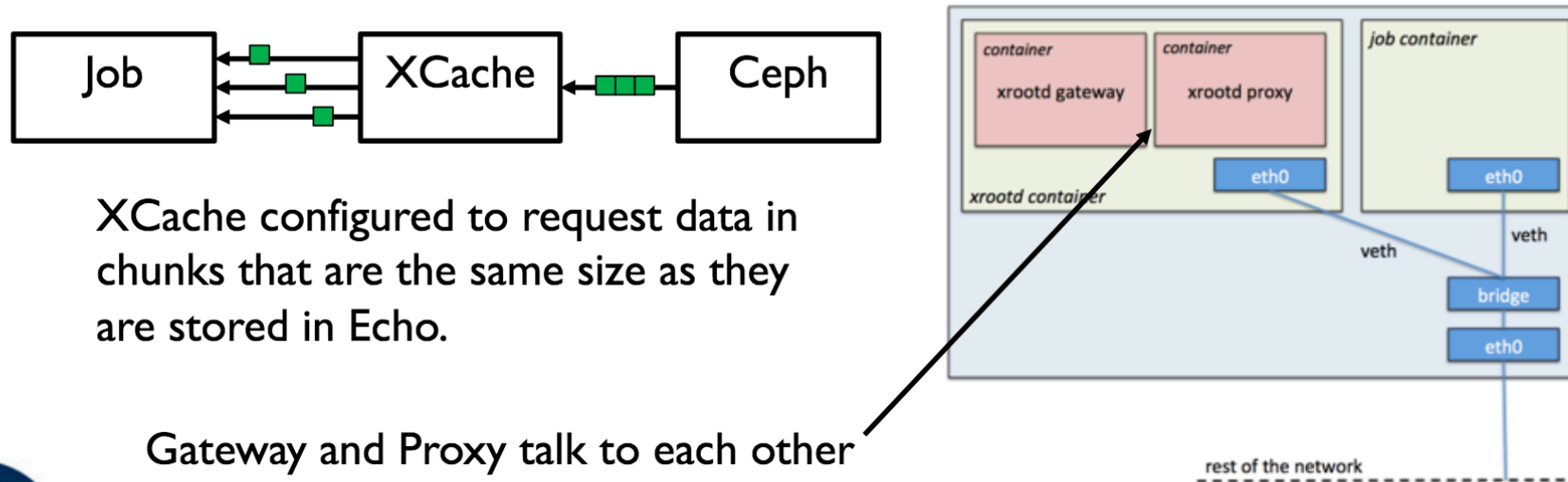


RAL XCache Setup

- A Ceph plugin has been written for XrootD.
- Object stores don't support vector reads.
- XCache is used to optimize access for jobs using Direct I/O.
- Container on each WN runs XrootD gateway which talks to Ceph and XrootD proxy which users will talk to.



XCache configured to request data in chunks that are the same size as they are stored in Echo.

Gateway and Proxy talk to each other using Simple Shared Secret Key.



Update on Xcache applications

Wei Yang, Andy Hanushevsky

Supporting Xcache applications

RAL (CEPH + Xcache integration, and many other things)

SLATE (Univ. of Chicago, containerize Xcache and other services)

EDCF (integration with ATLAS workflow - “site mover”, etc.)

Xcache for ESS

- Initial test show that Xcache and ESS can work together.
- Waiting of ESS itself to find out its position in the ES workflow.

Xcache in CVMFS and Singularity (Asoka, Rod, Wei) - **Done**

- Making Xcache available everywhere, even on WN
- Easy to deploy

Volatile RSE

- Started initial discussion with Cedric a long time ago but need to follow up.

Xcache for CVMFS to replace Squid

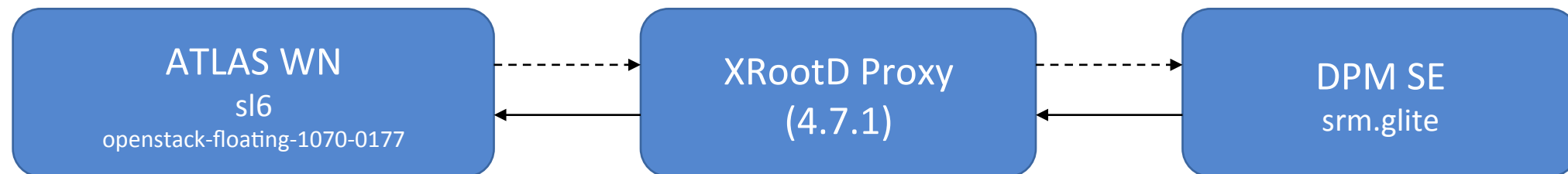
- Initial test and integration is quite successful. ([Presentation at the CernVM workshop](#))
- Need scale test
- And make a few architecture design choice.

