

# **CS3 2018 - Workshop on Cloud Storage Synchronization and Sharing Services**

Monday 29 January 2018 - Wednesday 31 January 2018

AGH Computer Science Building D-17



## **Book of Abstracts**



CS3 (Cloud Services for Synchronization and Sharing) is an active and growing community, which was initiated at the kick-off event in 2014 at CERN. The distinct mark of CS3 is that it grew bottom-up, without a single organization being a sponsor or providing top-down funding.

CS3 attracts a variety of contributors: companies prototyping the concept of on-premise cloud storage; universities and research labs offering such services for research and education activities; and representatives of the academic world. Over the past few years CS3 has become a point of excellence for cloud storage services.

The CS3 community is centered in Europe but it has significant links and collaborations with other parts of the world. It is a unique crossroads for promising SMEs to exchange their experiences and set up collaborations, for example, to allow proper cloud-storage interoperability.

In recent years we have witnessed a global transformation of the IT industry with the advent of commercial (“public”) cloud services on a massive scale. This clearly puts pressure on the on-premise services deployed using open-source software components. It is hence a good time to discuss the dangers, opportunities and future directions for our community. That is why we have added a panel discussion on the Future of Sync&Share to the programme of CS3 2018.

In previous CS3 conferences it progressively became clear that one of the strengths of the on-premise services for the research institutions is the integration of cloud storage services into generalized Cloud Infrastructure Service Stacks for data science (CISS). This year we have added a dedicated session to discuss CISS services to complement the session on Novel Applications.

A third important trend is the progressive incorporation of Collaborative Platform Services into the classic Sync&Share environments. Key players are joining CS3 2018 to present their solutions for real-time, collaborative editing tools that complement the on-premise Sync&Share services.

Thank you to all CS3 contributors and participants for being such a constructive and vibrant community.

Enjoy the excellent scientific programme of CS3 2018 and enjoy your stay in the wonderful city of Krakow!

Jakub T. Moscicki  
Massimo Lamanna



# Contents

SWAN: Service for Web-based Analysis . . . . .	1
Data lifecycle control through synch&share . . . . .	1
Keeper service: Long term archiving tool for Max-Planck institutions on base of Seafile . . . . .	2
Reproducible high energy physics analyses . . . . .	3
Cubbit: a distributed, crowd-sourced cloud storage service . . . . .	3
Owncloud EFSS beyond just "scale": a glimpse into the future . . . . .	4
Next steps in Nextcloud . . . . .	4
Space.Cloud.Unit - blockchain powered Open-Source Cloud-Space Marketplace . . . . .	4
Seafile - what's new in 2017 and future roadmap . . . . .	5
[Poster] Pydio Next Gen : micro-services and open standards . . . . .	6
Private, hybrid, public ... or how can we compete? . . . . .	6
Future Panel Discussion . . . . .	7
Sync and Share is Dead. Long Live Sync and Share. . . . .	7
The future of File Sync and Share . . . . .	8
EOS the CERN disk/cloud storage for science . . . . .	8
Cynny Space Cloud Object Storage: integrated hardware-software solution developed specifically for ARM® architecture . . . . .	8
dCache as a backend for cloud storage . . . . .	9
Onedata Virtual Filesystem for Hybrid Clouds . . . . .	10
iRODS: Data-Centric, Metadata-Driven, Data Management . . . . .	11
Improving swift as Owncloud backend to scale . . . . .	11
ownCloud and objectstore . . . . .	13
Scaling storage with cloud filesharing services . . . . .	13
Sync & share solution to replace HPC home? . . . . .	13

Contribution to a Dialog Between HPC and Cloud Based File Systems . . . . .	14
New sharing interface for SWAN . . . . .	15
Interoperability with sharing: Open Cloud Mesh status and outlook . . . . .	15
Document Processing and Collaboration without Boundaries . . . . .	16
Collabora Online: Real-time Collaboration on Documents . . . . .	16
From Sync & Share to Collaborative Platforms: the CERNBox case . . . . .	17
[Discussion] OCM plans and use-cases . . . . .	17
PowerFolder Open Cloud Mesh API (v0.0.3) integration . . . . .	18
Seamless Sharing in the ownCloud Platform . . . . .	18
Experience of deployment of the SWAN system on the local resources of the St. Petersburg State University . . . . .	19
CERNBox Service . . . . .	19
Fault tolerant Cloud infrastructure . . . . .	19
Sync and Share at SURF . . . . .	20
Experience with taking over a service running ownCloud on virtualized infrastructure (Openstack/CEPH), and surviving the experience . . . . .	20
Site report survey summary . . . . .	21
Running Nextcloud in docker/kubernetes: Load testing results . . . . .	21
Down the Rabbit Hole: Adventures in Bug-Hunting . . . . .	21
Troubleshooting Relational Database Backed Applications . . . . .	22
Rocket to the Cloud – A Faster Way to Upload . . . . .	22
An insiders look into scaling Nextcloud . . . . .	23
ownCloud scalability research . . . . .	23
Shibboleth authentication for Sync & Share - Lessons learned . . . . .	23
Global Scale - How to scale large instances beyond the typical limitations . . . . .	24
Distributed, cross-datacenter replicated relational database and file storage at ownCloud . . . . .	24
Container-based service deployment in private and public clouds . . . . .	25
Future architectures for sync and share based on microservices . . . . .	25
Neural Network Nanoscience Scanning Electron Microscope Image Recognition . . . . .	26
Integrated processing and sharing services for geospatial data . . . . .	26

Research Data Management with iRODS . . . . .	27
Up2University: Educational Platform with Sync/Share . . . . .	28
Towards interactive data analysis for TOTEM experiment using Apache Spark . . . . .	28
Keynote: multiwinner voting: theory, experiment, and data . . . . .	29
CS3 2018 Welcome . . . . .	29
Visit to CYFRONET Computing Center . . . . .	29





**Cloud Infrastructure&Software Stacks for Data Science / 19****SWAN: Service for Web-based Analysis**Danilo Piparo<sup>1</sup> ; Diogo Castro<sup>1</sup> ; Enric Tejedor Saavedra<sup>1</sup> ; Enrico Bocchi<sup>1</sup> ; Jakub Moscicki<sup>1</sup><sup>1</sup> CERN**Corresponding Author(s):** diogo.castro@cern.ch

SWAN (Service for Web-based ANalysis) is a CERN service that allows users to perform interactive data analysis in the cloud, in a “software as a service” model. It is built upon the widely-used Jupyter notebooks, allowing users to write - and run - their data analysis using only a web browser. By connecting to SWAN, users have immediate access to storage, software and computing resources that CERN provides, and that they need to do their analyses.

All these computing resources are isolated and provide users a secure place to run their work. The software provided is centrally managed, delivered in a distributed file system - called CVMFS - allowing users to forget about installation, configuration and compatibility of packages. Storage is provided by EOS, CERN’s mass storage solution, with a private area that is synchronizable through CERNBox - the cloud storage service.

Besides providing an easier way of producing scientific code and results, SWAN is also a great tool to create shareable content. From results that need to be reproducible, to tutorials and demonstrations for outreach and teaching, Jupyter notebooks are the ideal way of distributing this content. In one single file, users can pack their code, the results of the calculations and all the relevant textual information. By sharing them, it allows others to visualise, modify, personalise or even re-run all the code.

Given the importance of sharing and collaboration in our scientific community, the interface of SWAN has been enhanced to ease this task as much as possible. Up until now, besides the manual options (like sending notebooks in emails), users were able to use CERNBox to share their work. But they had to leave SWAN and go to the CERNBox interface, and look for the notebook that they were editing back in SWAN.

This approach worked, but it was not optimal. We wanted to offer a more integrated and simple model. Something that worked with the minimum clicks possible. With this in mind, we brought CERNBox sharing directly inside SWAN. And with it, we also brought a new and redesigned interface that also introduces the concept of a Project. A project is a special kind of folder that, besides the notebook(s), contains all other sorts of files, like input data or images. In order to simplify the process and keep it consistent, this is the only entity that can be shared among users, from within SWAN. And it can be shared from wherever the users are, either from the files view or inside the notebooks editor. Users just need to click on a button and write the names of whom they wish to share with (single users or groups), using an autocomplete that searches CERN’s directory. When someone gets a shared project, they can clone it to their storage - in order to open and edit the files - just by clicking on a button. All without switching services. And since this cloned project now belongs to the user, he can modify it as he wishes, and even share it again.

With the new approach described, sharing is now a first-class citizen in SWAN. A functionality that is very present and highlighted throughout the new user interface. Something that our users need to collaborate in a simpler manner.

**Cloud Infrastructure&Software Stacks for Data Science / 38****Data lifecycle control through synch&share**Guido Aben<sup>1</sup><sup>1</sup> AARNet

**Corresponding Author(s):** guido.aben@aarnet.edu.au

What is the DLCF?

The Data LifeCycle Framework (DLCF) is an Australian nationwide strategy to connect research resources and activities; predominantly those funded by national eInfrastructure funding.

The goal of the DLCF is to smooth over the complexity faced by ordinary researchers, when they have to piece together their own digital workflow from all the bits and pieces made available through funded eInfrastructure as well as commercial players. By simplifying the process, we want to improve data discovery, storage and reuse where possible.

The role of the synch&share service (in this case, AARNet's CloudStor) is to act as a central hub, where data comes to (e.g., from instruments) and is dispatched from (e.g., to compute, for processing) and received back into; eventually and at the end of the cycle, data is expunged; e.g., into a repository or into a publication.

The work on DLCF is currently ongoing. The first stage of the DLCF is the development and live testing of a handful of technologies we feel are missing before we can arrive at improved provenance, traceable collaboration across organisations and similar bridges required to smooth over the road towards open data science.

