CERNBox: sharing reloaded

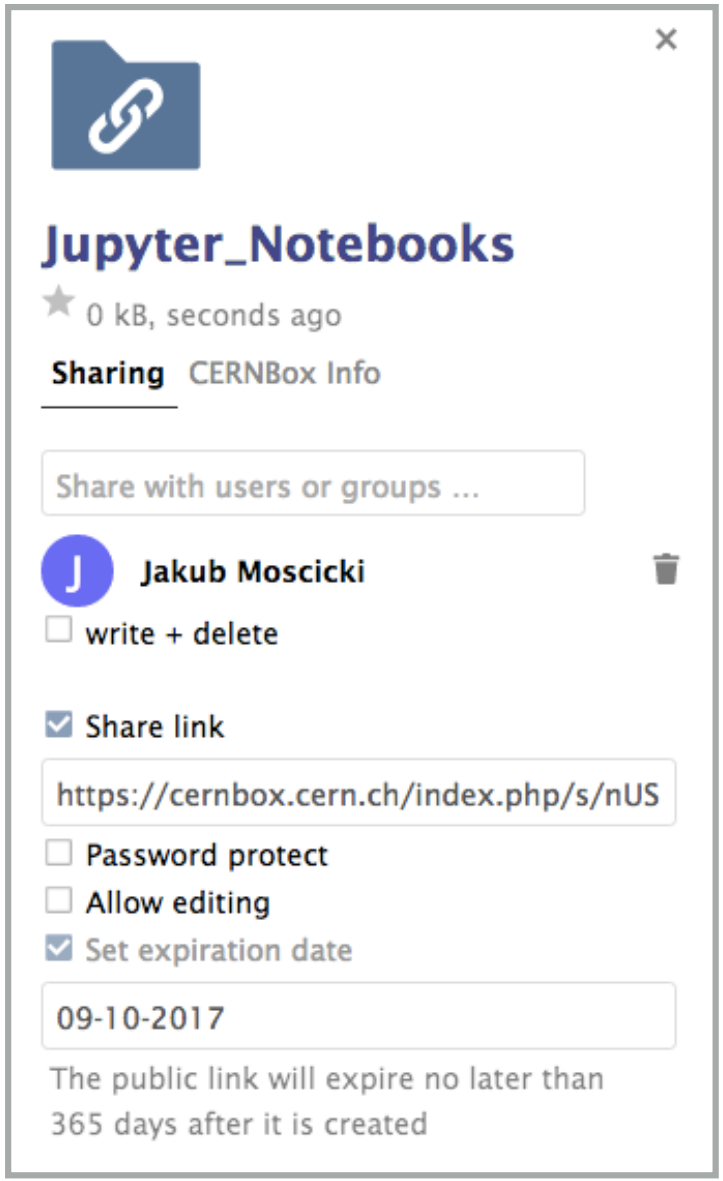
Jakub T. Mościcki (KUBA)

CERN IT-ST

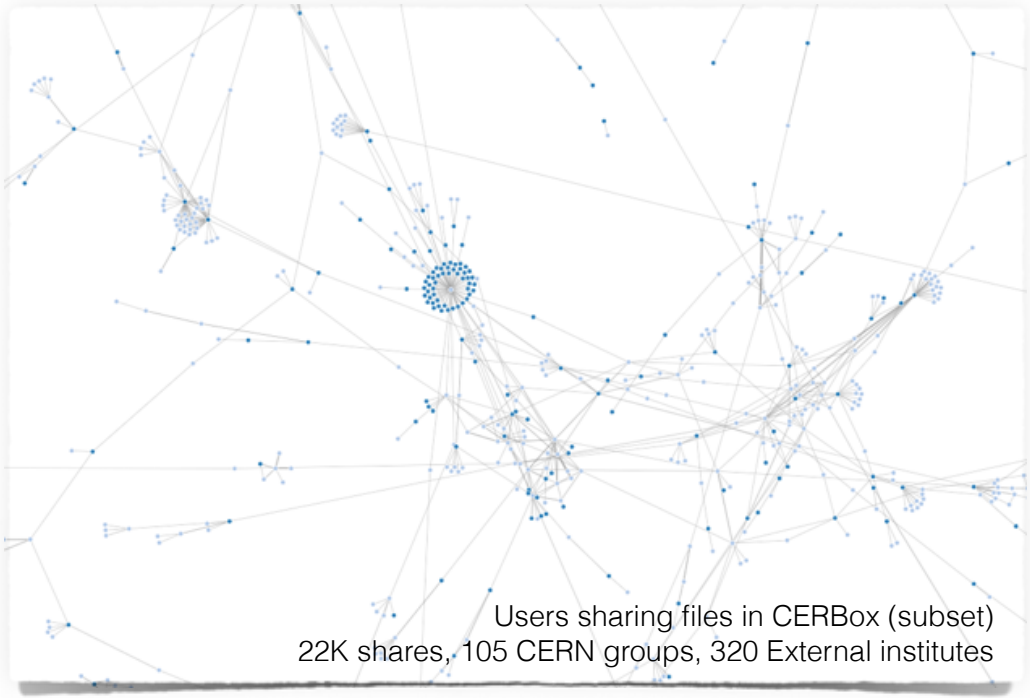
EOS Workshop at CERN 2018

Sharing in CERNBox

- Sharing made *very* easy

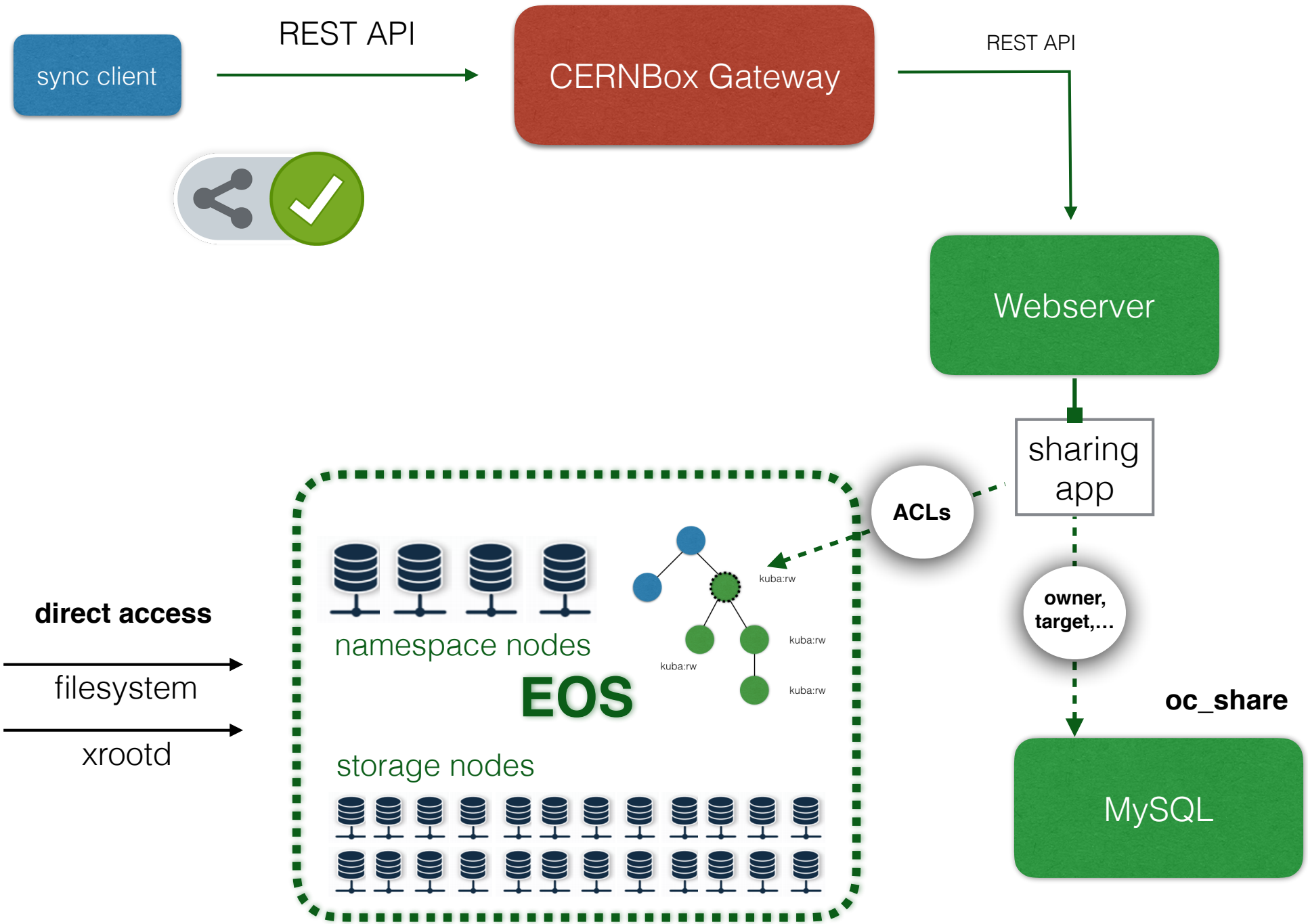


- A successful model



Users sharing files in CERBox (subset)
22K shares, 105 CERN groups, 320 External institutes

