

EOS CITRINE 
&
Future Storage R&D 

Andreas-Joachim Peters
IT-DSS-TD

• Summary of EOS Beryl

- what functionality is new

• Citrine

DRAFT

- what functionality is planned/on its way and what is relevant in the context of ALICE and future storage deployments

• Diamond R&D

DRAFT

- how to get a scalable (DB free) namespace/filesystem at low cost



- **cover accidental deletions**
recycle bin
- **improve reliability/high availability**
Master/Slave namespace
- **decrease cost, increase reliability**
ECC erasure encoding/RAIN
LRU cache & policy based file conversion
- **client for multithreaded applications**
new XRootD client
- **integrate remote CC according to IT planning**
GEO replication support Wigner/CERN
- **add standard interface**
WebDAV/HTTPS support with krb5 + X509 authentication





- Inter Group & Geo Balancing
(relevant for CERN/Wigner CC split)
- Scale-Out authentication
(relevant for CERN agile batch infrastructure)
- XRootD 4 + ReadV support with RAIN files
(relevant for analysis and space usage reduction in EOSALICE)
- Topology aware Scheduling & Placement
(relevant for file availability)

- **Infinity**
add concept of **VST** (volume storage)



- **Unity**
add concept of central **VST** namespace and site **VST** for a read-write federation
(global data management)



From Local to Global Storage

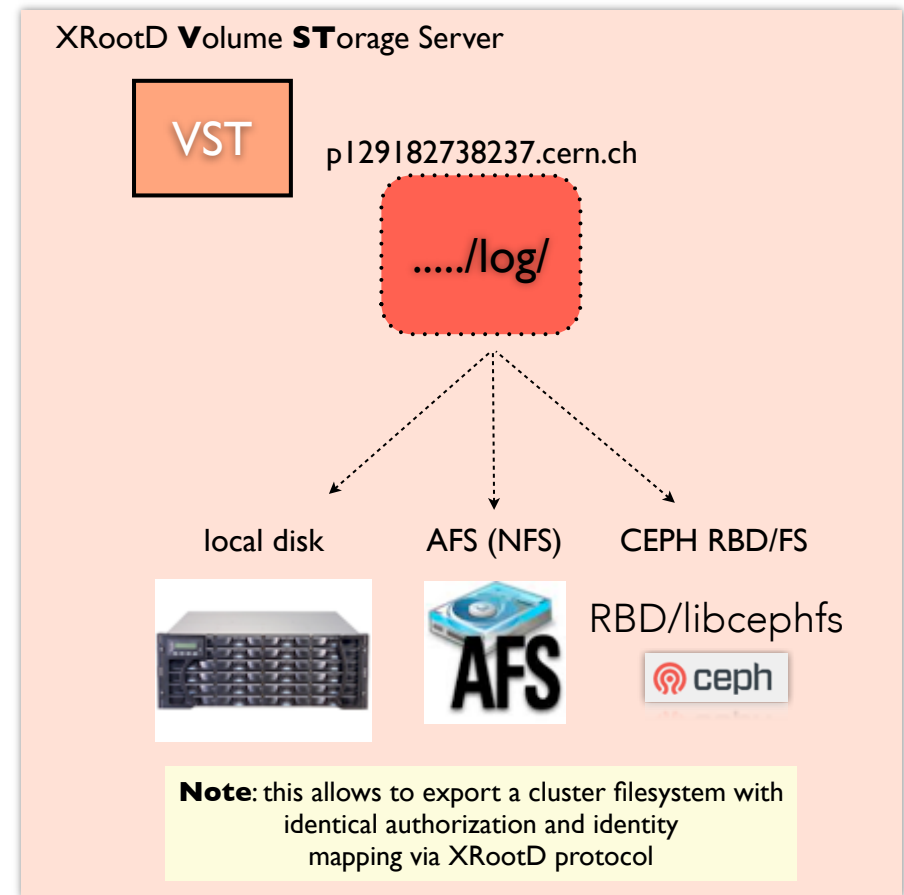
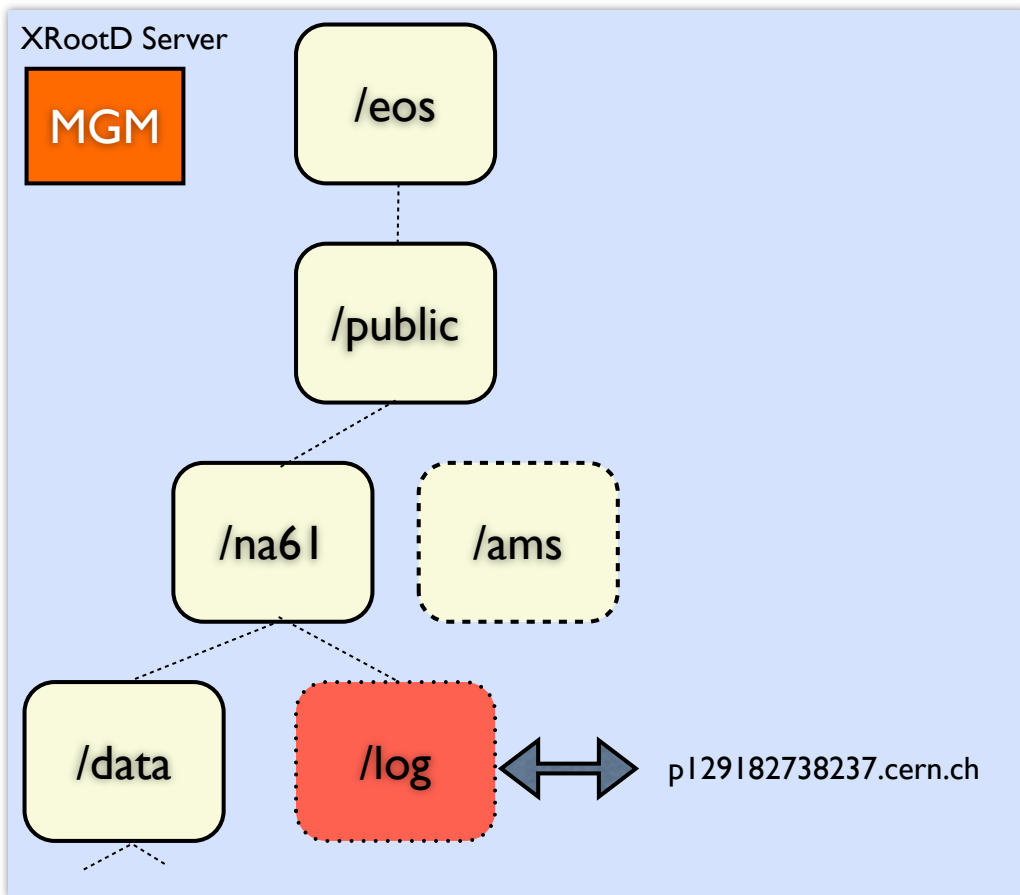


