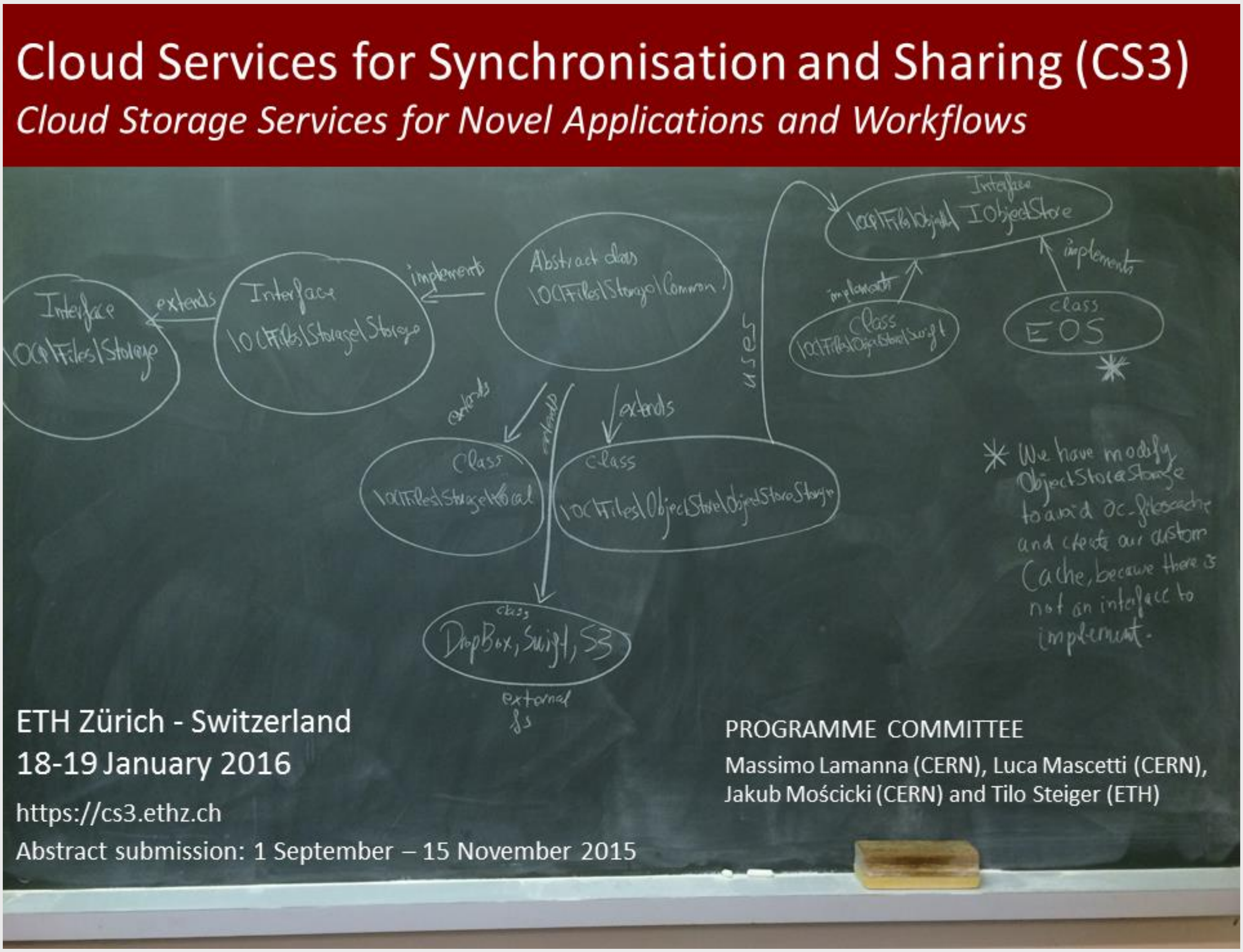
Cloud Services for Synchronisation and Sharing

Book of Abstracts

18-19 January 2016

ETH Zürich



Cloud Services for Synchronization and Sharing

18-19 January 2016

ETH Zürich, Switzerland

Programme Committee

Massimo Lamanna (CERN)

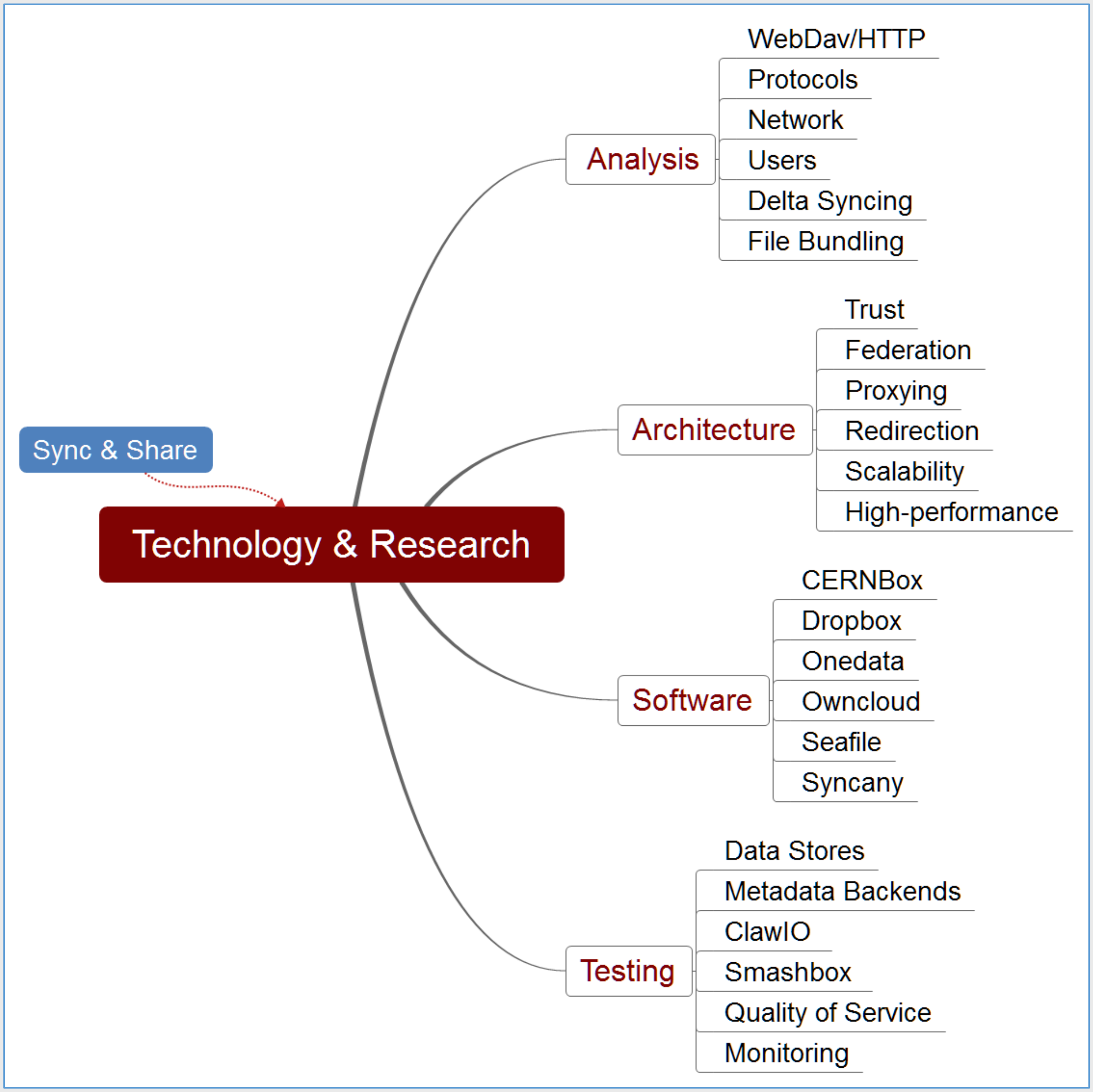
Luca Mascetti (CERN)

Jakub T. Mościcki (CERN)

Tilo Steiger (ETH)

Editor of the Book of Abstracts

Belinda Chan (CERN)


Sync & Share
Technology & Research
Analysis
WebDav/HTTP
Protocols
Network
Users
Delta Syncing
File Bundling
Architecture
Trust
Federation
Proxying
Redirection
Scalability
High-performance
Software
CERNBox
Dropbox
Onedata
Owncloud
Seafile
Syncany
Testing
Data Stores
Metadata Backends
ClawIO
Smashbox
Quality of Service
Monitoring

Technology and Research

Experiences of Cloud Storage Service Monitoring: Performance Assessment and Comparison, E. Bocchi (Politecnico Torino)

Onedata: High-performance Globally Distributed Data Access Platform, M. Orzechowski (AGH Krakow)

Syncany: Concepts for trustless abstracted cloud storage, P. Otte (Delft)