Current sharing architecture



Under the hood

- oc_share (MySQL): source of truth
- simplified permissions: read, read/write
- mapped directly to EOS ACLs
  - **eos attr set sys.acl …**
  - **sys.acl="u:laman:rwx!m,egroup:it-dep-st-fdo:rwx+d,u:kuba:rw"**

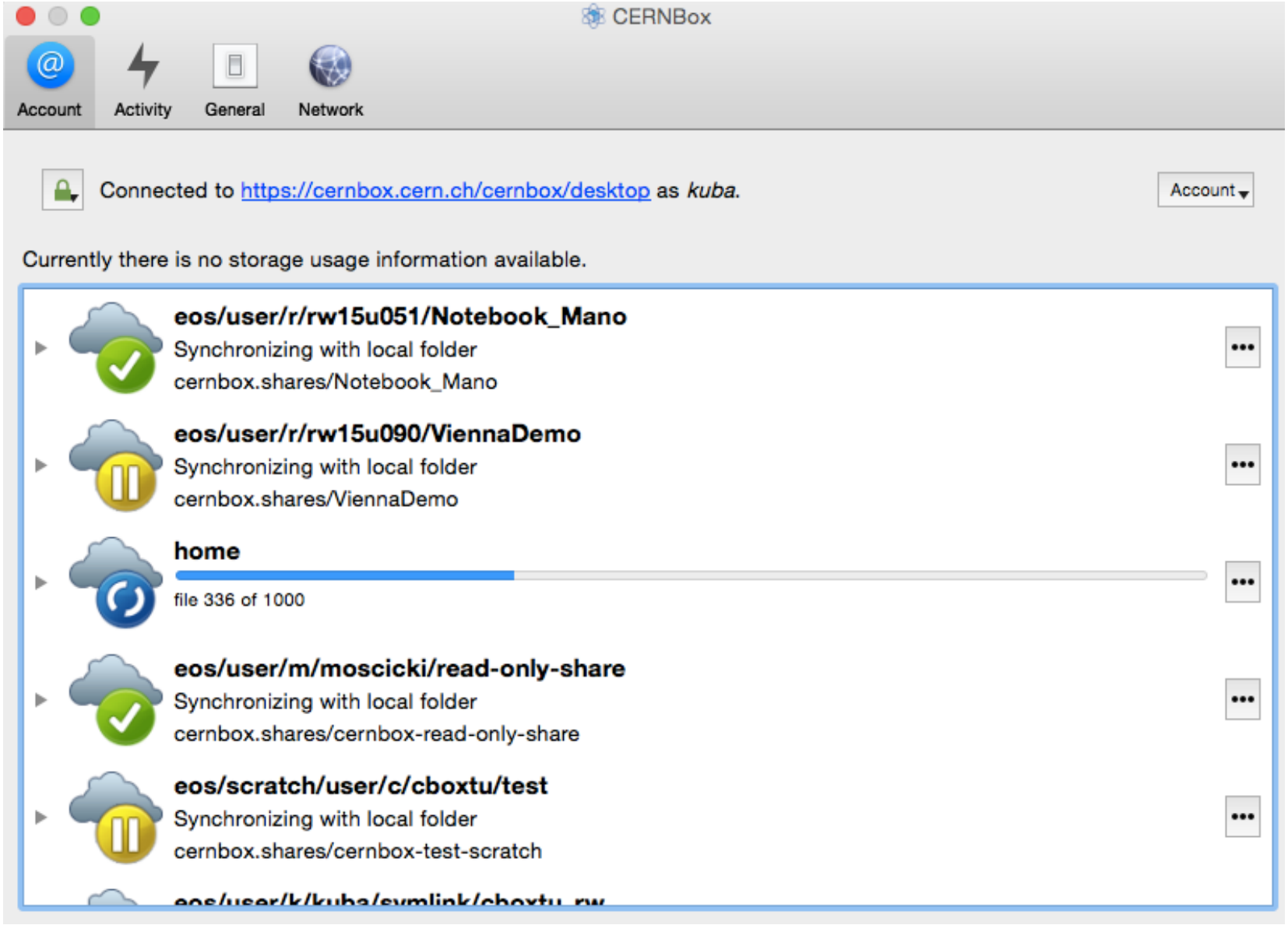
Admin tooling

- https://github.com/cernbox/cboxshareadmin
  - **cernbox-share**
    - **add** path owner sharee
    - **remove** path owner sharee
    - **list-shared-by** owner
    - **list-shared-with** user

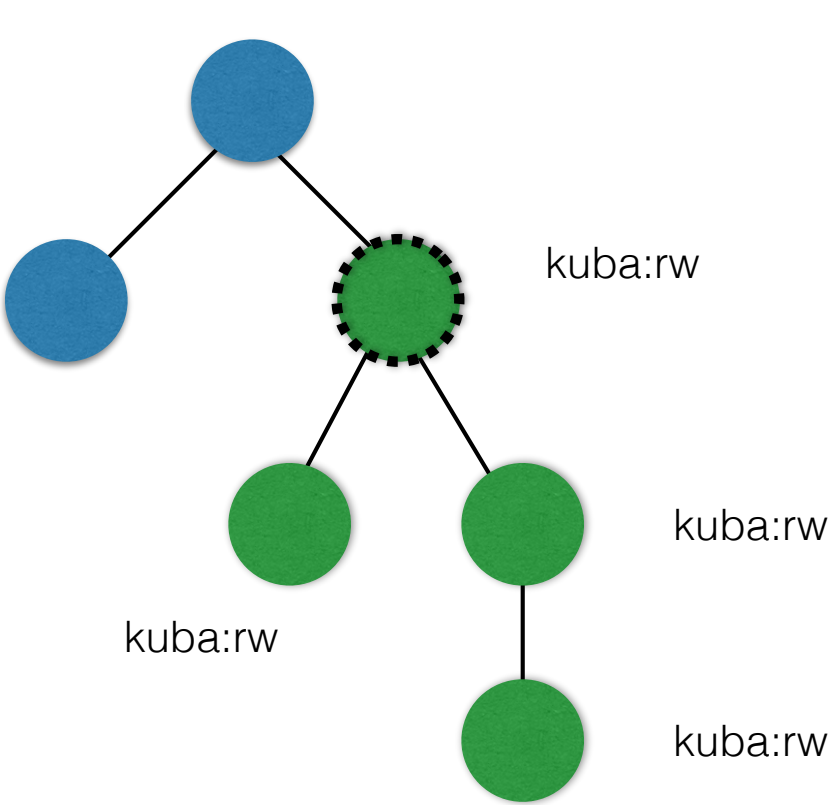
Issues

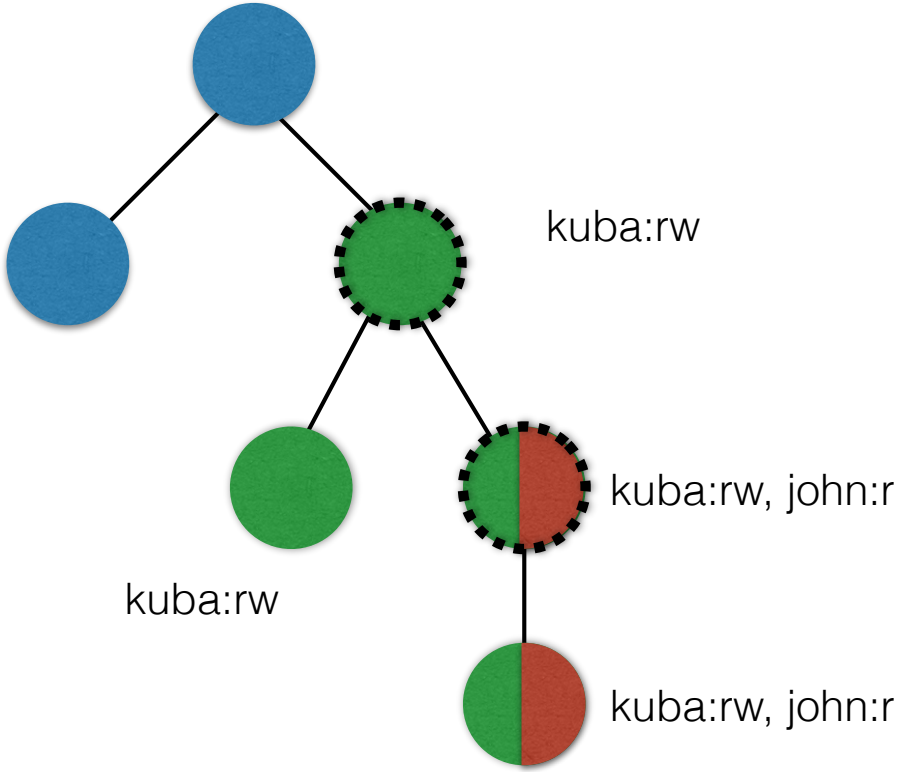
- Usability
  - hard to synchronize shares
- Consistency of ACLs
  - truth in DB vs ACLs in filesystem
- User expectation
  - filesystem view
  - sync/share view

