File locking in CERNBox and EOS

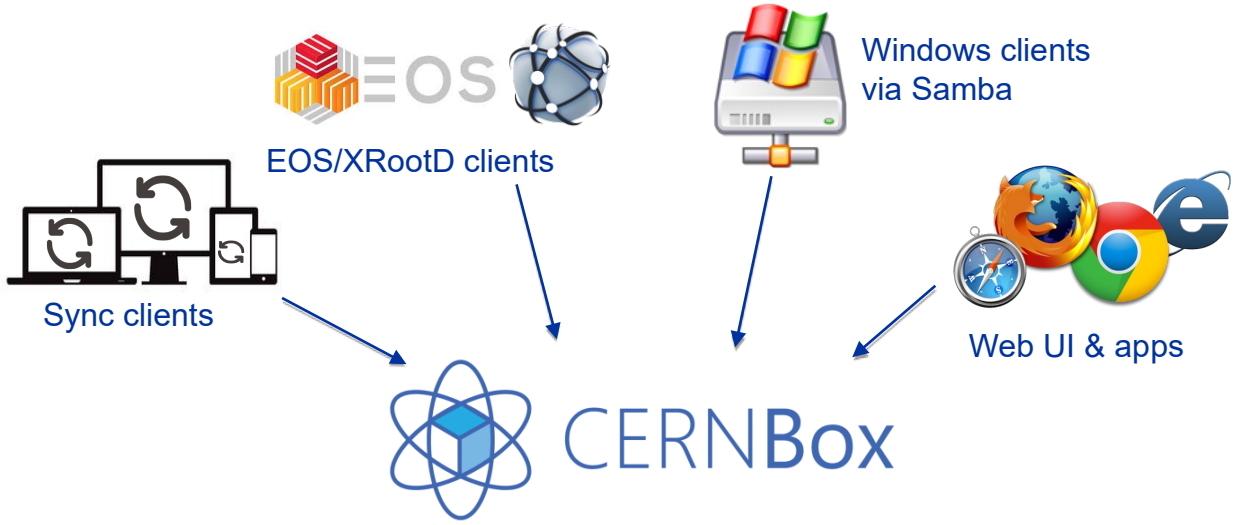
Giuseppe Lo Presti

CERN IT-ST

Last year: home folders at CERN

EOS/CERNBox proposed as “user home space” to all CERN users

- Introduced (advisory) locking via special lock files in order to support Office suites
- Typical case:
  Web vs Samba access



Now: new CERNBox features & integrations

- **CERNBox now supports single-file sharing in edit mode** – at last!
  - BUT lock files could not be read nor written as user has no access to the containing folder...
- **Need to go beyond web UI / online storage**
  - So far **sync clients** could always conflict with outstanding online / web clients
  - Long-standing request to support a check-out / check-in file (i.e. **lock**) feature
    - Must apply to Desktop sync clients as well

- **=> Consolidating lock support in the core of *Reva*, the CERNBox engine**
  - To be respected by all clients, including Desktop sync clients
  - Co-developed with ownCloud in the context of **CS3MESH4EOSC**

A(nother) new API

- Defined a *Lock API*, modeled after the **WOPI API**:
  `SetLock`, **`GetLock`**, **`RefreshLock`**, **`Unlock`**
  - Atomicity to be implemented by the storage providers

- Multiple lock types, modeled after WebDAV specs
  - Introduced *shared* locks, *write* locks (allow many readers) as well as *exclusive* locks
  - A lock includes an expiration time (typically 1h)
  - **Sync clients speak WebDAV and will be able to interact with locks**
    - Both to respect existing locks and to allow grabbing a lock for the user

Some implementation details (I)

- For EOS, we use **a dedicated user xattr to store the lock metadata**
  - **`user.iop.lock`**
  - Solves the constraints with permissions; external lock files could still be `stat`'ed
  - Just need to make setting xattrs atomic, similarly to the `O_EXCL` POSIX flag

    ✅ Available in recent EOS versions

- The payload is a base64-encoded JSON with the following structure:

```
{
    "lock_id": "<opaque_payload_from_app>",
    "type": LOCK_TYPE_WRITE,
    "user": { ... },
    "app_name": "Collabora Online",
    "expiration": { "seconds": 1665446400 }
}
```

Some implementation details (II)

- The ***Reva*** daemon features multiple storage drivers
  - The EOS implementation is included in a specific driver, maintained by the CERNBox team
  - The interaction to EOS happens via the `eos` CLI, and it will be moved to GRPC

- Goal: ensure all relevant clients adhere to those new locks
  - The WOPI server was made fully compliant
  - The CERNBox WebDAV interface is in the making
  - The Web UI is yet to be extended, to inspect locks and offer a check-out/check-in feature

...And some caveats with future work

- Not all race conditions are covered
  - In some corner cases the logic includes non-atomic set-after-check operations
  - Not a blocker, previous lockfiles-based implementation had similar issues
  - Work in progress – eventually building on top of Atomics support (cf. Abhi's presentation)

- File locked via FUSE are not yet taken into account!
  - The plan is to have EOS respect a "CERNBox lock" if it exists
    - A `flock` request would fail
  - Conversely, EOS would fail setting this special xattr if a `flock` exists on the file
    - The CERNBox layer would report the file "locked by another application or user"

Outlook

- A complete integration of Microsoft Office 365 in CERNBox is coming up
  - Such extensions further enhance Office support for our users

- The ongoing developments in the CS3MESH project will benefit from this feature
  - EOS playing the role of an advanced storage back-end
  - Further scenarios are being envisaged, e.g. with Jupyterlab notebooks

Thank you for your attention!
Time for questions

CS3MESH4EOSC has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no. 863353