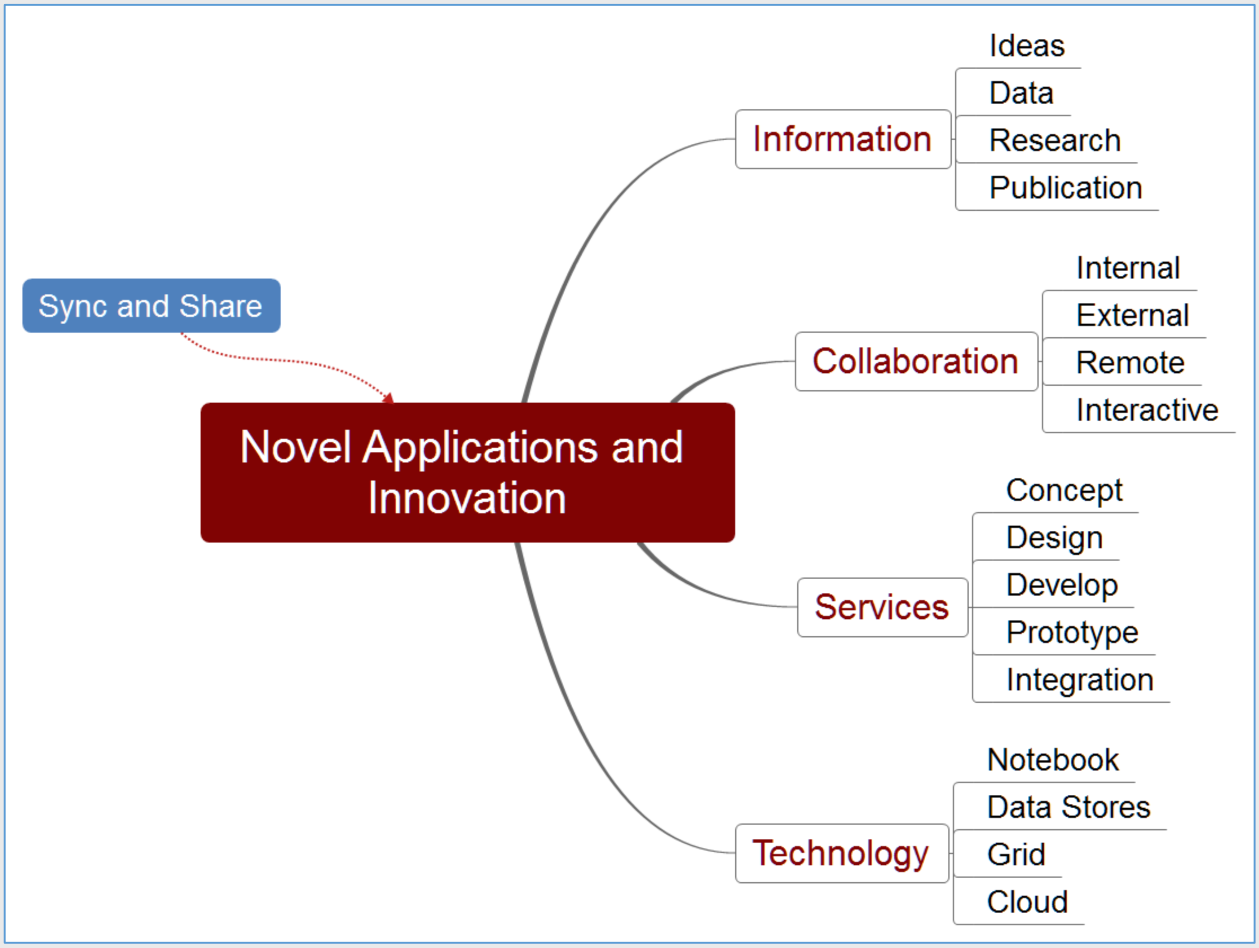
The next step in federated file sync and share, F. Karlitschek (ownCloud)

Benchmarking and testing ownCloud, Seafile, Dropbox and CERNBox using smashbox, P. Mrowczynski (Technical University of Denmark Copenhagen)

Scaling ownCloud – by user redirection and proxying, F. Orellana (Technical University of Denmark Copenhagen)

Testing storage and metadata backends with ClawIO, H. Labrador (Vigo University)

A study of delta-sync and other optimisation in HTTP/Webdav synchronisation protocols, W. Jarosz (AGH Krakow and CERN)


Sync and Share
Novel Applications and Innovation
Information
Ideas
Data
Research
Publication
Collaboration
Internal
External
Remote
Interactive
Services
Concept
Design
Develop
Prototype
Integration
Technology
Notebook
Data Stores
Grid
Cloud

Novel Applications and Innovation

Data Mining as a service: Innovative user applications and workflows with cloud storage services, D. Piparo (CERN)

Towards an infrastructure for interactive Earth Observation data analysis and processing, A. Burger (European Commission - Joint Research Centre Ispra)

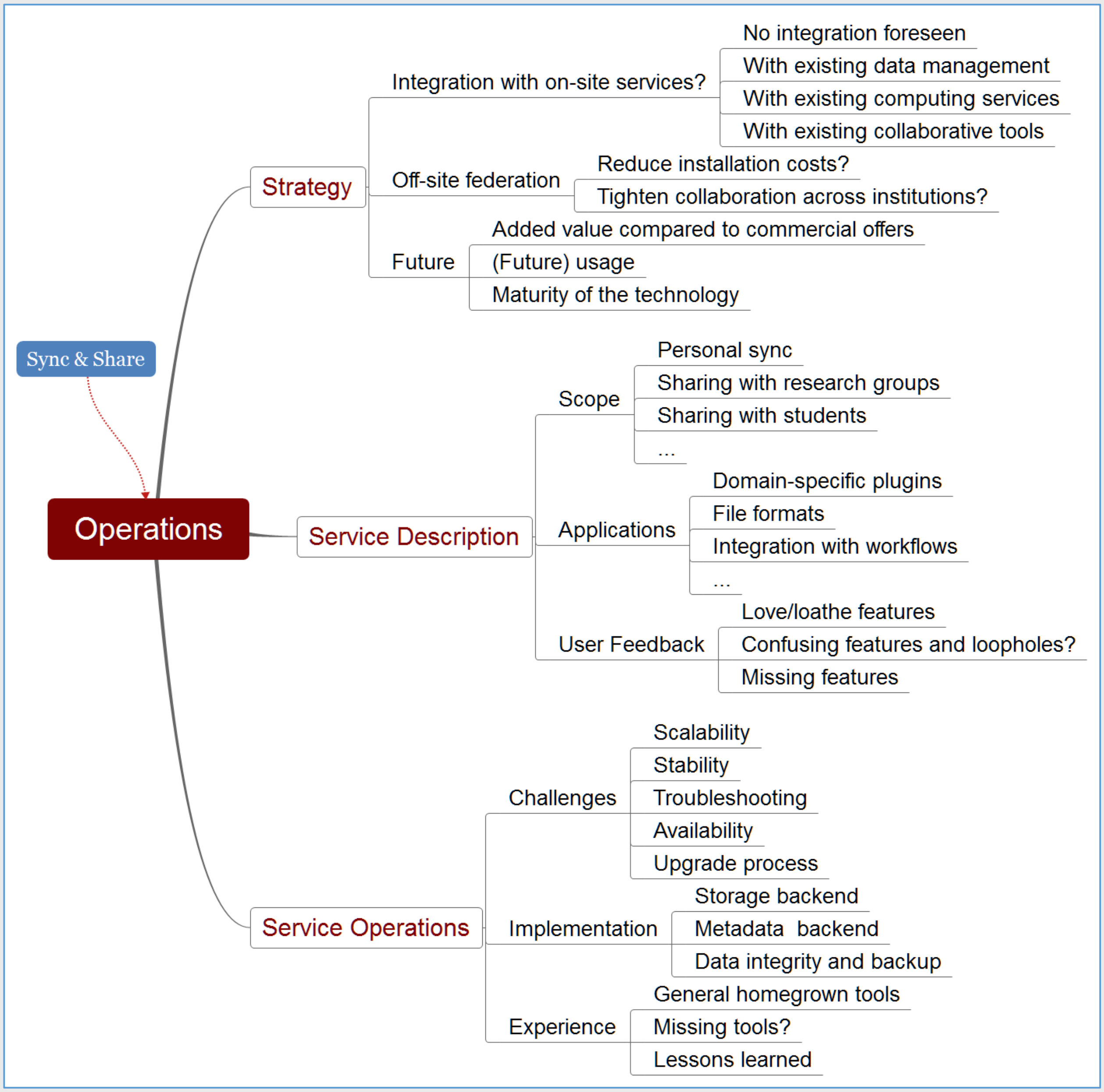
User friendly access to Grid and Cloud resources for scientific computing, A. Richards (Imperial College London)

Experience in integrating end-user cloud storage for CMS Analysis, H. Riahi (CERN)

Authorization in GARRbox: Delegated resource management and user data isolation, F. Farina (GARR, Italy)

Zenodo: enabling scientists, projects and institutions to share, preserve and showcase multidisciplinary research results, T. Smith (CERN)

Publishing, sharing and preserving your data: from synchronization to curation, F. Orellana (Technical University of Denmark Copenhagen)


Sync & Share
Operations
Strategy
Integration with on-site services?
No integration foreseen
With existing data management
With existing computing services
With existing collaborative tools
Off-site federation
Reduce installation costs?
Tighten collaboration across institutions?
Future
Added value compared to commercial offers
(Future) usage
Maturity of the technology
Service Description
Scope
Personal sync
Sharing with research groups
Sharing with students
...
Applications
Domain-specific plugins
File formats
Integration with workflows
...
User Feedback
Love/loathe features
Confusing features and loopholes?
Missing features
Service Operations
Challenges
Scalability
Stability
Troubleshooting
Availability
Upgrade process
Implementation
Storage backend
Metadata backend
Data integrity and backup
Experience
General homegrown tools
Missing tools?
Lessons learned

