My CoRe: ownCloud at CNRS

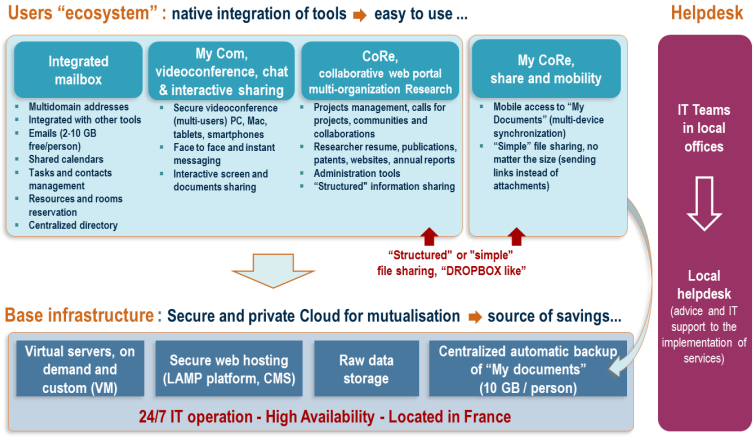
David Rousse
CNRS - DSI

ETH zürich

Content

Content

Global context



* *Laboratories, administrative units and functional departments can benefit from these services*

More information (in French) at `http://ods.cnrs.fr`

Why My CoRe?

Two different business needs[identified mainly through end users polls]

- Synchronization and sharing service to provide a secure alternative to Dropbox for CNRS users
- Backup service for CNRS users' documents

A unique choice

- ownCloud because it had the required functionality, it was already used in some local CNRS units with a good end users feedback and it was (and still is) open source (so easy to customize)
- Service deployed in CNRS' IN2P3 Computing Center, in order to keep the data safe and to be able to use the existing network bandwith[10GB/s link] and also the local backup and restore service[based on Tivoli]
- Service operated by an IT service provider, to be able to provide it as a 24-7 service

Service summary

Status:	Production(deployment steps in appendix 4)
Number of users (current/target):	4500(at 2016/01/11) / 15.000(for end 2016)
Default quota:	20GB
Linux/Mac/Win user ratio:	20/40/40(more metrics in appendix 3)
Desktop/mobile/web clients access ratio:	Unknown yet(but other metrics in appendix 3)
Technology:	ownCloud(web) with MariaDB-Galera(DB) and Scality(storage)
Target communities:	CNRS members
Integration in current environment:	None(except our existing Shibboleth SSO backend)
Risk factors:	Load on DB
Most important functionality:	Files synchronization and sharing, central files backup
Missing functionality:	Versioning, accounts for external users, sharing files temporary and then delete theses files (once downloaded)

Content

Functional choices (1/2)

ownCloud

- ownCloud core, community edition (version 7)
- Antivirus app
- Without Versions app (too much load generated on DB)

And some apps developed/forked by CNRS

- "Dashboard" app for metrics on service usage (current metrics in appendix 3)
- "Lotsofgroups" app for managing a lot of groups
- "Group custom" app for end users group management
- "Password Policy" app for password policy enforcement
- "GTU" app for GTU online agreement

Theses developments have been made on the ownCloud server part, nothing has been yet developed on the clients side
Source code is available on Github at `https://github.com/CNRS-DSI-Dev/`

Functional choices (2/2)

And some apps developed/forked by CNRS (continued)

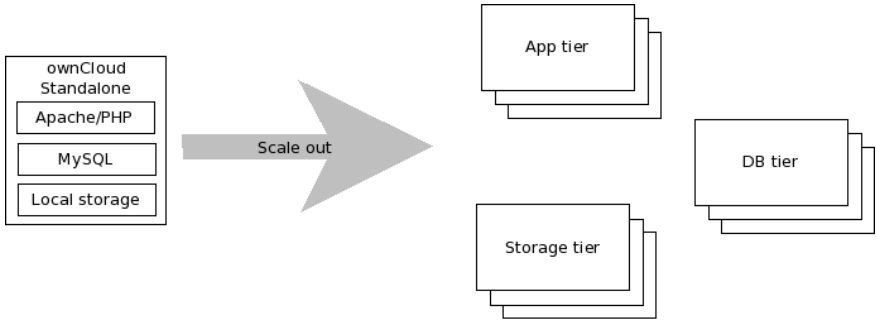
- "Gatekeeper" app for filtering access depending on end user groups
- "User Servervars 2" app for end users authentificate and account provisionning (in relation with Shibboleth SSO backend)
- "User account actions" developed for performing actions when users are created/deleted
- "User files migrate" app for transfering files between two accounts of a unique user
- "User files restore" app and shell scripts for managing backup and restore of end users files [a]
- Shell script "Mycore sympa" to integrate ownCloud users list to an SYMPA mailing list manager
- A specific theme
- Various housekeeping shell scripts

Theses developments have been made on the ownCloud server part, nothing has been yet developed on the clients side
Source code is available on Github at `https://github.com/CNRS-DSI-Dev/`

a. *See detail on appendix 2 of this document*

Content

Initial key points



Two key points

- Is ownCloud scalable and if so how (how much web, DB nodes do we need, ...)?
- Which components to choose in order to implement the service?

