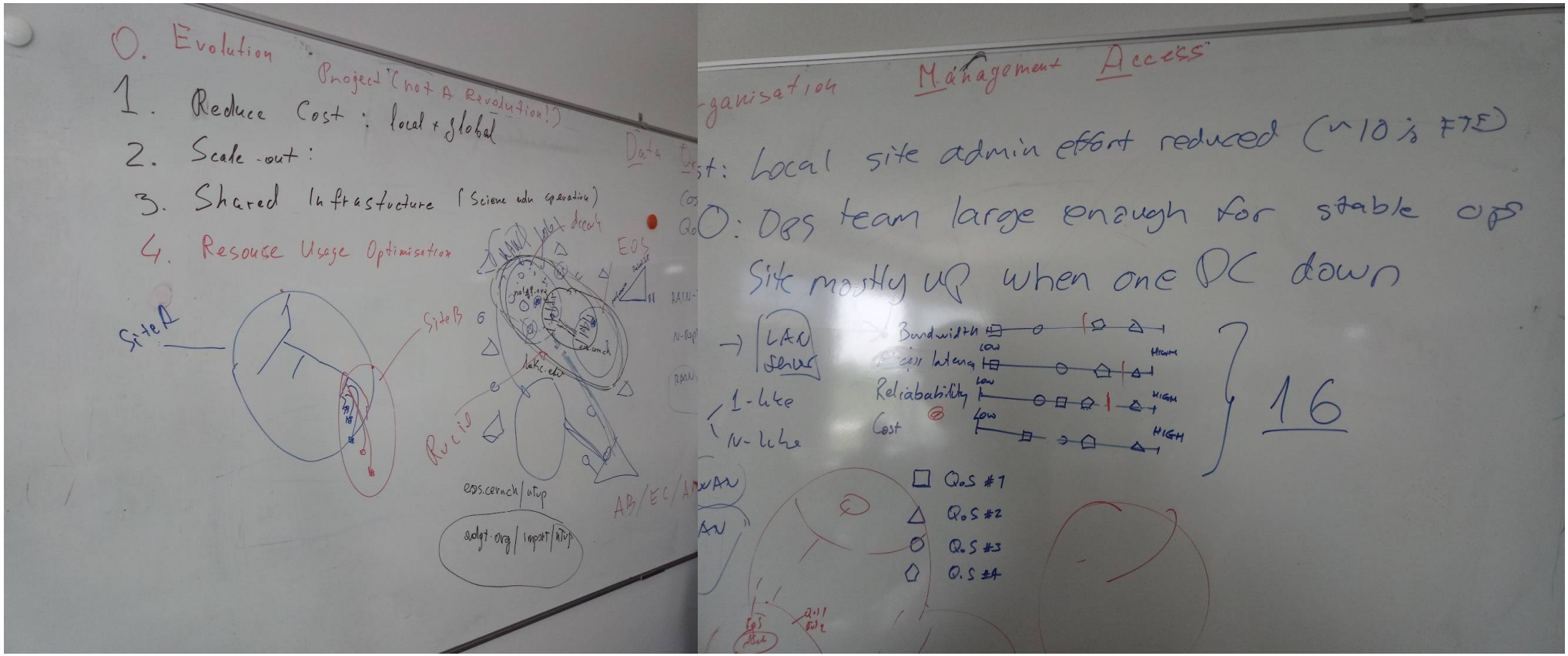
Daniele Cesini – INFN-CNAF
Patrick Fuhrmann - DESY

eXtreme DataCloud

Data Management for extreme scale computing

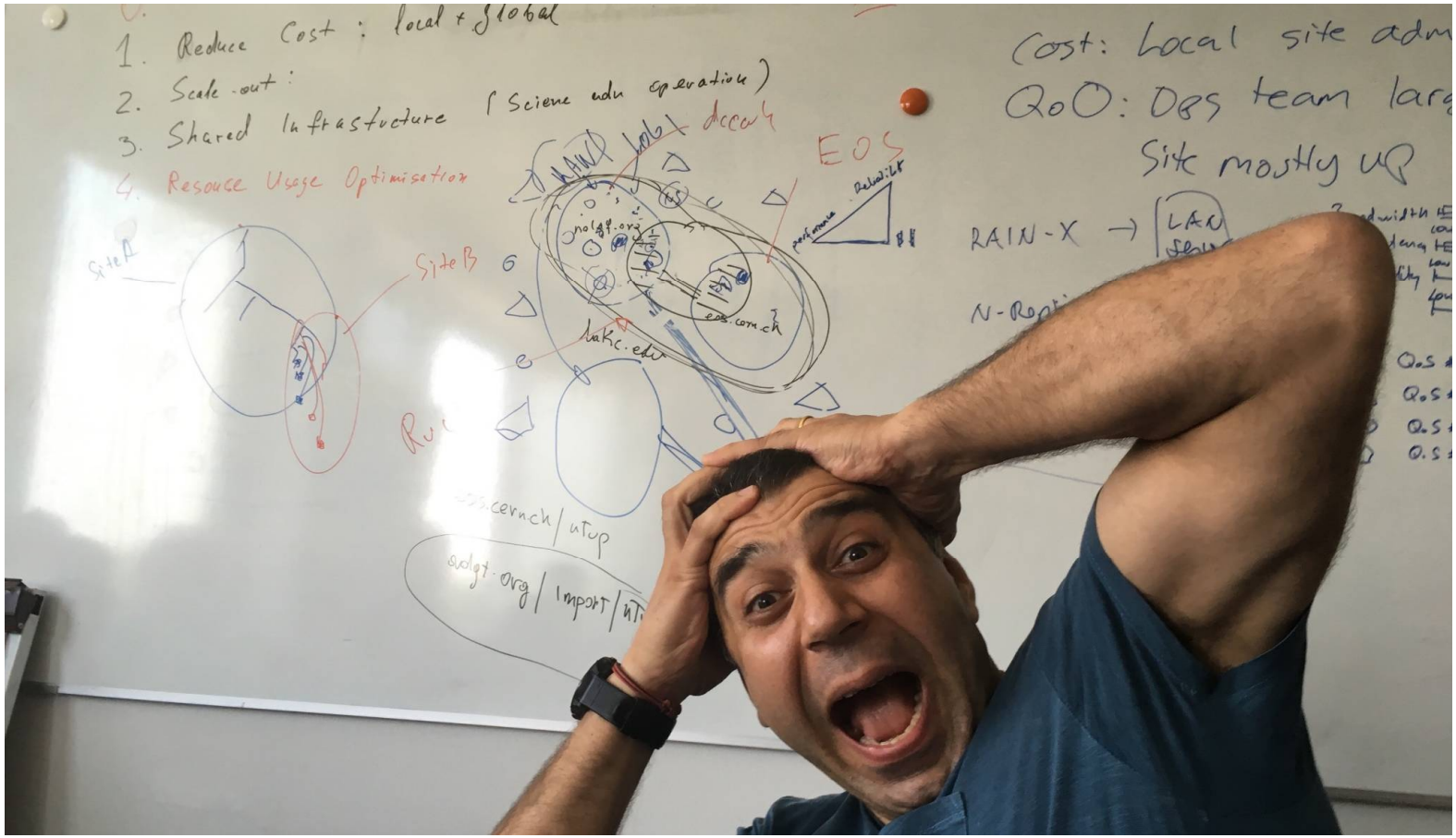
eXtreme DataCloud is co-funded by the Horizon2020 Framework Program – Grant Agreement 777367