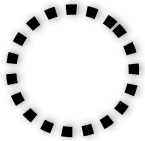
Requires multiple sync folders



ACL management: basic sharing

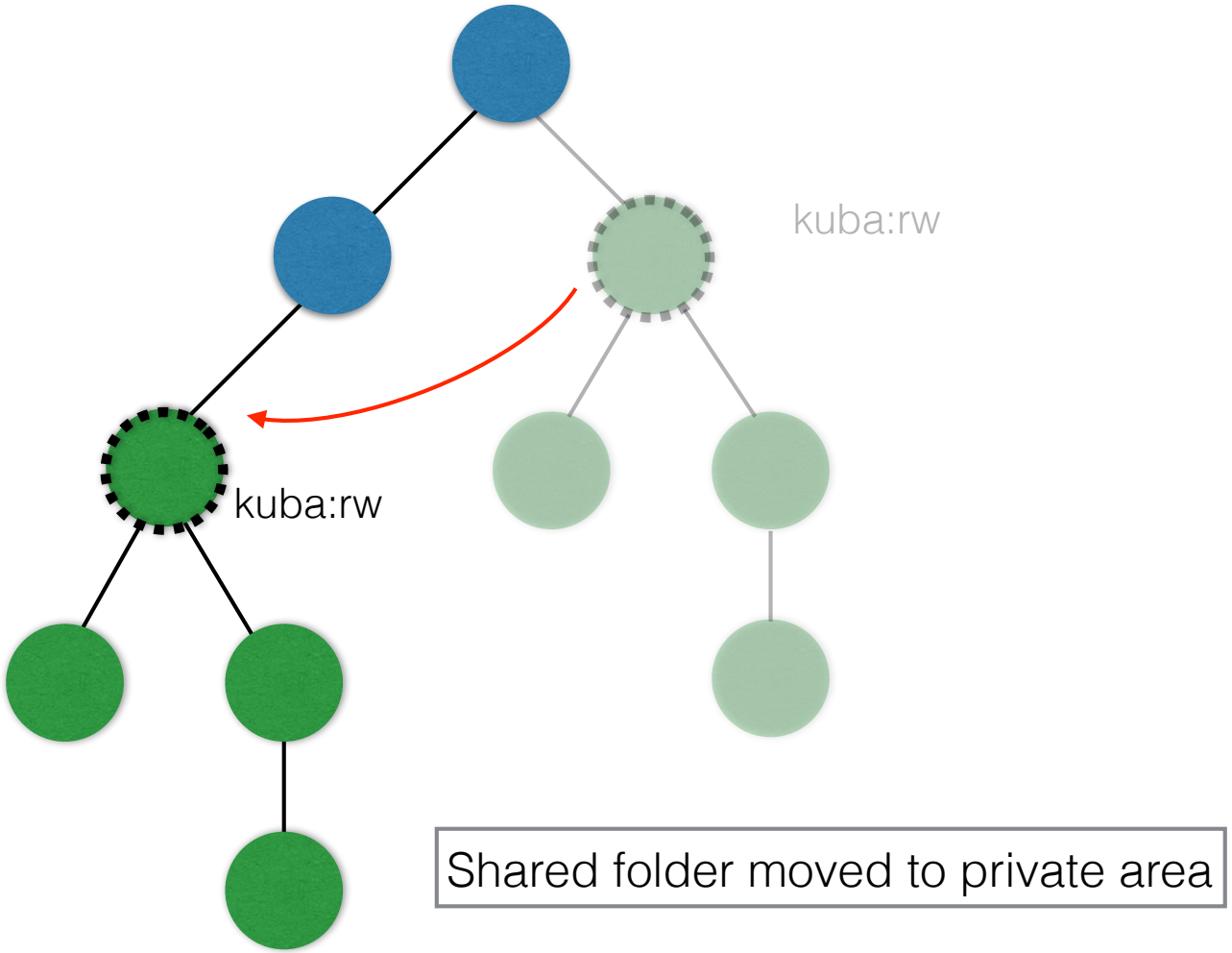




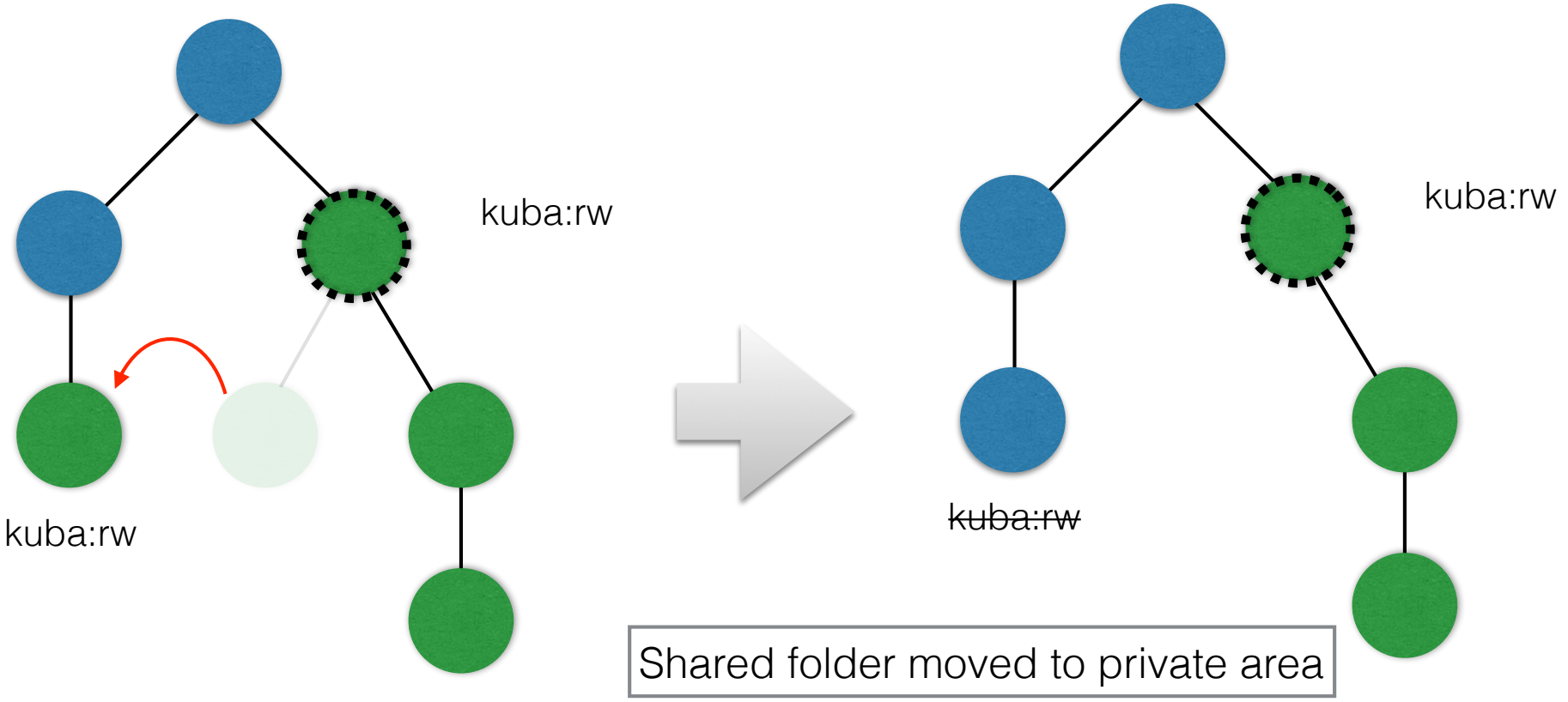


point of sharing/unsharing

ACL management: challenges



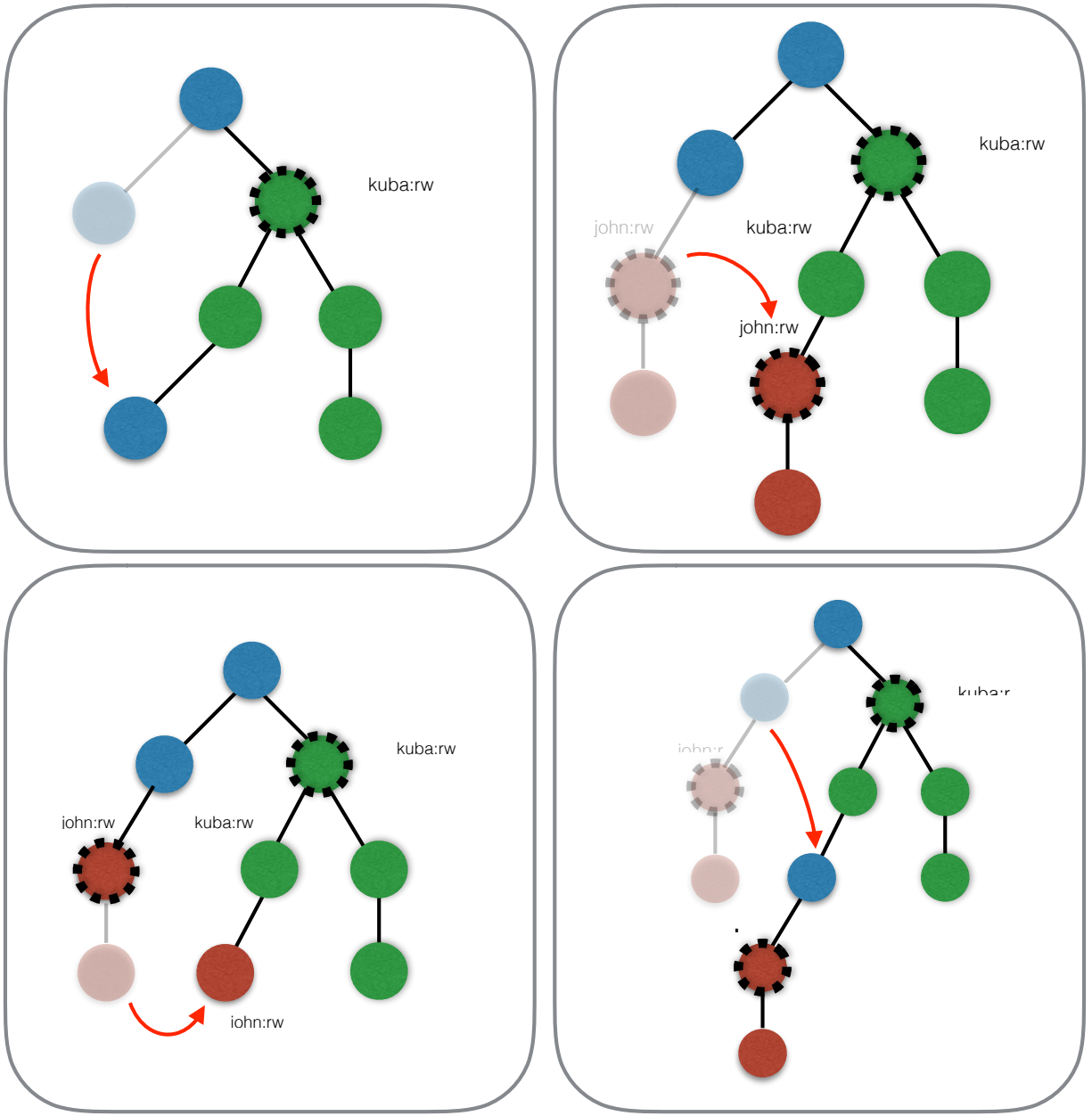
ACL management: challenges (2)



Competing strategies how to handle such cases

- local storage view via sync client
- filesystem users and expectations for *mv* semantics

ACL management: more corner cases for move



Keeping consistency