Load tests and sizing

Method

- Functional hypothesis on the service usage
- Deployment of a dedicated infrastructure for theses load tests studies
- Technical hypothesis on the web, DB and storage behaviours

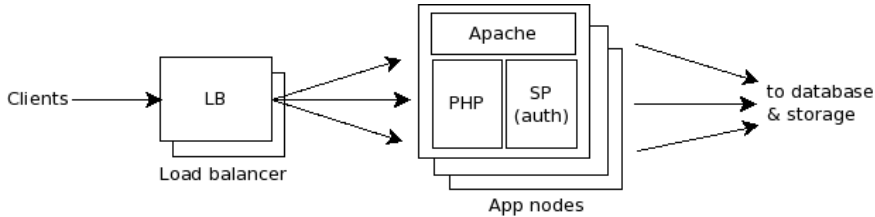
See in detail at https://github.com/CNRS-DSI-Dev/mycore_press/raw/master/CERN-CNRS-meeting-20141117.pdf, appendix 1

Results

- "Go" to deploy such a service, with an accurate visibility on the architecture evolution
  (number of nodes, number of servers, ... depending on number of users, number of devices per user, ...)
- Main concern: cost on the long term for such a service
- Functional hypothesis can have a huge impact on sizing

See in detail at https://github.com/CNRS-DSI-Dev/mycore_press/raw/master/CERN-CNRS-meeting-20141117.pdf, appendix 2

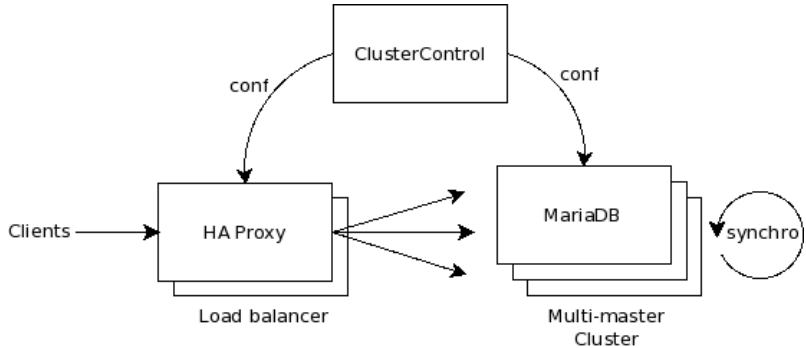
App tier choice



ownCloud, PHP, Shibboleth's SP and Apache choices

ownCloud for business reasons, Apache because of internal teams knowledge, Shibboleth's SP on the web nodes in order to have the same web nodes

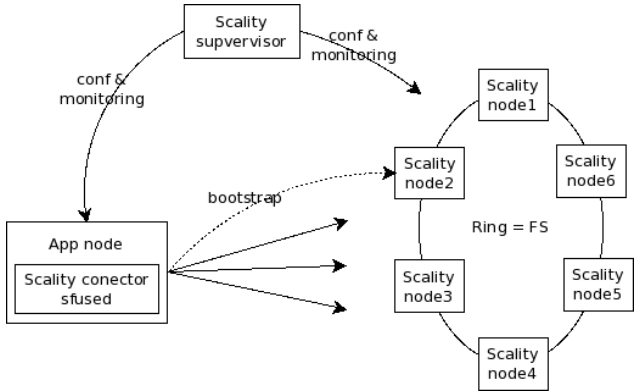
DB tier choice



MariaDB-Galera cluster choice

open source[except ClusterControl tool, detail at http://severalnines.com/product/clustercontrol], 32 nodes supported, already known internaly and easy to install and administrate through ClusterControl tool