The first of these enabling technologies is a Research Activity Identifier (RAiD); RAiD is an identifier for research projects and activities. It is persistent and connects researchers, institutions, outputs and tools together to give oversight across the whole research activity and make reporting and data provenance clear and easy. RAiD can be used to assist in reporting on institutional engagement with infrastructure providers and data output impact measures. Institutions can also use RAiD to locate data resources used by research projects and leverage that data into future projects through search and linking tools.

We do not aim to invent new standards and protocols however; we are keenly tracking work in SWORDv3, CERIF, OAI-PMH etc. and would like to compare notes with workshop participants about their experiences and opportunities for joint interoperability work.

**Cloud Infrastructure&Software Stacks for Data Science / 41**

## **Keeper service: Long term archiving tool for Max-Planck institutions on base of Seafile**

Vladislav Makarenko<sup>1</sup>

<sup>1</sup> *Max-Planck Digital Library*

**Corresponding Author(s):** makarenko@mpdl.mpg.de

Keeper is a central service for scientists of the Max Planck Society and their project partners for storing and archiving all relevant data of scientific projects. Keeper facilitates the storage and distribution of project data among the project members during or after a particular project phase and seamlessly integrates into the everyday work of scientists. The main goal of the Keeper service is to ensure sustainable and persistent access not only to the scientific project results but also to all data created during the research project, without any additional effort. All scientific projects stored in Keeper can be listed in the project catalog, which is only accessible within the Max Planck Society. The Keeper service fulfills the archiving regulations of the Max Planck Society as well as the German Research Foundation to ensure 'good scientific practice', takes care of project data after project ending and therefore is long term archiving (LTA) compliant. The specific features like Cared Data Certificate have been developed to support the LTA requirements.

This talk will cover a development evolution of the Keeper service: from target-setting to the building of service infrastructure as HA cluster on top of Seafile software. We will explain the Keeper use cases in the Max-Planck Society context and specific features developed to support them. An important

part of the talk will be dedicated to the long term archiving aspects of the project, institutional as well as technical.

## Cloud Infrastructure&Software Stacks for Data Science / 32

### Reproducible high energy physics analyses

Tibor Simko<sup>1</sup> ; Diego Rodriguez Rodriguez<sup>2</sup> ; Jose Benito Gonzalez Lopez<sup>1</sup>

<sup>1</sup> CERN

<sup>2</sup> Universidad de Oviedo (ES)

**Corresponding Author(s):** tibor.simko@cern.ch, diego.rodriguez.rodriguez@cern.ch

The revalidation, reuse and reinterpretation of data analyses requires having access to the original **virtual environments, datasets, software, instructions and workflow steps** which were used by the researcher to produce the original scientific results in the first place. The **CERN Analysis Preservation** pilot project is developing a set of tools that assist the particle physics researchers in structuring their analyses so that **preserving** and **capturing** the knowledge around analyses would lead to easier **sharing, reusing and reinterpreting** data. Assuming the full preservation of the original analysis environment, the user code and the computational workflow steps, the **REANA Reusable Analysis** platform enables one to launch **container**-based processes on the computing cloud (Docker, Kubernetes) and to rerun the analysis workflow jobs with new input. The REANA system aims at supporting several **workflow engines** (CWL, Yadage), several **shared storage** systems (Ceph, EOS) and **compute cloud** infrastructures (OpenStack, HTCondor). REANA was developed with the particle physics use case in mind and profits from synergies with general research data analysis patterns in other scientific disciplines such as life sciences.

## File Sync&Share Products for Home, Lab and Enterprise / 62

### Cubbit: a distributed, crowd-sourced cloud storage service

**Author(s):** Lorenzo Posani<sup>1</sup> ; Marco Moschettini<sup>1</sup>

**Co-author(s):** Alessio Paccioia<sup>1</sup>

<sup>1</sup> Cubbit

**Corresponding Author(s):** lorenzo.posani@cubbit.net, marco.moschettini@cubbit.net

Cubbit is a **hybrid cloud infrastructure** comprised of a network of p2p-interacting IoT devices (swarm) coordinated by a central optimization server. The storage architecture is designed to reverse the traditional paradigm of cloud storage from “one data center to rule them all” to “a small device in everyone’s house”.

Any IoT device that supports an Unix-based OS can join the swarm and share its storage space. Coordinated by our optimization server, the peers communicate via peer-to-peer data channels, collectively **virtualizing an object-storage service** over the swarm.

Through the employment of error-correcting-code algorithms and network monitoring, Cubbit achieves uptime and transfer performances comparable to traditional client-server solutions, while at the same time operating at **significantly-reduced managing costs** and environmental impact. The result is a **crowd-hosted and self-sustained data center** built over its very own final users, who can access cloud storage in exchange of local, unreliable physical storage.

In this talk we will present the general infrastructure of Cubbit and its main functionalities, with a particular focus on **environmental sustainability, security, and retrievability** of stored data on the distributed network.

**File Sync&Share Products for Home, Lab and Enterprise / 59**

## **Owncloud EFSS beyond just "scale": a glimpse into the future**

**Corresponding Author(s):** cschmitz@owncloud.com, felix.boehm@owncloud.com

ownCloud has been an excitingly successful service in the EFSS space since its breakthrough in 2013. Since customers deploy the solution in vastly different environments as public, private or hybrid cloud and utilizing different infrastructure components and identity providers, operational experience showed challenges with the previous design decisions.

This talk will reflect on the past experiences of the 100s of different large scale setups, challenges presented to operators and ownCloud project and company, and touch on specific topics related to the challenges in NREN/Research deployments.

Considering this active relection and synthesis of key learnings, the focus will shift towards an outlook on which lessons where learned and how that influences the timelines and plans in the years ahead. An important aspect will be reflection on actions taken to overcome technical debt and results obtained by the changes implemented since the new direction under the new CTO was undertaken and how this will proceed into the future.

**File Sync&Share Products for Home, Lab and Enterprise / 60**

## **Next steps in Nextcloud**

Frank Karlitschek<sup>1</sup>

<sup>1</sup> *Nextcloud*

**Corresponding Author(s):** frank@nextcloud.com

This talks covers the current state and functionality of Nextcloud. Especially the new and innovativ features of Nextcloud 12 and 13 are discussed and presented in detail. Examples are End 2 End encryption, collaboration and communication features and security and performance improvements. The second part of the talk presents the roadmap and strategic direction of Nextcloud for the coming releases. Another topic is the Nextcloud community. How to become a part of it, how to contribute and participate.

**File Sync&Share Products for Home, Lab and Enterprise / 34**

## **Space.Cloud.Unit - blockchain powered Open-Source Cloud-Space Marketplace**

Christian Sprajc<sup>1</sup>

<sup>1</sup> *PowerFolder*

**Corresponding Author(s):** sprajc@powerfolder.com

**Blockchain** is currently one of the hot topics. Developed as part of the cryptocurrency Bitcoin as a web-based, decentralized, public and most important all secure accounting system, this database principle could not only revolutionize the worldwide financial economy in the future; Blockchain is already an topic in electromobility, health care or supply-chain-management - just to name a few.

Our goal is to create a marketplace for cloud storage using blockchain technology. The project is to be financed via an Initial Coin Offering (ICO). The finished open source product will be named “Space.Cloud.Unit (SCU)”.

The concept behind **Space.Cloud.Unit** is to bring storage providers and storage consumers together in a virtual marketplace. The consumers, in need of storage, can specify and place their inquiries for storage size, number of copies, time frame, and availability. Storage providers, for their part, make corresponding offers; this can be both private individuals and professional storage providers. Inquiries and offers are automatically matched in the marketplace.

Likewise automated, a digital purchase and service contract (a **smart contract**) is then concluded, in which scope, redundancies and availability are defined and rewarded; the payment is made in SCUs. The provider must provide a technical proof that he has actually saved the data under the agreed conditions. Otherwise, a previously defined contractual penalty also filed on the Blockchain becomes due.

All data gets fully encrypted transmitted and stored to the provider . This happens off-chain, and thus independent of the blockchain and will be integrated into existing software solutions such as PowerFolder, Nextcloud, ownCloud, Seafile, Pydio, and more.

The **software package** developed for SCU will contain the following components:

- A **marketplace** that brings together storage providers and consumers
- The **blockchain** for generating coins and smart contracts
- The **apps** to integrate the solution into existing cloud storage software

The entire blockchain developed for this will be created as an **open source software**.

By integrating **Space.Cloud.Unit** as an app into already existing cloud storage software, the development effort and risk associated with similar projects on the market (such as Filecoin) is significantly reduced. Nevertheless, for the realization of the project additional developers and IT specialists are needed; These should also be financed through the ICO, as well as public relations, marketing and lawyers for the legal advice. As part of this ICO, users purchase Space Cloud Units (SCUs), which they can later use to buy storage on the marketplace.

Summing up the **key benefits** of Space.Cloud.Unit:

- Higher flexibility than traditional methods
- Precisely tailored conditions instead of rigid packages
- Standardized process and interfaces
- Easier access for providers compared to closed systems such as marketed vendors e.g. Amazon
- An additional way for cloud storage providers to offer their services
- Greater choice for storage consumers
- Financial reimbursement for users in case of non-compliance with agreed conditions (more safety through automated, smart contract based SLAs)
- Security through complete encryption and high decentralization
- Cost savings for the providers
- High transparency
- Automated execution

**File Sync&Share Products for Home, Lab and Enterprise / 10**

## **Seafile - what's new in 2017 and future roadmap**

Jonathan Xu<sup>1</sup>

<sup>1</sup> *Seafile Ltd.*

**Corresponding Author(s):** jonathan.xu@seafile.com

Seafile is an open source file sync and share solution. Thanks to its high performance, scalability and reliability, it has been successfully used by many organizations in Europe, North America and China.