Mainly a report form the Data Organization Management Access KickOff



WLCG DOMA kick-off: https://indico.cern.ch/event/729930/

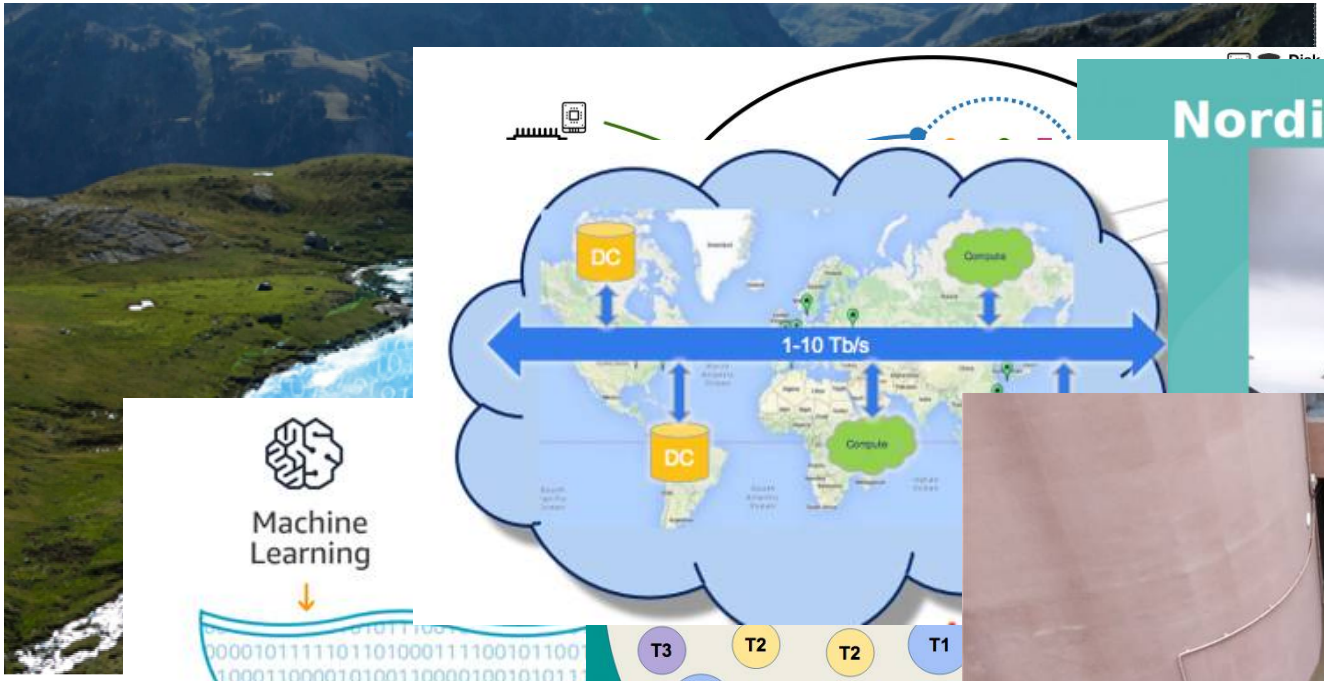
DOMA

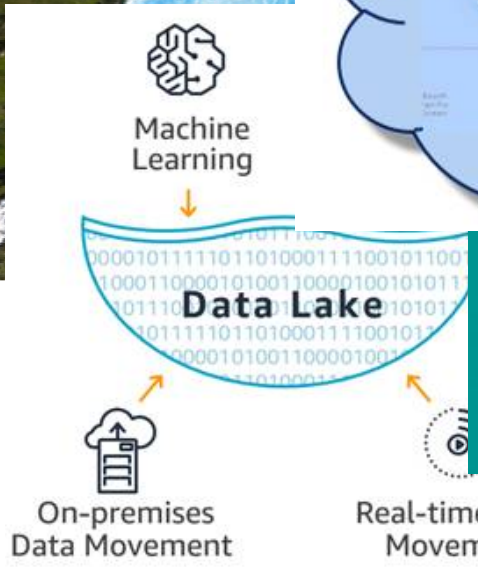


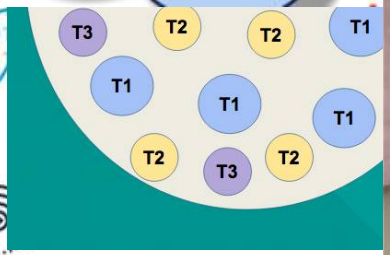
WLCG DOMA kick-off

Photo: Courtesy Xavier

The DataLake

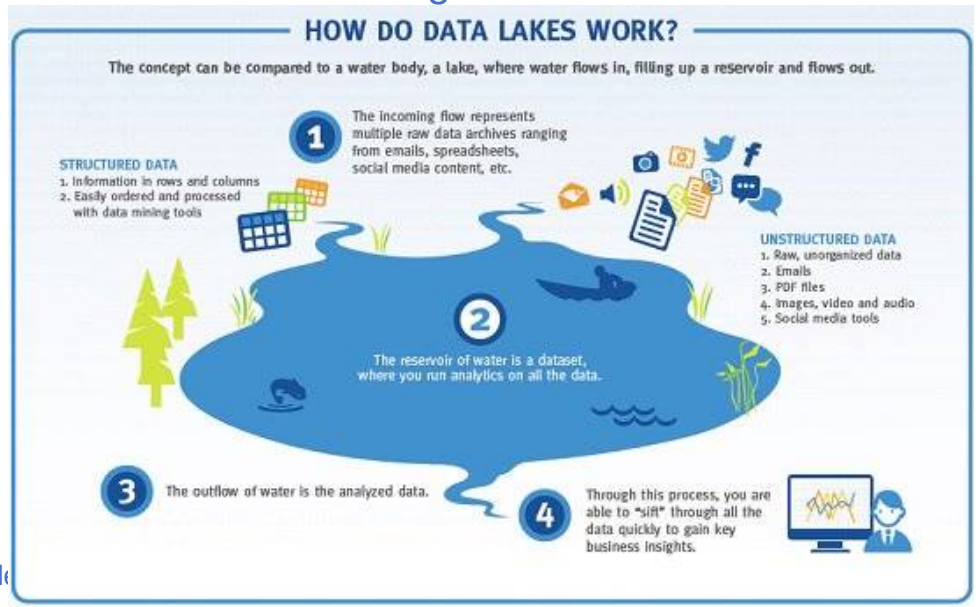






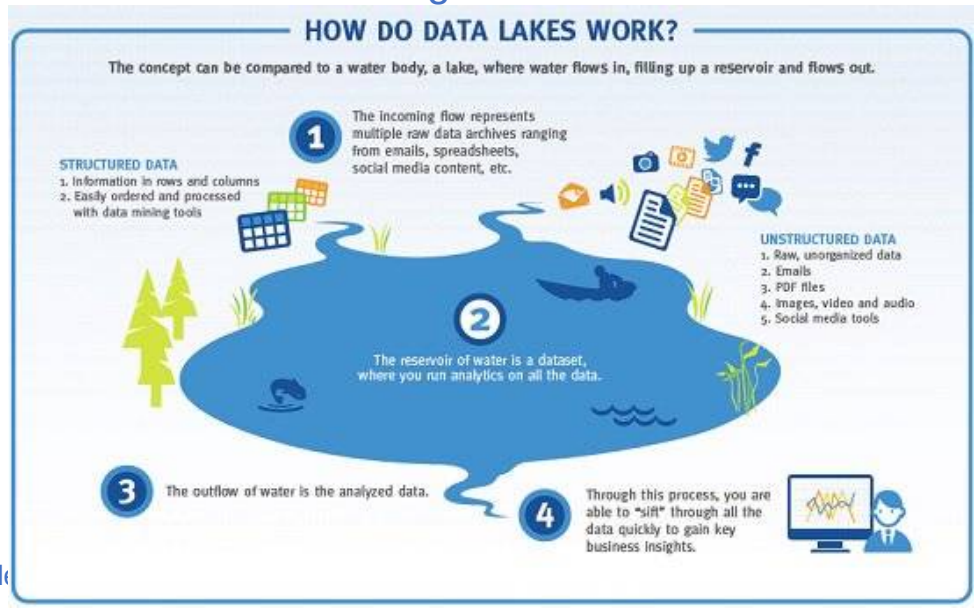
Data Lakes in Wikipedia

- A data lake is a method of storing data within a system or repository
  - in its natural format
  - facilitates the collocation of data in various structural forms
    - usually object blobs or files.
- The idea of data lake is to have a single store of all data
  - raw data
  - transformed data for Reporting, Visualization, Analytics and machine learning
- The data lake includes
  - structured data, i.e DB
  - semi-structured data, i.e. CSV, logs, XML, JSON)
  - unstructured data, i.e. emails, documents, PDFs
  - binary data, i.e. images, audio, video



Data Lakes in Wikipedia

- A data lake is a method of storing data within a system or repository
  - in its natural f
  - facilitates us structural forms
    - usua lobs or files.
