Storage Interoperability for DOMA (Actually a report on the DESY meeting)

Presenter: Patrick Fuhrmann
for Maria, Simone, Xavier, Andrea, Tigran, Paul, Al, Dmitry, Vincent, Mattias

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

Contributors

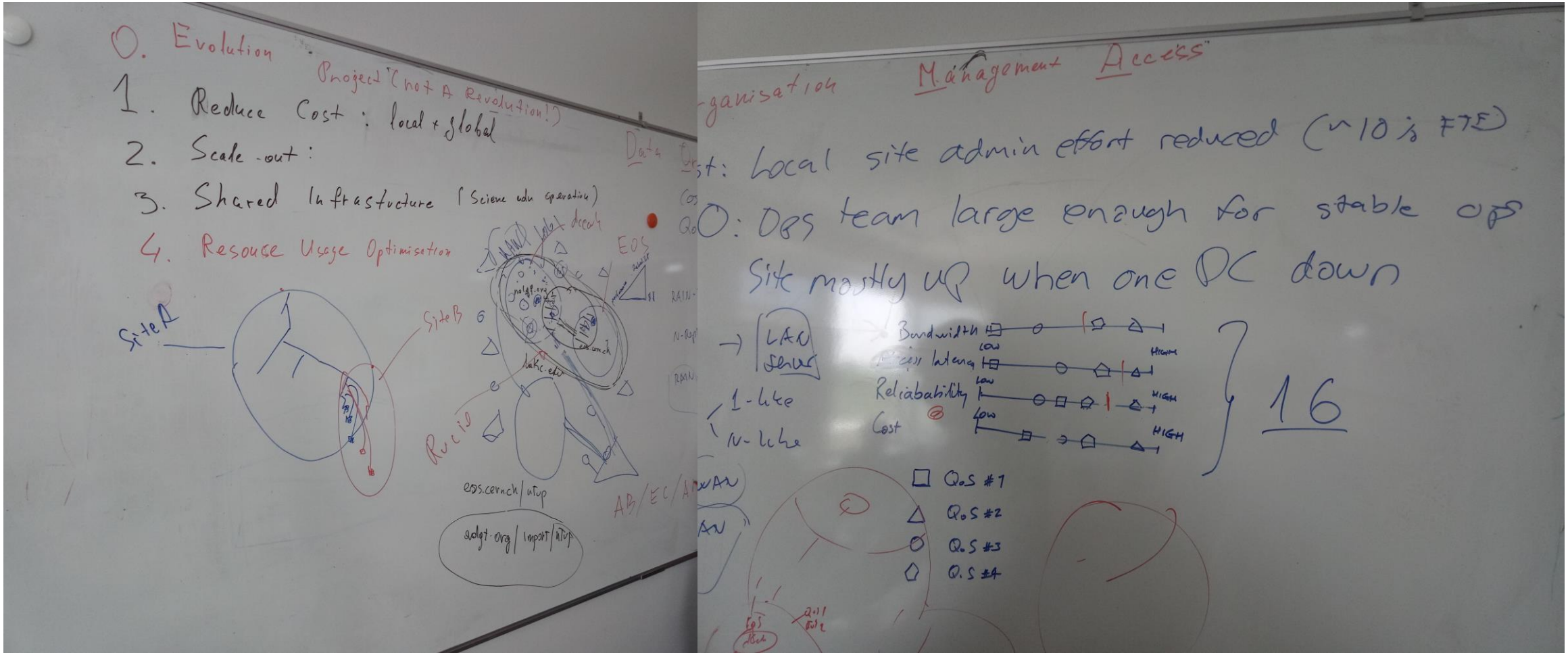
- Essentially the summary of the DOMA dCache meeting at DESY
  - dCache
    - Paul
    - Tigran
    - Patrick
    - Al, Fermi
    - Dmitry, Fermi
    - Vincent, NDGF
  - EOS
    - Andreas
    - Xavier
  - WLCG/DOMA
    - Maria
    - Simone
  - NDGF
    - Mattias Wadenstein