In this presentation, we'll provide a review of Seafile's development in 2017, and what we plan to accomplish in the future. We'll also present a site report from China with heavy usage, demonstrating the scalability of Seafile in real world scenario, and also some performance tuning lessons learned.

**File Sync&Share Products for Home, Lab and Enterprise / 42**

## **[Poster] Pydio Next Gen : micro-services and open standards**

Charles du Jeu<sup>1</sup>

<sup>1</sup> *Pydio*

**Corresponding Author(s):** charles@pydio.com

On-premise EFSS is now an established market, and open source solutions have been key-players in the last couple of years. For many enterprises or labs, the need for privacy and handling large volumes of data are show-stoppers for using saas-based solutions. Still, for these users, the experience speaks by itself: even with good software, **it is hard to deploy a scalable and reliable system** serving massive amount of data, massive amount of users and desktop sync on top of that.

Historically developed in PHP, Pydio started to dig alternative technologies 2 years ago, by introducing a dedicated companion (Pydio Booster, presented at CS3 last year) that would lower the load on the PHP fronts (for uploads, downloads and websocket). Acknowledging the success of such a tool, Pydio Team decided 6 months ago to make a major leap: **rewrite the whole backend in Go, following a micro-services architecture** that would fit the demands of nowadays infrastructure.

The poster will present this new architecture, how the team took profit of her knowledge of Sync&Share to organize data inside micro silos, and the choices made in terms of interface with the outside world (APIs) to stick to the most advanced and open standards. This major release shall be made available at the end of Q1 2018.

**Future of Sync&Share (Panel Discussion) / 51**

## **Private, hybrid, public ... or how can we compete?**

Jakub Moscicki<sup>1</sup>

<sup>1</sup> *CERN*

**Corresponding Author(s):** jakub.moscicki@cern.ch

Over the last years we have witnessed a global transformation of the IT industry with the advent of commercial ("public") cloud services on a massive scale. Global Internet industry firms such as Amazon, Google, Microsoft massively invest in networking infrastructure and data-centers around the globe to provide ubiquitous cloud service platforms for any kind of service imaginable: storage, databases, computing, web apps, analytics and so on.

This clearly puts pressure on the on-premise services deployed using open source software components and begs the question: what is the future of the on-premise services in the long run? Can we compete with the giants and how? What are the main selling points for on-premise deployment and are they still relevant? Computer security, confidentiality of data, cost of ownership, functionality, integration with other on-site services? What are the strong points of the CS3 community? What are the weaknesses? Is fragmentation and diversity of the community a problem? What about ongoing EU-funded efforts to build an open and pervasive e-science infrastructure?

Do our end-users get functionality and reliability that can compete with commercial clouds? Could it be envisaged that research labs store their data externally and are still able to do data-intensive science efficiently? Does the current model scale into the future for the SMEs delivering technology and software for on-premise service deployments? Can institutions take a risk of lock-in with external cloud service providers? Or can such risks be mitigated?

In this presentation I do not provide answers. I try to ask the relevant questions.

**Future of Sync&Share (Panel Discussion) / 58**

## Future Panel Discussion

**Future of Sync&Share (Panel Discussion) / 15**

## Sync and Share is Dead. Long Live Sync and Share.

David Jericho<sup>1</sup>

<sup>1</sup> *AARNet*

**Corresponding Author(s):** david.jericho@aarnet.edu.au

“Sync and Share is Dead. Long Live Sync and Share.” discusses the increasing disinterest users have in simple file storage, Simple storage is a commodity service, with Google, DropBox, and other big players who can legitimately resolve concerns about data centre security, legal control, administration and audit, and standards compliance. The competitive advantage for any given data storage service is the application stack and API enablement on that data set, as well understand the the mobility of users.

Without the development of extended features, file storage becomes a race to the bottom in terms of both price and minimal difficulty of access.

AARNet has a program of development involving deploying the SWAN programmatic notebooks, enabling the Australian NeCTAR cloud compute infrastructure on the storage, video integrations for a national archive of cinema and television media, the raison d'être for S3 gateway services, the backup frameworks, high speed data transfer, and other enablers on CloudStor. We see this program as strategic to the service, as simple storage services will most probably not have a future in the Australian education and research community.

Further, there's a balance to be found between the extremes of the US Department of Energy's Energy Science's Science DMZ concept (which is part of AARNet's CloudStor design), and the increase in edge computing (also part of CloudStor's design). The goal is to find the balance between bringing data to the user, and bringing getting data to the compute.

The goal of this presentation is to inspire conversation and collaboration around the progress forward with these platforms, and how the research and education communities can put value on top of data stores to avoid irrelevance.

**Future of Sync&Share (Panel Discussion) / 25****The future of File Sync and Share**Frank Karlitschek<sup>1</sup><sup>1</sup> *Nextcloud***Corresponding Author(s):** frank@nextcloud.com

We are heading into a world where the files of most users are hosted by 4 big companies. This is the case for most home users, companies but also education and research institutions. If we want to keep our sovereignty over our data, protect our privacy and prevent vendor lock-in then we need open source self hosted and federated alternatives.

A new challenge is the increasing blending of application hosting and storage as seen at Office 365 and Google Suite. This has the danger to lead to a very strong vendor lock-in.

This talk will discuss the ongoing trends in these areas and possible solutions. It will also give an overview of newest Nextcloud features and the long term roadmap to provide an alternative to centralised services.

**Scalable Storage Backends for Cloud and HPC: Foundations / 43****EOS the CERN disk/cloud storage for science**Luca Mascetti<sup>1</sup><sup>1</sup> *CERN***Corresponding Author(s):** luca.mascetti@cern.ch

EOS, the high-performance CERN IT distributed storage for High-Energy Physics provides now more than 250PB of raw disks and supports several work-flows from LHC data-taking and reconstruction to physics analysis.

The software is developed at CERN since 2010, is available under GPL license and it is also used in several external institutes and organisations.

EOS is the key component behind the success of CERNBox, the CERN cloud synchronisation service for end-users which allows syncing and sharing files on all major mobile and desktop platforms aiming to provide offline availability to any data stored in the EOS infrastructure.

Today EOS provides multiple views/protocols to the same namespace and storage backend - via the OwnCloud synchronisation client, via xrootd protocol for physics data analysis applications access, as a mounted filesystem, with latency optimised wide-area file access protocols or via SAMBA endpoint for Windows clients.

In addition it is possible to interact with the system using Jupyter Notebooks provided at CERN by the SWAN (Service for Web based ANalysis) platform which offers to scientists a web based service for interactive data analysis.

We report on our experience with this technology and applicable use-cases, also in a broader scientific and research context and its future evolution with highlights on the current development status and future roadmap.

**Scalable Storage Backends for Cloud and HPC: Foundations / 31****Cynny Space Cloud Object Storage: integrated hardware-software solution developed specifically for ARM® architecture**



Stefano Baldi<sup>1</sup> ; Andrea Marchi<sup>1</sup> ; Stefano Stalio<sup>2</sup>

<sup>1</sup> *Cynny Space*

<sup>2</sup> *INFN*

**Corresponding Author(s):** s.baldi@cynnyspace.com

The purpose of the presentation we'll propose during the CS3 conference in Krakow is to highlight the technological features of Cynny Space's cloud object storage solution and the results of performance and usability for a sync & share use case.

#### 1) Software specifically designed for ARM® architecture

The object storage solution is specifically designed and developed on storage nodes composed of fully-equipped ARM® based micro-servers and a single storage units. The 1:1 micro-server to storage unit ratio and the File System optimized for ARM® delivers optimal levels of scalability, resilience, reliability and efficiency.

#### 2) Peer-to-peer, independent storage network

Every storage node is fully independent and interchangeable, providing a storage solution with no single point failure. To maximize the parallelization of the operations, every node is responsible for some chunks (and not the whole file). The use of Dynamic Hash Tables and a distributed Swarm Intelligence on a peer-to-peer network minimizes the cooperation overhead and allows a high level of scalability, fault-tolerance and clusterization.

#### 3) Fully integrated hardware and software solution

The software upon which the system is built overcomes the limits that we meet by using a general-purpose solution since the latter is not optimized to work with a limited hardware and a large number of servers. Moreover, a software designed for a specific hardware configuration allows to optimize and parametrize all of the services involved (Operating System, File System, Database, internal communication protocols), and it permits to be very flexible towards several access protocols.

#### 4) Use case example: Cynny Space & NextCloud

A use case of the solution is the integration as an external storage service for NextCloud's sync & share client. The results will be presented during the CS3 conference, with highlights on performance, conflict resolution, federated cloud and multiple protocols accessibility.

## Scalable Storage Backends for Cloud and HPC: Foundations / 22

### dCache as a backend for cloud storage

**Author(s):** Peter van der Reest<sup>1</sup>

**Co-author(s):** Gerd Behrmann<sup>2</sup> ; Patrick Fuhrmann<sup>1</sup> ; Paul Millar<sup>1</sup> ; Tigran Mkrtchyan<sup>1</sup>

<sup>1</sup> *DESY*

<sup>2</sup> *NDGF*

**Corresponding Author(s):** peter.van.der.reest@desy.de

Over two years Data-Cloud team at DESY uses dCache as a backend storage for the ownCloud instance used in a production. As being a highly scalable storage system, dCache is widely used by many sites to store hundreds of petabytes of scientific data. However, the cloud-backend usage scenarios have added new requirements, like high availability and downtime less updates any software or hardware components.