Operations

User Management and Sharing in Sciebo, the Academic Cloud Storage Service in North Rhine-Westphalia, H. Angenent et al. (University of Münster)

INFNBox: a distributed and high reliable cloud infrastructure for data sharing at INFN, G. Donvito et al. (INFN Bari)

First experience with box.psnc.pl based on Seafile, M. Brzezniak (PSN Poznan)

CERNBox: Cloud Storage for Science, L. Mascetti (CERN)

Distributed Sync&Share Operations at AARNet, G. Aben (AARNET)

u:cloud - File Sync and Share in a heterogeneous scientific environment at the University of Vienna, C. Kracher et al. (Technische Universität Wien)

MyCoRe - ownCloud service at CNRS, D. Rousse (CNRS France)

Ad hoc Federated Storage with OpenStack Swift, J. Topjian (Cybera - Calgary)

Experience with running Owncloud on virtualized infrastructure (Openstack/Ceph), C. Schnidrig (SWITCH Zürich)

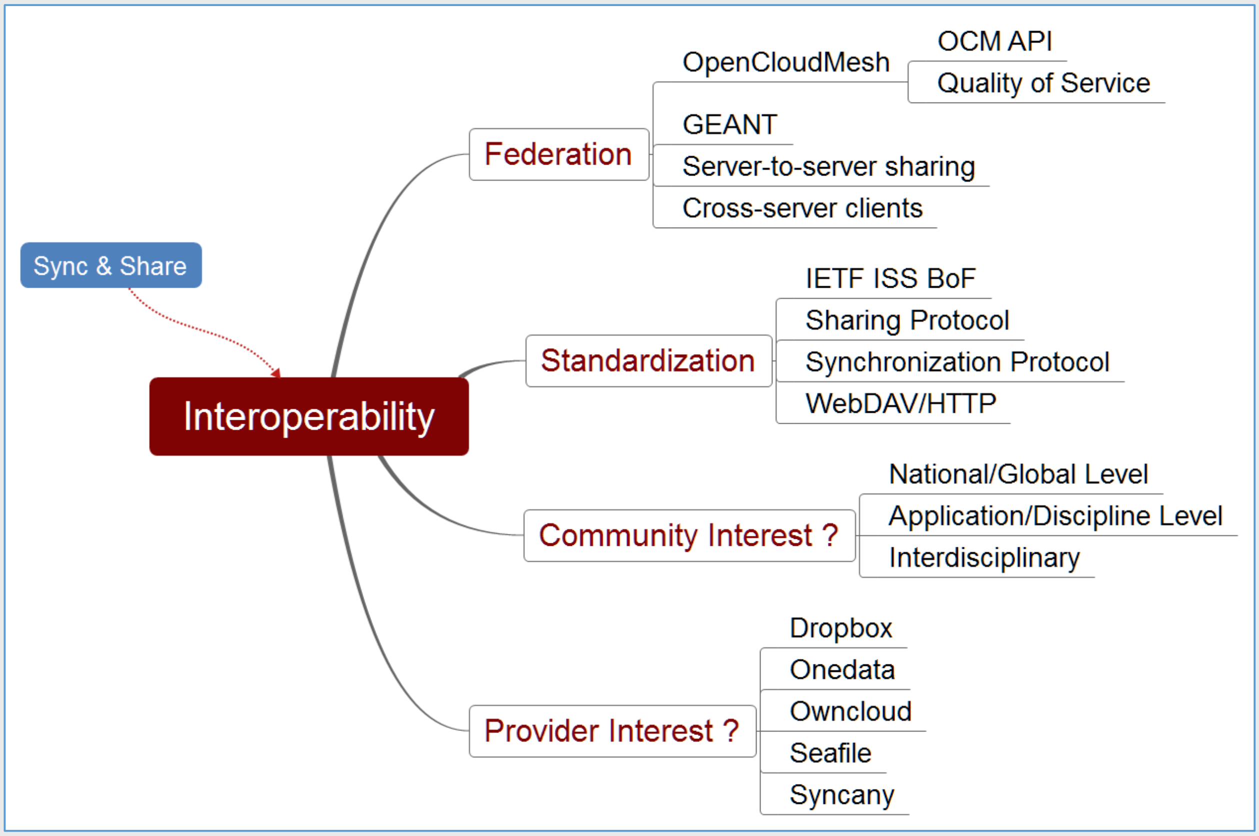
UnipiBOX: owncloud@unipi, S. Spinelli (Università di Pisa)

Two years of SURFdrive, lessons learned and future plans, R. Trompert (SURFSara Amsterdam)

Site Report LMU Faculty of Physics, K. Steinberger (Ludwig Maximilians Universität München)

DESY-Cloud: an ownCloud + dCache update, P. Millar et al. (DESY Hamburg)

Polybox: Sync and Share at ETH Zürich, T. Steiger (ETH Zürich)


Sync & Share
Interoperability
Federation
OpenCloudMesh
OCM API
Quality of Service
GEANT
Server-to-server sharing
Cross-server clients
Standardization
IETF ISS BoF
Sharing Protocol
Synchronization Protocol
WebDAV/HTTP
Community Interest ?
National/Global Level
Application/Discipline Level
Interdisciplinary
Provider Interest ?
Dropbox
Onedata
Owncloud
Seafile
Syncany

Interoperability

Open Cloud Mesh initiative, C. Schmitz (ownCloud), P. Szegedi (GEANT), J. Moscicki (CERN)

Sync & Share - Ripe for standardisation? S. Leinen (SWITCH Zürich)

Conference Keynote

Bitcoin: Synchronization and Sharing of Transactions

Prof. Dr. Roger P. Wattenhofer (ETH Zürich)

http://www.disco.ethz.ch/members/roger/bio.html

New cloud storage services allow [people] to share, transfer and synchronize data in simple but powerful ways." The description of the CS3 workshop sounds a bit like Bitcoin. In my talk, I will first give a short introduction to the Bitcoin system, explaining some of the basics such as transactions and the blockchain. Then, I will discuss some interesting technical aspects in more detail, regarding the stability, security, and scalability of Bitcoin.

Regarding stability, I will discuss Bitcoin's eventual consistency, and the related problem of double spending. Regarding security, I will shed some light into our findings regarding the bankruptcy of MtGox, previously the dominant Bitcoin exchange service.

Finally, if time, I will discuss scalability, and present duplex micropayment channels. Apart from scalability, these channels also guarantee end-to-end security and instant transfers, laying the foundation of a network of payment service providers.

Technology and Research

Experiences of Cloud Storage Service Monitoring: Performance Assessment and Comparison

E. Bocchi (Politecnico Torino)

Enrico Bocchi [1,2] , Idilio Drago[1] , Marco Mellia[1] 1 - Politecnico di Torino, Italy 2 - Telecom ParisTech, France {name.surname}@polito.it

Personal cloud storage services offer a large amount of space and the ease to synchronize devices with great simplicity. They help in sharing content and in backing up data by relying on the cloud to seamlessly store users' files. Despite the high public interest in such services, little information is available about design, implementation choices and, most of all, actual performance implications.

In this work, we design a methodology to run benchmarks of cloud storage services. We unveil how they are implemented, where their servers are located, and measure implication on performance as seen by end-users. By means of repeatable and customizable tests, we identify eventual advanced capabilities the cloud client implements, and measure performance implications. We consider realistic workloads (e.g., the exchange of text and binary files, compressed archives, the presence of file replicas, etc.) and network accesses (e.g., high speed university campus, or 3G mobile connectivity).

We use then the benchmarking methodology to compare 11 cloud services, including popular solutions like Dropbox or Google Drive, and two private storage solutions, i.e., the open source ownCloud and the commercial VMware Horizon, that we installed and configured in our campus network. We take the perspective of a customer located in Europe, and we benchmark each service.

Our case study reveals interesting differences in design choices. Results show no clear winner, with all services having potential for performance improvements. Some are limited by design choices, e.g., by artificially throttling upload and download speed, or by long application timers that slow down synchronization procedures. Others suffer TCP performance issues due to their data centers being located in other continents. In some scenarios, the synchronization of the same set of files can take 20 times longer. In other cases, we observe a wastage of twice as much network capacity, questioning the design of some services, especially in a bandwidth constrained scenario like 3G/4G connectivity.

Our results show the implications of design choices on performance, and of the tradeoffs faced when building cloud storage services. The developed methodology and the collected results are useful both as benchmarks and as guidelines for system design. In addition, they help the prospected customer in the choice of the best service by allowing the execution of independent performance tests before purchasing a storage offer.

Keywords: Cloud Storage; Dropbox; Measurements; Performance; Benchmarking

Onedata: High-performance Globally Distributed Data Access Platform

M. Orzechowski (AGH Krakow)

Łukasz Dutka, Michał Wrzeszcz, Renata Słota, Michał Orzechowski, Jacek Kitowski AGH University of Science and Technology, Academic Computer Centre Cyfronet AGH, Krakow, Poland AGH University of Science and Technology, Department of Computer Science, Krakow, Poland

Keywords: data access, distributed system, file sharing, open data

Onedata [1] is a complete high-performance storage solution that unifies data access across globally distributedenvironments and multiple types of underlying storages, such as Lustre, GPFS, or other POSIX-compliant filesystems. It allows users to share, collaborate and perform computations on the stored data. Globally Onedata comprises Onedata Providers, that expose storage systems to Onedata and provides actual storage to the users. Onedata introduces the concept of Space, a virtual volume, owned by one or more users. Each Space has to be supported with a dedicated amount of storage by multiple Onedata Providers. The Spaces are accessible to users via an intuitive web interface, that allows for Dropbox-like file management and file sharing, fuse-based client that can be mounted as a virtual POSIX file system, REST and CDMI standardized APIs. Fine-grained management of access rights, including POSIX-like access permissions and access control lists (ACLs), allow users to share entire Spaces, directories or files with individual users or user groups. Furthermore, Onedata features local storage awareness that allows users to perform computations on the data located virtually in their Space. When data is available locally, it is accessed directly from a physical storage system where the data is located. If the needed data is not present locally, the data is fetched in real-time from remote facility, using a dedicated highly parallelized dedicated protocol.