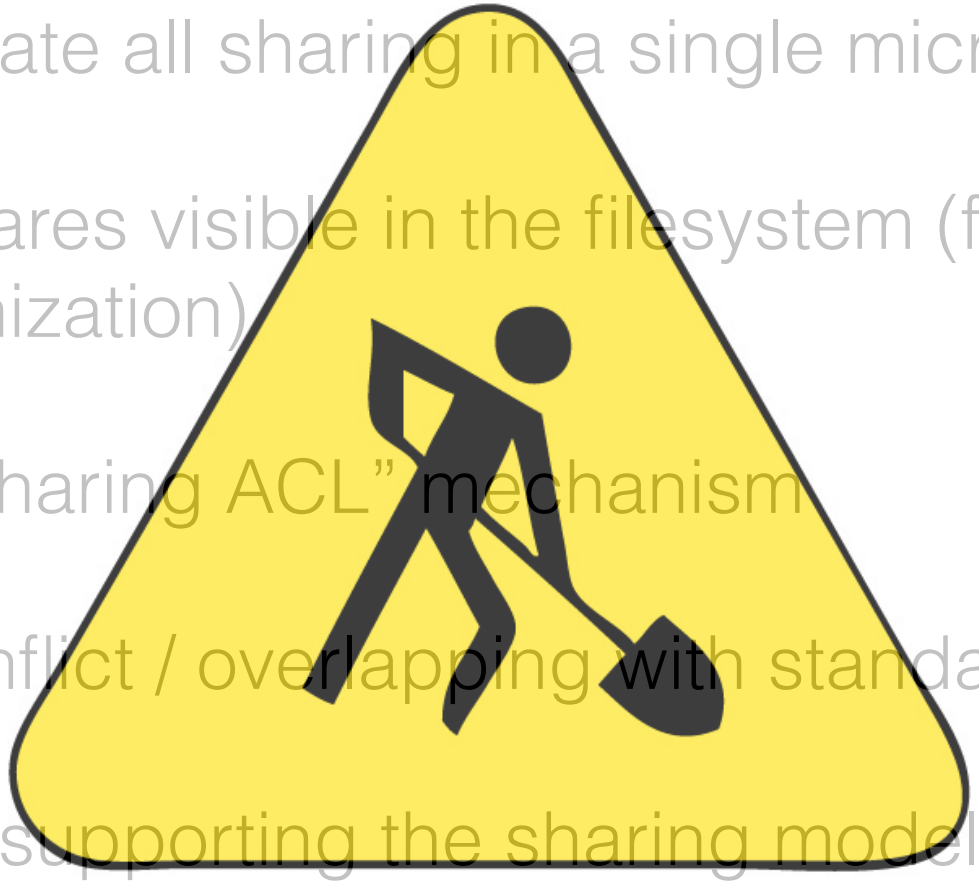
- https://github.com/cernbox/cboxshareadmin
  - **cernbox-share**
    - **verify** [--fix] owner
  - A useful admin tool (“fix a problem”)
  - Regular reconciliation of inconsistencies *a posteriori*

Reloading…

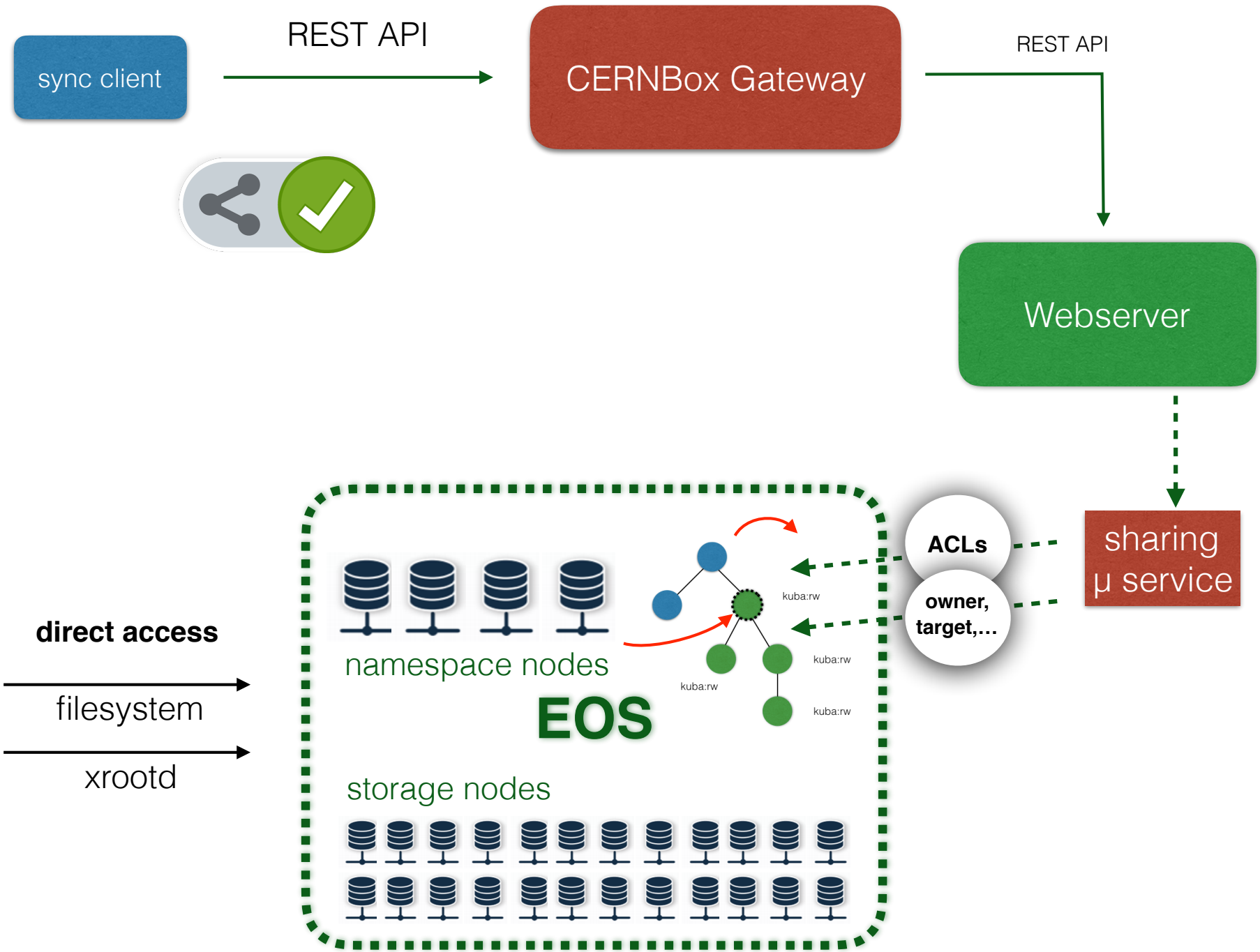
- Consolidate all sharing code in a single micro service
- Make shares visible in the filesystem (for easier synchronization)
- Add a “sharing ACL” mechanism
  - No conflict / overlapping with standard ACLs
  - Better supporting the sharing model (entire subtree)