Since version 2.16 dCache team have made a big effort to move towards redundant services in dCache to remove a single point of failure. Moreover, the low level UDP based service discovery is replaced with widely-adopted Apache-ZooKeeper - a persistent, hierarchical directory service with

strong ordering guarantees. As being itself a fault-tolerant service with strong consistency guarantee ZooKeeper becomes the natural place to keep a shared state or take a role of service coordination.

With this presentation we will show technical solutions applied to achieve a highly-available, fault-tolerant dCache deployment. We will touch some aspects of ownCloud and dCache integration. Finally, we want to point cloud software provides to the shortcomings of currently available solutions that limits functionality that scalable storage system like dCache can provide.

## Scalable Storage Backends for Cloud and HPC: Foundations / 33

### Onedata Virtual Filesystem for Hybrid Clouds

**Author(s):** Lukasz Dutka<sup>None</sup>

**Co-author(s):** Bartosz Kryza<sup>1</sup> ; Michał Orzechowski<sup>2</sup> ; Lukasz Opiola<sup>3</sup> ; Michał Wrzeszcz<sup>3</sup>

<sup>1</sup> ACC Cyfronet-AGH

<sup>2</sup> AGH University of Science and Technology, Academic Computer Centre Cyfronet AGH, Krakow, Poland

<sup>3</sup> ACK Cyfronet AGH

**Corresponding Author(s):** lukasz.dutka@gmail.com

Onedata is a complete high-performance storage solution that unifies data access across globally distributed environments and multiple types of underlying storages, such as NFS, Lustre, GPFS, Amazon S3, CEPH, as well as other POSIX-compliant file systems. It allows users to share, collaborate and perform computations on their data.

Globally Onedata comprises of: Onezones, distributed metadata management and authorisation components that provide entry points for users to access Onedata; and Oneproviders, that expose storage systems to Onedata and provide actual storage to the users. Oneprovider instances can be deployed, as a single node or a HPC cluster, on top of high-performance parallel storage solutions with ability to serve petabytes of data with GB/s throughput.

Onedata introduces the concept of Space, a virtual directory, owned by one or more users. The Spaces are accessible to users via an intuitive web interface, which allows for Dropbox-like file management and file sharing, Fuse-based client that can be mounted as a virtual POSIX file system, or REST and CDMI standardized APIs. Onedata does not provide users with any physical storage, each Space has to be supported with a dedicated amount of storage by one or more providers, who are running Oneprovider component.

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. Onedata user groups are particularly suitable for managing communities of users that wish to share common resources. Access to Spaces is managed by flexible authentication and authorisation methods such as: access tokens, OpenID, X.509 certificates and Macaroons.

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 available locally, the remote data is fetched in real-time from remote facility, using a dedicated highly parallelized dedicated protocol with block-level data transfer, that also provides common features such as pre-staging, data migration and data replication.

Currently Onedata is used in Indigo-DataCloud and PLGrid projects as a federated data access solution, aggregating either computing centres or whole national computing infrastructures; and in EGI-Engage, where it is a basis for Open Data Platform prototype for dissemination and exploitation of open data sets.

Open Data Platform prototype is capable of publishing Spaces as data containers with assigned globally unique identifier, such as DOI (Digital Object Identifier) or PID (Persistent Identifier), to selected communities or public open data portals. It features metadata management editor with ability to add custom metadata-schemes to open data containers, files, and folders. Detailed data management plan, can be defined for individual containers characterising the overall data lifetime: from generation, generated data formats, curation, licensing and long term preservation policies.

## Scalable Storage Backends for Cloud and HPC: Foundations / 7

### iRODS: Data-Centric, Metadata-Driven, Data Management

Jason Coposky<sup>1</sup>

<sup>1</sup> *iRODS Consortium*

**Corresponding Author(s):** jasonc@renci.org

iRODS is Open Source Data Management that can be deployed seamlessly onto your existing infrastructure, creating a unified namespace, and a metadata catalog of all the data objects, storage, and users on your system. iRODS allows access to distributed storage assets under the unified namespace and frees organizations from getting locked into single-vendor storage solutions. iRODS can represent data at rest in object, tape, and POSIX filesystems all within the same logical collection. Within the same catalog, iRODS provides the ability to annotate every data object, logical collection, storage resource, and user in the namespace. Through the use of this metadata these entities become actionable and may be operated upon by the integrated rule engine framework. The rule engine framework allows for any operation within your iRODS zone to be a trigger or hook for your code. This affords the creation of automated data management policy which may prevent the operation, provide context to the operation, log the operation or many other use cases. Additionally, iRODS provides the ability to federate any number of zones, allowing for the sharing of not only data across administrative boundaries, but the sharing of infrastructure as well.

Given these features iRODS provides a number of capabilities: automated data ingest, storage tiering, compliance, indexing, auditing, data integrity, provenance, and publishing.

This talk will cover the four core competencies of iRODS, which afford the implementation of these many capabilities. We will then cover emerging data management design patterns as well as existing use cases deployed in production. Finally, we will cover the iRODS software roadmap as well as an overview of the iRODS Consortium.

## Scalable Storage Backends for Cloud and HPC: Integration / 40

### Improving swift as Owncloud backend to scale

Ricardo Makino<sup>1</sup> ; Thomas Diniz<sup>2</sup> ; André Castro<sup>2</sup> ; Guilherme Maluf<sup>2</sup> ; Rodrigo Azevedo<sup>1</sup>

<sup>1</sup> *RNP*

<sup>2</sup> *Anolis*

**Corresponding Author(s):** ricardo.makino@rnp.br, rodrigo.azevedo@rnp.br

The National Education and Research Network (RNP) is an organization that plans, designs, implements and operates the national network infrastructure under contract with the Ministry of Science, Technology, Innovation and Communications (MCTIC). A current government program includes five ministries - MCTI, Education (MEC), Culture (MinC), Health (MS) and Defense (MD), and annually define the objectives of the contract and its plan.

The increasing production of scientific data (eg, environmental monitoring, biodiversity databases, a variety of simulation and visualization systems such as climate forecasting, high physical energy data collection, astronomy and cosmology), cultural datasets and others, brings the need for a scalable, sustainable and high availability IT infrastructure to support these demands, and these facilities must be located in a distributed manner and in locations offering telecommunications, energy and safety services, as well as appropriate physical space and infrastructure.

In addition to the question of where these data are stored and processed, there is a need to create different institutions in research programs, such as the Large-scale Biosphere-Atmosphere Experiment in the Amazon (LBA), which is composed of 280 institutional and international, with about 1400 Brazilian scientists, 900 researchers from Amazonian countries and 8 European nations and from American institutions, aiming to study and understand the climate and environmental changes in the Amazon. In this type of communities sharing information securely and in compliance with legislation, whether from data collected from sensors, or in the production of articles or articles, is vital, and a cloud file synchronization and sharing solution meets this demand.

edudrive@RNP is cloud file synchronization and sharing offered by RNP to your community, it allows users to sync your data between your desktops, notebooks, smartphones and tablets and share this data with others users of the service or not, supporting researchers, teachers and students in research projects and during his academic studies.

The service is developed by RNP, in partnership with Anolis IT and is based on Owncloud software, which acts as the frontend of the service offering a web portal, desktop and mobile clients that users uses to synchronize and share their files. In addition, Openstack Swift is also used as a multi-tenant object storage backend, which provides high scalability, cost savings and significant resiliency, as a Software Defined Storage (SDS). Additionally, one of the key service requirements it's the integration with Shibboleth-based SAML federation for user authentication and authorization.

During the development of the service was identified a problem in the way that Owncloud use Openstack Swift as a storage backend, which caused a great slowness in file upload, download and delete operations when the number of files in the service grows.

This problem occurs in the connection between Openstack Swift and Owncloud, because when configuring Openstack Swift as the Owncloud storage backend it is mapped a single tenant and a single container from Openstack Swift where all the data are stored, it do not like a problem with a few number of files, but when the number of files grows, the search and replication activities of this data become slow, this occurs because the growth of sqlite metadata databases, which are synchronized between the storage nodes in each upload and delete operations, and this performance issue increase when you have a geographically separated infrastructure, negatively impacting the use of the service.

In addition, another important issue identified is the security of the information stored in the service, because all data of all users are stored in the same tenant and container, and this do not guarantee a logical segregation of this data, and in case of a leak of the credentials of the tenant and container used in the backend all the data stored on this can be exposed.

Based on this performance issue, it was necessary a development effort, in order to balance that storage backend load and use all features that is offered by this type of storage backend properly. To accomplish this, for each institution (university, research center, etc) was mapped as a tenant inside the storage backend, and each user was mapped as a single container inside your institution (tenant).

With this new mapping, the performance problems were solved, in addition, the data of the institutions and users of the service were segregated in a more appropriate way, bringing more security to the service. All changes were made to OwnCloud's OpenStack Object Storage plug-in, more specifically in the swift class, with changes to the default flow of user access to the service, and the execution of operations such as uploading or downloading of files.

All this new flow, which originally was controlled by OwnCloud, is now controlled by the Federated Access Control System (FACS), managing all the institutions which are authorized to access the service in an identity federation and creating the tenant of each institution based on its identifier in the identity federation and the user container during his first access, based on its unique identifier in the identity federation (EPPN - EduPersonPrincipalName). After that, all user data is saved in its own container within the tenant of your institution.

## Scalable Storage Backends for Cloud and HPC: Integration / 27

### ownCloud and objectstore

Thomas Müller<sup>None</sup>

**Corresponding Author(s):** thomas.mueller@owncloud.com

To allow better scalability of ownCloud in large installations we spent some time to leverage the ownCloud integration with s3 based objectstores like Ceph and Scalify.  
At the ownCloud Conference 2017 we have been presenting the vision where to go.  
At CS3 we will present the results!