Currently Onedata is used in PLGrid [2] and Indigo-DataCloud [3] projects as a federated data access solution, aggregating either computing centres or whole national computing infrastructures; and in EGI-Engage [4], where it is a prototype platform for dissemination and exploitation of open data sets, including capabilities such as: publishing data to selected communities or public, metadata-based data discovery, and open data DOI (Digital Object Identifier) generation.

Acknowledgements

This work is supported partly by the European Regional Development Fund program no.POIG.02.03.00-12-137/13 as part of the PL-Grid CORE.

References

[1] Ondata: Global data storage solution for science. http://onedata.org
[2] PL-Grid (Polish Infrastructure for Supporting Computational Science in the European Research Space). http://projekt.plgrid.pl/en
[3] INDIGO-DataCloud (Integrating Distributed data Infrastructures for Global Exploitation). https://www.indigo-datacloud.eu/
[4] EGI-Engage (Engaging the Research Community towards an Open Science Commons). https://www.egi.eu/about/egi-engage/

Syncany: Concepts for trustless abstracted cloud storage

Pim Otte, Syncany

otte.pim@gmail.com

In the post-Snowden world it is more clear than ever that we cannot allow to pass data around the internet without encryption. This poses a particular problem in the case of sharing files and offsite backups.

Syncany is an open source project that aims to offer users the ability to use their own cloud storage provider efficiently with client-side encryption and versioning. This implies that whoever they are using as a provider is trusted to ensure availability of data, but they are not trusted with readable data of any kind.

The three key properties of Syncany are the ability to do offsite backups, with efficient versioning and encryption. Other tools offer the ability to do two of these three, but not all of them. Because there is no way to do server side logic, everything needs to be done on the client side. The implication of this is that the cloud is treated as just dumb storage. This allows Syncany to employ a design that is flexible in the choice of backend by means of plugins.

This talk will detail the concepts that allow Syncany to achieve all of this. Among the topics of discussion will be: Abstraction of the cloud backend, deduplication of data, client-side encryption and, finally, decentralized versioning and conflict resolution through vector clocks. All of these concepts will be introduced and explained by demoing step-by-step how Syncany uses each of these to allow users to be sure their private data stays private. Finally, the talk will be concluded by touching on open areas of research and development surrounding Syncany.

The next step in federated file sync and share

Frank Karlitschek, Founder and CTO ownCloud

frank@owncloud.com

ownCloud has rapidly grown, within a few years it went from an open source project to a company to some of the largest private deployments with often tens of thousands of users per installation and millions of total users.

In the course of these very agile years, the scope of deployments and actual challenges in transcending beyond singular clouds into a strategical direction of federation has led to significant architecture adjustments and also lessons learned.

We believe that the future of global file sync and share is federation of independent cloud servers instead of centralized and monolithic services. It should be possible that decentralized and distributed cloud server work together and behave from a user point of view as one big virtual service. Similar to Email where different organizations can host their own mail server but they exchange mails as if it one big centralized service.

Because of that federation is one of the core strategic goals for the last 2 years. The first basic feature was available in ownCloud 7. In ownCloud 8 we add a second way to do federation and improved this features in 8.1 and 8.2 I'm happy to announce that in ownCloud 9 we will reach the next milestone that I'm happy to call 'full federation'. The focus areas are performance improvements be introducing push notification between federated server, extended auto completion of users on remote servers and advanced monitoring and security settings.

ownCloud published a draft of the Federation API last year and is actively working with partners and standardization bodies to establish this as an open standard. This should become an open standard that can be implemented by different software vendors.

This talk will cover ownCloud 9 and beyond, giving an outlook on the strategical and architectural decisions and directions ownCloud will take in the years to come.

Benchmarking and testing ownCloud, Seafile, Dropbox and CERNBox using smashbox

P. Mrowczynski (Technical University of Denmark, Copenhagen)

F. Orellana (Technical University of Denmark), J. Moscicki (CERN)

File synchronization and sharing services are increasingly used in research collaborations. In order to support this, there are three basic demands that the underlying architecture needs to meet: reliability, scalability and security. The aim of the present work is to allow service providers to better assess the operational status of their service, better understand the service behavior from a user perspective and to identify possible bottlenecks.

Concretely, we have extended Smashbox [1], a testing framework for ownCloud-based services, with benchmarking capabilities and new supported clients. We have used the benchmarking tools to continuously measure the performance of the ownCloud deployments in real-time and compared with existing commercial cloud storage services. Currently, the benchmarking tool [2] supports ownCloud, Seafile and Dropbox clients and allows simulating specific user behavior via a set of customizable tests. Collected benchmarking data can be visualized using a Grafana web interface or analyzed using a set of scripts to generate summary reports for a specific period of time.

In this work we present possible use cases of the benchmarking framework. The test cases are mostly focused on monitoring the operational status of the commercial services, gauging available resources to a specific user under different loads at different times of the day and less focused on general comparison of the capabilities of the synchronization clients.

Keywords:
Cloud synchronization, Benchmarking, ownCloud, Dropbox, Seafile, Operational status, Quality of service, Scalability, Reliability.

References:

1. https://github.com/mrow4a/smashbox-analytics
2. https://hub.docker.com/r/mrow4a/smashbox/

Scaling ownCloud - by user redirection and proxying

Frederik Orellana (Technical University of Denmark)

fror@dtu.dk

ownCloud is built on the LAMP stack. For traditional LAMP web sites, scaling techniques have been developed over the past decades: HTTP load balancing, reverse proxying, separation of web servers and DB server, MySQL scaling and tuning. ownCloud however, is nonstandard, in the sense that it serves nonstatic content using both file and DB servers and not only serves, but also receives data (i.e. it is stateful). One way of scaling this is by standard web scaling on top of a distributed file system and a distributed DB server. This still leaves the filesystem and DB server as potential bottlenecks. We propose a different approach: HTTP load balancing with redirects according to user ID i.e. always sending the same user to the same physical server. This eliminates the need for distributed file system and a central DB. Concretely, we're writing an ownCloud app implementing this. An obvious complication is files shared between users on different physical servers. In the web interface, for the time being, we deal with this via proxying.

Testing storage and metadata backends with ClawIO

H. Labrador (Vigo University)

Hugo Gonzalez Labrador[1] [a]LIA2, University of Vigo

Open Source Cloud Sync and Share software provides synchronisation layer on top of a variety of backend storages such as local filesystems and object storage. In case of some software stacks, such as ownCloud, a SQL database is used to support the synchronisation requirements.

We tested how different technology stacks impact the ownCloud HTTP-based Synchronisation Protocol.

Efficiency and scalability analysis was performed based on benchmarking results. The results have been produced using the ClawIO framework prototype.

ClawIO [1] is a Cloud Synchronisation Benchmarking Framework. The software provides a base architecture to stress different storage solutions against different cloud synchronisation protocols.

Such architecture is based on the IETF Storage Sync draft specification [2]. The synchronisation logic in divided into control and data servers. This separation is done by the use of highly-decoupled micro services connected to each other using high performance communication protocols such a gRPC [3].

Keywords: Cloud Synchronisation, Benchmarking, ownCloud

References

[1] http://clawio.github.io/
[2] L. S. Y. Cui, Z. Lai, Internet Storage Sync: Problem Statement
[3] K. Varda, Protocol Buffers: Google's Data Interchange Format

A study of delta-sync and other optimisation in HTTP/Webdav synchronisation protocols

W. Jarosz (AGH Krakow and CERN)

Recent advances in provisioning of cloud storage services enable geographically distributed teams of researchers to store, share, transfer and synchronize files between their personal computers, mobile devices and large scientific data repositories. One of the challenging issues is synchronization of large-scale data repositories and how to achieve it efficiently using the existing, open protocols.

The main objective of this research is to analyse existing open-source synchronization and sharing protocols (with a strong focus on Webdav / HTTP sync) and to identify enhancements required to better support of the following use cases: simple office document workflows ("dropbox", low-frequency modifications), home directory extension, novel scientific workflows - integration with existing infrastructure and services for data storage, analysis and processing. The following possible enhancements are considered: delta-sync, bundling, data chunking, deduplication, checksumming, network-awareness. The research is based on analysis of the real-life usage of CERNBox service, which bases on an open source Webdav / HTTP synchronization protocol. Results were gathered using monitoring data, server logs and network usage statistics. Metrics used in the research include: Percentage of versioned files HTTP PUT/GET ratio Time needed for simple operations File types and sizes distribution and more.