- The idea ata lake is t ingle store of all data
  - raw d
  - trans data visuali nalytics and machine learning
- The data des
  - struct a, i.e DB
  - semi-st ogs, XML, JSO
  - unstructured d ails, documents, PDFs
  - binary data, i.e. images, audio, video



Data Lakes in WLCG

- Storage Consolidation
  - Storage services are the most complicated to operate in WLCG
    - Investigating the possibility to reducing the number of storage endpoints
  - Consolidate, especially at the national level, different endpoints into a single distributed instance

- Data lakes are an extension of storage consolidation

- Geographically distributed storage centers, potentially deploying different storage technologies, are operated and accessed as a single entity

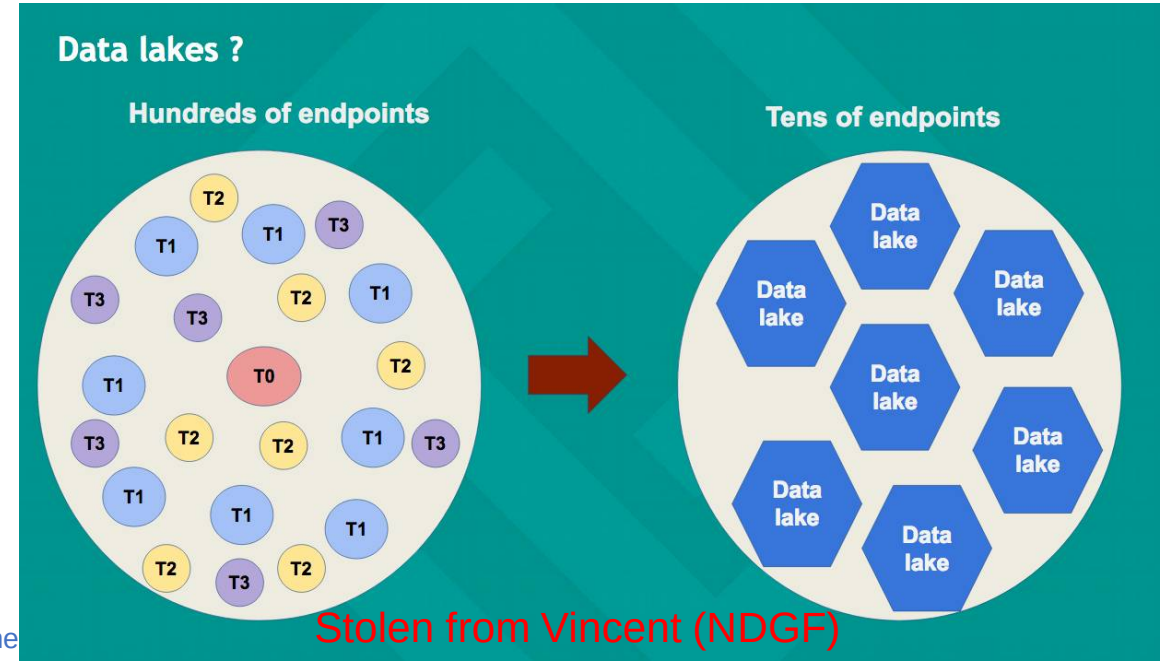
- Storage centers in a lake should be connected through high bandwidth links (> 10Gb/s in 2017) and relatively low latency (< 50 ms), consequently it is expected that a single data lake would not span more than one continent.

On the definition of the *Data Lake*

- Very diverse understanding of the expression *data lake*.
  - ⇢ Attempts to define it by ‘name space’ or ‘region’ or ’country’ all failed.