Infinity

- EOS *Infinity*

AFS-like attached volumes hosting data+meta data of a subtree

- small/many file use cases
- allows to attach any mountable FS tree into EOS namespace
- allows to have extended attributes on file and directory level for meta data tagging



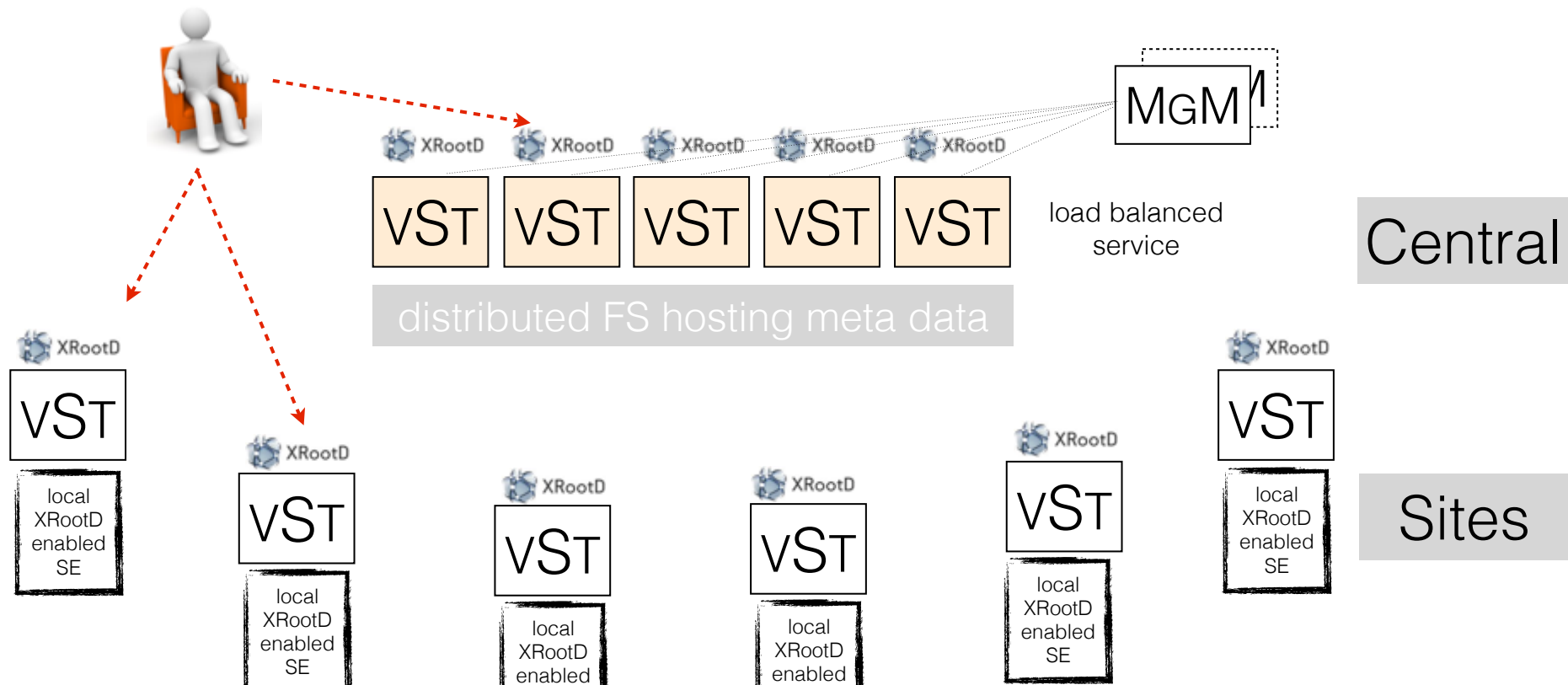


Unity



- **EOS *Unity***



Today's Federations provide a redundant functionality via a read-only overlay network. A complete storage federation should have also placement capabilities, honor replication policies and a global reliable namespace. We can use a group of VSTs to host the global logical namespace redirecting read and write requests to VSTs hosting a logical or physical namespace (sites). A site VST is just a redirection and report gateway to any regular XRootD enabled SE or a local EOS setup. For placement and file access we can extend the already existing geo placement/scheduling capabilities of EOS used for the CERN/Wigner CC setup.