Novel Applications and Innovation

Data Mining as a Service: Innovative user applications and workflows with cloud storage services

Danilo Piparo (CERN)

Massimo Lamanna, Luca Mascetti*, Jakub Moscicki*, Danilo Piparo**, Enric Tejedor***
** CERN IT-DSS*
*** CERN PH-SFT*

Data Mining as a Service (DMaaS) is a software and computing infrastructure that allows interactive mining of scientific data in the cloud. It lets users run advanced data analyses in a browser with a single click, also leveraging the widely adopted Jupyter Notebook interface. Furthermore, the system makes it easier to share results and scientific code, access scientific software, produce tutorials and demonstrations as well as preserve the analyses of scientists. The single components of DMaaS are characterised: from the high level user interface to the backends linked to services which are commonly provided by IT departments such as files synchronisation, mass storage, conference management tools, development portals or batch systems. The added value acquired by the combination of the aforementioned categories of services is discussed.

To conclude, the experience acquired during the implementation of DMaaS within the portfolio of production services of CERN is reviewed, focusing in particular on the opportunities offered by the CERNBox synchronisation service and its massive storage backend, EOS.

Towards an infrastructure for interactive Earth Observation data analysis and processing

A. Burger (European Commission - Joint Research Centre Ispra)

A. Burger[1] and P. Soille[2]
European Commission, Joint Research Centre (JRC)
[1]Institute for Environment and Sustainability, Digital Earth and Reference Data Unit
[2]Institute for the Protection and Security of the Citizens, Global Security and Crisis Management Unit

The Copernicus programme of the European Union with its Sentinel fleet of satellites operated by the European Space Agency are effectively making Earth Observation (EO) entering the Big Data era by delivering massive amounts of image data in various domains of the electromagnetic spectrum and with global revisit time as short as five days. This means that most current application projects require new tools and enhanced data sharing capabilities.

After a brief presentation of the concept of the Earth Observation Data and Processing Platform that is being developed at the Joint Research Centre (JRC), we focus on the development of an interactive framework for EO data processing and analysis based on IPython Notebooks and present possible scenarios to benefit from public cloud storage service in view of achieving results faster and more efficiently.

Index Terms - Earth Observation, IPython, Jupyter, Docker

User friendly access to Grid and Cloud resources for scientific computing

Alex Richards (Imperial College London)

We will demonstrate a complete system that provides for easy access to grid/cloud resources through a mature python based framework. Computational tasks (jobs) are defined and they can subsequently be sent for execution at a large number of different computing resources without that any detailed understanding of these is required. Interaction with file storage systems such as Grid storage elements, mass storage file systems and the CERNbox directly from worker nodes will be discussed. The underlying technology is based on the Ganga user interface and the Dirac workload management system. A number of examples will be given from user communities across different areas of science and a simple demonstration of how this could be integrated into a Jupyter notebook framework will be given.

Experience in integrating end-user cloud storage for CMS Analysis

H Riahi[1] (Hassen.Riahi@cern.ch)

A Aimar[1], A A Ayllon[1], J Balcas[2], D Ciangottini[3], J M Hernández[4], O Keeble[1], A Kiryanov[1], N Magini[5], A Manzi[1], L Mascetti[1], M Mascheroni[6], A J Tanasijczuk[7], E W Vaandering[5]

[1]CERN, Switzerland
[2]Caltech, USA
[3]INFN, Perugia, Italy
[4]CIEMAT, Spain
[5]FNAL, USA
[6]INFN, Milano-Bicocca, Italy
[7]UCSD, USA

End-user cloud storage is increasing rapidly in popularity in research communities thanks to the collaboration capabilities it offers, namely synchronisation and sharing. CERN IT has implemented a model of such storage named, CERNBox, integrated with the CERN AuthN and AuthZ services. To exploit the use of the end-user cloud storage for the distributed data analysis activity, the CMS experiment has started the integration of CERNBox as a Grid resource. This will allow to CMS users to make use of their own storage in the cloud for their analysis activities as well as to benefit from synchronisation and sharing capabilities to achieve results faster and more effectively.

In this paper, we present the integration strategy and infrastructure changes needed in order to transparently integrate end-user cloud storage with the CMS distributed computing model. We describe the new challenges faced in data management between grid and cloud and how they were addressed, along with details of the support for cloud storage recently introduced into the WLCG data movement middleware, FTS3. The commissioning experience of CERNBox for the distributed data analysis activity is also presented.

Authorization in GARRbox: Delegated resource management and user data isolation

F. Farina (GARR, the Italian NREN)

Co-authors: Massimo Carboni, Cristiano Valli, Paolo Velati, Simone Visconti
Consortium GARR, Via dei Tizii 6 – 00185 Rome Italy
{name.surname}@garr.it

GARRbox [1] is the synch and share service built and operated by Consortium GARR, the Italian National Research and Education Network [2]. GARRbox is not a general purpose cloud storage. The service is the GARR response to a specific commitment from the Italian Ministry of Health for supporting the needs of the biomedical research community. While ease of use is important for researchers, collaboration on health data can't come at the risk of breaching patient privacy. That's why security and resiliency are top priority.

The GARRbox architecture is a three tier web application with modern design in mind: a light Twelve-Factor approach has been adopted to grant elasticity, resiliency and standard based access at every tier. Physical resources are metro-distributed over two datacenters. The VMs are resilient through active-active load balancing on the same datacenter and active-passive instances between the sites. The presentation layer is based on ownCloud Enterprise Edition. ownCloud has been adopted for different reasons: clients availability on most OS platforms and support for encryption. Moreover, Consortium GARR has a rich history of supporting the Italian identity federation [3] and ownCloud is able to leverage the existing infrastructure.

Users are not allowed to access GARRbox directly. Their home institutions must subscribe the service first, then they receive a number of user slots and a global amount of storage. The service lets every institution manage the resources. The local Virtual Resource Managers (VRM) control the user access and customize quota. When an institution needs more resources, they will be assigned with just a click. Furthermore, physical location where users' data are located is parametric. The VRM chooses at subscription time whether data will be hosted on a separate volume or on a storage shared among other subscribers.

Data location is transparent to the end-user during the registration workflow, enabling the customizability of the service according to the subscribing institution needs. This is implemented with a custom ownCloud authentication plugin and a federated authorization service, called Primologin. GARRbox combines federated authentication and local credentials to support of encryption also on non-web clients. Sensible operations benefit of separate authentication channels: for example, password reset needs also authentication through the Identity Federation.

At subscription, Institutions provide information about their Identity Providers and endorse the users that will be VRM. Federated Discovery Service has also been customized to interact with Primologin and expose only the subscribers' IdP. When a new end user registers to GARRbox, federated authentication is performed. According to the attributes, the user is associated with the institution, the local ownCloud account is created, and personal data area is mapped on the institution virtual storage as stated by the VRMs. When the registration completes, the VRM receives a notification asking for the user authorization. To simplify the management, Primologin provides a domain specific language to state authorization ACLs.

In this work we present how Primologin authorization service interacts with ownCloud to support access control and virtual resources delegation in GARRbox. We give an overview of the adopted resource model and the services interactions, showing how the approach can be generalized to other applications. Finally, we provide a brief report of the service adoption, discussing the steps needed to open the service to a larger user base like the Italian Universities and the Cultural Heritage Community.

References

[1] GARRbox: https://garrbox.garr.it/
[2] Consortium GARR: http://www.garr.it
[3] IDEM Identity Federation: https://www.idem.garr.it

Zenodo: enabling scientists, projects and institutions to share, preserve and showcase multidisciplinary research results

T. Smith (CERN)

Lars Holm Nielsen, Tibor Simko, Tim Smith; CERN

The dream of Open Science is to ensure the fruits of scientific endeavor are available to all, anywhere in the world, to ensure solid ground for the wheels of science to turn on. But today data is rarely shared when publishing results since the processes which were designed for papers can't handle the scale and complexity needed for data. This undermines the very foundations of the scientific process since without access to research results, it is difficult to validate, filter or reuse the research. Zenodo [1] is an Open Data service conceived to fulfill this need and power the move to Open Science. Having access to data is only one part of the story however, as it is often difficult or even impossible to interpret the data without also having access to the code used to perform the published analyses. So Zenodo offers the same open services for software, as well as any other output produced by researchers.

Zenodo was created at CERN as part of it's mission to make available the results of its work, with the backing of the European Commission as part of the OpenAIRE project [2]. Zenodo is built on the Invenio [3] digital library software developed by CERN for its own institutional and subject repositories, as an open source project. Zenodo is built on CERN's cloud compute and storage infrastructure, and will soon migrate to EOS [4] backend data store used to store the vast LHC data.

To facilitate and encourage the publication of research data and software it is desirable to make the process trivial by connecting Zenodo with the tools and repositories where the researchers are exchanging their work-in-progress versions during the collaborative analysis phase. For software, connecting with Github [5] was a good first step! For data, the possibility for download from DropBox was immediately offered. Moreover, I will propose in this talk that a more integrated connection with OwnCloud would be highly desirable, given its growing popularity in academic circles.

References

[1] http://zenodo.org
[2] http://openaire.eu
[3] http://invenio-software.org
[4] http://eos.web.cern.ch
[5] https://guides.github.com/activities/citable-code/

Publishing, sharing and preserving your data: from synchronization to curation

F. Orellana (Technical University of Denmark Copenhagen)

Frederik Orellana (Technical University of Denmark), Lars Holm Nielsen (CERN ITCIS), Lars Christensen (Technical University of Denmark)

ownCloud allows easy uploading, storing and sharing of research data, and several deployments exist that are used used for precisely this. One example is the ownCloudbased Danish research data service, data.deic.dk, provided by the Danish eInfrastructure Cooperation. Typically, however, such services are not intended as persistent publishing or longterm archiving platforms.