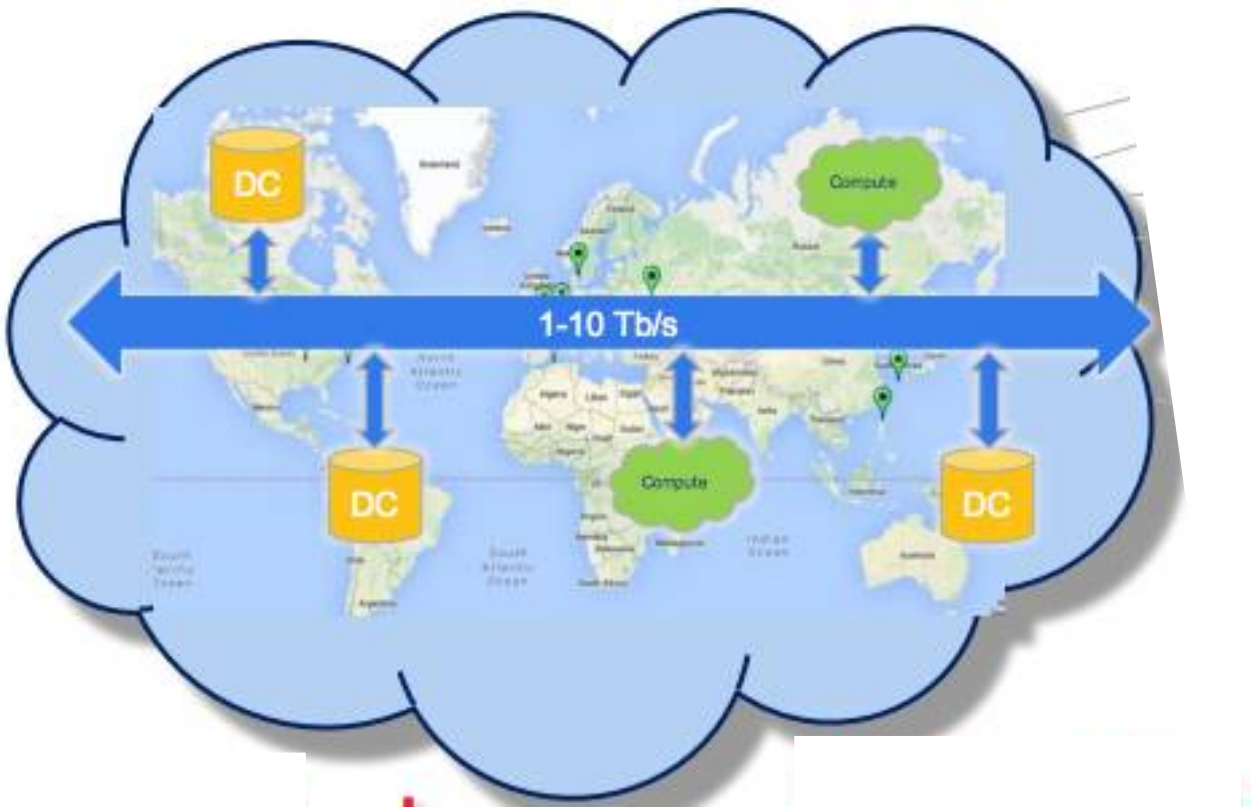
- Good suggestion by Maria Girone: We should avoid using the expression

- **Looks like one thing, but is composed of many**



Or…

- Looks like one thing and is composed by one thing…



- Caching
- Network Performance

Data Lakes in WLCG

- Experiments already handle very complex data distribution workflows and rely on a certain level of non co-location between data and processing units
- They all implemented techniques and workflows for remote reading of data
- The point here is to move this “intelligence” and “orchestration” of geographically distributed data **at the infrastructure level**

QoS: What is that ?

- Ongoing work started with INDIGO. Great progress based on work from KIT and CNAF.
- Instead of caring about disk and tape on the experiment framework level, it would be less complex and more future proof define the quality of local storage, concerning persistency, access latency and price
- These ‘storage classes’ need to be well and commonly defined
- Example:
  - Storage centers define two (or more) classes with names and properties:
    - ‘*archive*’: Probability of data loss < (1 in 1.000.000) and max access latency 2 hours
    - ‘*archive-online*’ same data loss probability but max access latency 1 ms.
- QoS classes need to be discoverable

QoS Advantages

- We don’t care how storage provides fulfill their QoS SLA: e.g. Tape, CEPH with 20 copies or engraved in stone.
- We are prepared for storage technology changes, w/o changing our data persistency model.
  - Eg. Tape could be replaced by disk arrays or cloud storage
  - HPC might need new storage qualities, like super low latency (BeeGFS)
- We could directly map to cloud providers
  - Amazon: Glacier (cheap, safe, high latency), S3 (low latency, expensive)
- QoS will become discoverable and orchestration middleware (Rucio) can use matching or AI algorithms to select the right combinations of ‘classes’ at the storage provider endpoints.
- This kind of model is attractive to new (younger) communities. They essentially don’t know anyway what a tape is.

INDIGO-DataCloud/XDC CDMI QoS

Quality of Service in storage – broker page

Logout

Available Qualities of Storage

	Name	Access Latency [ms]	Number of Copies	Storage Lifetime	Location	Storage type	Available Transitions
	disk	100	1		DE	Processing	tape, disk+tape
	disk+tape	100	2		DE	Processing	tape
KIT	DiskAndTape	50	3	20 years	DE	Processing	TapeOnly
	DiskAndTape	50	2		IT	Processing	
KIT	DiskOnly	50	3	20 years	DE	Processing	
	DiskOnly	50	1		IT	Processing	
	profile1	10	3	20 years	DE	Processing	profile2
	profile2	10000	2		DE	Archival	profile1
	SSDDisk	10	1		IT	Processing	StandardDisk, Tape
	StandardDisk	1000	3	20 years			

DESY
KIT
PSNC
INFN

- Access Latency [ms]
- Number of Copies
- Storage Lifetime
- Location
- Available Transitions