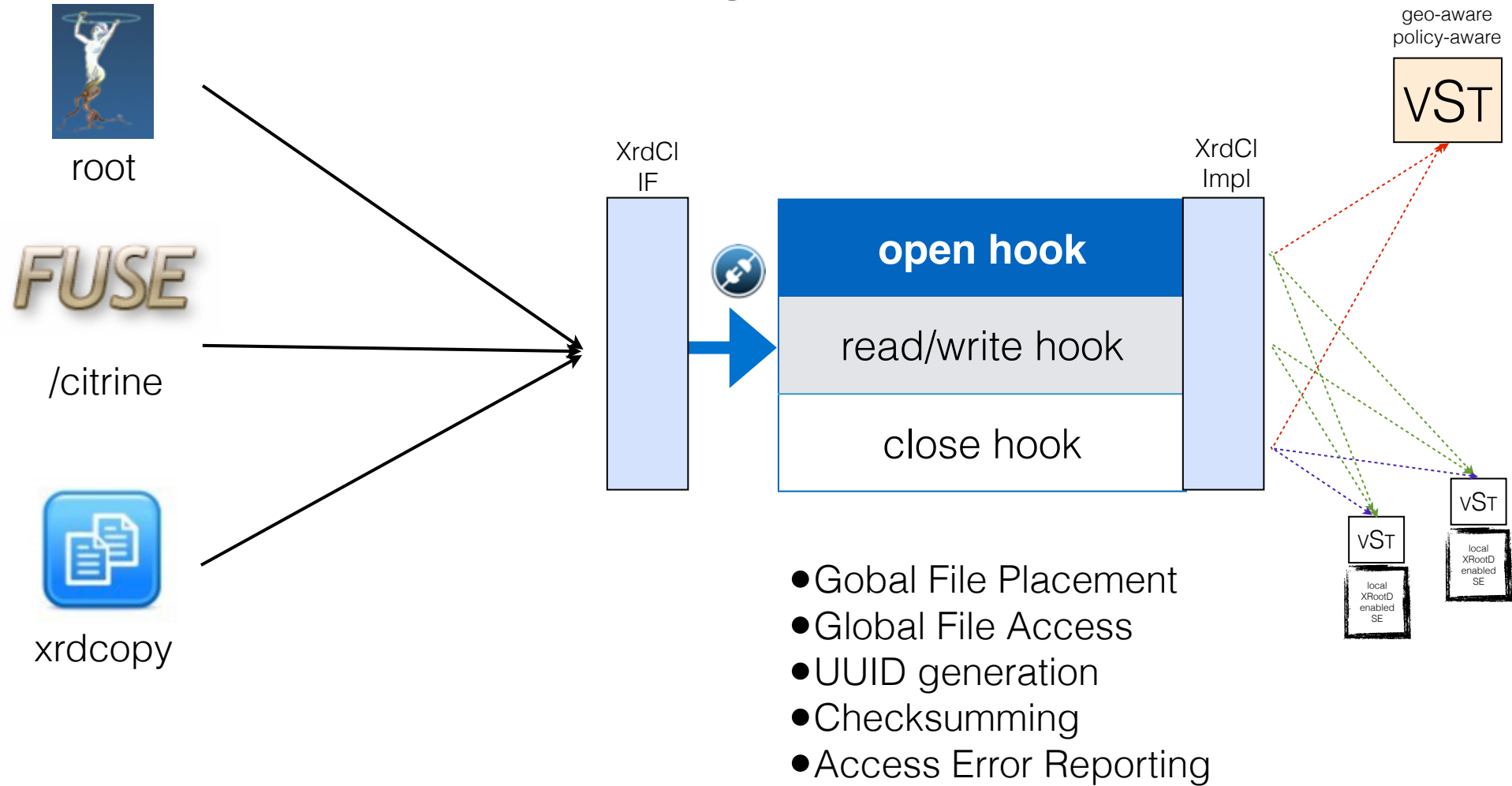
Unity



- **Keys to Unity** 
- **XrdCI** is going to provide a plugin mechanism in the IO path
- Diamond  R&D will provide **scale-out filesystem** including a fast query engine (following slides)