Reloading…

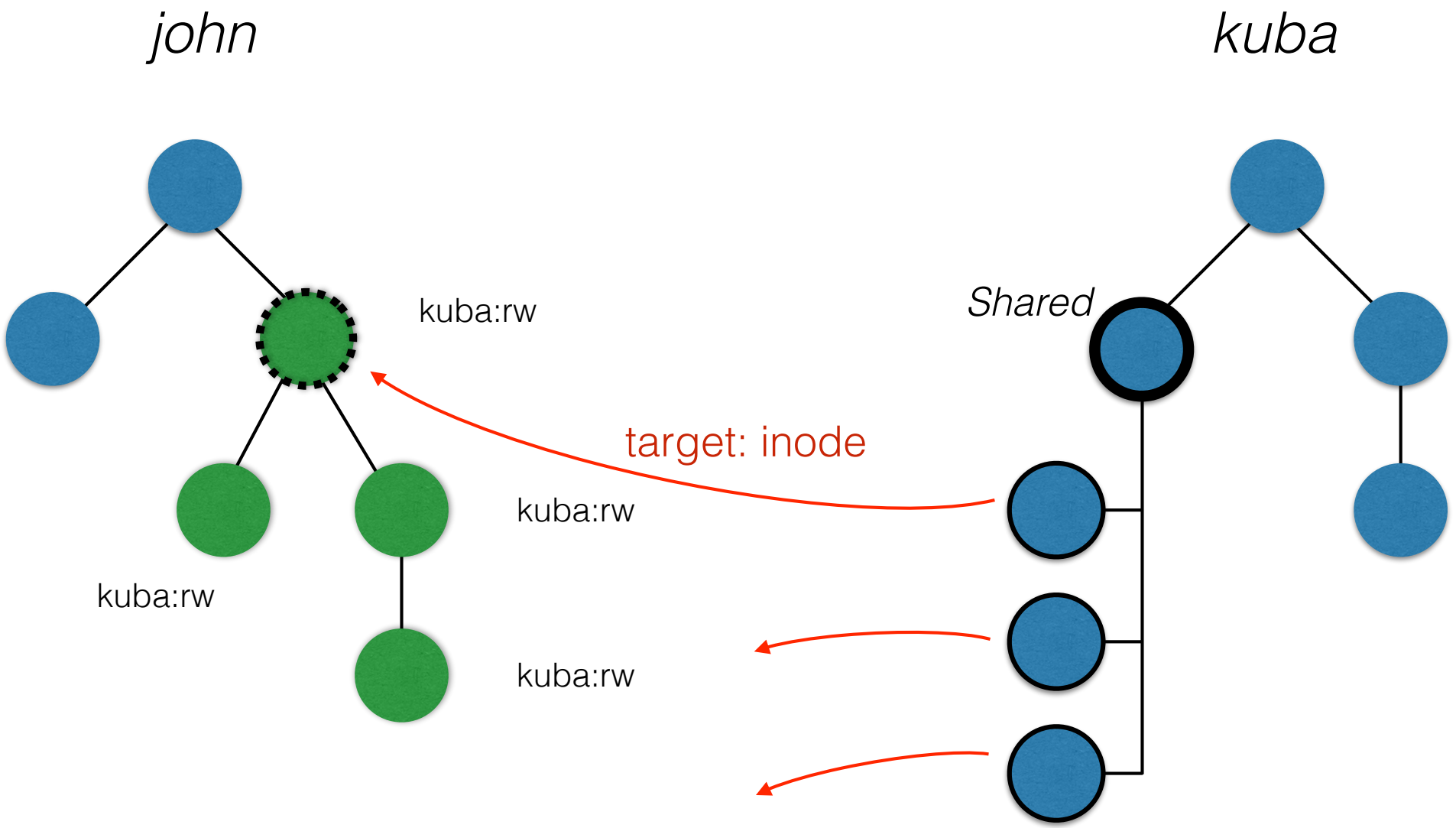
- Consolidate all sharing in a single micro service
- Make shares visible in the filesystem (for easier synchronization)
- Add a “sharing ACL” mechanism
  - No conflict / overlapping with standard ACLs
  - Better supporting the sharing model (entire subtree)



Sharing architecture 2.0



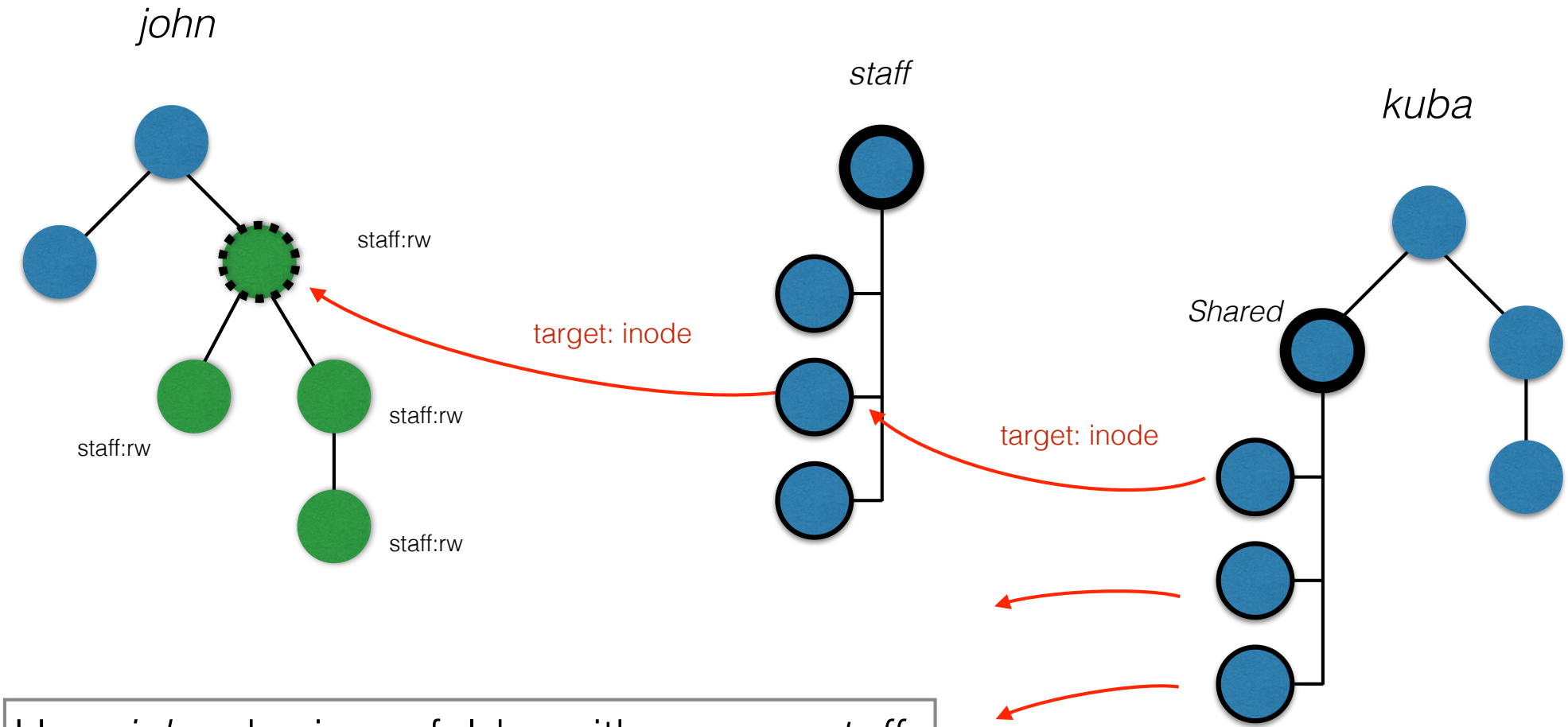
Direct access to shared folders on storage
Challenge: scaling and infinite recursion