## Scalable Storage Backends for Cloud and HPC: Integration / 24

### Scaling storage with cloud filesharing services

Gregory Vernon<sup>1</sup>

<sup>1</sup> SWITCH edu-ID

**Corresponding Author(s):** gregory.vernon@switch.ch

SWITCH has been running cloud-based filesharing services since 2012, starting with an experiment where we hosted FileSender in the Amazon Cloud. After this experience, we decided to build a cloud service for ourselves, SWITCHengines, which runs upon an OpenStack Infrastructure. The challenge with our SWITCHengines infrastructure and filesharing is the Ceph storage that we use for our user files. Today we successfully host more than 30,000 users on our SWITCHdrive service.

In this presentation, we will look at the following:

- The history of SWITCH's cloud based filesharing services, with a special focus on SWITCH's ownCloud service SWITCHdrive
- Our experiences with storage backends, with an strong emphasis on Ceph
- What worked well for us
- What challenges we faced and how we overcame them
- Future challenges and solutions we are currently investigating

## Scalable Storage Backends for Cloud and HPC: Integration / 39

### Sync & share solution to replace HPC home?

Maciej Brzezniak<sup>1</sup> ; Krzysztof Wadówka<sup>1</sup> ; Marcin Pośpieszny<sup>1</sup> ; Piotr Brona<sup>1</sup> ; Radosław Januszewski<sup>1</sup>

<sup>1</sup> PSNC Poznan Poland

**Corresponding Author(s):** maciekb@man.poznan.pl

Handling 100s of Terabytes of data at the speed of 10s of GB/s is nothing new in HPC. However, high performance and large capacity of the storage systems rarely go together with their ease of use. HPC storage systems are specifically difficult to access from outside the HPC cluster. While researchers and engineers tolerate the fact that they need to use rigid tools and applications such as textual SFTP client or globus-url-copy and SSH/text consoles in order to store, retrieve and manipulate the data

within HPC system, such inconvenience does not support productivity. Oposite, the burden related to exchanging data with HPC systems makes learning curve for HPC systems adopters steep and is often a source of errors and delays in implementing and running computing workflows.

Moreover, in the classical HPC storage architecture, the data flow to and from the HPC system has to be explicitly managed by users and their applications or workflow systems. For instance, downloading the computations results from the cluster to the user workstation or mobile device has to be triggered manually by user or steered by a mechanism that is synchronised with computing jobs management system. Compared to ease of use of cloud storage systems, such a data flow control reminds 80's or 90's of the previous century.

In our presentation we will demonstrate how a high-performance and robust sync & share storage system based on well-optimised synchronisation mechanism and equipped with functional and well-optimised user tools such as GUI and virtual drive can provide efficient, convenient and reliable interface among to HPC storage systems. We will also show that such an interface is also capable of handling really large volumes of data at proper speed as well as offer relevant responsiveness to user I/O operations.

The discussed solution is based on Seafile, a scalable and reliable sync&share software deployed in PSNC's HPC department servers and storage infrastructure. Seafile has been for years a basis of sync & share services that HPC Department at PSNC offers internally and to academic community in Poland since 2015 through the PIONIER network. While, this service was initially targeted mostly regular users who store their documents, graphics and other typical data sets, it attracted also researchers. These power users started to challenge our systems with 10s of Terabytes of data. However, the real challenge was yet to come.

In autumn 2017, we started a pilot deployment of Seafile for research groups within CoeGSS EU project. They use our synchronisation and sharing solution as an equivalent of the home directory in HPC systems, in order to store, access and exchange results of the simulations rrun in the PSNC's flagship HPC cluster 'Eagle' (1,4 PFlops, #172 at Top500 on Nov 2017).

While the computations are performed using high-performance Lustre filesystem attached through Infiniband to the HPC cluster, the input and output data are transferred, accessed and synchronised over regular network using Seafile. This solution provides ease of access to data sets, as users can interact with their files stored in the HPC system using Web interface, desktop GUI applications and virtual drive solution available accross Windows, Linux and MacOS. Automation of the data flow is also ensured as the data can be selectively synchronised with the HPC storage as they are produced by computing jobs in the cluster. In the same time high performance of storage and retrieval is possible as Seafile scales to many MB/s of transmission througput and seamlessly serves lots of small files.

## Scalable Storage Backends for Cloud and HPC: Integration / 16

### Contribution to a Dialog Between HPC and Cloud Based File Systems

Jean-Thomas Acquaviva<sup>1</sup>

<sup>1</sup> *DDN Storage*

**Corresponding Author(s):** [jacquaviva@ddn.com](mailto:jacquaviva@ddn.com)

IME I/O Acceleration layer is one of the latest efforts of DDN in order to satisfy the never ending needs for performance of the HPC community. We propose to discuss some of the latest advancements of IME product in respect to the larger evolution of Software Defined Storage has it is observed outside of the HPC market.

The arrival of the Flash has pushed existing HPC file systems to their limits. Therefore, IME could be seen as a response to the drastic reduction of storage latency. It could be argued that the only way to adapt from a reduction of latency by about 1000 is a complete design overhaul.

However, a driving force since IME inception has been the emerging requirements observed in the HPC community. The evolution of usages, the needs for more intense data sharing or collaborative access, the necessity of implementing not only file access but work-flows and specificities of HPDA were keys in the design decision of the IME.

Similarly, cloud storage is passing now through a challenging transformation, looking to leverage cloud from a simple storage backup to a backend solution for running traditional applications. The changing paradigm from a sequential access to a large and cheap repository to a latency to a performance sensitive and predictable environment requires evolutions close to HPC storage solutions.

Taking the HPC IO500 as an illustration, we will debate of the relevance of several performance metrics, the way they reveal underlying mechanisms, and how they are correlated to technological orders of magnitudes. We will discuss how these technological parameters impact both metadata management and data payload capability.

Conducting a similar exercise for Cloud based file systems, we will analyze existing similarities, emphasize convergences as well as differentiators between Cloud based FS and HPC oriented storage software.

We will conclude with some possibilities toward shared futures.

## Sharing and Collaborative Platforms / 54

### New sharing interface for SWAN

**Corresponding Author(s):** diogo.castro@cern.ch

This presentation gives details and demonstrates the new SWAN sharing interface. See also: “SWAN: Service for Web-based Analysis” in “Cloud Infrastructure&Software Stacks for Data Science” session.

## Sharing and Collaborative Platforms / 61

### Interoperability with sharing: Open Cloud Mesh status and outlook

**Corresponding Author(s):** peter.szegedi@geant.org

As part of the collaboration effort between GÉANT, CERN and ownCloud, in January 2015, an idea (aka. OpenCloudMesh, OCM) has been initiated to interconnect the individual on-premises 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, in the first place, 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.

The ultimate goal is to design and implement this protocol as an open federated sharing standard that can be deployed by different file-based sync and share services and providers. To achieve this, GÉANT should start engaging with a wide variety of stakeholders as well as participate in the related global standardisation efforts.

The OCM Project is implemented in phases:

- Phase I. (Oct 2015 – Feb 2016): Demonstrating the first working prototype of the OCM protocol (API v1.0 BETA) functionally working between two separate administrative ownCloud domains (i.e.

between two NRENs).

- Phase II. (Feb 2016 – June 2016): Demonstrating the OCM protocol first implemented and working between two independent sync&share software vendors' domains. A live demonstration happened at TNC'16 in June 2016.
- Phase III. (June 2016 – Jan 2017): Creating a protocol description/definition that is compliant, described, neutral, modular, minimal, secure and robust in order to be implemented by any vendors.
- Phase IV. (Oct 2017 – May 2018): Aims at paving the way towards standardization. Explore patent and IPR issues, as well as the potential fora for initiating the standardization discussion. Need a reference installation (proxy) fully compliant with the latest specs.

Phase IV. is currently running. Contributions are very welcome!

## Sharing and Collaborative Platforms / 55

### Document Processing and Collaboration without Boundaries

**Author(s):** Olga Golubeva<sup>1</sup>

**Co-author(s):** Galina Godukhina<sup>2</sup>

<sup>1</sup> *Ascensio System SIA*

<sup>2</sup> *Ascensio System SIA*

**Corresponding Author(s):** olga.golubeva@onlyoffice.com

Global academic community tends to show more interest in using cloud technologies for scientific data processing, which is determined by the need for quick joint access to the data.

This presentation will deal with the question of the convenient and effective cloud editing of documents as the main form of storing and exchanging the information.

ONLYOFFICE, a project by Latvian software development company Ascensio System SIA, is aimed at creating an innovative office suite for comprehensive online work with text documents, spreadsheets and presentations.

The presentation will cover the following key problems of online editing and methods of solving them using HTML5 Canvas:

- Limited editing toolset;
- Insufficient formatting quality;
- Weaknesses in compatibility with popular formats;
- Unequal display of content depending on browser or device;
- The need in complete collaborative work on documents;
- Resources for extending editing functionality.

The separate topic will be devoted to data protection methods and mechanisms realized in ONLYOFFICE web solutions.

We as well admit the importance of integrating this functionality in familiar instruments for working with documents. The final part of the presentation will touch upon integration with popular file sync&share platforms.

## Sharing and Collaborative Platforms / 48

### Collabora Online: Real-time Collaboration on Documents

Jan Holesovsky<sup>1</sup>



<sup>1</sup> *Collabora Productivity*

**Corresponding Author(s):** kenny@collabora.com

Come and hear about what is Collabora Online and how it integrates into many File Sync&Share Products to create a powerful, secure, real-time document editing experience. Hear about the improvements over the last year, catch a glimpse of where we are going next, and hear how you can get it integrated into your product - if you haven't integrated it yet.