- Client needs **KRB5** or **X509** credentials and one can add new authentication mechanism to XRootD (like grid job authentication by job ID...)



• *XrdCI IO Plugin*

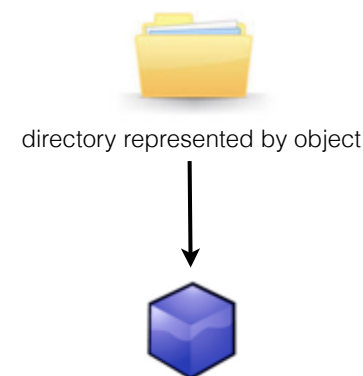
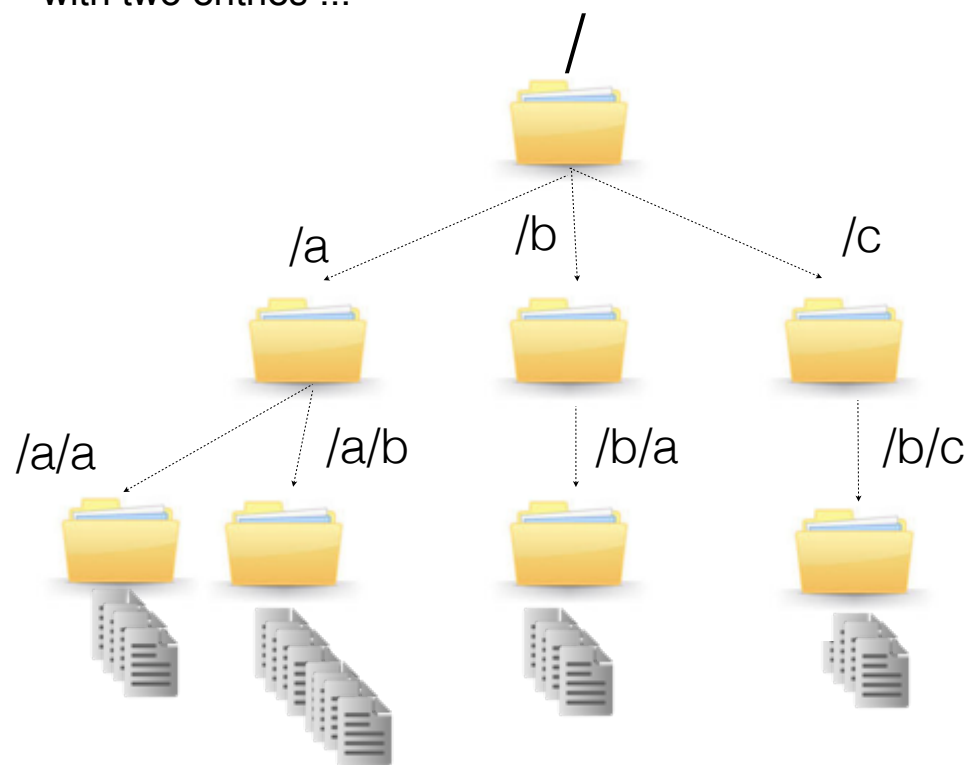


