Preparing EOS for Enterprise Users

Example: **EOS proposed for CareHD and vINCI**

Gregor Molan

Comtrade Digital Services

COMTRADE

Background: CERN EOS + Comtrade

Background

- EOS productisation project
  - http://openlab.cern/project/eos-productisation
  - Since 2015
- Coordinators
  - CERN: Luca Mascetti
  - Comtrade: Gregor Molan
- Goal:
  - EOS productisation

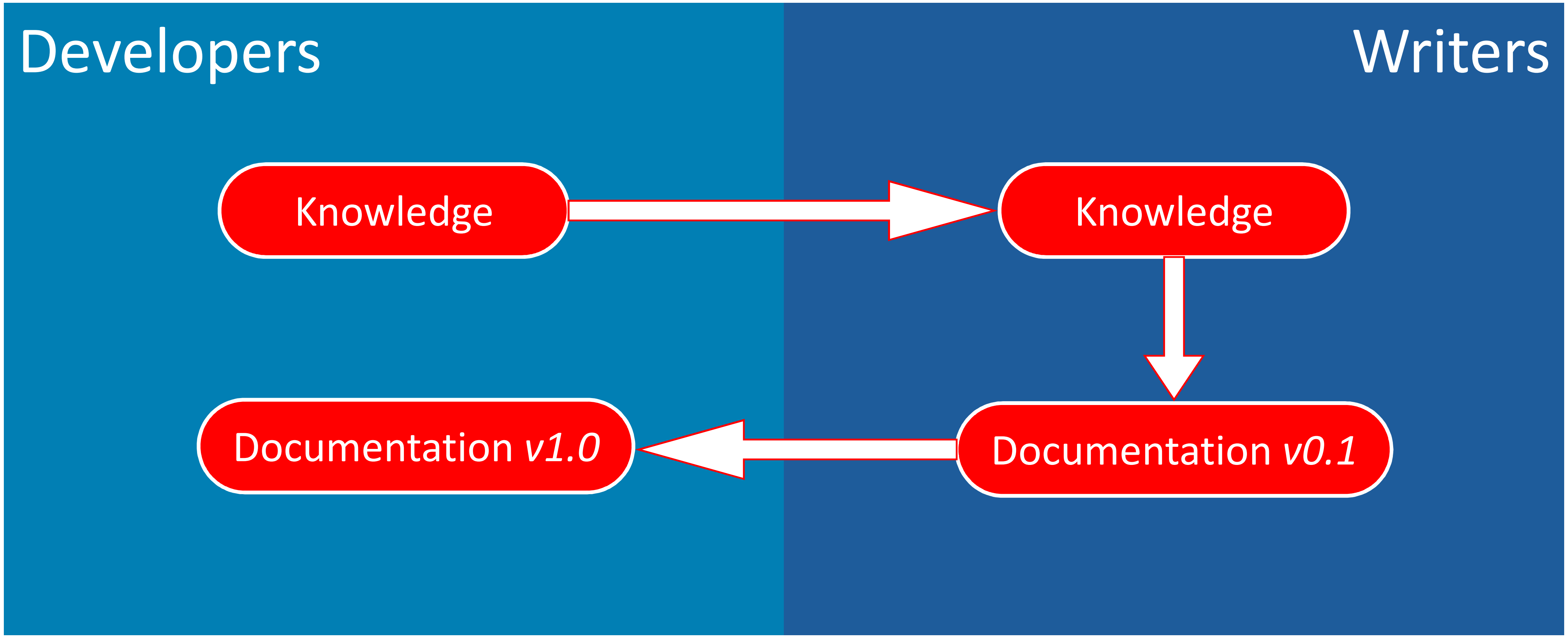
CERN EOS: Platform for IoT Data

CareHD	vINCI
• H2020 MSCA-RISE project • Patient-centred Connected Health • Target: • People living with Huntington's Disease (HD)	• H2020 AAL project • Clinically-validated Integrated Support for Assistive Care and Lifestyle Improvement • Target: older adult

Enterprise Software Development

- Architecture process
- Development process
- **Documentation**
- Support
- Marketing
- Sales