Zenodo is a service, provided by CERN, intended as a longterm publishing and archiving platform with appropriate storage, preservation, presentation and publishing features along with a sustainability commitment.

The aim of the present project is to expose the combined benefits of two services in a seamless way to researchers effectively shortening the path from storing and sharing to publishing and preserving research data.

Concretely, we're implementing an ownCloud app, that will allow uploading and publishing data to Zenodo with a few mouseclicks. The aim is for the app to provide ORCID integration in ownCloud and to make use of Zenodo's API and ORCID integration.

Operations

User Management and Sharing in Sciebo, the Academic Cloud Storage Service in North Rhine-Westphalia

Holger Angenent, holger.angenent@uni-muenster.de, University of Muenster

Andreas Wilmer, Christian Schild, Raimund Vogl

Sciebo [1] is a federated cloud storage service for most of the higher education institutes in North Rhine-Westphalia. Each of the participating institutes is represented by a separate ownCloud instance on the servers. Thus the federated cloud sharing is crucial for sharing between two instances. It was chosen to create a single instance for each institution because the service is hosted at three different sites. Furthermore, this distribution model makes the service more scalable.

We will give details about the federated cloud sharing mechanism in connection with sciebo as well as the concept of user management which allows to in- or exclude institutes in a flexible manner. For this purpose, we have established a portal for self-registration via Shibboleth to give them accounts for sciebo that are stored in a central LDAP server. Those accounts can have expire dates independent from those of the home institutions of the users. We guarantee the users to have a six month courtesy time to migrate to other services after leaving their institution.

The account names are composed of the user names at the home institution in addition with the domain of the institution, separated by an @, for example john.doe@uni-muenster.de.

The desktop and mobile clients make use of this naming scheme. A lookup server is used to send the user clients to their individual institutions, so we can distribute the same branded clients with pre-entered URL for all users. For the access of the Web GUI, it is still necessary to know the URL of one's own sciebo instance.

References

[1] "sciebo - the Campuscloud" for NRW. http://sciebo.de/imperia/md/content/ziv/pdf/sciebo_eunis-paper.pdf

INFNBox: a distributed and high reliable cloud infrastructure for data sharing at INFN

G. Donvito et al. (INFN Bari)

Giacinto DONVITO (INFN-Bari) giacinto.donvito@ba.infn.it Stefano STALIO (INFN-LNGS) stefano.stalio@lngs.infn.it

Data Sync and Share public tools that allow for improved collaborative work are not optional anymore in scientific communities. Commercial products, like DropBox and GoogleDrive, provide ease of use and advanced functionalities but often limit the quota usage to free accounts and, much more important, do not guarantee data persistence, ownership, and privacy to the end user.

ownCloud is the tool that many INFN sites chose for their local desktop sharing needs, but such systems do not meet some of the requirements of INFN scientific collaborations that are, most of the times, distributed over multiple sites and over universities and research institutions all over the world.
INFN IT departments have today the expertise and the infrastructure needed for creating a nation-wide cloud storage to be used by the about 6000 staff and associates based in the 20 plus institution sites all over Italy. Federating this infrastructure with that of partner institutions would represent an important improvement to the way information and knowledge are shared.

This talk will describe the distributed, fully redundant and highly available architecture we plan to implement for the realization of the INFN sync and share platform and will give an overview of the technologies that will be adopted, highlighting innovative approaches and solutions. We will also describe how ownCloud, CEPH and Swift will be implemented in order to fulfill the users requirements of having a Data Sync and Share tool that is able to survive also in the case of a complete failure of a datacenter and to be scalable at the level required by a small physics experiment.

First experience with box.psnc.pl based on Seafile

Maciej Brzeźniak (maciekb@man.poznan.pl), Poznań Supercomputing and Networking Center, Supercomputing Department

Gracjan Jankowski, Michał Jankowski, Stanisław Jankowski, Norbert Meyer, Rafał Mikołajczak, Krzysztof Wadówka, Paweł Woszuk, Sławomir Zdanowski
{gracjan,jankowsk,staszek,meyer,fujisan,kwadowka,pwoszuk,slawek}@man.poznan.pl
Poznań Supercomputing and Networking Center, Supercomputing Department

Data synchronisation services are among the most widely adopted cloud applications. Everyone uses Dropbox or similar cloud services. Academic community is not an exception. However, while academics need ease of use and convenience, similarly to most users, they also need high scalability and good reliability due to the size or value of their datasets.

PSNC takes the effort to provide Polish academic community with relevant solution. PSNC entered the field by implementing National Data Storage [1] project where secure sync & share client CryptoBox [2] supporting clientside encryption was developed. While limited in scope this experience helped us to understand challenges of the sync & share applications.

Since December 2014 PSNC runs a sync and share service box.psnc.pl [3] based on Seafile[4]. It is a limitedscale production service for PSNC staff and collaborators from EU R&D projects as well as selected universities and institutions in Poznan.

In this presentation we will discuss the motivation to choose Seafile as a promising, reliable, scalable and lightweight solution. We will also share experience from operating and using it over a year. In addition we will overview its interesting features (such as Thunderbird's Filelink integration extensively used in our daily business) as well as the latest developments at Seafile, (including these related to data security clientside data encryption).

References

1. National Data Storage: https://nds.psnc.pl
2. M.Brzeźniak, G.Jankowski, M.Jankowski, S.Jankowski, N.Meyer, R.Mikołajczak, P.Woszuk, A.Zawada, S.Zdanowski, K.Guryn, T.Chmiel, L.Redynk. Secure alternative to Dropboxlike sync and share solutions. TERENA Networking Conference 2014 Presentation. https://tnc2014.terena.org/core/presentation/4
3. Box@PSNC service: https://box.psnc.pl
4. Seafile. https://www.seafile.com/en/home/

CERNBox: Cloud Storage for Science

Luca Mascetti (CERN)

Massimo Lamanna (CERN), Jakub Mościcki (CERN)

CERNBox is a cloud storage service for end-users that allows researchers to share, transfer and synchronise files between their personal computers, mobile devices and large scientific data repositories. The service offers “sync and share” capabilities on all major mobile and desktop platforms (Linux, Windows, MacOSX, Android, iOS).

CERNBox is based on the ownCloud software stack integrated with EOS*, the CERN large-scale data storage repository which provides 70 petabytes of usable storage capacity.

CERNBox provides out-of-the-box functionality of synchronization and sharing of files using standard ownCloud client software and web platform. These capabilities, as well as offline access to all files already stored in the CERN EOS infrastructure, offer an opportunity for new scenarios for data analysis. The evolution of the service is driven by a high (and growing) demand in the community for an easily accessible cloud storage solution. The service allows users to share big volumes of data across working groups and provides extremely easy to use interface available on all client platforms.

In this report we present lessons learnt in offering the CERNBox service with the emphasis on scalability and quality of service. We also focus on the key technical aspects of EOS-ownCloud integration: direct users access to the underlying storage backend with a guarantee of consistency between the data store and the metadata namespace used for synchronization. CERNBox provides consistent metadata namespace independently of the access protocol (filesystem, sync clients, webdav mounts or web layer). This significantly improves the “external storage” concept originally introduced in ownCloud software stack which requires periodic rescanning of external data sources. We also introduce modifications in the sharing component of the ownCloud web layer to allow CERNBox to provide sharing in a more scalable way.

We report on emerging usage possibilities and past showcases, in particular the ongoing integration of “sync and share” capabilities with the LHC data analysis and transfer tools in the CERN service ecosystem. We also discuss how this affects the future of the CERN home directory service.

* "EOS as CERNBox backend is supported for collaborating user communities. Optional commercial support is being prepared by a company".

Distributed Sync&Share Operations at AARNet

G. Aben (AARNET)

As will be obvious to all participants in this workshop, the availability of a suitable sync&share frontend and client suite allows one to begin offering a decent scale sync&share service. A fair number of participants will by now share our experience that, as soon as this service begins to scale to significant numbers of users (~>1000) spread across significant geographical expanses (~>50ms), the effects of both the synch as well as the share property become manifest and need to be addressed before one can scale further.

Synching, after all, implies bidirectional traffic, and given the end user TCP stack even of 2016, this implies one has to locate the data repository proximal to the user (indeed all major commercial CDN systems do so). On the other hand sharing, and more specifically collaborating on files, implies data consistency; hence, taken together, we now have the requirement to keep consistency across nodes which individually may be ingesting data at ~10GBit/s; one proximal to the originating user, one proximal to a sharing user (who may be >50ms away).

AARNet, the operator of the Australian R&E network, has been striving to cater to these requirements for a number of years now. We have been rather activist in using our staging platform to swap out unsatisfactory componentry for newly available open source modules. As of late 2015, most of the architecture stack is performing to our satisfaction; the components we now hope to swap out for further improvements are a) the underlying replicating filesystem (which at the moment doesn't replicate in realtime), b) the memory caching system (which at the moment doesn't share state across nodes), c) horizontal scaling by on-demand launching of new worker nodes such as database, webserver frontends etc (currently static; new nodes must be hand-cranked). We have identified candidates and are actively testing with these and our presentation will detail experiences so far.