QoS in the Lake

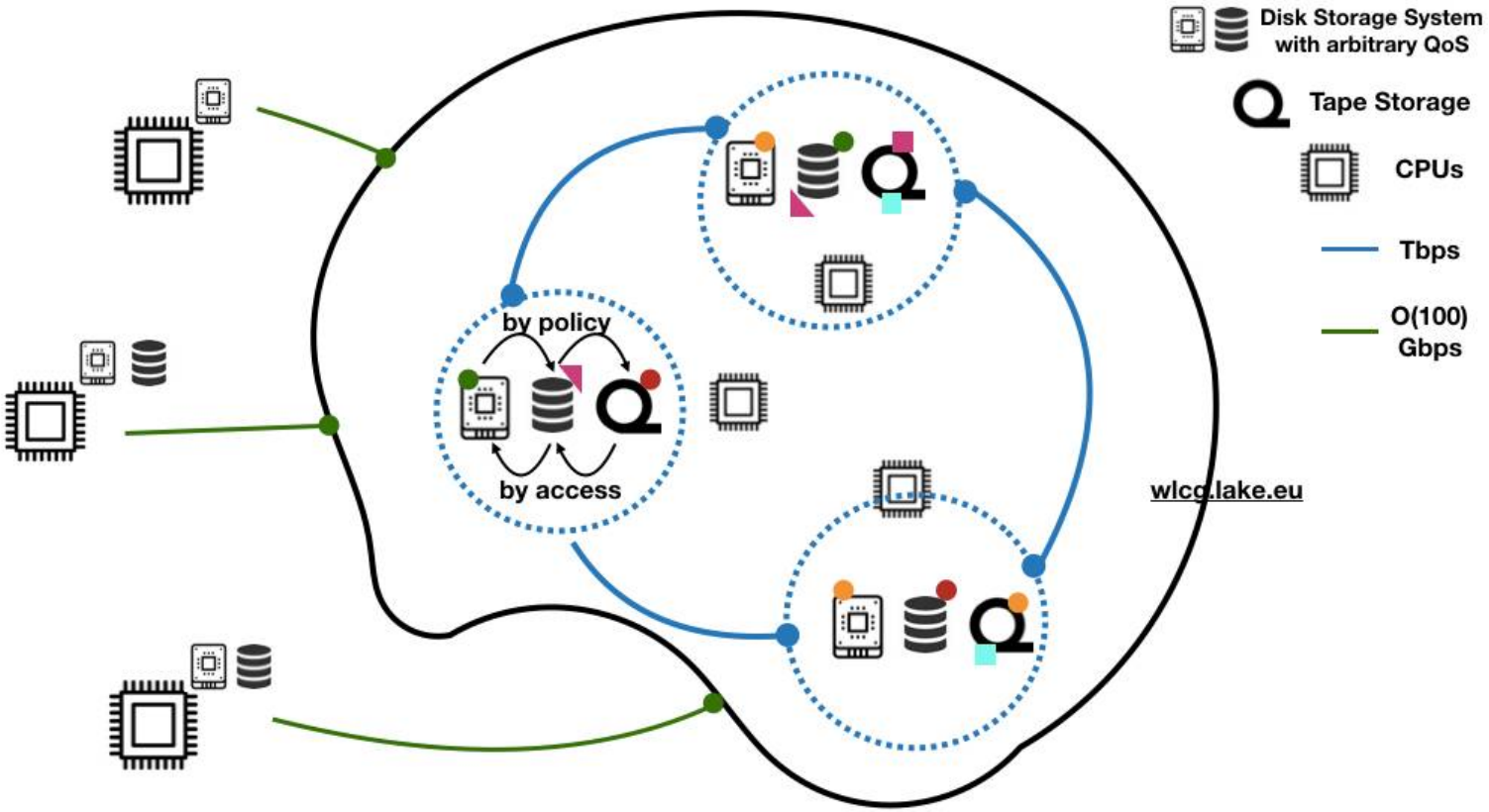
- We are looking for a solution in which the lake behaves as hierarchical system and optimizes the data organization based on policy (first) and usage (after) exploiting different QoS

- Moreover, the solution should foresee the possibility to attach a volatile storage to the lake
  - It should be used as tactical storage, hosting a redundant set of data, to optimize data access and the system should auto-recover in case the volatile storage disappears

A QoS–based workflow in the Lake

- One can imagine a data workflow where files that need several levels of processing start with a fast disk (hot) requirements and eventually transition to a nearline disk (warm) to finally end up archived on tape (cold) with a low likelihood they get recalled shortly after.
  - The three tiers mentioned before strongly relates to performance, reliability and resiliency which are metrics dominated by cost.
- The transition between the storage tiers (or *hierarchy* change) can be either:
  - *passive* (ie. estimations on processing/access/replication time for a particular data/set)
  - *active* by implementing triggers at metadata level once the needs for the data/set change.
    - This approach opens the possibility to steer data workflows in real-time through extended attributes “flags” on the namespace.

The Data Lake



What are we told would justify changes ?

- Reduce cost for storage
  - Global (WLCG level) and local (Site level)
  - Hardware
  - Operations
  - Number of replicas is already below 2 on avg (Atlas reported 1.3)
- Scale out
  - Does the current model really have a scale-out problem ?
  - Which architecture would solve that ?
- Or simply : Evolution forced by ’external’ technologies or methodologies. (Best example is the ‘cloud’)

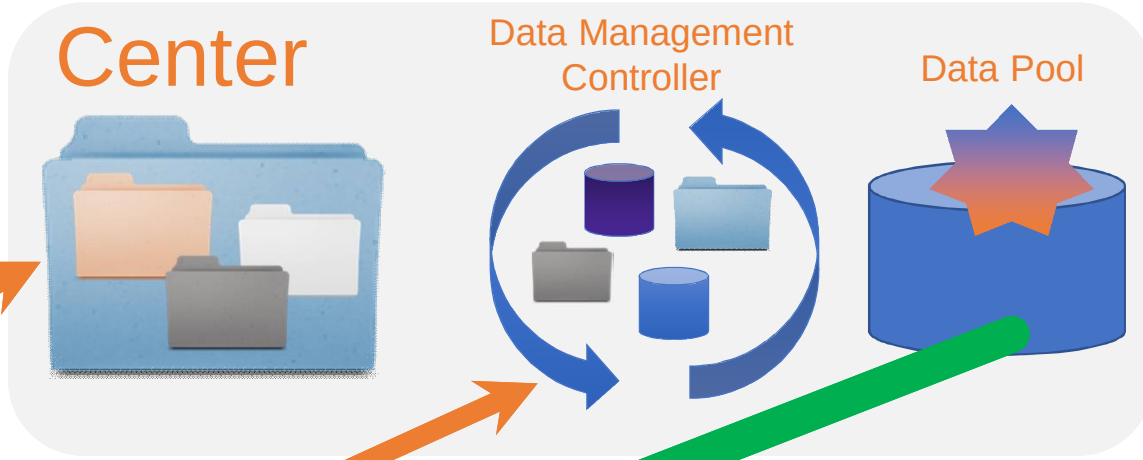
On Cost

- Saving costs by reducing operational complexity. Or better: getting more storage for the same money
  - Focus on larger sites, assuming small sites are ineffective
    - There are no numbers available, how much that would save us
    - There are ideas to run smaller sites ‘operator-less’ regarding storage
    - Probably it’s better to avoid drying-out universities
- Operational cost savings are not automatically transferred to more storage space. That funding could simply disappear
- Saving cost by providing fewer copies or smarter caching of data
  - Improved high level orchestration. (data placement)
  - Well defined storage retention (QoS, see later)
  - Balancing network against storage (depends on network costs)
  - With or w/o smarter caching.