Topics at the DESY mini Workshop

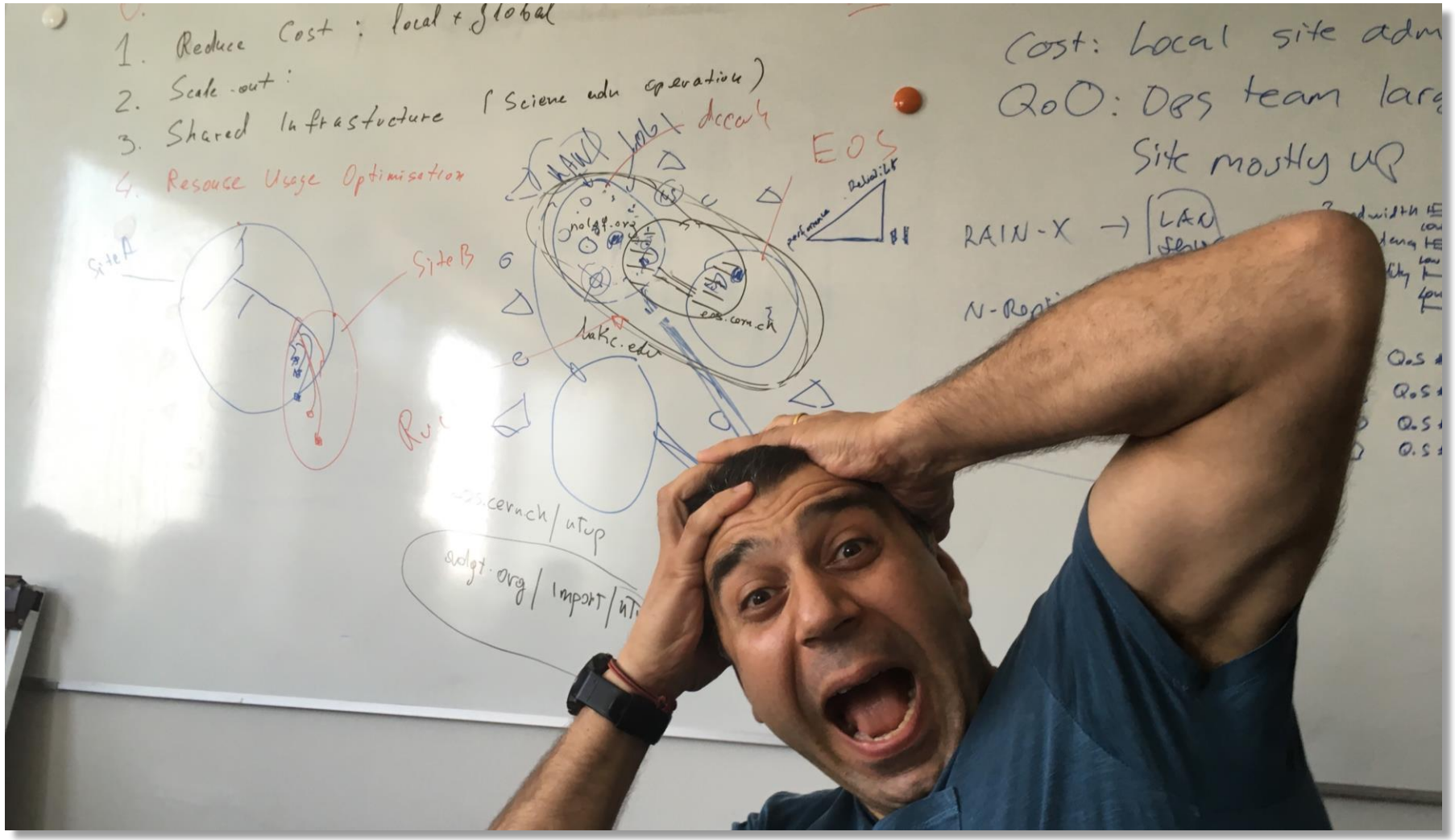
- Reasons why changes might be needed.
- Storage Interoperability in terms of storage quality definitions and API or protocol specification
- Which kind of caching could be helpful. (Only touched on)
- Not enough time to discuss more.

In a nutshell

Data Organization Management Access



DOMA Workshop (DESY)



Courtesy Xavier

Now, in more detail

What are we told would justify changes ?

- Reduce cost for storage
  - Global (WLCG level) and local (Site level)
  - Hardware
  - Operations
- Scale out
  - Does the current model really have a scale-out problem ?
  - Which architecture would solve that ?
- Shared Infrastructure
  - E-Infrastructure
  - Research Infrastructure
- Resource Usage Optimization (Summary of above)
- 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.
    - My opinion: We need 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.

Storage Interoperability (see Oliver)

- **Who** is working on Storage Interoperability
  - EOSCPilot
  - **eXtreme DataCloud (see next slide)**
  - Research Data Alliance
  - ESCAPE proposal
- **Storage Access** seems to be well covered
  - GridFTP, http(s)/WebDAV and xrootd
  - Possibly moving away from GridFTP?
  - FTS will hide those details.
  - Storage access is not a big issue anymore
- **AAI** is on a good way, possibly moving to OIDC (See Oliver's talk)
- How is the quality/preciousness of the data resp. storage communicate between customers and provider: **QoS**

Side remark: What is this XDC WP 4 about ?

- Implementing a configurable data workflow orchestration, in terms of data location and storage quality (QoS).
- Providing managed and unmanaged data caching services at all levels.
- Providing event based interfaces to external systems
  - Generating events to the XDC orchestration services when data is entering the XDC system.
  - Generating events to external compute clusters when data is ready to be processed.
- Federating heterogeneous data sources, building a virtual horizontal infrastructure-specific data space.

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
  - To let the storage infrastructure provider simply define the quality of local storage, concerning persistency, access latency and price.
  - To let he research infrastructure combine those classes to their needs.
- On both levels those ‘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.
  - RUCIO defines two or more classes visible to the experiment framework:
    - Class ‘*RAW-COLD*’ which is a combination of two ‘archive’ classes in different countries or geographical areas. (> 100 KM apart)
    - Class ‘*RAW-HOT*’ which is one site with ‘*archive*’ plus one site with ‘archive-online’
- QoS classes on both levels 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)
  - HNSciCloud
- 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.