Apart from these architecture field nodes, we have recently engaged with the Australian research cloud provider (NeCTAR) to put in place trans-organisational ownCloud-to-OpenStack integration (as in, they don't share physical nodes, backend or policy), making use of a SAML2 based federation inbetween. This is now in a state where entitled users can use the AARNet ownCloud frontend and user clients to ingest and extract data into/from a 3rd party OpenStack cloud; we will report on this work as well.

u:cloud - File Sync and Share in a heterogeneous scientific environment at the University of Vienna

Christian Kracher (University of Vienna)

Christian Kracher (Project Manager at Computer Centre of the University of Vienna)

In the field of research, the University of Vienna is subdivided into 15 faculties and four centres. Faculties are responsible for research and teaching, while centres fulfil specific tasks for the University or predominantly serve either as research or as teaching bodies. For these overall 10.000 staff members of the University, of which are approx. 7000 scientists, u:cloud is the designated File Sync and Share service since January 2015.

File Sync and Share Services are widely used to facilitate certain workflows and are a de facto standard tool on many electronic devices. They're also enabling collaboration over geographical borders and are therefore predestined to help scientists with their research.

Collaboration across disciplines, the increasing size of those collaborations, and the application of information technologies on scientific problems have contributed greatly to the data-intensive nature of scientific research. These trends have reduced the significance of distance, allowed scientists to share resources remotely, and increased both the complexity and the quantity of research data.

A substantial amount of interviews with scientific groups at the University of Vienna were done before and after the roll out of the u:cloud service. Initially, we wanted to know, which demands the scientists had - and afterwards, what they were actually doing with the service.

Some of the results include, that scientists are much more flexible and secure in their research with File Sync and Share Services. For example, archeologists using the service at excavations for synching photos and directly sharing it with colleagues for analysis.

Keywords: Cloud, Collaboration, Share, Sync

My CoRe - ownCloud service at CNRS

David Rousse (CNRS France)

david.rousse@dsi.cnrs.fr
CNRS/DSI, 358 rue Pierre Gilles de Gennes, Bât. Tour GAIA, Quartier Bouysset, 31 670 Labège, France

Background and Context

CNRS has launched since the end of 2012 services dedicated to Higher Education and the research communities (http://ods.cnrs.fr/):

- messaging service (based on MS Exchange)
- chat and videoconference service, called My Com (based on MS Skype for Business)
- secure cloud service (virtual machines, web hosting and data raw storage)
- collaborative portal service, called CoRe (based on MS Sharepoint)

A new service called My CoRe (http://ods.cnrs.fr/mycore.html), based on the open source software ownCloud (http://www.owncloud.org/) has been launched in production since October 2015, after a 9-months-beta period: the goal is to serve CNRS users to easy mobility and nomadism at work.

Functional Choices

The business need was divided into two functionalities: a files synchronization and sharing service on the one hand, and a backup service for end users files on the other hand.

To fit theses needs, we chose to deploy a service based on ownCloud software (community edition), because it had the required functionalities in terms of synchronization and sharing, and it is open source, so easy to customize (check https://github.com/CNRS-DSI-Dev to see what was developed on the top of ownCloud core software). Moreover, the new technical infrastructure deployed for My CoRe service has been located under CNRS' IN2P3 Computing Center, in order to secure the storage and to also use the existing IN2P3's internal backup service.

Technical Choices

Is ownCloud scalable for a large number of users and if so, what is the cost of such a service on the long term? That was the first and main question we had to answer at the beginning of the project. To achieve this task, we ran load tests and were able to provide a result sheet where functional hypothesis on the ownCloud service usage are linked to the number of web and database nodes, and also to network bandwidth usage (see https://github.com/CNRS-DSI-Dev/mycore_press/blob/master/CNRS-LINAGORA-ownCloud-load-estimate.ods). The cost of such an infrastructure was calculated based on this estimate sheet.

Another huge work during the technical deployment of the project was to choose the web, database and storage components: we decided, after some internal tests and theoretical comparisons, to use:

- Apache (with ownCloud, PHP and Shibboleth service provider) as web tier
- MariaDB/Galera as database tier
- Scality as storage tier

Main Project Feedbacks

From a functional point of view, the main difficulty was to provide a solution for two different needs: a file synchronization and sharing service, and a files' backup service.

The length of the project, two years between the kick off and the go live in production, was also a main concern: difficulties during the implementation step, with a new dedicated infrastructure to deploy, were the root cause of this latency.

Regarding the architecture, the sizing of the platform on the long term, to serve a large number of users, and as a consequence the cost of the service, has been the main concern of the technical study. Some improvements of the architecture, for instance accessing the storage tier directly in an object mode or bypassing the oc filecache ownCloud's table, are still to be done.

Ad hoc Federated Storage with OpenStack Swift

Joe Topjian, joe.topjian@cybera.ca, Cybera

Cybera, part of the Canadian NREN, has been actively involved with OpenStack since the very beginning. We deployed the first public OpenStack cloud in Canada in 2011 and have seen numerous other clouds deployed since then. In Alberta alone, we know of at least five OpenStack clouds, and it's safe to say there will be several more in the future.

Many of these clouds share the same users and some of these users want to leverage their multi-cloud access. But what does "multi-cloud" access look like? Federated Identity is one solution: clouds under the same federation can allow users to authenticate with the same set of credentials. However, their cloud resources are still isolated. Additionally, some clouds may not be able to join a federation.

Swift, OpenStack's object storage component, has an interesting feature called "Container Sync" that solves both of these issues. It enables a user to synchronize a storage container from a local Swift service to one or more external Swift services. This works regardless if the user authenticated through a federation or by using separate local accounts.

We will discuss how to enable Swift's Container Sync, how it works, and our experience using it to date. We will also discuss different synchronization models that can be created with Container Sync and the advantages and disadvantages of each.

Experience with running Owncloud on virtualized infrastructure (Openstack/Ceph)

C. Schnidrig (SWITCH Zürich)

Christian Schnidrig, Cloud Engineer, SWITCH christian.schnidrig@switch.ch

Owncloud is quite popular among sync & share service providers. As the name implies, Owncloud was build with home users in mind. It can run on devices as small as a raspberry pi. At the same time this product is also sold to service providers who support with one single Owncloud instance more then 10k users. This already being an astounding achievement, it is not yet good enough. Service providers would need Owncloud to scale up to 100k users or even more.

We at SWITCH have a sync & share service (SWITCHdrive) based on Owncloud as well as an IAAS offering (SWITCHengines) based on Openstack/Ceph. Having this Openstack cluster, it seemed natural to us to put an ever---growing cloud service like SWITCHdrive on it.
In order to run well on Openstack, an application has to scale horizontally. The cloud deployment should also not introduce single points of failure.

Reality is of course different. Originally written with small deployments in mind, Owncloud is not really well suited for Openstack/Ceph.

We want to present our Owncloud deployment on SWITCHengines, discuss its advantages but also its limitations. The work---arounds we've implemented/planned and future Owncloud features that would help with running it on an Openstack cluster.

UnipiBOX: owncloud@unipi

Simone Spinelli, simone.spinelli@unipi.it, University of Pisa

Davide Vaghetti, Samuele Tognini

The University of Pisa is a middle-sized university in the center of Italy with about 70.000 students and 8.000 employees. Departments and research facilities are distributed all over the city and are connected by a proprietary optical network which was born around the year 2000 and is still growing, connecting other research and government institutions. Students are connected wired or wireless, inside the departments, in their accommodations and in the outside areas. All users are present on the network: everybody with their own data to share, specificities and requirement.

The world of research is probably the biggest producer of data. In this case, the need is to store data and share it with the outside world, like other universities or research centers: this should be done in an easy and secure way. On the contrary, employees and administrative staff want to share most of the data inside the university and their concern goes to privacy and data security. All these different needs and perspectives find an answer in a private sync&share platform.

Sync and share solutions are widely offered in many different flavors in terms of functionalities, software licensing and architecture. Cloud-based solutions are really attractive: requirements like availability and disaster recovery are automatically met by relying on professional cloud services and, especially for educational institutions, cost analysis show that this choice is convenient, decreasing the CAPEX. For a university, anyway, user's data are a primary asset and the absence of a real ownership of policies and data make the (public) cloud unsuitable.

The University of Pisa already invested a lot of resources in realizing and maintaining an IaaS platform based on Openstack, using Ceph as the storage backend. Right now the infrastructure is distributed over three datacenters and it provides 2PB raw storage, 512 CPU and 2TB RAM.
The considerations about data ownership and the actual possibility to use an in-house computing and storage infrastructure already in production, make it convenient and reasonable to implement a sync&share platform on premise using opensource software.
The aim of this contribution is to give an overview of how the University of Pisa intends to realize and offer a private sync&share service, based on existing storage and computing infrastructure: the target is to reach all the students and employees of our community (around 75K).

The chosen solution is Owncloud: at the moment of writing this document, a testbed is running for a small group of users (around 50) to verify software functionalities: it is realized with a single virtual machine hosted on openstack and it uses Ceph RBD volume as local storage.

The future production infrastructure is under development. The main idea is to use Ceph object-storage capabilities for realizing the owncloud primary storage: this should assure a robust and scalable backend. The use of a virtual infrastructure and proper automation mechanisms to host and deploy the owncloud instances, permit to achieve the necessary elasticity and scalability for the frontend.