XCache Deployment @ ED

5/2/2018

Tests on our site

- Work node
 - Single SL6 OS virtual machine running ATLAS ana jobs
 - Use NAT to bend network from our SE head node to the proxy server
- XRootD Proxy
 - XRootD proxy server (4.7.1), pointing directly to our DPM storage
- Tests with:
 - standard proxy (without caching functionality), r/w enabled
 - standard proxy, r/o enabled
 - proxy cache

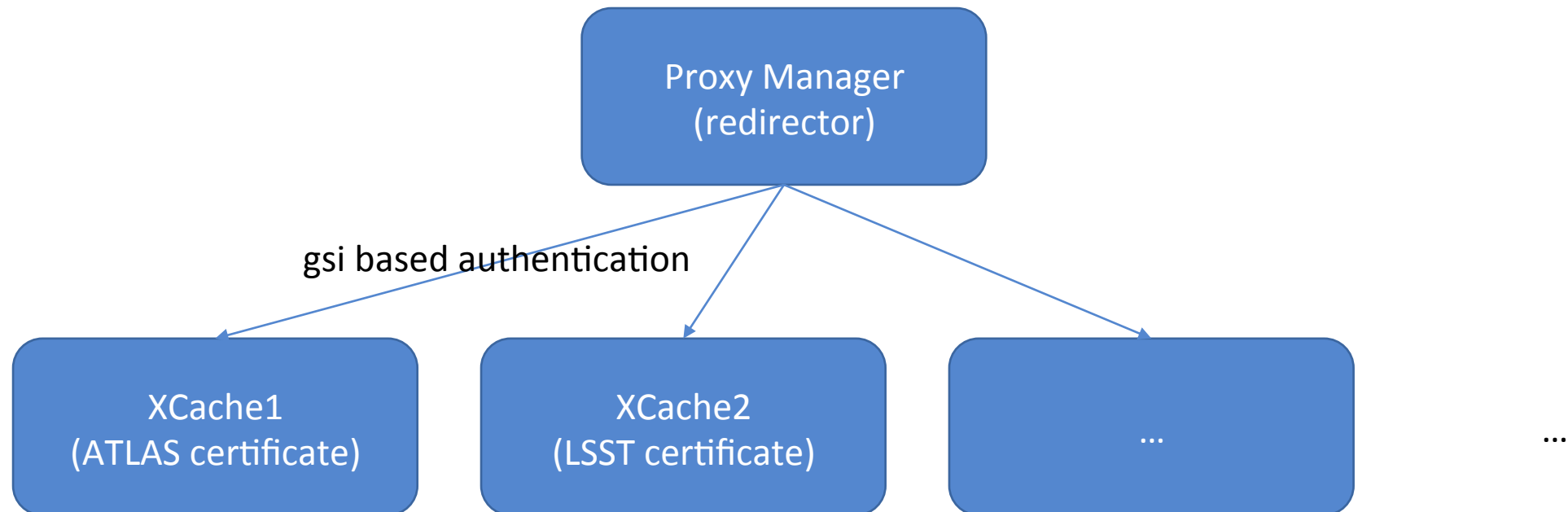


Under investigation/development

- Client plugin
 - Prefix url of files
 - Separate read/write activities
 - Reading: root://xcacheAddr//root://url/file
 - Writing: root://url/file
 - Under testing
- Site-mover
 - Under investigation

Thoughts of XCache for multiple VOs

- Use of multiple instances of XCache as a cluster
- Under tests to find potential defects
- What about forwarding clients' credentials?



Automated Cache Deployment & Operation

Ilija Vukotic • TCB

February 5, 2018

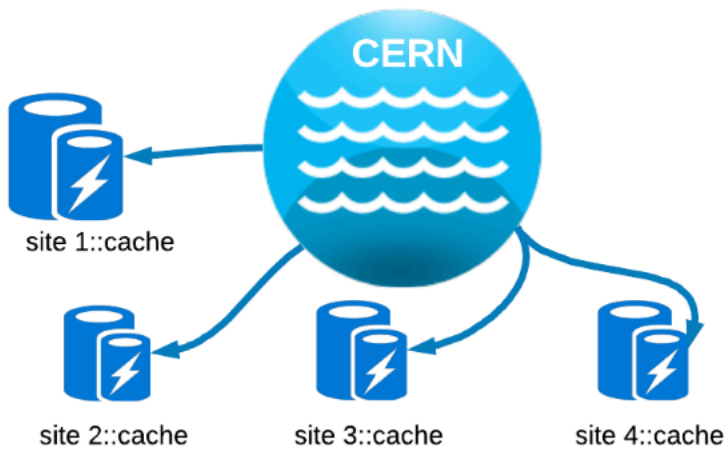
Future storage model: lakes & caches

In the context of HL-LHC we aim to have fewer storage elements in order to reduce costs.