Storage tier choice



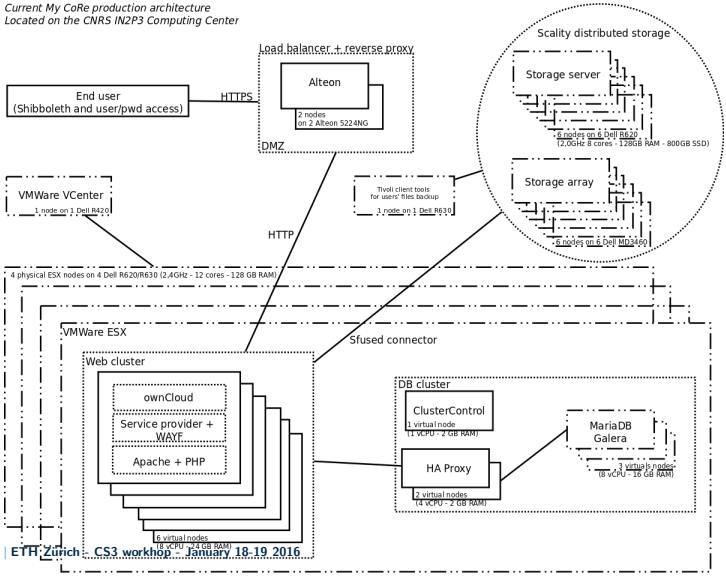
Scality software defined storage choice

Reliable file system access, hardware agnostic solution and scalable by design

See in detail at https://github.com/CNRS-DSI-Dev/mycore_press/raw/master/CERN-CNRS-meeting-20141117.pdf, appendix 3

Current architecture

Current My CoRe production architecture
Located on the CNRS IN2P3 Computing Center



Content

Positive points ... and negative ones

Positive points

- Useful project, so real motivation to make it a success
- Very positive end user feedbacks
- Meaningful exchanges between project team members and with other partners (especially CNRS' IN2P3 Computing Center)

Negative points

- Hard to conciliate the needs for a "Dropbox like service" and for a pure files backup and restore service
- An agile process instead of a classical V-model approch would have been a better choice to reduce the project length(length mainly due to difficulties encountered during implementation)
- Deploying a dedicated backend (web, DB, storage) just for My CoRe was time consuming : re-using an IaaS platform and building the My CoRe service on top of that would have been more appropriate

Next steps (1/2)

Functional roadmap

- App for setting local password just after the first login [a]
- ownCloud 8.2 migration
- "Share manager" app for an easy management of the files shares [b]
- Guest accounts [c]
- App Versions activated
- Enhanced end users administation backend
- Users UID migration, because of new CNRS referential "Réséda"
- Specific mobile and PC clients [d]

---

a. See detail at `https://github.com/CNRS-DSI-Dev/user_set_password`
b. See detail at `https://github.com/LydSC/sharewatcher`
c. Accounts for non-CNRS members, with a 0GB quota, in order to ease collaborations between CNRS users and external CNRS users
d. For instance to have a unique access through our Shibboleth SSO, whatever the client is

Next steps (2/2)

Technical roadmap

- Architecture evolution
  - basic option with more web and/or DB nodes, see detail at = `https://github.com/CNRS-DSI-Dev/mycore_press/raw/master/architecture/schema_2016.pdf`
  - alternative option with DB migration from virtual to physical nodes = `https://github.com/CNRS-DSI-Dev/mycore_press/raw/master/architecture/schema_2016_option.pdf`
- Access storage backend through an object interface instead of a file system access
- Try to investigate the "Server to server sharing" ownCloud function
- End-to-end testing tool, to test the ownCloud core functionalities [a]
- File encryption (on server side)

a. For instance with the CERN' smashbox tool

Questions?

Projet team (previous and current members)

David Bercot (IT department manager) Jonathan Bouchiquet (end users support), Marc Dexet (developer), Philippe Dubrulle (communication), Gilian Gambini (system administrator), Eric Gervasoni (end users committee manager), Jérôme Jacques (system administrator), Nadine Marouzé (previous IT department manager), Paulo Mora de Freitas (end users committee manager), Olivier Lenormand (IT coordinator), Jean-Yves Lopez (previous IT department manager), Patrick Paysant (developer), David Rousse (previous project leader), Lyderic Saint-Criq (developer) et Alexandre Salvat (project leader(since 2015/11))

Partners involved in the project

DSI' CNRS IT department, DSI' CNRS graphist, CNRS' IN2P3 Computing Center and the following external partners Atos, Dell, Linagora, Scality and SeveralNines