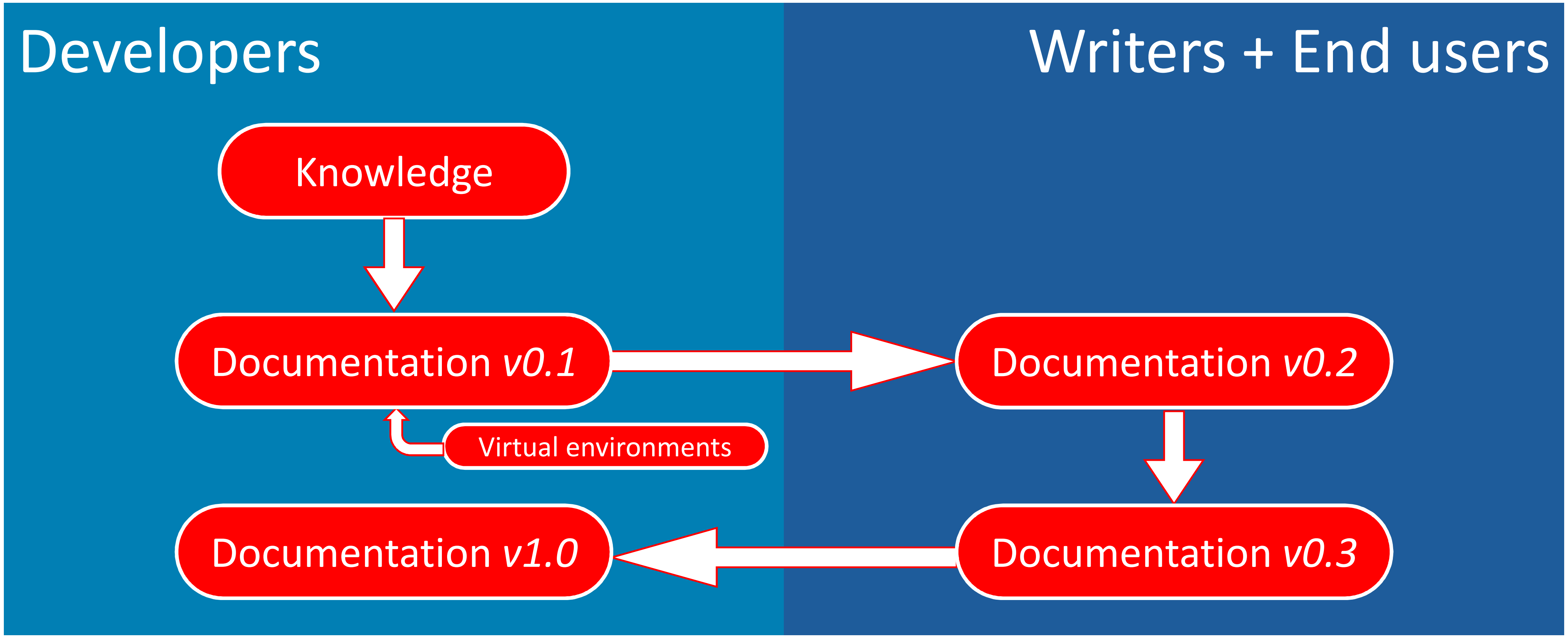
Classical Enterprise Documentation



Documentation Development Steps

- Developers: Knowledge about EOS
  - Existing workflow
  - Known functionalities
- Writers: Documentation verification
  - Live EOS storage system at CERN
  - Feedback from end users
- EOS virtual environments
  - Exercise functionalities (Geo-scheduling, Balancing, Upgrades)

CERN EOS Documentation



Summary

- Never underestimate documentation need
- Different quality of documentation
- Documentation is the first contact with customers