**Sharing and Collaborative Platforms / 47**

## **From Sync & Share to Collaborative Platforms: the CERNBox case**

Giuseppe Lo Presti<sup>1</sup>

<sup>1</sup> *CERN*

**Corresponding Author(s):** giuseppe.lopresti@cern.ch

In this contribution, the evolution of CERNBox as a collaborative platform is presented.

Powered by EOS and ownCloud, CERNBox is now the reference storage solution for the CERN user community, with an ever-growing user base that is now beyond 12K users.

While offline sync is instrumental for such a widespread usage, online applications are becoming more and more important for the user community: to this end, the integration of Microsoft Office Online will be presented, which complements the existing offer for Physics analysis (SWAN).

Online applications on the other end open up a new dimension for a sync & share storage system. Collaborative authoring of documents is now becoming standard practice with public cloud services, and within CERNBox we are looking into several options: from collaborative editing of shared office documents with different solutions (Microsoft, OnlyOffice, Collabora) to integrating mark-down as well as LaTeX editors, to exploring the evolution of Jupyter Notebooks towards collaborative editing, where the latter leverages on the existing SWAN Physics analysis service.

The vision is thus to aim towards a 'web desktop', where the wider scientific community is enabled to process, share, and collaboratively author data from the web browser.

**Sharing and Collaborative Platforms / 53**

## **[Discussion] OCM plans and use-cases**

Peter Szegedi<sup>1</sup>

<sup>1</sup> *GÉANT*

**Corresponding Author(s):** peter.szegedi@geant.org

This panel discussion session will be focusing on the actual use cases that can drive the adoption and further development of the OCM protocol. Panellists will be requested to provide their views and vision for the future with regard to interoperability between private cloud domains.

## Sharing and Collaborative Platforms / 29

### PowerFolder Open Cloud Mesh API (v0.0.3) integration

Jan Wiegmann<sup>None</sup>

**Corresponding Author(s):** jw@jwiegmann.de

Open Cloud Mesh (OCM) is a joint international initiative under the umbrella of the GÉANT Association that is built on the 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. OCM defines a vendor-neutral, common file access layer across an organization and/or across globally interconnected organizations, regardless of the user data locations and choice of clouds.

The concept of the OCM integration within PowerFolder aims to combine installed Cloud Services on several locations to seamlessly share and sync files between end-users of those services.

In this context, we will present how the OCM standard (defined at <https://rawgit.com/GEANT/OCM-API/v1/docs.html>) has been integrated into the PowerFolder server application and how the corresponding OCM API endpoints are requested. PowerFolder has also made some individual adaptations to the OCM API endpoints in order to enable concepts that were previously not dealt with, such as the transfer of permissions between two or more services, discovering accounts in an federated network and building trust relationships between different locations.

The initiative presented here also has the following aims:

- Requesting OCM v0.0.3 endpoints in PowerFolder.
- Sharing permissions with OCM v0.0.3.
- Building mutual trust relationships between services: Request validation!
- Account discovery between services: Federated login.
- The pf\_classic protocol as an alternative to WebDAV within the OCM standard.

The presentation of these concepts is intended to advance the further development and finalization of the OCM standard.

## Sharing and Collaborative Platforms / 30

### Seamless Sharing in the ownCloud Platform

Patrick Maier<sup>None</sup> ; Tom Needham<sup>None</sup>

**Corresponding Author(s):** pmaier@owncloud.com, tom@owncloud.com

Current sharing in ownCloud does not allow seamless access of shared data. Media disruptions and inefficient communication methods reduce productivity for teams through a lack of information. Sharing 3 introduces a new bi-directional request-accept flow for streamlining collaboration within the ownCloud platform. This gives users further control over their data, allows them to request access to other data, and receive notifications of requests for collaboration.

Site reports / 63

## Experience of deployment of the SWAN system on the local resources of the St. Petersburg State University

**Author(s):** Andrey Erokhin<sup>1</sup>

**Co-author(s):** Andrey Zarochentsev<sup>1</sup>

<sup>1</sup> *St Petersburg State University (RU)*

**Corresponding Author(s):** andrey.erokhin@cern.ch

The report focuses on the deployment of the CERN SWAN-like environment on top of existing EOS storage. Our setup consists of a local cluster with Kubernetes to run JupyterHub and single-user Jupyter notebooks plus a dedicated server with CERNBox. The current setup is tested by our colleagues in the Laboratory of ultra-high energy physics of the St. Petersburg State University, but there are plans to adapt the system to other SPbU departments. Some details, problems and solutions are described.

Site reports / 45

## CERNBox Service

Hugo Gonzalez Labrador<sup>1</sup>

<sup>1</sup> *CERN*

**Corresponding Author(s):** hugo.gonzalez.labrador@cern.ch

CERNBox is a cloud synchronisation service for end-users: it allows synchronising and sharing files on all major desktop and mobile platforms (Linux, Windows, MacOSX, Android, iOS) aiming to provide universal access and offline availability to any data stored in the CERN EOS infrastructure. With 12000 users registered in the system, CERNBox has responded to the high demand in our diverse community to an easily and accessible cloud storage solution that also provides integration with other CERN services for big science: visualization tools, interactive data analysis and real-time collaborative editing.

We report on our experience managing the service and on the underlying technology that will allow us to move towards a future unified CERN Home directory service.

Site reports / 21

## Fault tolerant Cloud infrastructure

**Author(s):** Peter van der Reest<sup>1</sup> ; Quirin Buchholz<sup>1</sup>

**Co-author(s):** Birgit Lewendel<sup>2</sup> ; Patrick Fuhrmann<sup>1</sup> ; Ralf Lueken<sup>1</sup> ; Tigran Mkrtchyan<sup>1</sup>

<sup>1</sup> *DESY*

<sup>2</sup> *Deutsches Elektronen-Synchrotron (DE)*

**Corresponding Author(s):** peter.van.der.reest@desy.de

Over two years Data-Cloud team at DESY provides a reliable ownCloud instance for a selected set of users. While service is still officially in a pilot phase, it's has the same support and priority level as any other production services provided by the IT group. However, before removing it's "beta"

status some extra actions have to be taken: the instance must be fault tolerant and allow software and hardware updates without any downtime. Moreover, an extra disaster recovery plan have to be defined to make users data to be safe, even if data centre becomes unavailable - the end-users store the most precious data - the documents!

To achieve this ambitious goal, all involved components have to provide the same level of stability and data protection. In this presentation we will share our deployment setup, including HA-dCache installation, which is used as the storage backend, database replication with automatic fail-over configuration, load-balanced set of ownCloud servers and data offloading to an external location for disaster recovery. The sites, which still planning deployment with a similar level of availability, can make use of our experience to build and provide fault tolerant service to their customers.

**Site reports / 35**

## Sync and Share at SURF

Ron Trompert<sup>None</sup>

**Corresponding Author(s):** ron@sara.nl

The past year we were able add a number of extra features to the SURFdrive service in order to make it more attractive to users and institutes and there is more to come. Another thing that we have observed that several institutes and research groups have a need for a SURFdrive in a version more tailored to their needs. SURFdrive is fine as it is but it is a one size fits all solution. Different researchers have different needs and we will cater for that with a separate service that we are currently setting up. This service also provides the capability to serve data to other compute and archiving resources at SURFsara.

**Site reports / 18**

## Experience with taking over a service running ownCloud on virtualized infrastructure (Openstack/CEPH), and surviving the experience

Gregory Vernon<sup>1</sup>

<sup>1</sup> SWITCH edu-ID

**Corresponding Author(s):** gregory.vernon@switch.ch

In the summer of 2017, I inherited SWITCHdrive, SWITCH's ownCloud-based filesharing system. SWITCHdrive is a fairly complex service including a set of docker based microservices. I will describe the continuing story of our experiences with running such an environment. We had some interesting developments in tuning our MariaDB/Galera database infrastructure, and we have also greatly improved the performance with our CEPH cluster. Additionally, I will also tell the exciting story of my baptism of fire with taking over responsibility for running the service.

Topics covered will include:

- Experiences with our NFS server storage backend, and what we are considering for the future
- Our continuing saga with CEPH storage
- Database improvements over the past year
- Implementing Docker into our environment

Site reports / 57

## Site report survey summary

Dr. Tilo Steiger<sup>1</sup>

<sup>1</sup> *ETH Zuerich*

**Corresponding Author(s):** steiger@id.ethz.ch

Technology: deployment, testing and optimization / 23

## Running Nextcloud in docker/kubernetes: Load testing results

D Pennings<sup>1</sup> ; Martijn Kint<sup>None</sup>

<sup>1</sup> *360 ICT*

**Corresponding Author(s):** d.pennings@360ict.nl, martijn@360ict.nl

We started looking at the site reports of CS3 with the goal of designing a large OwnCloud/Nextcloud solution. We learned that the main product used for sync and share is Nextcloud/Owncloud and looking at that large user base implementations, saw that most are implementations are large monolithic installs. The site reports also showed that these large installs have some weaknesses like scaling with 10.000's of concurrent users, database scalability and the need for storage distributed over multiple datacenters. Even the Nextcloud Global Scale presented on CS3-2017 is based on multiple large monolithic installations to further scale out while maintaining the large installs and their complexity.

We tried an alternative approach with lots of small installations based on microservices and presented the design on CS3-2017. The idea is to separate the application from the data and infrastructure, for which microservices are a good candidate. In our opinion this reduces the overall complexity of the infrastructure and should scale horizontally. In this talk we want to present our preliminary load test results on how scalable this design is.