QoS Steps needed

- We need to agree on a expandable set of well defined storage classes and their capabilities.
  - Ongoing ‘working group’ in RDA
- We need to agree on a protocol to discover and communicate the requested QoS for incoming data or on how to change the QoS for existing data.
  - Transition from ‘fast’ to ‘archive-online’ when data is declared ‘precious’
  - Transition form ‘archive-online’ to ‘online’ to save money.
- Reminder: How the data is finally stored or how the transition is done is completely up the storage providers.

Status of QoS

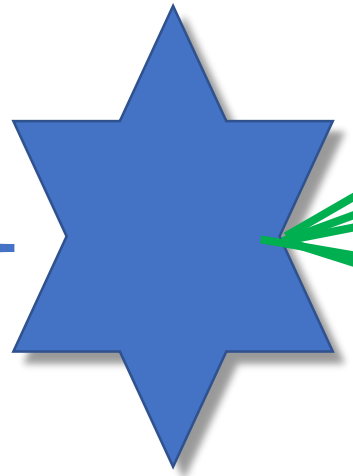
- dCache already offers QoS, combining one or multiple tape copies and one or multiple disk copies. So there is experience.
- Result of INDIGO DataCloud and XDC:
  - dCache provides a proprietary REST based protocol to communicate QoS plus transitions plus a SNIA acknowledged CDMI extension of a QoS protocol.
  - There is a modern ‘reference implementation’ in the official SNIA github repository, developed by INDIGO, available for the CDMI protocol extension, describing QoS in storage.
  - There are already plug-ins for dCache, GPFS, CEPH and StoRM.
- Within XDC dCache and EOS are in the process of agreeing on a common REST full protocol which can be converted to CDMI by the INIDGO/XDC CDMI reference implementation.
- Paul is leading a working group within RDA on the definition of QoS storage classes.

Example of an evaluation testbed for Qos

INDIGO-DataCloud/XDC CDMI QoS

Storage Broker Webservice

curl://



As example for a PaaS service

CDMI

IBM GPFS / HPFS

ceph

StoRM

dCache

INDIGO-DataCloud/XDC CDMI QoS

Quality of Service in storage (Broker Page)

Logout

Available Qualities of Storage

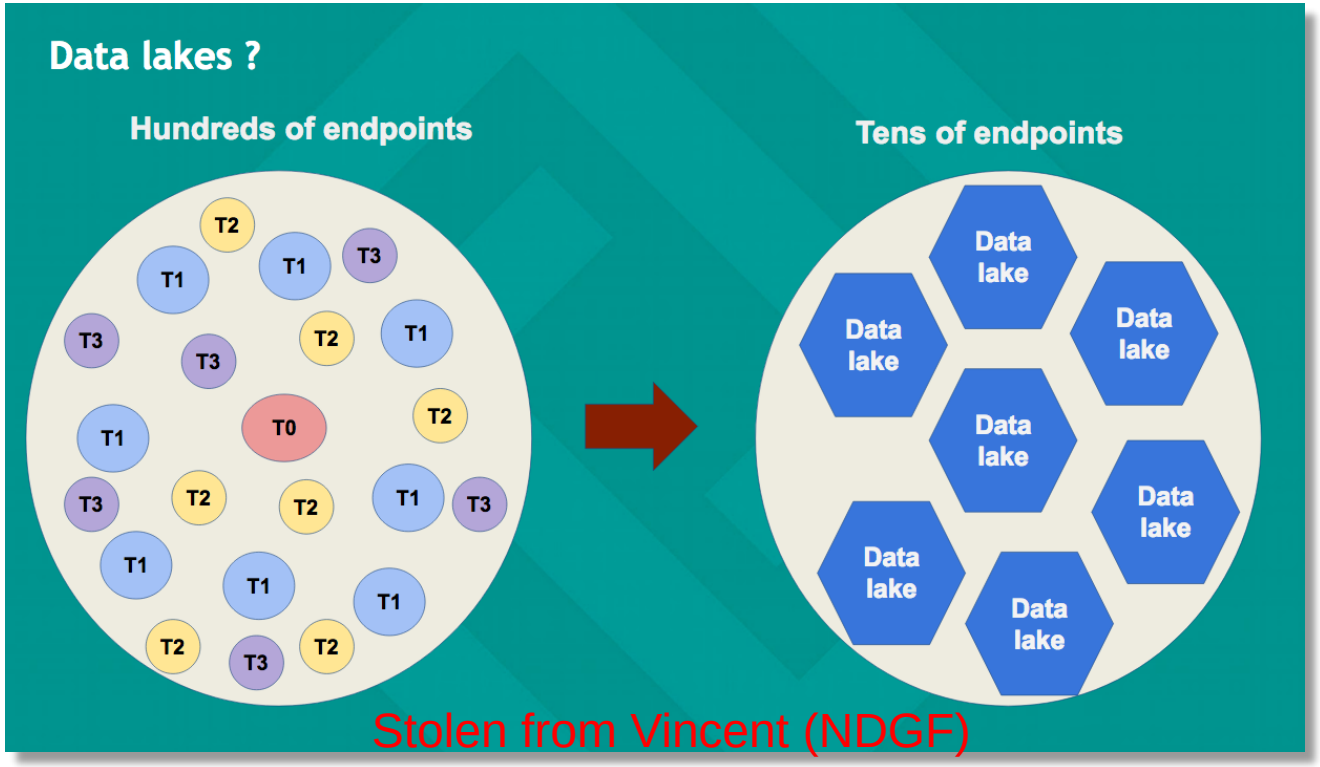
- Access Latency [ms]
- Number of Copies
- Storage Lifetime
- Location
- Available Transitions