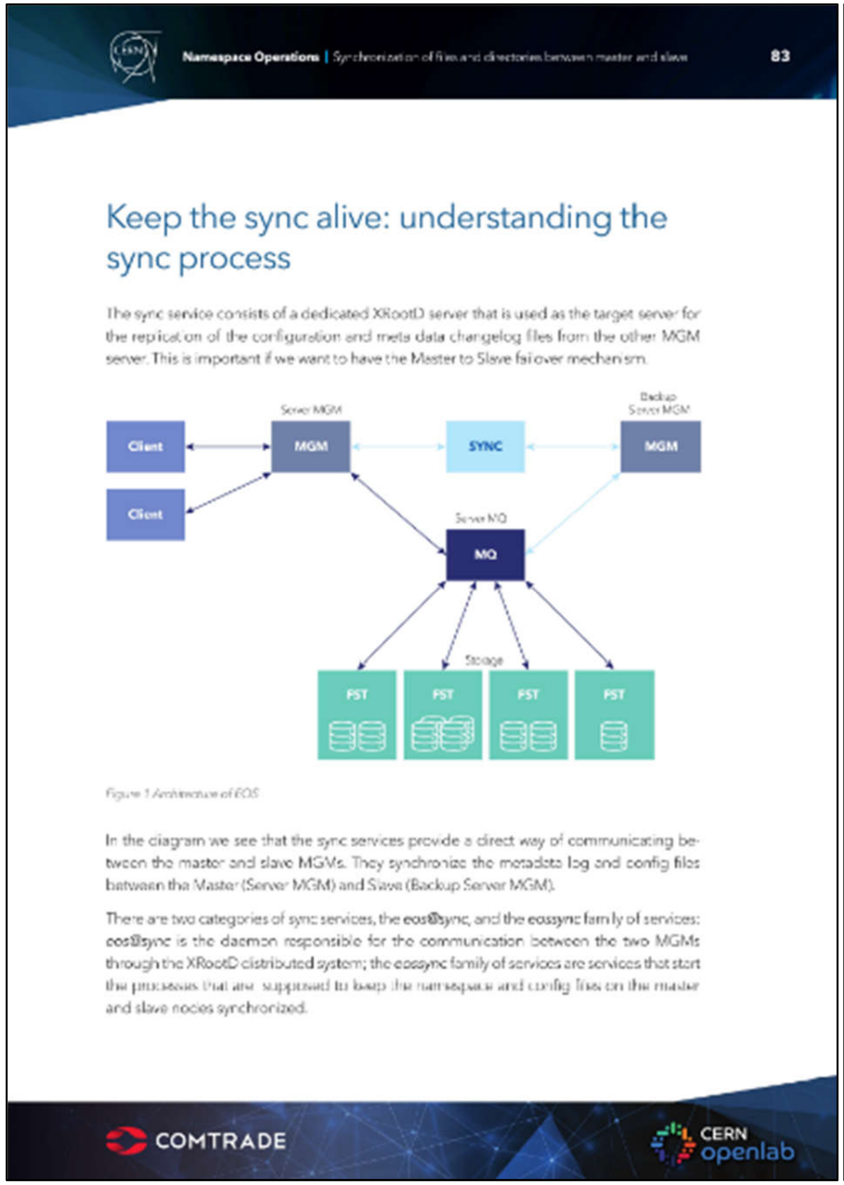
- Enterprise quality documentation
  - CERN EOS productisation project (more than 1200 pages)
  - Test run with CareHD project: www.carehd.eu
  - Test run with vINCI project: https://vinci.ici.ro

CERN COMTRADE

EOS

Namespace Operations

CERN openlab



Keep the sync alive: understanding the sync process

The sync service consists of a dedicated XRootD server that is used as the target server for the replication of the configuration and meta data changelog files from the other MGM server. This is important if we want to have the Master to Slave failover mechanism.