Content

URLs in relation with My CoRe

- Previous My CoRe presentation during 1st CERN workshop = https://github.com/CNRS-DSI-Dev/mycore_press/raw/master/CERN-CNRS-meeting-20141117.pdf
- My CoRe presentation at JRES conference on Dec 2015 (in French) = https://github.com/CNRS-DSI-Dev/mycore_press/tree/master/JRES2015
- ownCloud load test in detail = https://github.com/CNRS-DSI-Dev/mycore_press/blob/master/CERN-CNRS-meeting-20140513.pdf
- Comparison between theoretical load tests and real load after beta period service = https://github.com/CNRS-DSI-Dev/mycore_press/raw/master/myCore_comparison_estimate-load_real-load_on_ownCloud.pdf
- JoSy conference (in French), Strasbourg 2014 May = https://github.com/CNRS-DSI-Dev/mycore_press/blob/master/CNRS-JoSy-20140519.pdf
- Scality in detail, press made for the CNES (in French) = https://github.com/CNRS-DSI-Dev/mycore_press/blob/master/CNES-CNRS-Scality-20140619.pdf
- My CoRe, how is it built ? = https://github.com/CNRS-DSI-Dev/mycore_build
- Other My CoRe press to done/to come = https://github.com/CNRS-DSI-Dev/mycore_press

Appendix 2: details on backup/restore function (1/2)

End users files backup in My CoRe service

- The backup is made through a shell script that interfaces My CoRe infrastructure and Tivoli backup service = `https://github.com/CNRS-DSI-Dev/mycore_backup_restore_user_files/blob/master/backup_mycore.sh`
- The backup is scheduled in background, without end users interaction: files are backuped in background and stored by the CNRS' IN2P3 Computing Center Tivoli service

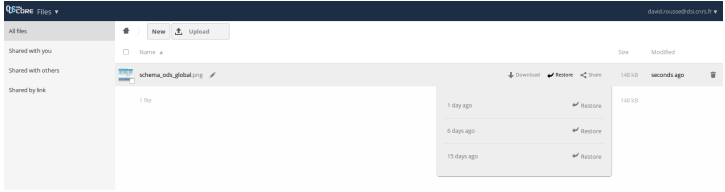
End users files restore in My CoRe service

- The restore is also made through a shell script that interfaces My CoRe infrastructure and Tivoli backup service = `https://github.com/CNRS-DSI-Dev/mycore_backup_restore_user_files/blob/master/mycore_restore.sh`
- An ownCloud app allows end users to ask for files restore = `https://github.com/CNRS-DSI-Dev/user_files_restore`

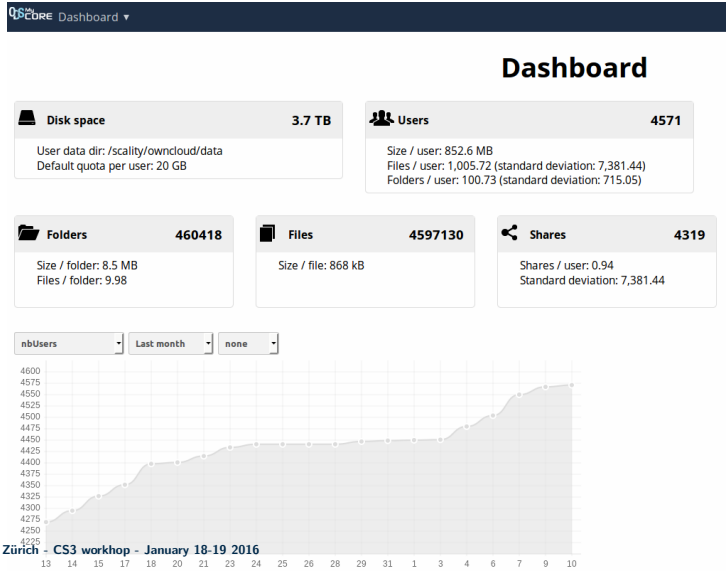
Global principle of a restore

- An end user can ask for a file/folder restore, at day-1, day-6 and day-15. Once restored, the restored file/folder overwrites the existing one (if exists)
- Restore requests are not processed in real time: instead, each request is written in a DB table and a scheduled shell script reads this table and execute the restore in background
- A user can cancel a pending restore request

Appendix 2: detail on backup/restore function (2/2)



Appendix 3: detail on service usage (screenshot of Dashboard app on 2016/01/11)



Appendix 4: My CoRe schedule

Schedule and deployment steps

- January to September 2013: market survey
- October 2013 to May 2014: ownCloud functional and technical evaluation (in collaboration with Linagora)
- May to December 2014: implementation (in collaboration with Atos)
- January to September 2015: beta service for 2.000 end users (5GB per user)
- Since October 2015: service opened to all CNRS laboratories (20GB per user)