DESY
KIT
PSNC
INFN

Name	Access Latency [ms]	N		tion	Storage type	Available Transitions
disk	100	1			Processing	tape, disk+tape
disk+tape	100	2			Processing	tape
DiskAndTape	50	3			Processing	TapeOnly
DiskAndTape	50	2		IT	Processing	
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	IT	Archival	SSDDisk, Tape

On the definition of the *data lake*

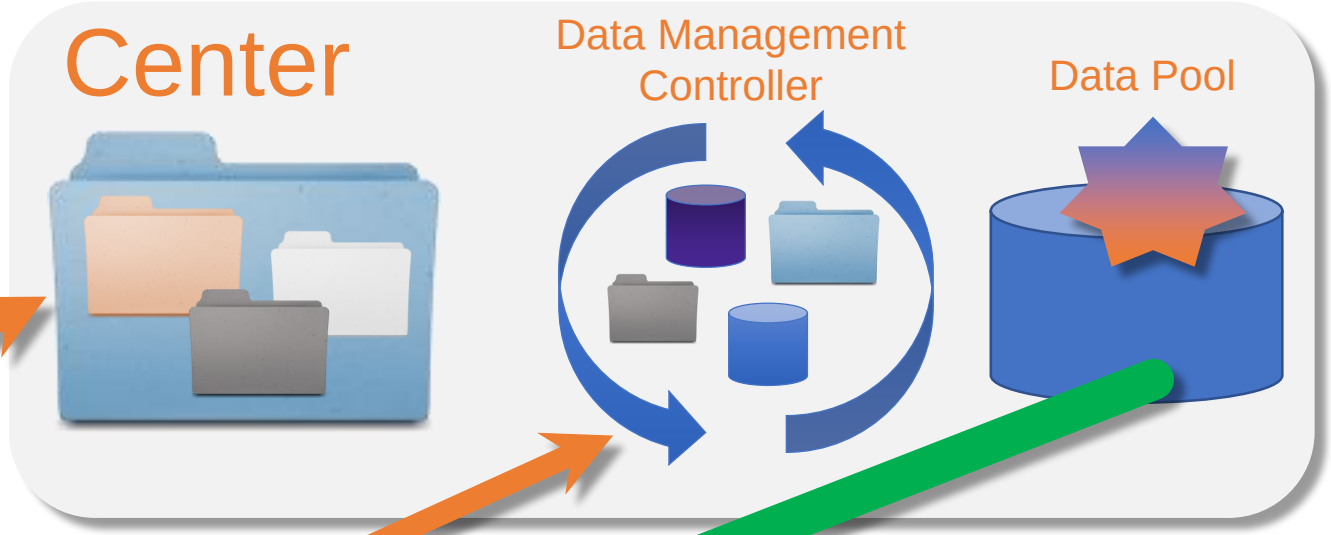
- Very diverse understanding of the expression *data lake*.
  - Attempts to define it by ‘name space’ or ‘region’ or ’country’ all failed.
- Good suggestion by Maria : We should avoid using the expression.
- My understanding is close to the one of NDGF (see Vincent's presentation)
- **NDGF is definitely a ‘data lake’**
- **Looks like one thing, but is composed of many.**



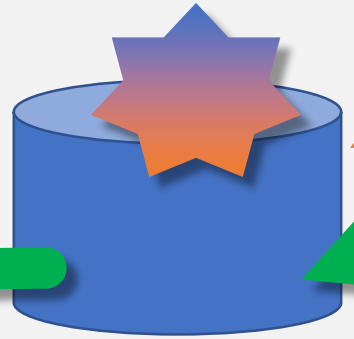
Some scenarios (Taken from XDC)