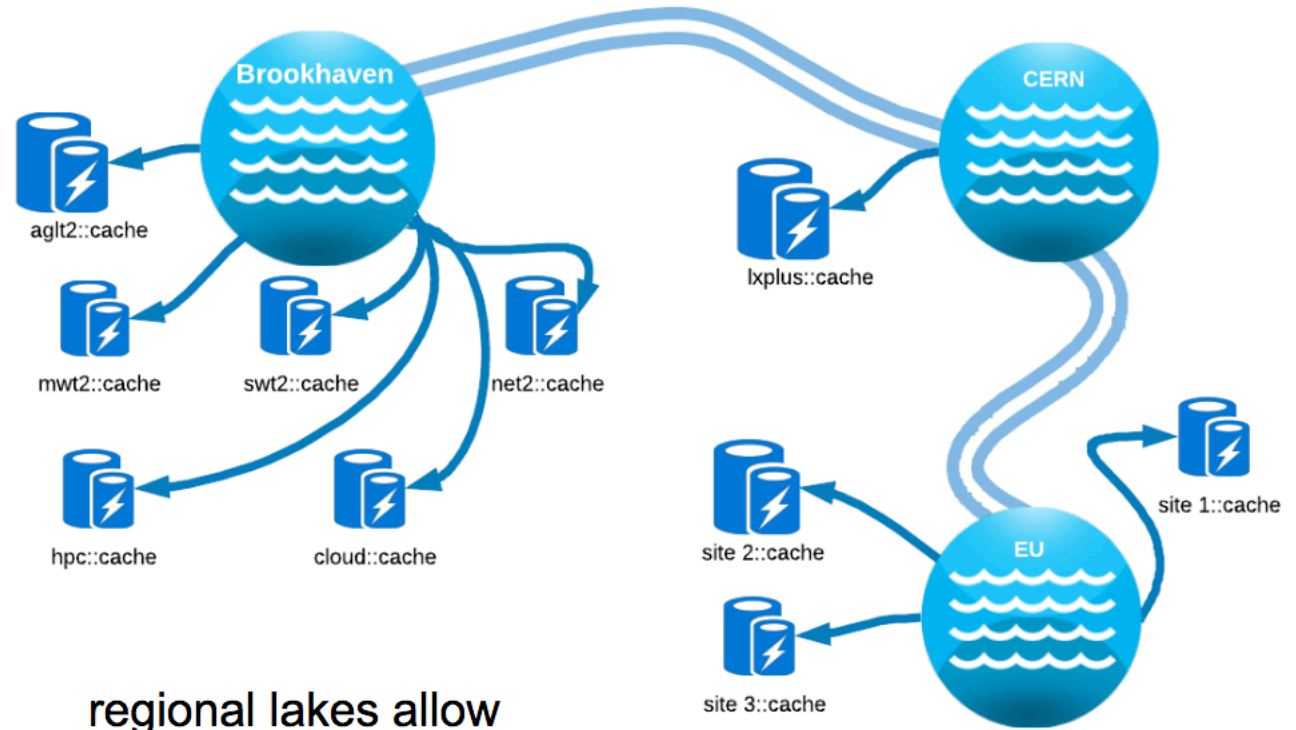
Provide a single logical SE provisioned by a large volume of highly performant storage (a "lake")

- Sites outside the data lake have no persistent storage. All data consumed by processors at those sites is either streamed or cached.
- Data products from processing centers may be either managed (they return to a lake) or un-managed (e.g. small analysis output) and can be sent directly to a user's storage site (or cloud account)

Lake models require caching



a single lake
model for
ATLAS



regional lakes allow
more access modes
from reduced latency

Automate & centralize deployment

- From FAX deployment we learned the hard way that it is too much to expect site administrators to:
 - Read xrootd config manuals
 - Correctly map it in AGIS
 - Keep software running and up to date
- This time around our goal is to have a single person managing it all. We would use the SLATE platform (next slide) to do it.
- Elements:
 - Slate edge cluster nodes at places that want to have xcache
 - Dockerized xcache (& yaml and potentially Helm charts)
 - Kubernetes deployment(s)
 - Establish new operational procedures

SLATE (services layer at the edge)

Slate (NSF project, Chicago/Michigan/Utah) will provide infrastructure to run edge services.

At the core it is a distributed kubernetes cluster.

It would enable an expert to configure, start, stop, recreate a caching server at a remote site, monitor it, check logs, etc., following best practices for containerization.

Will come in three levels of authorization:

- low level - executing Kubectl commands (advanced, for the slate team)
eg. `kubectl create -f xcache_mwt2.yaml`
- medium level - CLI abstraction (for authorized VO managers)
eg. `slate xrootd start --site=MWT2`
- high level - web interface (stretch goal) or Helm charts (VO managers)

Dockerized XCache

Dockerize XCache - in progress (80% done)

Kubernetes deployment - in progress (40% done)

Documentation - continuously updated

Monitoring - not started

Integrate running services into AGIS - not started. Will do it manually at the start.

All code in [GitHub](#), docker auto built at [DockerHub](#).

MKdocs - [Here](#)