User *john* sharing a folder with user *kuba*

point of sharing/unsharing

Challenge: update propagation scaling for group sharing



User *john* sharing a folder with egroup *staff* of which user *kuba* is member

point of sharing/unsharing

Sharing ACL

- Peaceful coexistance of POSIX/Classic-ACL with Sharing ACLs
  - Take appropriate actions with Sharing ACL on mv
- CERNBox-only user gets a consistent system
- POSIX-only user is not bothered
- Power-users have to know what they are doing

Implementation ideas

- **attr set sys.attr.link**=/eos/dev/origin-attr
  - you can update an existing share in ~O(1) in namespace
  - mv is more complicated, may result in ~O(logN) operations
  - everything else works by a standard ACL inheritance mechanism

Sharing of Notebooks

- New sharing interface for notebooks
  - SWAN frontend + CERNBox backend
  - https://github.com/cernbox/cboxswanapid
- More details: CS³ http://cs3.cyfronet.pl

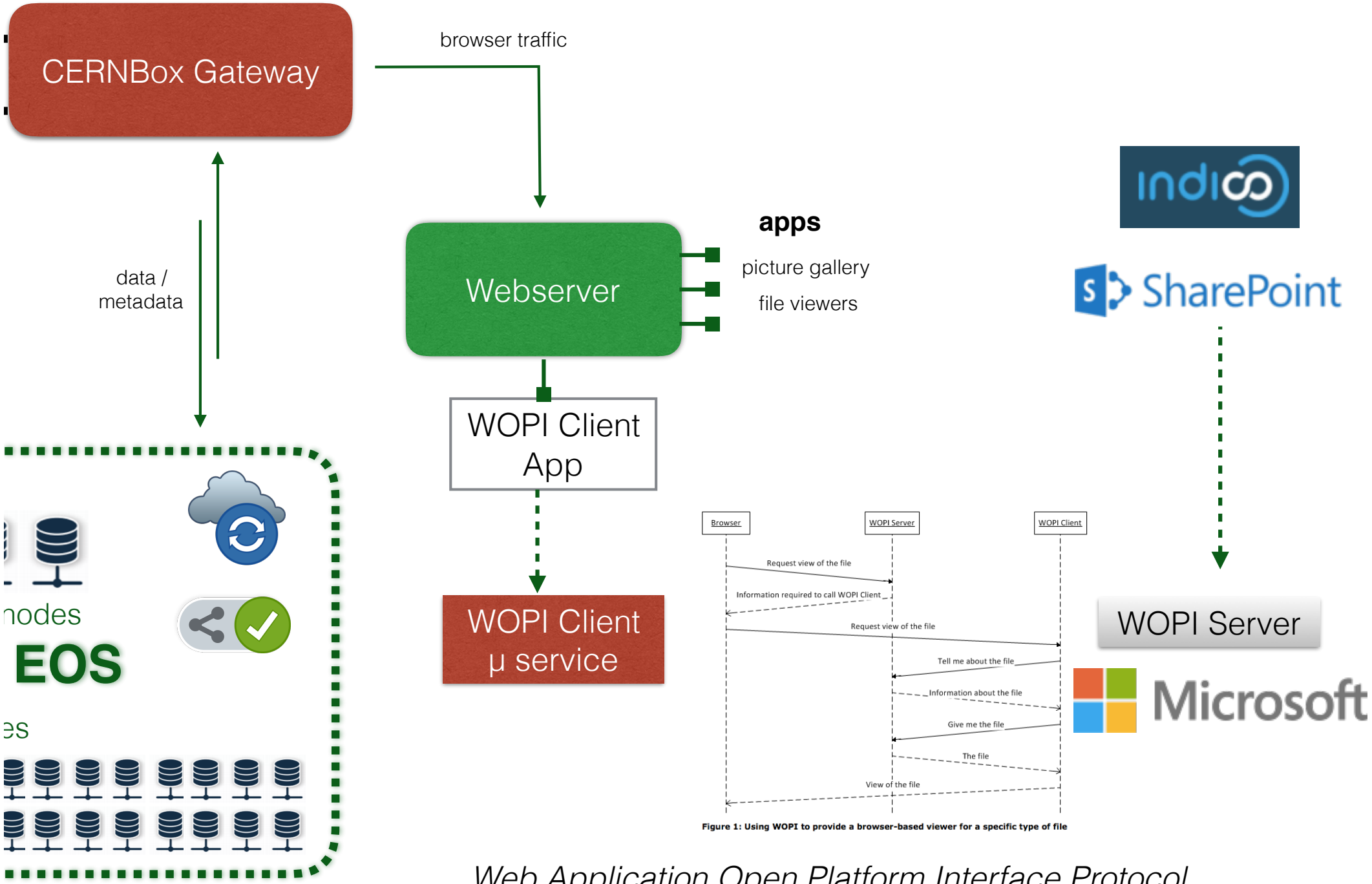
- https://indico.cern.ch/event/663264/contributions/2820873/attachments/1592406/2520626/Presentation_2.pptx

Summary

- We are improving CERNBox-EOS sharing functionality&integration
- Easier, better overall experience also for sophisticated filesystem users
- This is a gradual transition, first steps have been made, still a lot of work ahead