Technology: deployment, testing and optimization / 12

## Down the Rabbit Hole: Adventures in Bug-Hunting

Crystal Chua<sup>1</sup>

<sup>1</sup> *AARNet*

**Corresponding Author(s):** crystal.chua@aarnet.edu.au

This talk covers a journey through fuzz-testing CERN's EOS file system with AFL, from compiling EOS with afl-gcc/afl-g++, to learning to use AFL, and finally, making sense of the results obtained. Fuzzing is a software testing process that aims to find bugs, and subsequently potential security vulnerabilities, by attempting to trigger unexpected behaviour with random inputs. It is particularly effective on programs or libraries that handle file or input parsing as these areas are often susceptible to buffer overflow or other vulnerabilities, for example libxml2, ImageMagick and even the Bash shell. This approach to automated bug discovery dates back to the early 1950s, and has been steadily gaining popularity in recent years as fuzzing tools become more sophisticated - and more importantly, easier to use. Of particular note is american fuzzy lop (AFL), a genetic fuzzer written by Michal Zalewski (lcamtuf@google), which has seen massive success - to date, it has been used in

the discovery of over three hundred CVEs and many other non-exploitable bugs, in programs such as firefox, nginx, clang/llvm, and irssi. Initial experimental fuzzing attempts against EOS with AFL have been promising, and it is hoped that further efforts to establish a process around this will be greatly beneficial in the long run.

**Technology: deployment, testing and optimization / 17**

## Troubleshooting Relational Database Backed Applications

Gregory Vernon<sup>1</sup>

<sup>1</sup> SWITCH edu-ID

**Corresponding Author(s):** gregory.vernon@switch.ch

Managing the database where you store your application data is always an interesting challenge. As the scale of your service grows, so does the challenge of keeping a healthy database service. However with just a few tools and techniques it is possible to implement some serious performance improvements with just a little bit of effort. Using the performance tools included with MariaDB, at SWITCH we were able to significantly reduce our load on our a few of our MariaDB databases and even retire some of our database servers. In this presentation as an example we will show you how we improved the performance of our Owncloud database. However, these methods are applicable to any database-backed application, and with databases other than MariaDB. This session will equip you with the ability to identify issues with your database and give you ideas as to what to do to fix these issues.

**Technology: deployment, testing and optimization / 14**

## Rocket to the Cloud – A Faster Way to Upload

**Author(s):** Michael D'Silva<sup>1</sup>

**Co-author(s):** David Jericho<sup>1</sup>

<sup>1</sup> AARNet

**Corresponding Author(s):** michael.dsilva@aarnet.edu.au

Rocket is the first attempt at handling one of the particular problems that other tools have failed to solve. This presentation will demonstrate AARNet's experiences and tools used high-speed data transfers of different kinds of research data.

The research community in Australia is spread far and wide geographically, resulting in some cases to be physically far from one of our three CloudStor sites spread across the country. In addition, the data sets researchers store can be very varied, ranging from ephemeral data, to archival data. From many small files, to fewer very large files. This has meant that AARNet's software infrastructure needs to be spread in order to minimise network latencies between nodes, and this has created its own challenges in providing a reliable and reusable platforms for data sharing and transfer. Managing these requirements has resulted in more than one way to upload and share data.

Rocket helps some users run scientific instrumentation and require a tool to quickly upload vast amounts of datasets quickly. Usually the ownCloud sync client is used but for some it is not quick enough because it uploads files one by one with a single thread via the ownCloud webdav gateway, which can choke when presented many little files. In addition, the sync client is geared more to synchronisation rather than just upload, meaning that both client and server store the same data.

This is undesired by instrument users as it causes issues and interrupts the natural workflow. In some cases the instrument PC's disk becomes full of synchronised data they do not need. For this reason, we have developed a product called Rocket, which integrates directly into ownCloud and EOS.

Rocket is an upload only tool that bundles and uploads payloads of data into CloudStor using parallel threads. A payload can consist of bundles small files and chunks of large files. Settings such as payload sizes, number of threads, maximum number of files per payload, payloads to buffer in memory is all user modifiable. This means users can fine tune settings to best utilise their local network and PCs.

By keeping payload sizes consistent and by using parallel threads, in the right conditions, we are able to upload data as fast as the PC can read off the local disk. Rocket uploads files into our ownCloud, so it is possible to upload into a shared space and have files arrive at a group of users as files are uploaded.

**Technology: deployment, testing and optimization / 26**

## An insiders look into scaling Nextcloud

Matthias Wobben<sup>1</sup>

<sup>1</sup> *Nextcloud GmbH*

**Corresponding Author(s):** matthias@nextcloud.com

Nextcloud can be scaled from very small to very big installations. This talk gives an insiders look on how to deploy, run and scale Nextcloud in different scenarios. Discussed will be a very big installation in the research space, an installation in a global enterprise and the implementation of Nextcloud at one of the largest service providers in the world. The different infrastructural choices of servers, databases, storage backends, operating systems and other architectural differences are presented and compared. Also the various requirements around scalability, performance, high availability, backup, costs and functionality are considered.

**Technology: deployment, testing and optimization / 28**

## ownCloud scalability research

**Author(s):** Jörn Dreyer<sup>1</sup>

**Co-author(s):** Thomas Müller ; Patrick Jahns<sup>2</sup>

<sup>1</sup> *ownCloud GmbH*

<sup>2</sup> *ownCloud*

**Corresponding Author(s):** jfd@owncloud.com, thomas.mueller@owncloud.com

Over the past year we dropped the requirement that ownCloud should run on every PHP platform. This allows us to research architectural changes, like push notifications, microservices, dockerized deployments, HSM integration and storing metadata in POSIX or object storages. On the client side we are exploring E2EE, virtual filesystems and delta sync. Together with feedback from our community this will allow us to ease deployment, maintenance and scaling of an ownCloud instance.

**Technology: deployment, testing and optimization / 36**

## Shibboleth authentication for Sync & Share - Lessons learned

Enno Gröper<sup>1</sup>

<sup>1</sup> *Humboldt-Universität zu Berlin*

**Corresponding Author(s):** enno.groeper@cms.hu-berlin.de

The talk will introduce the main concepts of Shibboleth, advantages and disadvantages and show the integration of Shibboleth with a Sync and Share service (webapp with own session handling, not designed for using the Shibboleth session as webapp session) with Seafile as an example. Furthermore it will discuss the problems of Shibboleth federations and possible mitigations. A special focus will be on suitable Shibboleth attributes for identifying a user in a Sync & Share service and on Single Logout.

**Technology: deployment, testing and optimization / 13**

## Global Scale - How to scale large instances beyond the typical limitations

Björn Schießle<sup>None</sup>

**Corresponding Author(s):** bjoern@nextcloud.com

The typical Nextcloud setup for large installations includes a storage and a database cluster attached to multiple application servers behind a load balancer. This allows organisations to scale Nextcloud for thousands of users. But at some point the shared components like the storage, database and load balancer become a expensive bottleneck. Therefore Nextcloud introduced “Global Scale”, a new way to scale large installations based on the unique federated sharing feature beyond this limitations. Federated sharing combined with our new developed “Lookup Server” and the “Global Site Selector” allows you to set up many small Nextcloud server based on commodity hardware and connect them to one large system. This enables organisations to scale the system to hundreds of millions of users and reduce complexity and costs dramatically. This talks will discuss the current state of the Global Scale architecture.

**Technology: deployment, testing and optimization / 56**

## Distributed, cross-datacenter replicated relational database and file storage at ownCloud

Piotr Mrówczyński<sup>1</sup> ; Jörn Dreyer<sup>2</sup> ; Thomas Müller<sup>2</sup>

<sup>1</sup> *ownCloud GmbH, CERN, KTH Stockholm, TU Berlin*

<sup>2</sup> *ownCloud GmbH*

**Corresponding Author(s):** mrow4a@yahoo.com

There is growing interest for self-hosted, scalable, fully controlled and secure file sync and share solutions among enterprises. The ownCloud has found its share as free-to-use, open-source solution, which can scale on-premise from a single commodity class server to a cluster of enterprise class machines, and serve from one to thousands of users and PB of data. Over the years, it has grown a user base of over 10 million community members and users, with over 400 enterprise customers in 2017. To address challenges of globally distributed offices, companies are looking for solutions which could address the issues imposed by single datacenter architectures. In this contribution, I will address the above challenges with cross-datacenter replicated architecture. Remote client after authentication can contact any from multi-region servers. These in turn delegate some of the responsibilities to external services as log warehouse service, user/group management service, and



globally replicated storage and relational SQL database. This allows to not only mitigate the latency-to-file, but also provide very good data availability and load balancing properties. Additionally, the architecture provides very high data protection against disasters, partially allowing instant disaster-recovery. In the contribution I will look into existing solutions as CockroachDB and IBM Spectrum Scale.

**Technology: deployment, testing and optimization / 44**

## Container-based service deployment in private and public clouds

Enrico Bocchi<sup>1</sup>

<sup>1</sup> CERN

**Corresponding Author(s):** enrico.bocchi@cern.ch

Container technologies are rapidly becoming the preferred way to distribute, deploy, and run services by developers and system administrators. They provide the means to create a light-weight virtualization environment, i.e., a container, which is cheap to create, manage, and destroy, requires a negligible amount of time to set-up, and provides performance equatable with the one of the host.

Containers are particularly suitable for distributing and running software following a microservice-based architecture: Complex services are broken down into fundamental applications responsible for specific tasks and running independently on the others. In this context, one container constitutes a building block of the entire architecture and hosts a single application with its dependencies, libraries, and configuration files. The final service is assembled by running and orchestrating multiple containers at the same time, each responsible for a specific application.