To verify the model, we are rolling out another testbed for a bigger group of users (1K) that uses the object-storage backend and all the redundancy and load-balancing mechanisms. The feasibility of this approach in terms of performances and manageability is the main subject of the presentation.

Despite that a sync&share service is valuable by itself, such a massive deployment, theoretically done for all the users, makes it possible to use this platform as a center for other services to pivot on.

A simple example is to use owncloud to connect other public sync&share services (Dropbox, GoogleDrive, etc.). This embedded function overturns the user's perspective on the service: from another tool that you have to learn and configure, to a hub that connects all your storage accounts, making them all accessible and manageable from a single point.

Another integration scenario under development, is to use owncloud as a reseller of storage for other services using webdav protocol, in first instance for a personal webpage platform. Every user will have a special directory in the owncloud space where they can put personal website files. The results of these integrations will be discussed in the presentation.

Even in a small environment, like a single university, it is strategic to implement and maintain internally a sync&share platform. It allows a development that follows the real requirements of the organization. This approach, of course, originates challenges in both technical and administrative standpoints, but to face these challenges, originates the know-how that makes it possible to better support research and users in the future.

Two years of SURFdrive, lessons learned and future plans

R. Trompert (SURFSara Amsterdam)

ron.trompert@surfsara.nl

SURFdrive is an Owncloud-based personal sync-and-share service provided by SURF to the Dutch science and higher education community.

In this presentation we will discuss our experiences we have had and the lessons we have learned running this service in production. This will include the SURFdrive system as a whole, including storage backend, datatases, web servers and load balancing.

In addition we will discuss the plans we have for the future developments of the service as well as other domains where we intend to deploy Owncloud.

We intend to couple a storage system called dCache to an Owncloud instance. The dCache system has a transparent interface to our HSM system and it supports a wide variety of protocols to access data. We want to use this system to enable users to store and access their data from jobs running on our clusters and super computer as well as use Owncloud to enable data sharing.

Site Report LMU Faculty of Physics

Klaus Steinberger (Ludwig Maximilians Universität München)

Klaus.Steinberger@physik.uni-muenchen.de
RBG Fakultät für Physik

As many of our Users asked for services like Dropbox, we started evaluation some years ago. Major selection criterias where hosting on-site and licensing cost. After sorting out some commercial products because of missing features, we decided to go the open source way.

Starting with Version 4 we tested ownCloud and beginning with Version 6 we invited for a friendly user test. Since Version 7 we are in production with currently 350 users.

I will report about our (small) deployment, and future plans.

DESY-Cloud: an ownCloud + dCache update

Paul Millar, paul.millar@desy.de, Deutsches Elektronen-Synchrotron (DESY)

Quirin Bockholz, Patrick Fuhrmann, Tigran Mkrtchyan and Peter van der Reest

DESY is a research institute with users that have demanding storage requirements. In the past, these requirements led to forming the dCache collaboration, resulting in dCache: scalable software for storage. As has already been presented, DESY recently started a project to build a sync-and-share service using ownCloud and dCache: the DESY-Cloud service.

In this presentation, we will provide the current status of the DESY-Cloud project and report on recent work, both within dCache and ownCloud software stacks. We will describe the current pain-points within this stack and what we see as potential mechanisms to resolve them. Finally, we will describe the direction we see future development in this area.

Polybox: Sync and Share at ETH Zürich

T. Steiger, steiger@id.ethz.ch, (ETH Zürich)

Gianluca Caratsch, gianluca.caratsch@id.ethz.ch, (ETH Zürich)

Late in 2012 IT services at ETH Zürich started with an owncloud pilot for some selected users in order to test functionality. In summer 2013 the service "polybox" got operative and the new service was appreciated by most of the ETH users – students, PHD students, Professors and all staff members.

The growth rate in early months: in July 2014 the user quota was increased from 5GB to 50GB per user. With 50GB quota per user the acceptance raised again and today we have about 16300 users using nearly 50TB of storage. 75% of lecturers are using the service and a lot are sharing their teaching materials and exercises with the students.

In scientific groups polybox is a helpful tool to share data within and outside the campus. polybox is branded and offers desktop clients for all recent OS. Branded iOS and Android mobile clients are available through Google Play and Apple App Store. New desktop clients will be tested intensely before being deployed to central managed clients. We will show the developing numbers of the polybox service and the testing process.

polybox is built on RHEL multiple virtual servers being connected to an IBM SoNAS storage system. The scalable infrastructure helps us to cover the increasing performance requirements of our users. Nevertheless the performance is closely coupled with the ownCloud SW release management. The setup of the infrastructure will be shown in the contribution.

Keywords: infrastructure; service; storage; users; requirements

Interoperability

OpenCloudMesh (OCM) initiative

P.Szegedi, GÉANT peter.szegedi@geant.org

C.Schmitz, ownCloud, cschmitz@owncloud.com, J.T.Mościcki, CERN jakub.moscicki@cern.ch

OpenCloudMesh is a joint international initiative under the umbrella of the GÉANT Association that is built on ownCloud's open Federated Cloud sharing application programming interface (API) taking Universal File Access beyond the borders of individual Clouds and into a globally interconnected mesh of research clouds - without sacrificing any of the advantages in privacy, control and security an on-premises cloud provides. OpenCloudMesh provides a common file access layer across an organization and across globally interconnected organizations, whether the data resides on internal servers, on object storage, in applications like SharePoint or Jive, other ownClouds, or even external cloud systems such as Dropbox and Google (syncing them to desktops or mobile apps, making them available offline).

Around early 2012, TF-Storage participants started to actively look into data storage software platforms in order to provide on-premise file-based sync&share (aka. Dropbox-like) services to their constituencies. A number of commercial tools have been evaluated including ownCloud, PowerFolder, Joyent as well as some NRENs even ventured into the development of a proof-of-concept tool called the Trusted Cloud Drive (TCD) under TERENA. Among all these tools, by mid-2013, ownCloud appeared to be the most promising one with a growing open-source development community behind.

In December 2013, the GÉANT Association (formerly known as TERENA) and ownCloud GmbH. made an agreement that serves to facilitate the desire of various National Research and Education Networking organisations (NRENs) to introduce services based on ownCloud technology and/or to offer the technology to their constituencies.

The participating NRENs and other organisations came together under GÉANT not only to benefit from the joint buying power of the community but also to share deployment details and operations practices about providing on-premise file-based sync&share services to their constituencies using the ownCloud software. The input form the GÉANT community on these aspects are very much appreciated by ownCloud as they don't have any first-hand service deployment and operational experiences. This led to a mutually beneficial technical collaboration between GÉANT and ownCloud Inc.

As part of this collaboration effort, in January 2015, ownCloud initiated an idea (aka. OpenCloudMesh) to interconnect the individual on-premise private cloud domains at the server side in order to provide federated sharing and syncing functionality between the different administrative domains. The federated sharing protocol can be used to initiate sharing requests. If a sharing request is accepted or denied by the user on the second server than the accepted or denied codes are sent back via the protocol to the first server. Part of a sharing request is a sharing key which is used by the second server to access the shared file or folder. This sharing key can later be revoked by the sharer if the share has to be terminated.

ownCloud Inc. developed the first working prototype of this protocol and made the full specification available as a non-public preview for the GÉANT community in August 2015. A collaborative project was established under the umbrella of GÉANT called the OpenCloudMesh project on 22 October 2015. The kick-off meeting was held in Vienna, Austria. The project is co-managed by Peter Szegedi (GÉANT), Jakub Moscicki (CERN) and Christian Schmitz (ownCloud). This combination of project management ensures that all the major stakeholders – GÉANT National Research and Education Networks, CERN research community and ownCloud Inc. as a commercial company with its open source developers community – are equally represented in the project and the technical, management and business aspects are well-covered. The collaborative project is fully open to any participation and in-kind contributions.

Sync & Share - Ripe for standardisation?

S. Leinen (SWITCH Zürich)

Simon Leinen, Cloud Architect, SWITCH simon.leinen@switch.ch

File synchronization and sharing services are used by hundreds of millions of people and has become synonymous with “cloud” for many. Most users use these services via proprietary applications and protocols offered by commercial providers, though some use opensource systems to build their own.

These various systems don’t interoperate: The choice of a storage service generally dictates which client(s) have to be used—often there are no third party implementations. For collaboration, users must usually agree on a common service. Therefore users wanting to share files with different circles will often have to run multiple clients, which creates many issues ranging from learning overhead and confusion to increased load and battery drain.

If there were widely agreed upon protocols, users could freely choose between competing service providers and (independently) between competing client software implementations.

But can we get to such standards? The market is very dynamic, with providers adding functionality far beyond the basic functions of synchronization and sharing. A successful standard must allow further development and differentiation.

In addition, the dominating vendors are neither motivated to give up control over their systems, nor are they interested to make it easier for their users to switch platforms. On the other hand, new and smaller suppliers have an interest in standards, both to share development effort and to reduce their disadvantage due to network effects. Users also have an interest in standards. While individual users find it difficult to contribute to a standardization process, large (institutional) users or user communities could help drive it.

We will try to identify promising areas for standardization in file synchronization and sharing, and consider suitable fora for such an effort, in particular among existing standards development organizations (SDOs). Insights will be drawn from past standardization efforts in related areas (e.g. WebDAV in the IETF) as well as ongoing initiatives such as OpenCloudMesh and the ISS BoF at IETF 94.

A possible format would be a moderated panel with participants from the various stakeholder groups: commercial service providers, software producers, operators, users, and SDOs.