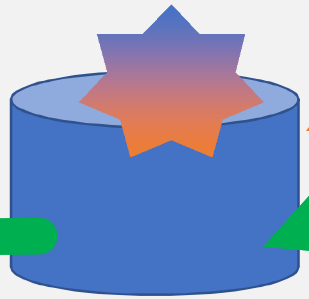
Some scenarios (Taken from XDC)

Data Lake, simple cache site

In production for years
e.g. NDGF, Michigan

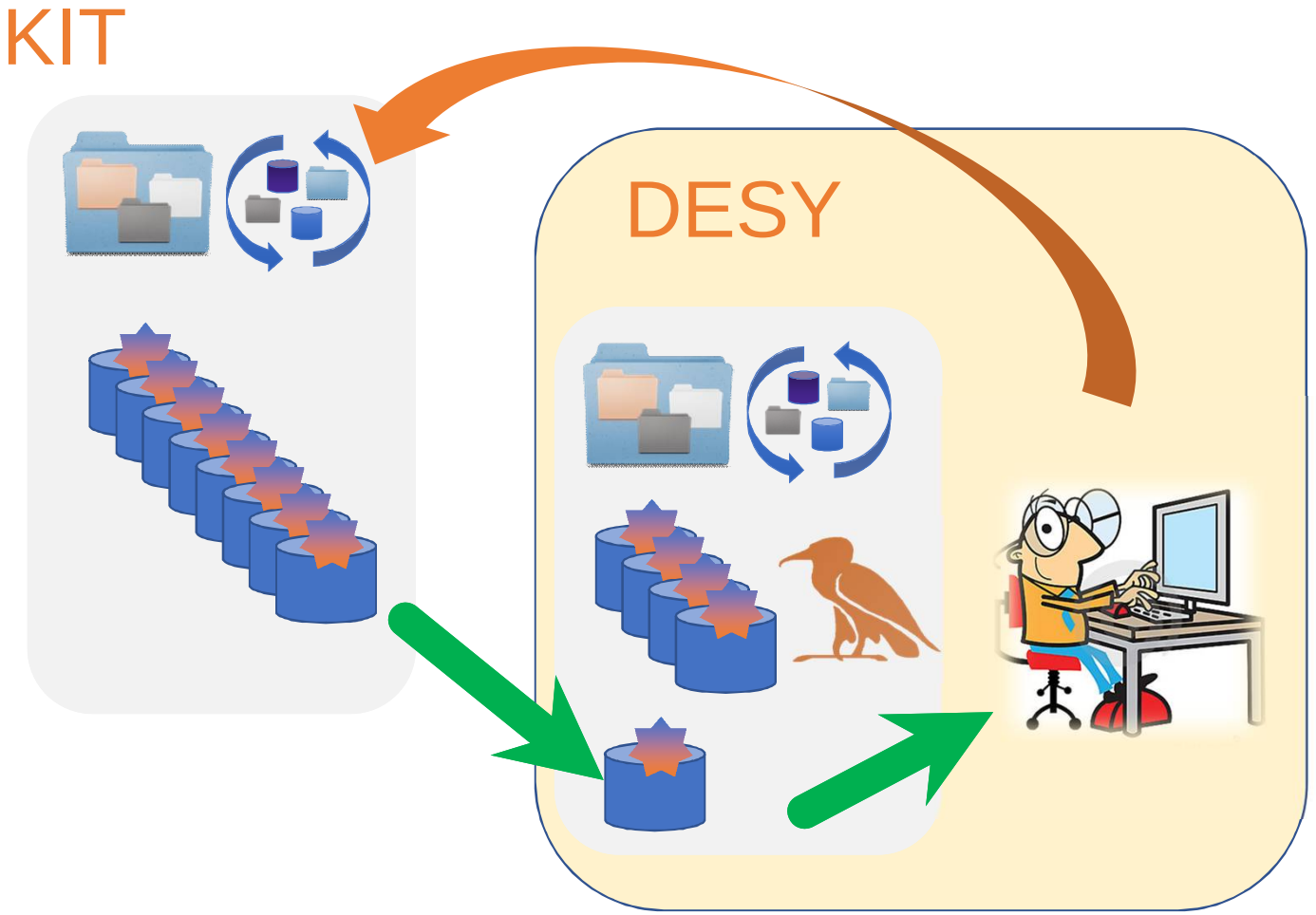


Satellite Site



Remote Data Pool

Complex multi provider scenario 1



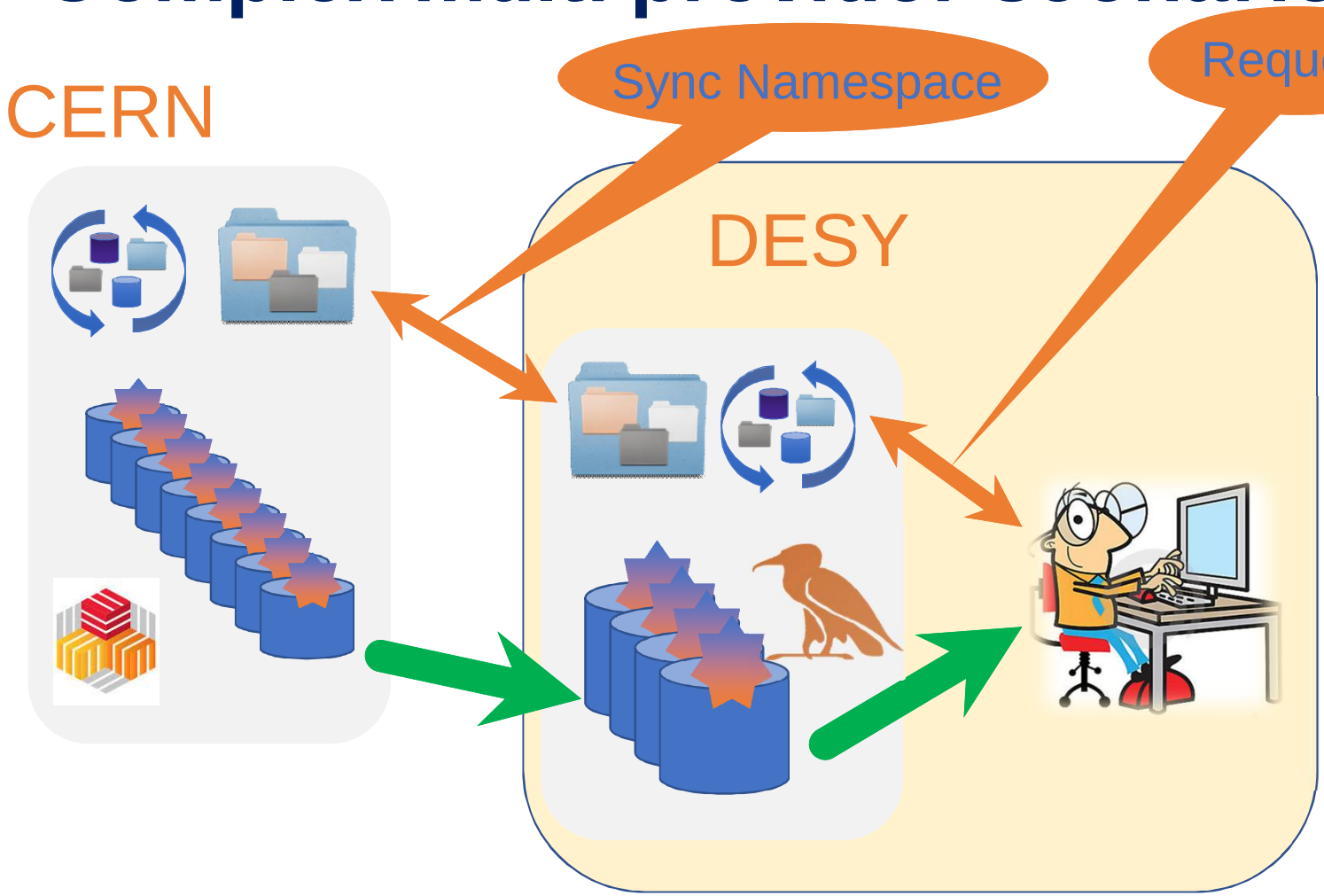
Advantages

- no additional software stack needed at sites.