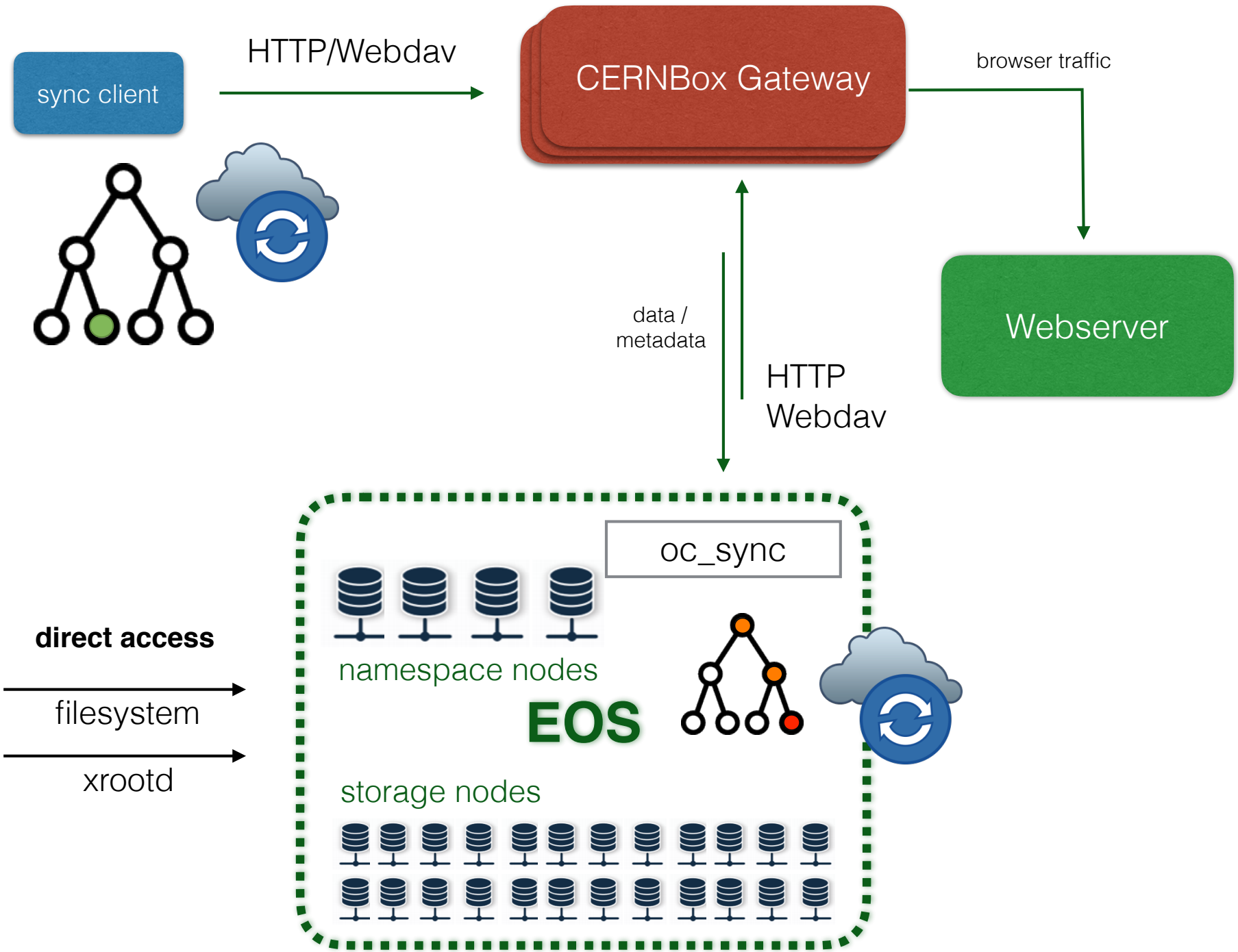
The end

Web Apps, Office and Collaborative Editing

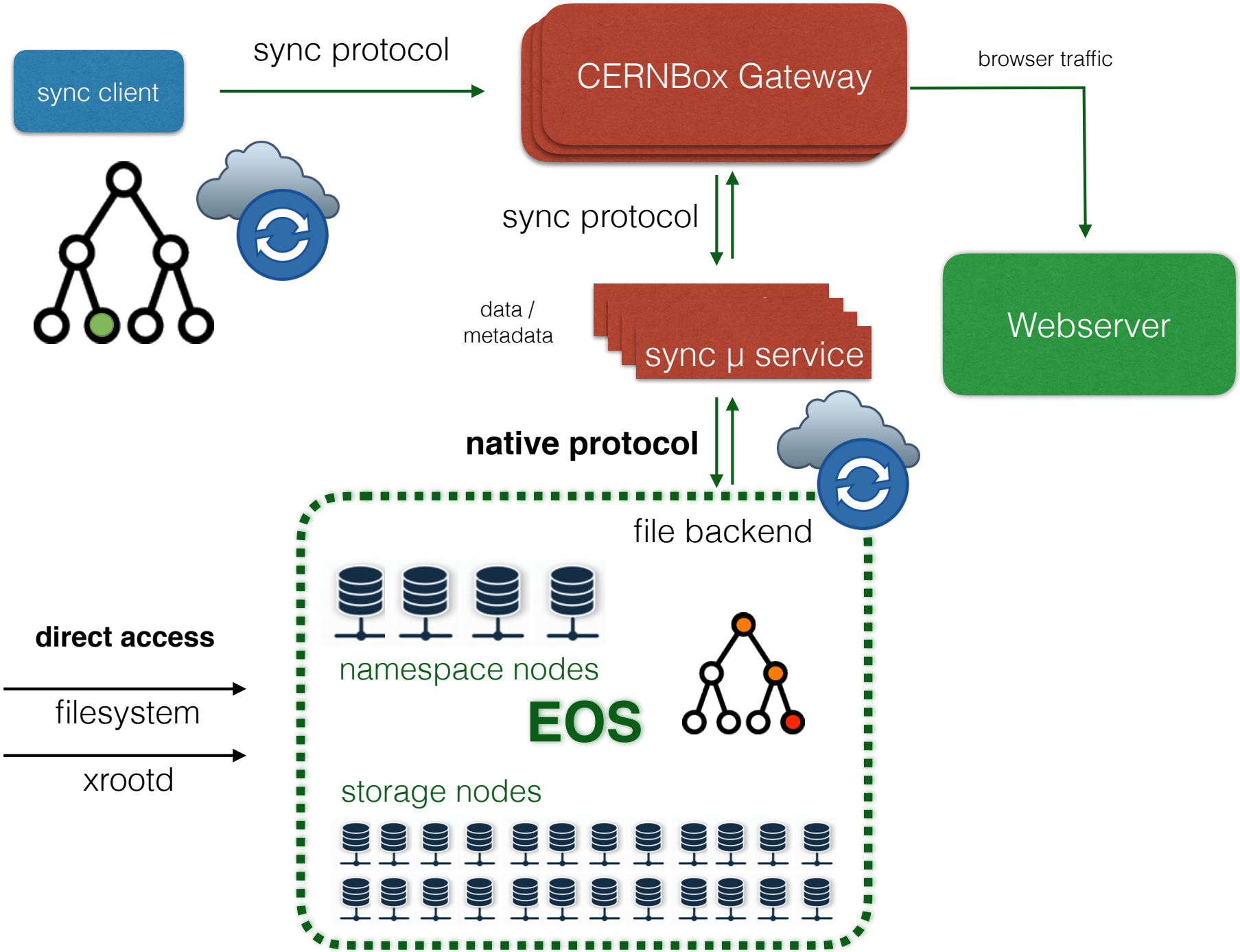


Web Application Open Platform Interface Protocol

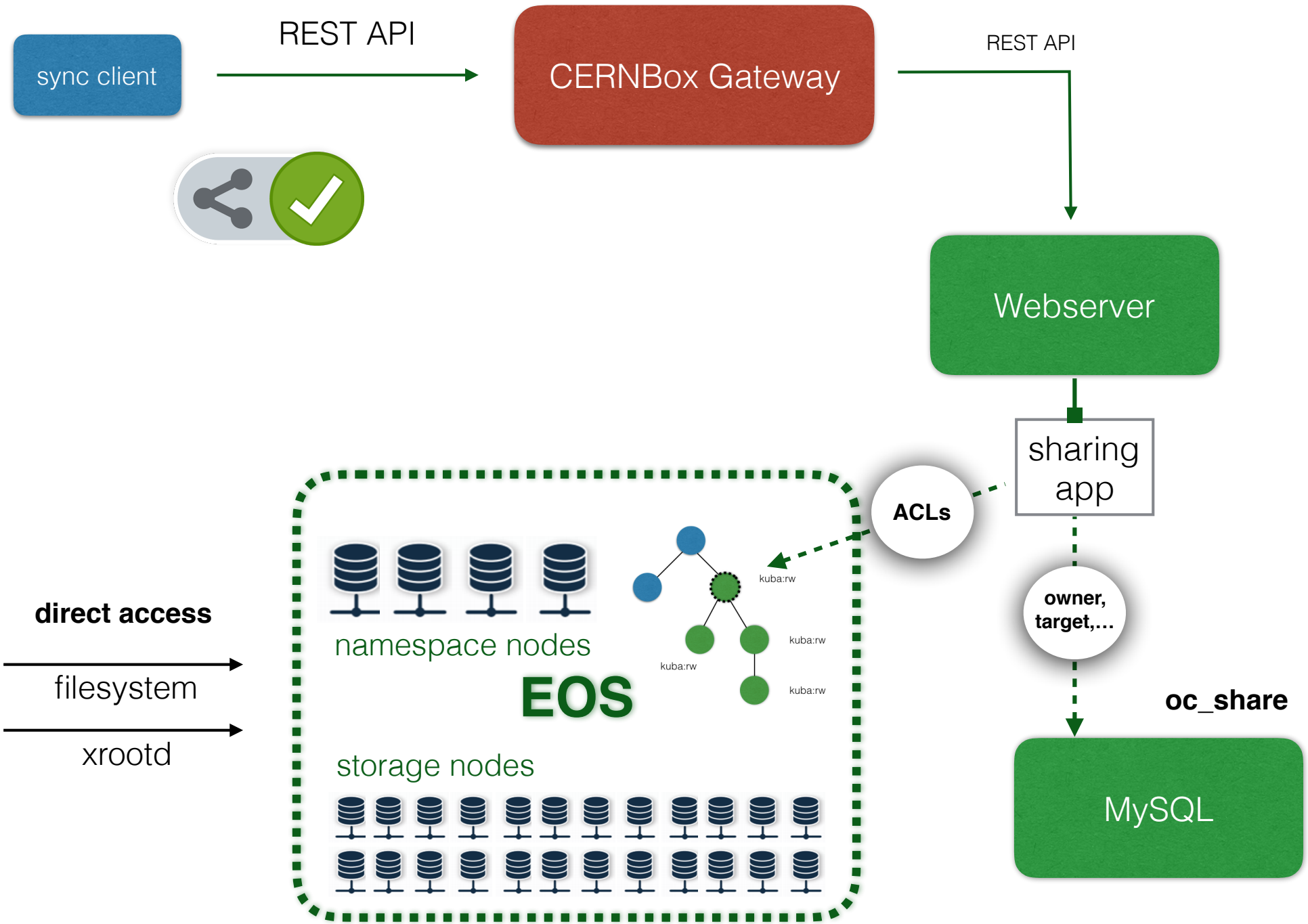
Synchronization & Data Flow



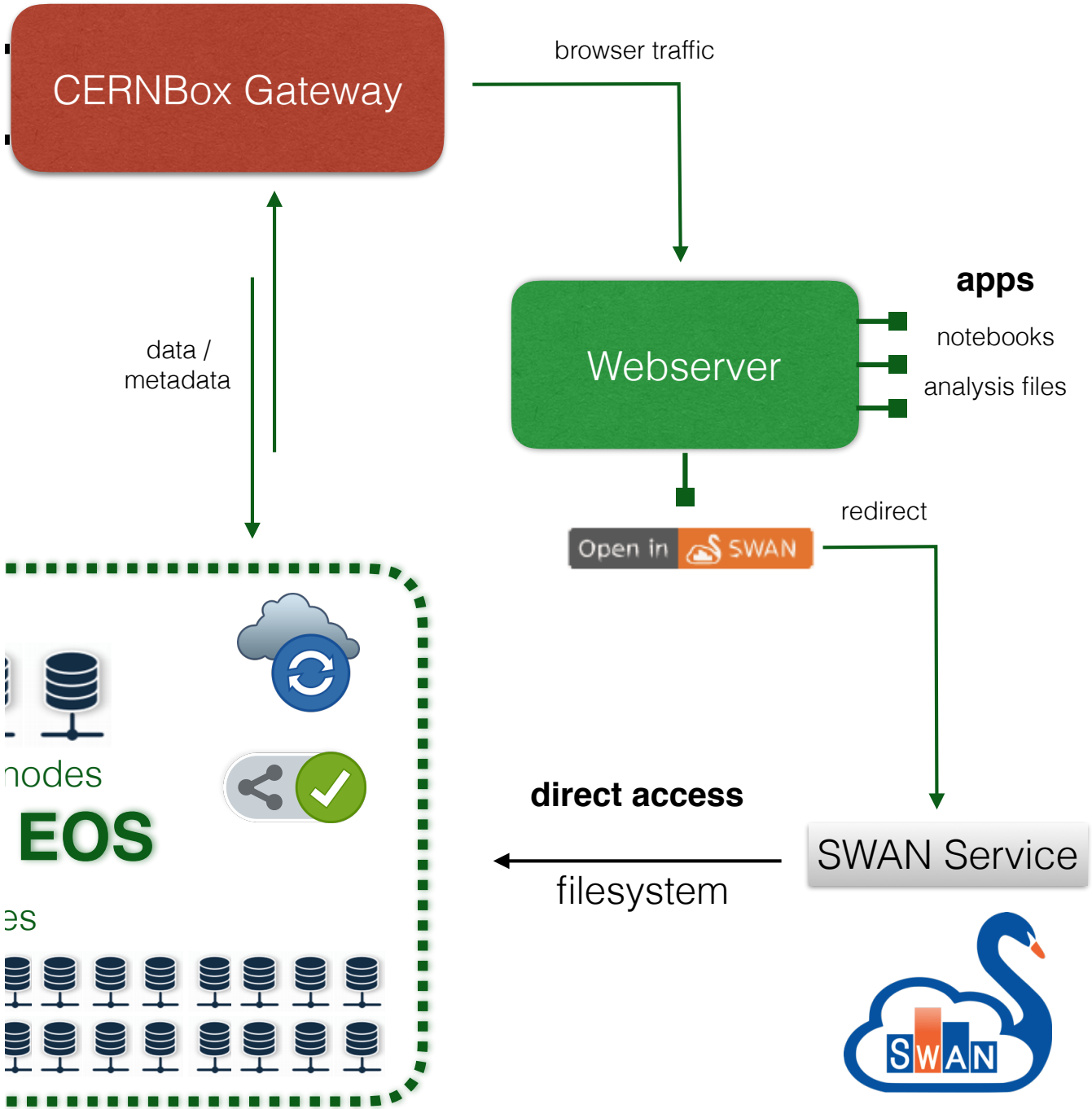
Synchronization & Data Flow 2.0

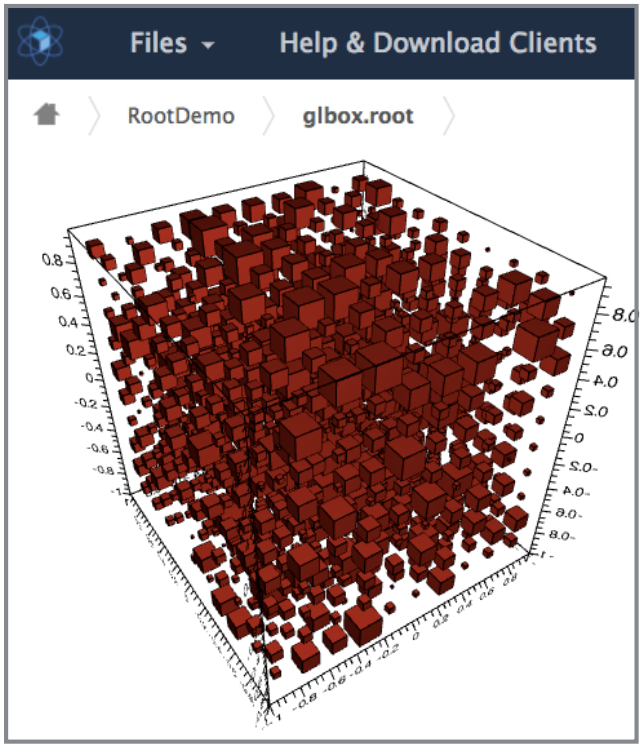


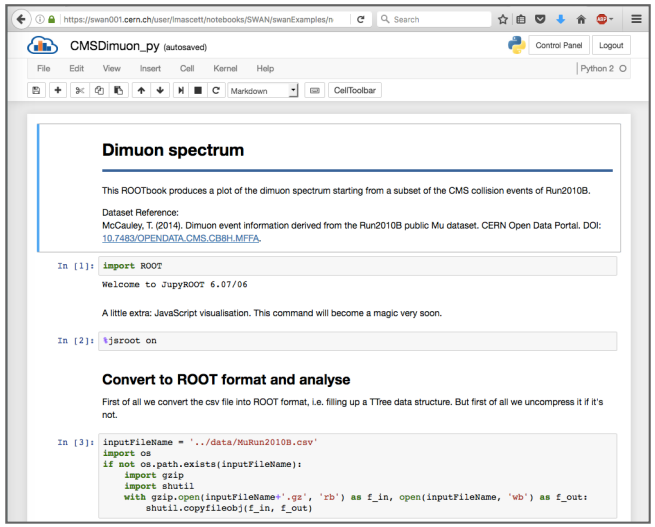
Sharing architecture



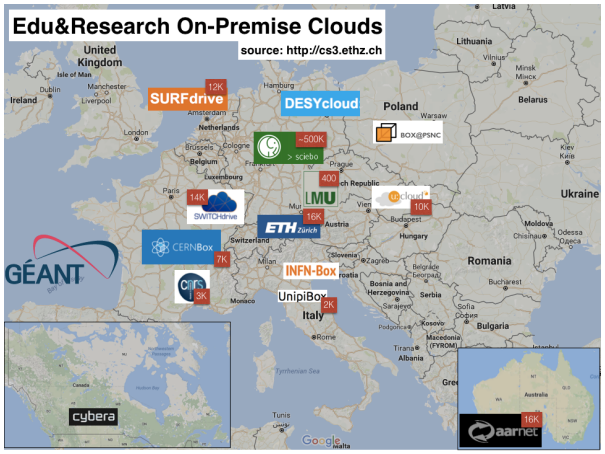