Data Lake, simple cache site

In production for years
e.g. NDGF, Michigan

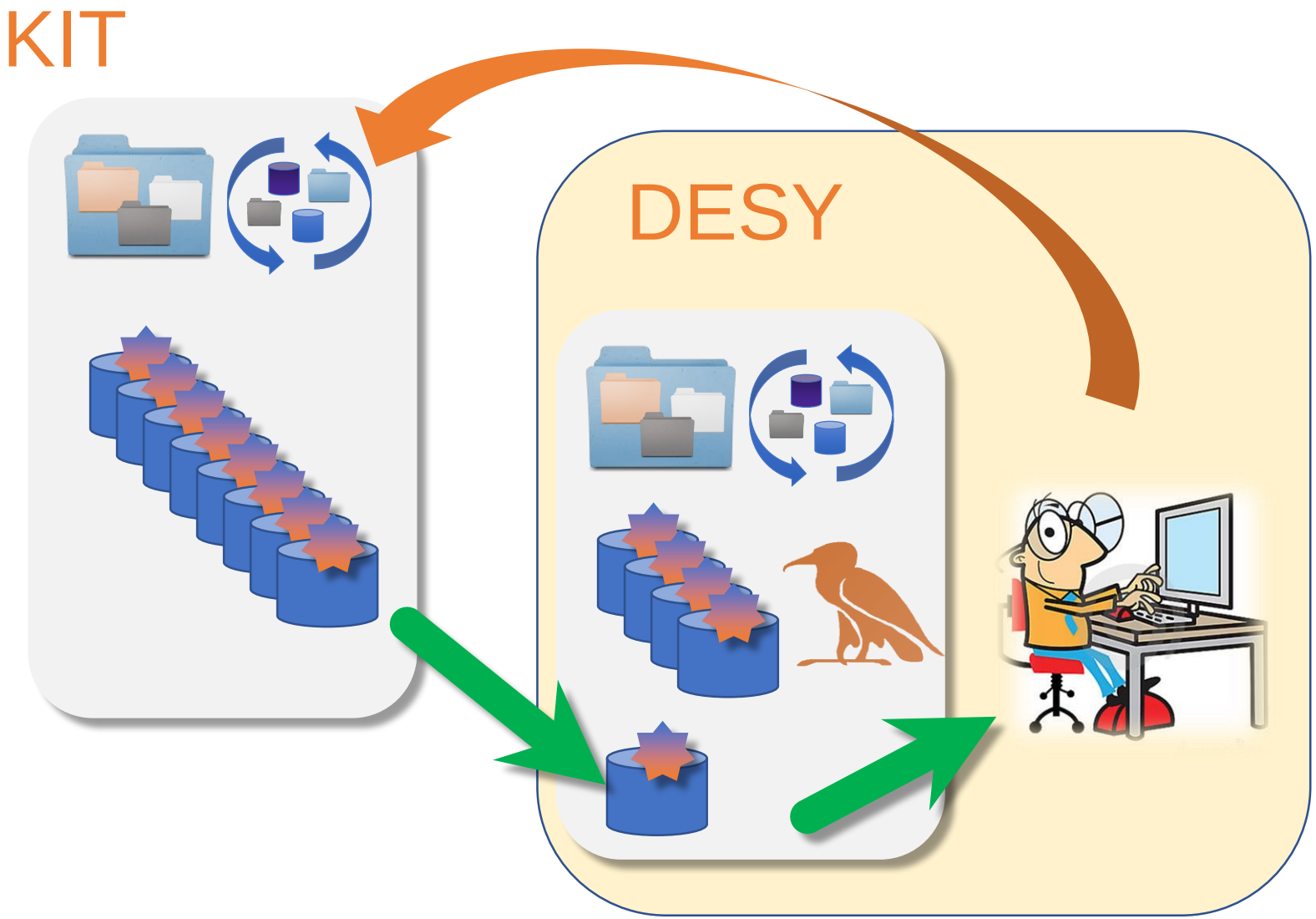


Satellite Site



Remote Data Pool

Complex multi provider scenario 1



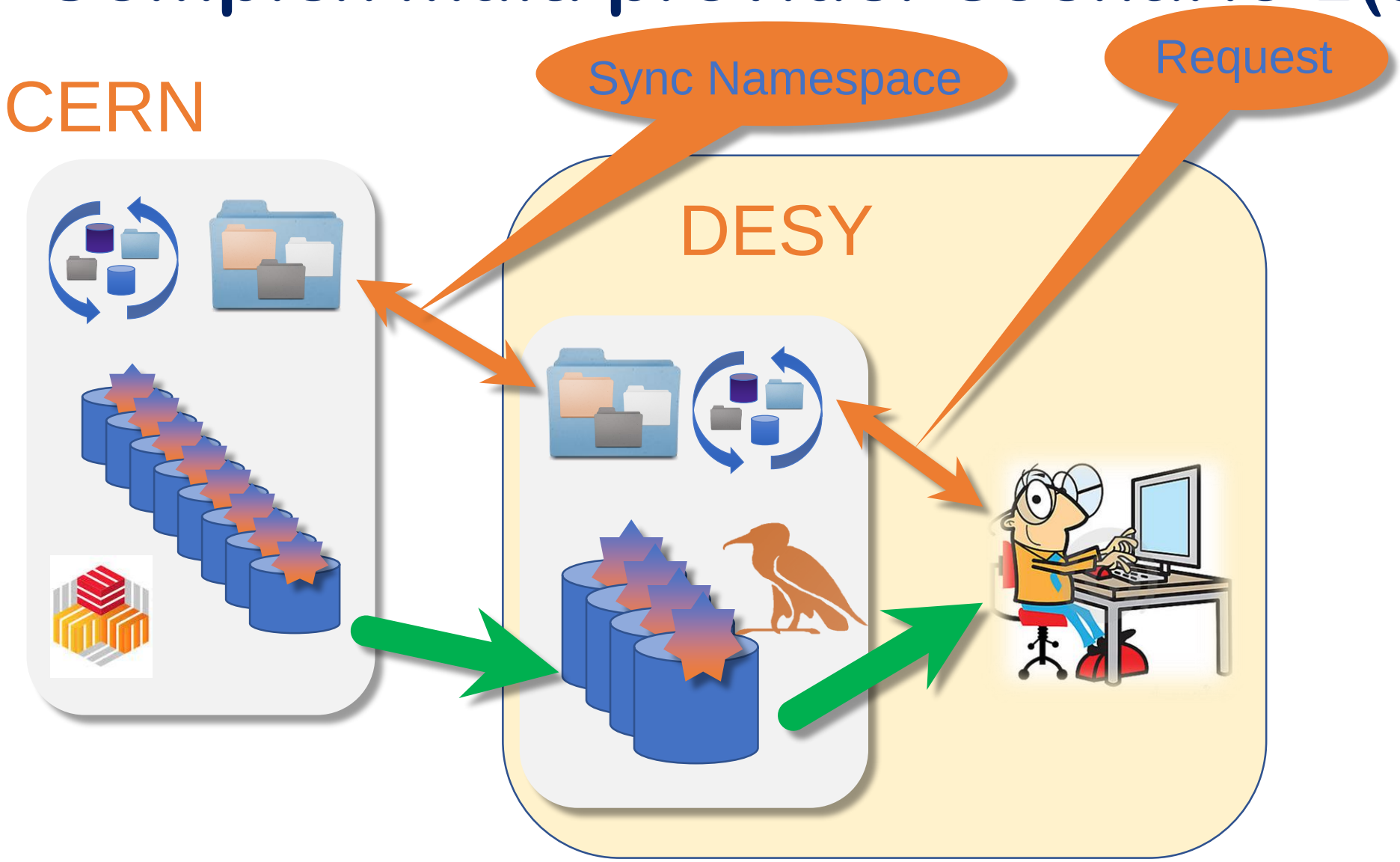