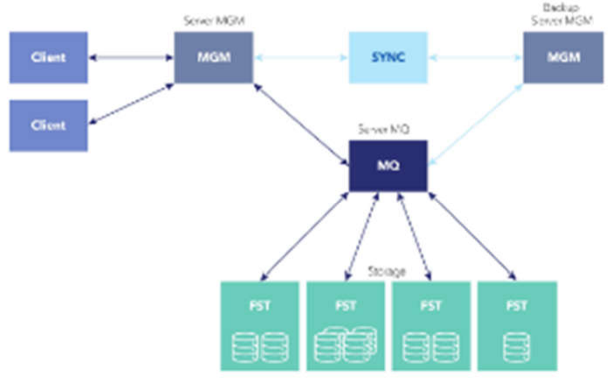
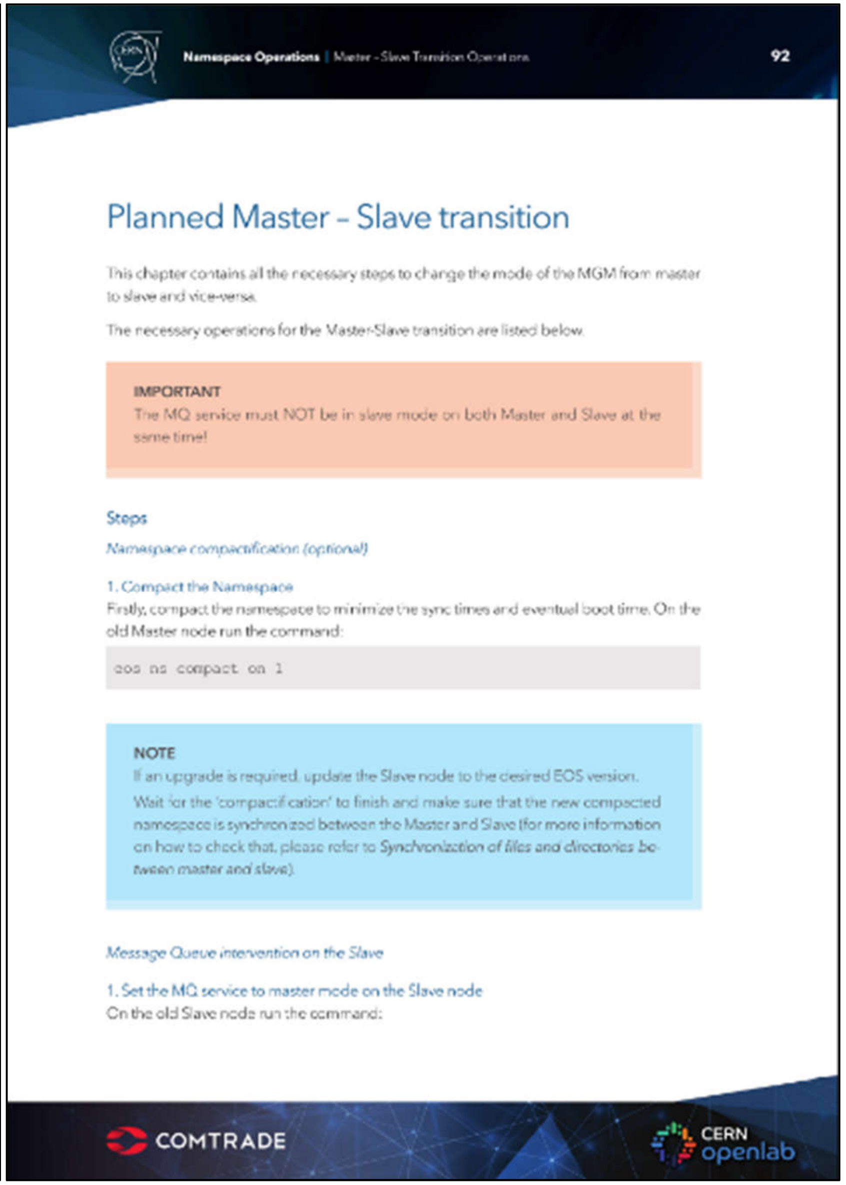


Figure 1 Architecture of EOS

In the diagram we see that the sync services provide a direct way of communicating between the master and slave MGMs. They synchronize the metadata log and config files between the Master (Server MGM) and Slave (Backup Server MGM).

There are two categories of sync services, the **eos@sync**, and the **eossync** family of services: **eos@sync** is the daemon responsible for the communication between the two MGMs through the XRootD distributed system; the *eossync* family of services are services that start the processes that are supposed to keep the namespace and config files on the master and slave nodes synchronized.

COMTRADE CERN openlab



Planned Master – Slave transition

This chapter contains all the necessary steps to change the mode of the MGM from master to slave and vice-versa.

The necessary operations for the Master-Slave transition are listed below.

IMPORTANT
The MQ service must NOT be in slave mode on both Master and Slave at the same time!

Steps

Namespace compactification (optional)

1. Compact the Namespace
Firstly, compact the namespace to minimize the sync times and eventual boot time. On the old Master node run the command:

```
eos ns compact on 1
```

NOTE
If an upgrade is required, update the Slave node to the desired EOS version.
Wait for the 'compactification' to finish and make sure that the new compacted namespace is synchronized between the Master and Slave (for more information on how to check that, please refer to *Synchronization of files and directories between master and slave*).

Message Queue intervention on the Slave

1. Set the MQ service to master mode on the Slave node
On the old Slave node run the command:

COMTRADE CERN openlab

Author

- Gregor Molan (Comtrade, Slovenia) <Gregor@comtrade.com>

COMTRADE

Thank you!