Still disadvantage

- Local data not accessible in case data link is down or central service not available

Complex multi provider scenario 2(a)



Advantages

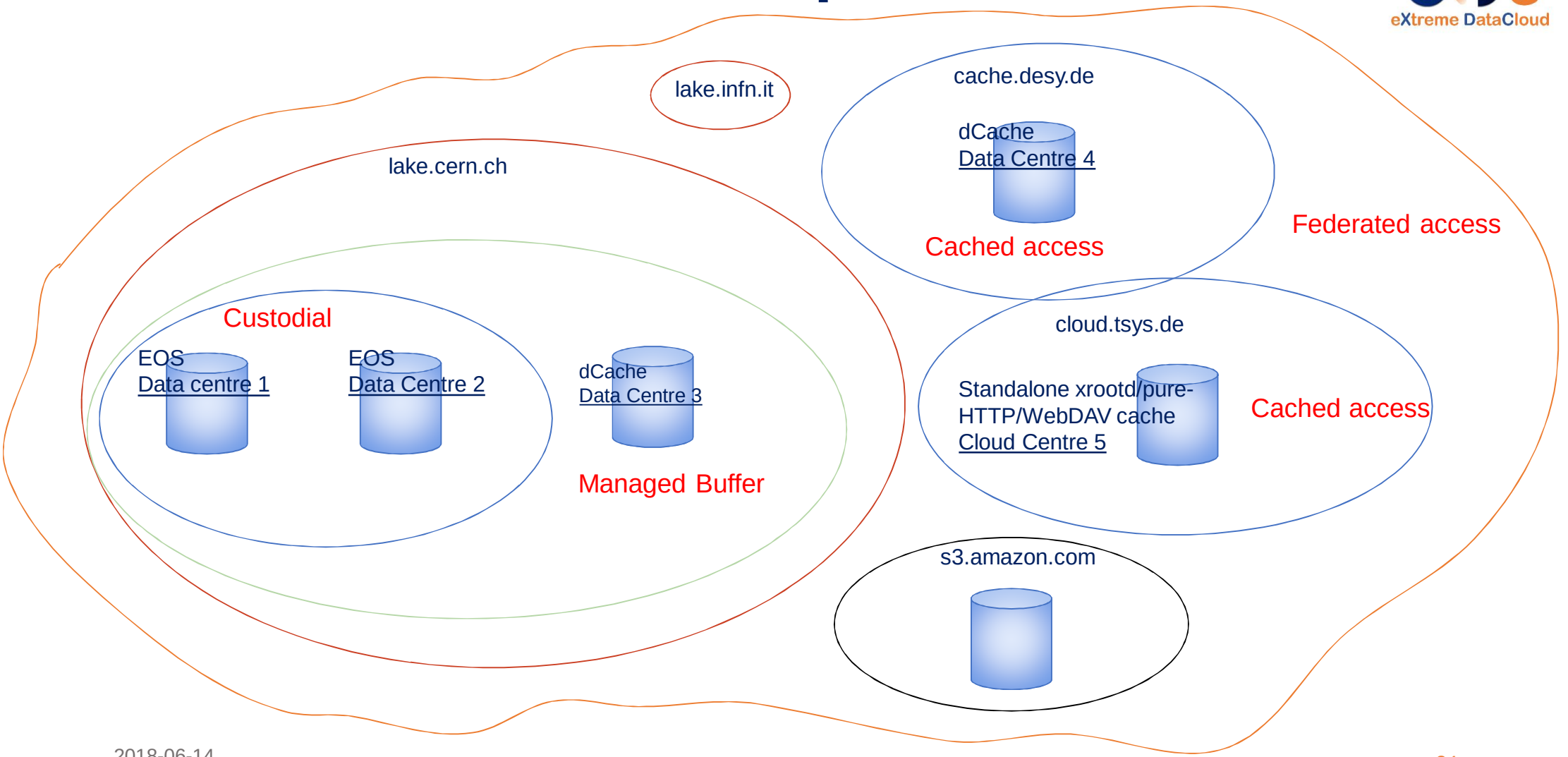
- Same software stack as we currently have at the sites.
- After the data has been transferred to the local storage system, a name space entry has been created locally and the data is available at the local site independently of the remote network link and the availability of the central service.

Requirement for the complex cases

- We would need to see a benefit
- We would need to agree on
  - A namespace synchronization protocol
    - On client request
    - On client request plus subsequent lazy background fill (like Dynafed)
    - Message queues for selected namespace events (producer – consumer)
  - ACL synchronization
  - Identity Mapping

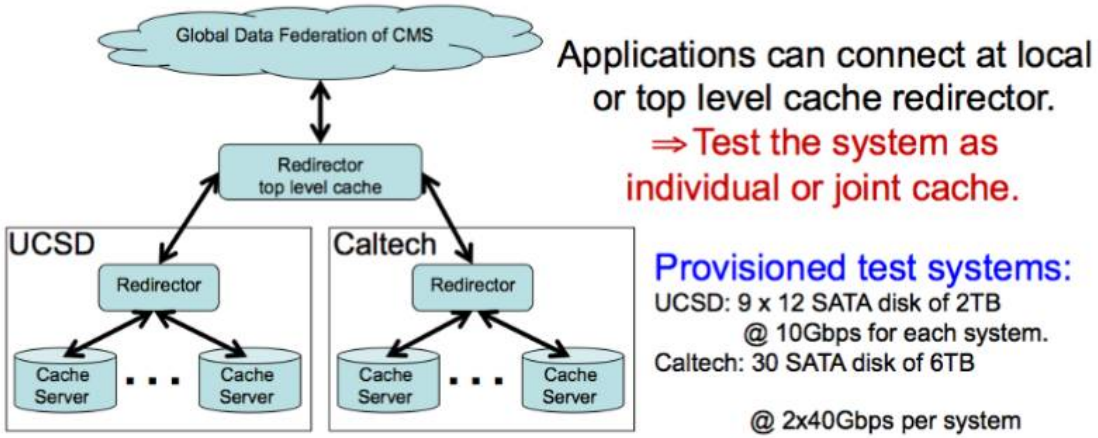
XDC minimum viable product":
- read-only operation
- fetch data on miss with service credentials
  -- data can be chunked or full-file
- manage cache residency, evicting data when necessary
- HTTP frontend with group-level authorization

XDC “Data Lake” concepts



Caching infrastructure with XROOTD

A Distributed XRootd Cache



Applications can connect at local or top level cache redirector.
⇒ Test the system as individual or joint cache.