Advantages

- no additional software stack needed at sites.

Still disadvantage

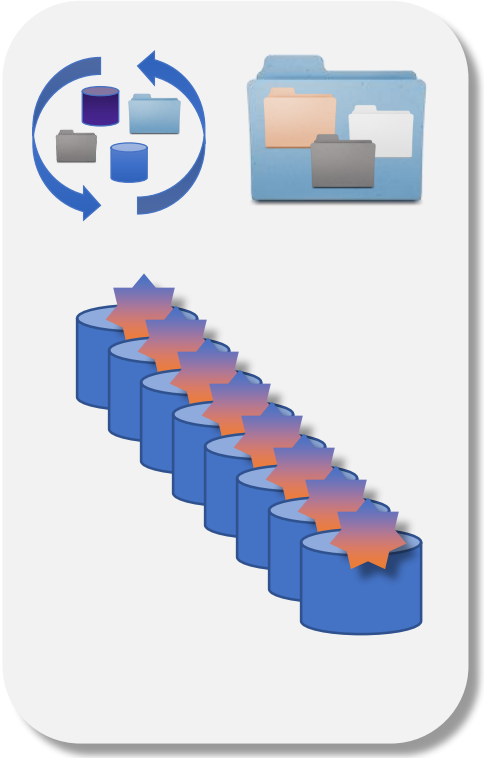
- Local data not accessible in case data link is down or central service not available

