

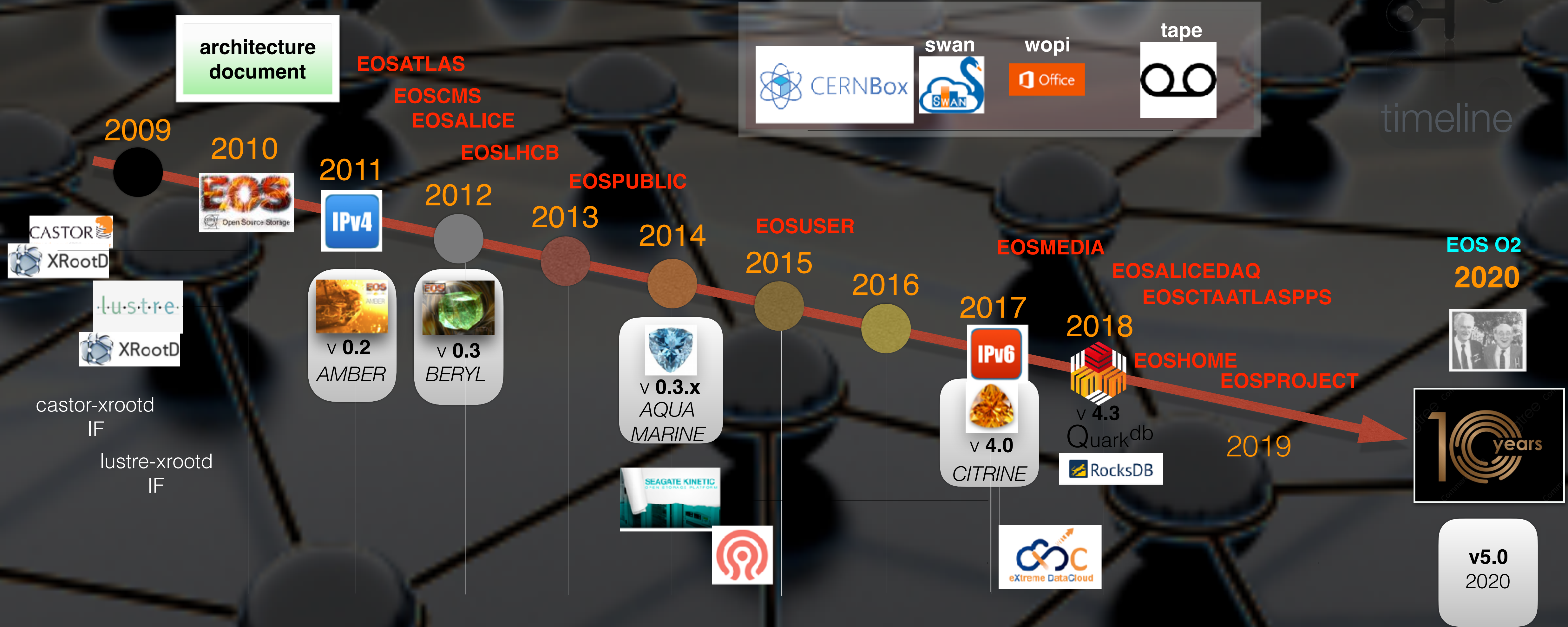
Appraisal



Andreas-Joachim Peters
CERN IT Storage Group

Project History

timeline



some decisions shaped our fate

the in-memory namespace

- started really cool because it was very fast compared to everything around, but soon became a curse the bigger instances grew at CERN - who likes to wait more than one hour to get back to work ... not starting the mutex discussion ...
- an in-memory cache with a RDMS backend could have been a simpler choice, ... and lock-free ... but then we wouldn't have QuarkDB now

a

XRootD as a framework

- XRootD is/was an extremely stable platform, but when you have problems like the one mentioned before, people quickly changed their opinion whenever a new bug appeared on client or server side ...
- it was the best fitting data transfer protocol for WAN & LAN back then and probably still is - WLCG targets now HTTPS, but most of the arguments why fell flat by now...
- and we had the most responsive support you can imagine for a project



some decisions:

the FUSE client

- XRootD had already a FUSE client when we started, so it was natural to try to evolve that one ...
- but someone in our group always warned: **if you give people once this, they will never let it go and it sets high expectations** - he was right, it is hard to meet the expectation of something which nowadays works always and can never fail: your filesystem!
- after re-writing FUSE already twice, a possible summary is: it will always be a compromise to squeeze a completely orthogonal implementation of a storage system into the world of a POSIX filesystem API - *it is much easier to write a distributed POSIX filesystem with the fundamental concepts of a POSIX filesystem when you start ...*
- the question is, if this compromise is good enough for users?



some really good experiences:

Collaboration

- we have built some really fruitful relationships over the time with friends from AARNET, JRC, LHC and other CERN experiments and their connected institutes - thank you all! Every workshop and the participants have helped the project to evolve!

a

CERN ST Team

- we are/were not a huge development/operations team at CERN - everybody in this team works/worked with a lot of dedication, every time it was needed - during & out of office hours - thanks to all of you! Moreover we also received also a lot of support and trust from our management!

The HIGGS discovery

- we certainly had a part in that - sometimes *because of EOS*, sometimes *despite* 😊

The number of EOS instances ...

- we are supposed to report every week in the s.c. C5 report, how many sites run EOS somewhere, but since we disabled the **spy** functionality (VST network) by default, it is difficult to know that precisely ... we probably should make a survey 🤪

Stability

- stability was always depending on what people were doing - things which are not used/developed are often very stable, EOS is/was always used and was under active development. With non-FS clients client stability was never an issue, with FS clients like FUSE it became one. In general there were great improvements in 2019!

Scalability

- we never have seen problems to scale the data part - we were/are running with 12k disks in a single instance, with QuarkDB the MD scalability is given by money for SSD space - *still the MD operation rate is not really easy scalable ...*

And now ... what about the future ... what is important now?

- over the last ten years a lot of technologies have evolved there are
 - new technologies & frameworks
 - new protocols
 - new storage systems
 - new filesystems
- for any long running project it is important to revise the substance it is made of, the architecture it is based on, the ideas & requirements who have led to its existence
- it is evident, there is a need for what EOS is providing - we have to make sure, that we use best available technologies and fill open gaps
- re-focus, maybe sometimes do less but better
- after years have courage to drop functionality, replace or on-board technology



A good starting point for the time to come ...

What is the reason, that EOS is still an evolving project? Couldn't we have finished by now?

How would we implement/provide EOS today, if we start from scratch ?

Which frameworks, protocols & technologies do we have today available?

What do we need today and tomorrow?



“YOUR FUTURE
IS WHATEVER
YOU MAKE IT,
SO MAKE IT A
GOOD ONE.”



EMMETT “DOC” BROWN