Scalable Namespace

Diamond R&D



- trivial idea: store a namespace in a scalable object store
 - we can represent data in a *hierarchical structure* using directories and files and we *don't need* to group an infinite amount of files into a single directory
 - each *file* is a *list entry* with meta data in a directory
 - each *directory* is represented as an *object* in an object store
 - to circumvent central locking we can allow a conflict if two files get created with the same name and different contents and make it visible in the namespace like a conflict in DropBox with two entries ...



dir.attributes



owner	acl	xattr
root root	xyz	user.x sys.y

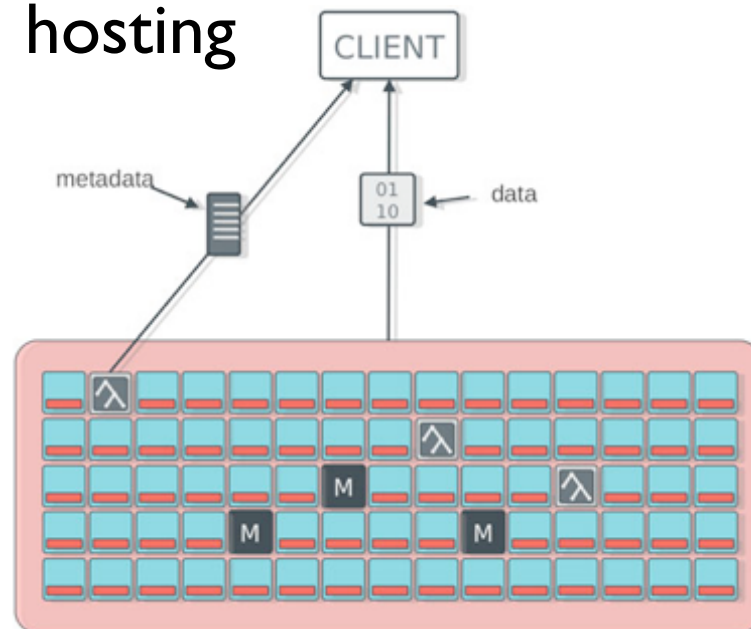
file table

Name	Size	Cks	Locatio	UUID
a	1	0xa	1:2	A
b	2	0xb	2:3	B
c	3	0xc	3:4	C
d	2	0x4	4:5	D
e	1	0x5	5:6	E

An existing Object Store ...



-  is an open source implementation of an object store providing features like *dynamic resizing*, *self-healing*, *guaranteed consistency*, *low read latency*, *async object IO*, *extended attributes + key-value map per object*, *object notifications*
- IT-DSS provides now a  (rados) object store **service** with 1 PB capacity [x3] (~50 nodes) - initially for VM hosting





- two options for a scalable namespace implementation
 - full-POSIX: [cephFS](#) provides the previously described model of a namespace where directories are mapped to objects
 - today it is approx. stable with a single namespace gateway machine, the design allows up-to 128 namespace gateway machines serving each a subtree of the namespace
 - POSIX-lite: [cephFSlite](#) would be CERN R&D to provide a similar but simpler model without the strong consistency constraints of POSIX and without the need of gateway namespace servers
- meta-data queries/views without a D&B ?
 - a meta data search is equivalent to set off an avalanche on the object store in a subtree
 - CEPH allows to implement plugin-functions on objects e.g. it is easy to query attributes in a very efficient way on the server hosting an object
 - to query a subtree one descends the levels of subtrees and executes asynchronous queries on all directories



Putting the things together ... Citrine + Diamond



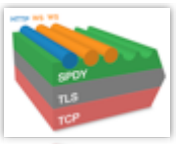
- **Citrine** *Infinity & Unity* features

aimed to improve scalability and GEO support in EOS combined with Diamond R&D towards a scalable filesystem

(possible longterm future EOS+AFS merging/replacing product?)

would provide as a by-product

- opportunity to run a *global central scalable namespace* replacing e.g. the AliEn File Catalogue and AliEn transfer services
- *site-replication policies* on subtrees/directories and verification
- XRootD would contribute with
 - a unified access via **XROOT** or **HTTP(S)** protocol to a global storage system *RW federation*
 - a low-level mechanism to move replicas between sites
TPC *third party copy*
 - an *easy to deploy* client side plugin to support global storage in the experiment framework (ROOT)



Hint of the day:
Ever heard of SPDY?

Questions?

Thank You