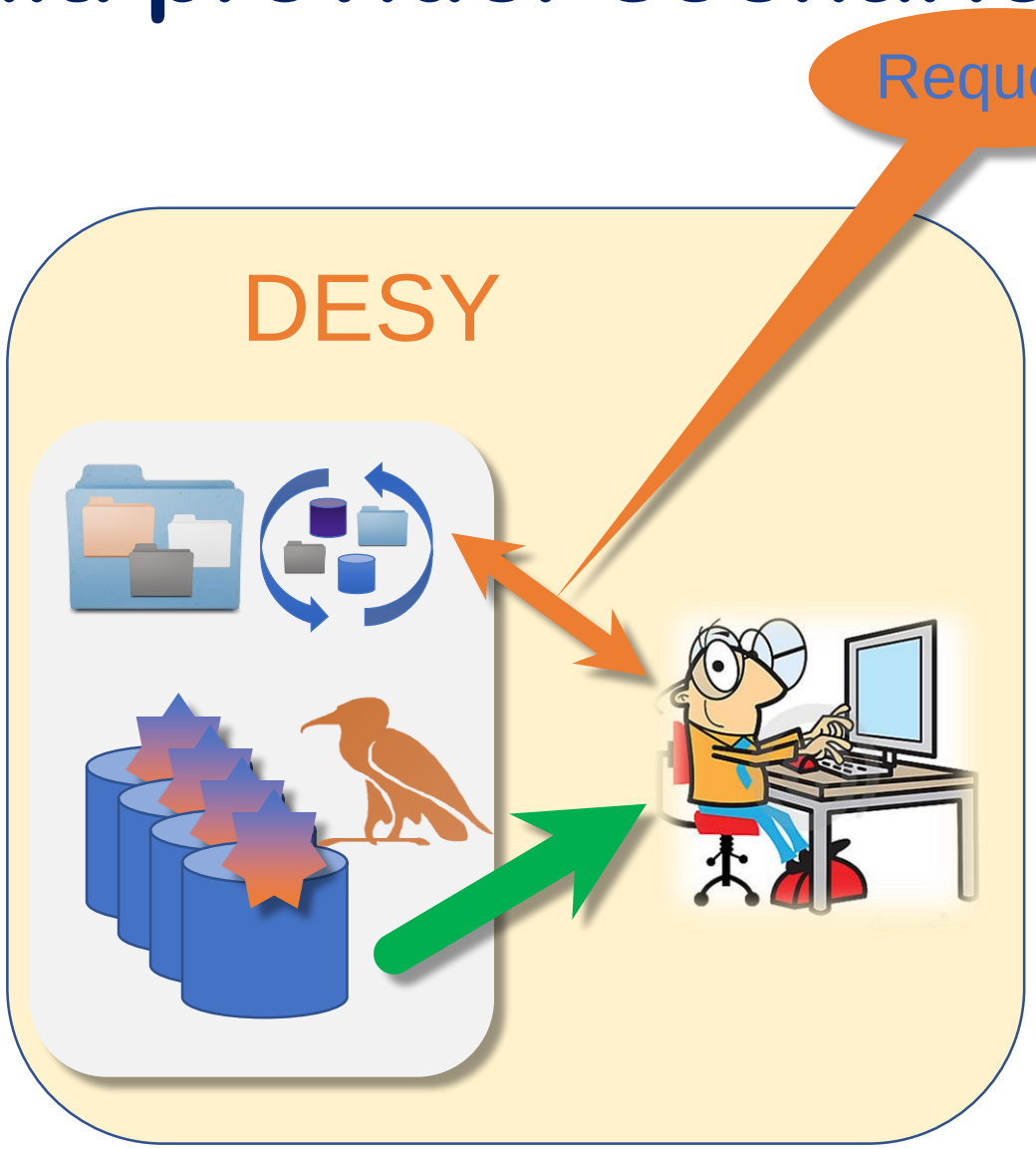
Complex multi provider scenario 2(a)



Complex multi provider scenario 2(b)