Web Apps and Scientific Computing







Federated Sharing, OCM



OCM protocol

sync client

REST API
sync protocol

CERNBox Gateway

REST API

Federated Sharing App

sync protocol

data / metadata

sync μ service

Webserver

native protocol

direct access

namespace nodes

kuba:rw

kuba:rw

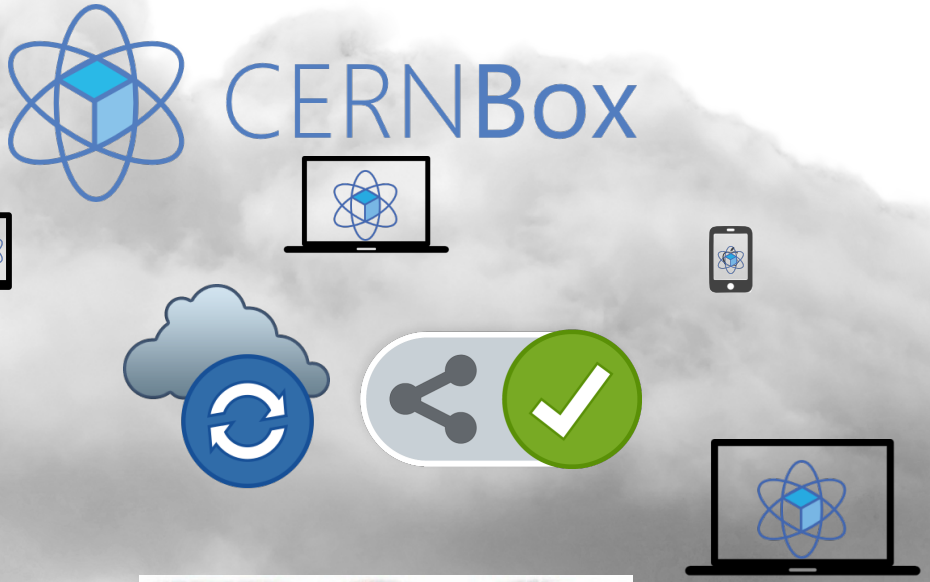
kuba:rw

ACLs

owner, target,…

sharing μ service

iOS



/home

/eos/cms

/eos/atlas

/eos/project

~1PB

~100 PB