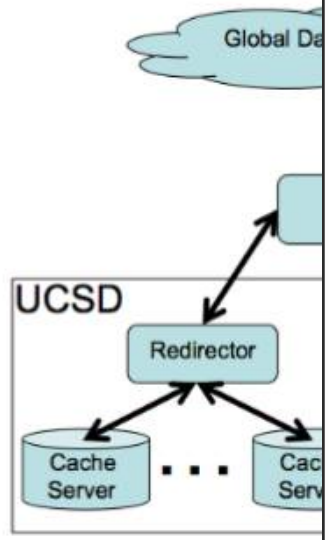
Provisioned test systems:
UCSD: 9 x 12 SATA disk of 2TB
@ 10Gbps for each system.
Caltech: 30 SATA disk of 6TB
@ 2x40Gbps per system

Production Goal:
Distributed cache that sustains 10k clients reading simultaneously from cache at up to 1MB/s/client without loss of ops robustness.

https://indico.cern.ch/event/729930/contributions/3021531/attachments/1661988/2663057/WLCG_DOMA.pdf

Caching infrastructure with XROOTD

A Distributed XRootd Cache



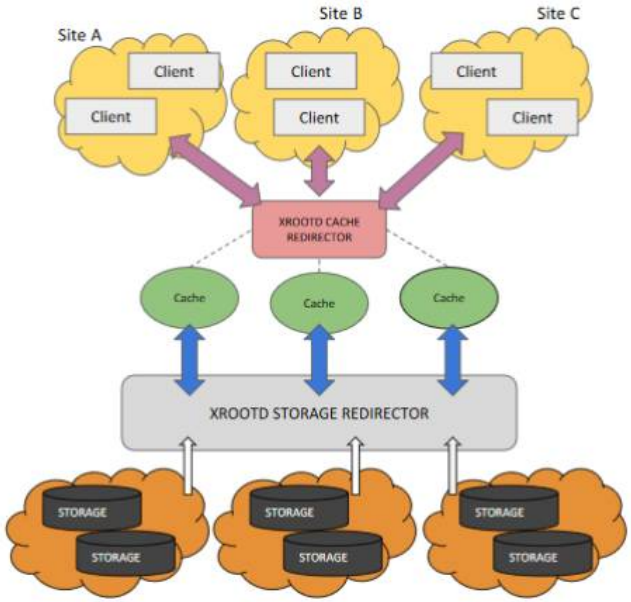
Distributed ca
from cache a

Distributed scenario with XROOTD CACHE redirector

Credits: D. Ciangottini, D.Spiga, T.Boccali – CMS and XDC

Geographically distributed cache

- The very same technology used on local scenario can be **geo-distributed**
- Use **ephemeral storages to enhance jobs efficiency**
- Leverage high speed links to **reduce the total amount of allocated space**



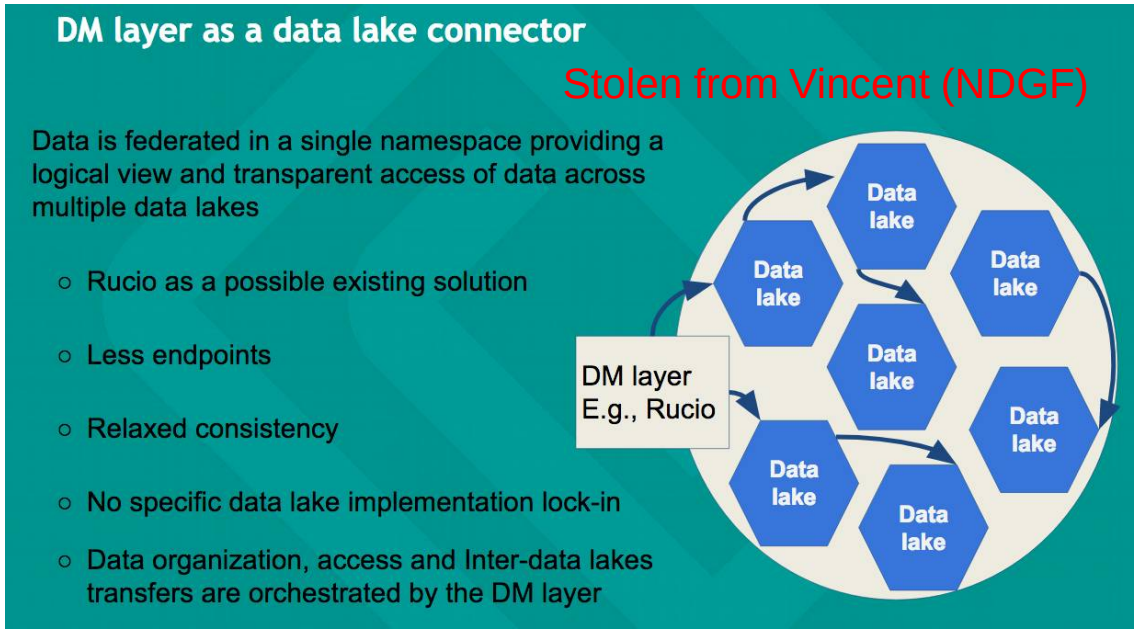
18

Discussion on caching @DOMA

- Caching will cost hardware
  - caching will have to be used wisely
  - making sure that one caches most of the reuse data.
- We do not have today clear numbers that caching is really effective
  - facilities w/o storage (HPCs, CPU only sites)
  - facilities with storage, but dealing with non local data
  - We even do not have a forum to discuss this
- Caches need to offer the capability to be pre-filled (pre-placement) or be passive caches (cache based on access)

Where is the data lake in those scenarios ?

- The complex scenarios above could either
  - be inside a data lake
  - connecting data lakes.
- Optionally data lakes are connected by DM solutions (Rucio). See Vincent's presentations for the NorduGrid Meeting.



- Two "RUCIOs"
  - Internal to the DataLake
  - Externel, orchestrating among DataLakes

Summary

- Data Lake
  - Expose unique storage entry point – looks like one but it's (or can be) made of many
  - Support different **QoS** for the internal components
    - Hierarchical storage
  - Caching at (every?) storage and CPU location
    - Pre-placement vs access-based
  - Intelligent management of data movement
    - RUCIO?
  - Unique namespace? (maybe)
  - To be deployed in parallel to the current infrastructure
  - Cost improvement to be understood