KIT



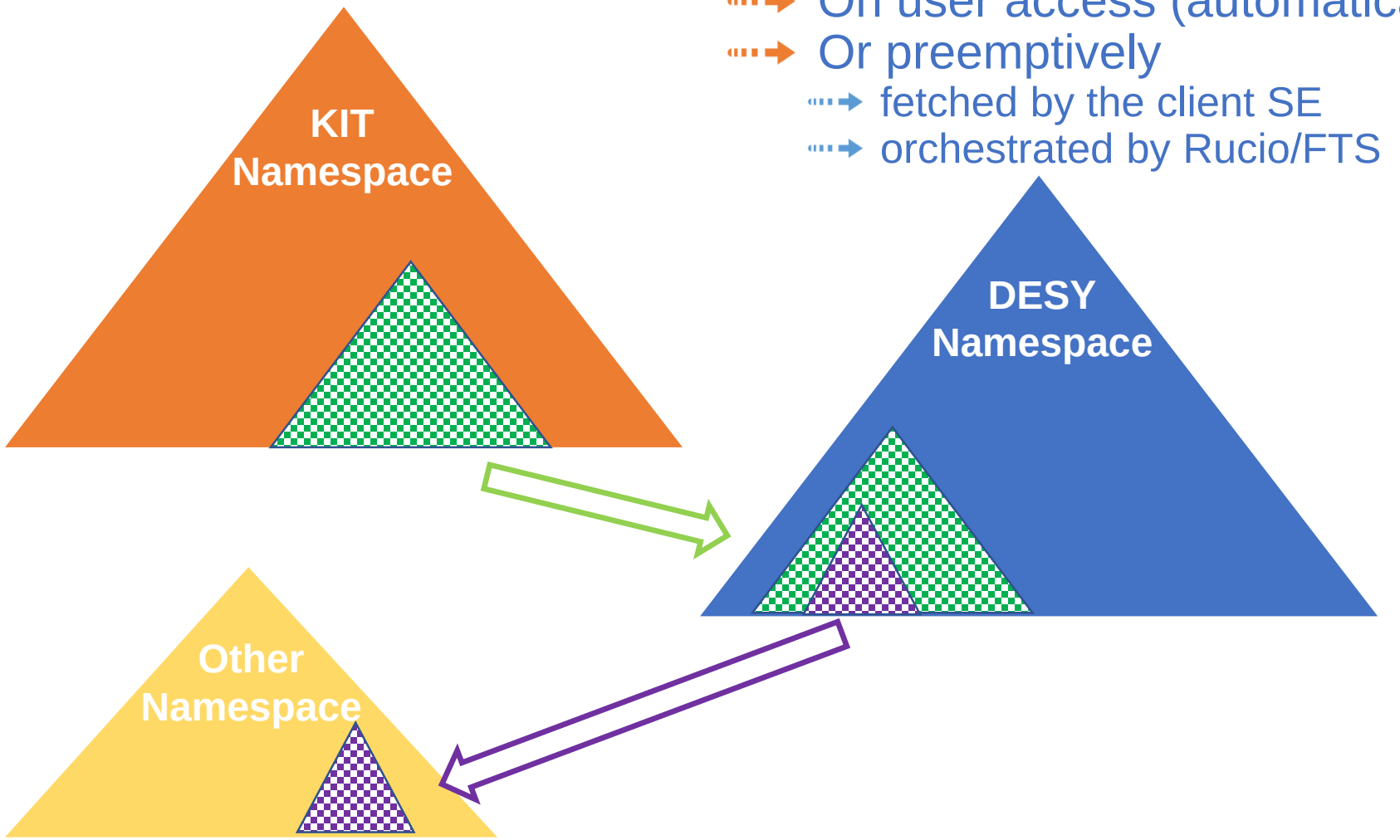


Advantages

- 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.

Namespace View

- Namespace is “Mounted = Visible”
- Data is cached at the remote site
  - On user access (automatically)
  - Or preemptively
    - fetched by the client SE
    - orchestrated by Rucio/FTS

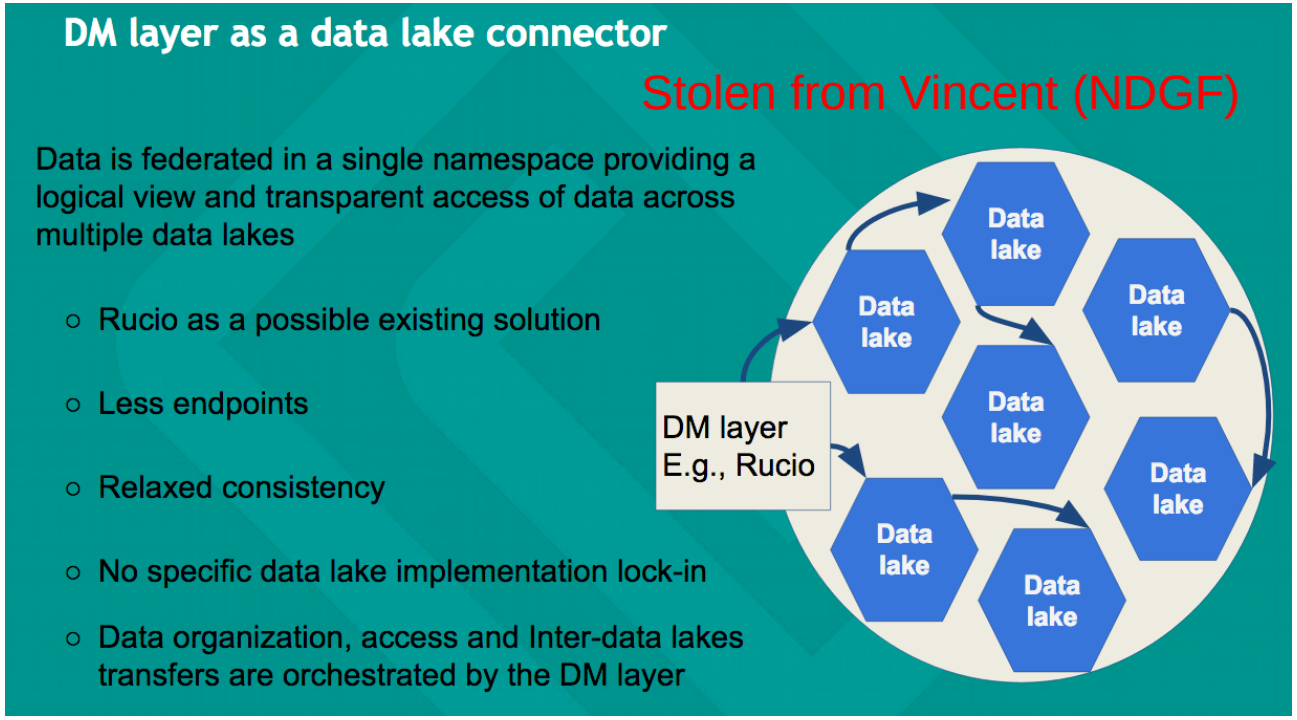


Requirement for the complex cases

- 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

Where is the data lake in those scenarios ?

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



Summary