In this work, we introduce Boxed: A container-based version of EOS (the CERN disk/cloud storage for science), CERNBox (Cloud storage & synchronization service), and SWAN (Service for Web-based ANalysis). Boxed is available in two flavors: (i) A one-click setup for personal use where all services run on a single host; and (ii) a production-oriented deployment with the ability to scale out according to the storage and computing needs.

Boxed demonstrates how CERN core services can be deployed in diverse scenarios, ranging from desktop and laptop computers to private and public clouds. In all contexts, Boxed delivers the same fully-fledged services used daily by CERN scientists in demanding scenarios. We report on our experience in the development of Boxed, on its future evolution and large-scale testing, and on its adoption in the context of educational and outreach activities.

**Technology: deployment, testing and optimization / 46**

## Future architectures for sync and share based on microservices

Hugo Gonzalez Labrador<sup>1</sup>

<sup>1</sup> CERN

**Corresponding Author(s):** hugo.gonzalez.labrador@cern.ch

Microservices are an approach to distributed systems that promote the use of finely grained services with their own lifecycles, which collaborate. The use of microservices facilitates embracing new technologies and architectural patterns. Sync and share providers could increase the modularity and facilitating the exchange of components and best practices adopting the use of microservices.

In CERNBox we have currently started replacing functionality in the monolithic software with small microservices to improve the management of the service and ease the integration with other services. We report on the current architecture and on the future evolution of the platform based on microservices.

**User Voice: Novel Applications / 8**

## Neural Network Nanoscience Scanning Electron Microscope Image Recognition

Stefano Cozzini<sup>None</sup>; Aversa Rossella<sup>1</sup>; Giuseppe Piero Brandino<sup>2</sup>; Alberto Chiusole<sup>3</sup>; Cristiano De Nobili<sup>4</sup>

<sup>1</sup> CNR/IOM

<sup>2</sup> eXact-lab srl

<sup>3</sup> eXact-lab srl

<sup>4</sup> MHPC sissa

**Corresponding Author(s):** stefano.cozzini@exact-lab.it

We present our recent work \1 where we applied state of the art deep learning techniques for image recognition, automatic categorization, and labeling of nanoscience images obtained by scanning electron microscope (SEM). Roughly 20,000 SEM images were manually classified into 10 categories to form a labeled training set, which can be used as a reference set for future applications of deep learning enhanced algorithms in the nanoscience domain. The categories chosen spanned the range of 0-Dimensional (0D) objects such as particles, 1D nanowires and fibres, 2D films and coated surfaces, and 3D patterned surfaces such as pillars. The training set was used to retrain and train from scratch on the SEM dataset and to compare many convolutional neural network models (Inception-v3, Inception-v4, ResNet). We obtained compatible results by performing a feature extraction of the different models on the same dataset. We performed additional analysis of the classifier on a second test set to further investigate the results both on particular cases and from a statistical point of view. Our algorithm was able to successfully classify around 90% of a test dataset consisting of SEM images, while reduced accuracy was found in the case of images at the boundary between two categories or containing elements of multiple categories. In these cases, the image classification did not identify a predominant category with a high score. We used the statistical outcomes from testing to deploy a semi-automatic workflow able to classify and label images generated by the SEM. Finally, a separate training was performed to determine the volume fraction of coherently aligned nanowires in SEM images. The results were compared with what was obtained using the Local Gradient Orientation method. This example demonstrates the versatility and the potential of transfer learning to address specific tasks of interest in nanoscience applications.

**User Voice: Novel Applications / 20**

## Integrated processing and sharing services for geospatial data

**Author(s):** Armin Burger<sup>1</sup>; Paul Hasenohr<sup>1</sup>; Davide De Marchi<sup>1</sup>

**Co-author(s):** Pierre Soille<sup>1</sup>

<sup>1</sup> European Commission - Joint Research Centre

**Corresponding Author(s):** armin.burger@ec.europa.eu, paul.hasenohr@ec.europa.eu

The Joint Research Centre (JRC) of the European Commission has set up the JRC Earth Observation Data and Processing Platform (JEODPP) as a pilot infrastructure to enable the knowledge production

Units to process and analyze big geospatial data in support to EU policy needs. The very heterogeneous data domains and analysis workflows of the various JRC projects require a flexible set-up of the data access and processing environments.

The basis of the platform consists of a petabyte-scale data storage system running EOS, a distributed file system developed and maintained by CERN. Three data processing levels have been implemented on top of this data storage and are delivered through a cluster of processing servers. The batch processing level allows running large-scale data processing tasks in a parallelized environment. The web-based remote desktop level provides access to tools and software libraries for fast prototyping in a standard desktop environment. The highest abstraction level is defined by an interactive processing environment based on Jupyter notebooks.

The interactive data processing in notebooks allows for advanced data analysis and visualization of the results on interactive maps on-the-fly. The processing in the notebooks is delegated via HTTP requests to a pool of processing servers for deferred parallel execution. As a response to the requests, the servers return the results of the data processing as JSON stream or map tiles which are rendered in the notebook. The processing is based on a custom developed API for analysis and visualization of geospatial data. The notebook-based approach gives users also the possibility to share data analysis and processing workflows with other users instead of merely sharing the output data of processing results. This facilitates collaborative data analysis.

All the processing levels are inter-connected through data access and data sharing interfaces based either on traditional file system access or on HTTP-based access provided by a CS3 software. Users can access data on the central platform storage from their office desktop via CIFS through a dedicated gateway, or from the Internet via a dedicated NextCloud instance. Access to data from the farm of processing servers is provided through a FUSE client specific to EOS in order to offer the highest possible data throughput. A major challenge is to set up consistent and reliable access control via all data access methods and tools. The migration of the cloud service to CERNBox is going to be assessed since it would facilitate the management of multi-protocol data sharing.

**User Voice: Novel Applications / 9**

## Research Data Management with iRODS

Hurng-Chun Lee<sup>1</sup> ; Robert Oostenveld<sup>1</sup> ; Erik van den Boogert<sup>1</sup> ; Eric Maris<sup>1</sup>

<sup>1</sup> *Donders Institute, Radboud University*

**Corresponding Author(s):** h.lee@donders.ru.nl

Research Data Management (RDM) serves to improve the efficiency and transparency in the scientific process and to fulfil internal and external requirements. Three important goals of RDM are:

- long-term data preservation,
- scientific-process documentation,
- data publication.

One of the tasks in RDM is to define a workflow for data as part of the research process and data lifecycle. RDM workflows usually consists of data-management policies that are considered complex by the researchers that have to implement them. A challenge of the data-management system is to strike the balance between procedural standardization and domain-specific customisation, so that different data-management workflows can be implemented.

In this presentation, we will discuss a data-management workflow designed for the research field of cognitive neuroscience, the data of which usually contains sensitive information. We will outline the authorisation policies around it, and present an approach of using a rule-based data management system, iRODS, to realise the workflow and achieve the three RDM goals.

User Voice: Novel Applications / 52

## Up2University: Educational Platform with Sync/Share

Corresponding Author(s): peter.szegedi@geant.org

The "Up to University" (Up2U) Project – coordinated by GÉANT – is to bridge the gaps between secondary school and university by providing European schools with a Next Generation Digital Learning Environment that helps students developing the knowledge, skills and attitudes they need to succeed at university. This student-centered digital learning environment forms the Up2U ecosystem, which integrates formal and informal learning scenarios and applies project-based and peer-to-peer learning methodology to develop transversal skills and digital competences crucial for success at university. Up2U believes in a digital learning ecosystem that is a set of autonomous tools represented by portable containers integrated together via open standards and protocols. Tools can be picked by the user in any shape and form to best support their individual learning path. Up2U is a flagship use case for the sync&share domain interoperability via the community standard OpenCloudMesh protocol.

ICT security and students' privacy are very critical aspects and there is a strong need for schools and the higher education system to have a better understanding of the context: What kind of threats and weaknesses schools face and what related countermeasures apply to protect networks and infrastructures? School staff (principals, teachers, ICT officers, etc.) all need to be involved in a public discussion on common best practices for security and privacy protection to improve understanding of the gaps to be filled.

The federated File Sync & Share functionality of the Up2U architecture is implemented by the ownCloud software. It provides added value to the traditional Learning Management Systems (such as Moodle) in the area of document handling. CERNbox is a particular File Sync & Share platform developed by CERN using the ownCloud engine. SWAN is based on the technology provided by Project Jupyter and it uses CERNBox as the home directory for its users. As a consequence, all the files (e.g., pictures, videos, data sets) available in CERNBox can be easily imported in a SWAN notebook and, vice versa, the notebook itself will be synchronized and stored in the cloud. Notebooks can then be linked to Moodle courses easily. The stand-alone File Sync & Share service domains offer federated sharing of files and folders via the community standard OpenCloudMesh protocol.

Up2U glues together its open architecture elements implemented in Docker containers that can be deployed on top of a wide variety of cloud infrastructures. Thanks to the modular and portable design, the local deployments can easily be customized to exploit the available infrastructure elements, educational resources and other specificities of the particular country or region as much as possible.

User Voice: Novel Applications / 37

## Towards interactive data analysis for TOTEM experiment using Apache Spark

Grzegorz Bogdał<sup>1</sup> ; Piotr Gawryś<sup>1</sup> ; Paweł Nowak<sup>1</sup> ; Łukasz Plewnia<sup>1</sup> ; Leszek Grzanka<sup>2</sup> ; Maciej Malawski<sup>1</sup>

<sup>1</sup> AGH University of Science and Technology

<sup>2</sup> AGH University of Science and Technology (PL)

Corresponding Author(s): , , , , leszek.grzanka@cern.ch, malawski@agh.edu.pl

Data analysis in High Energy Physics experiments requires processing of large amounts of data. As the main objective is to find interesting events from among those recorded by detectors, the typical operations involve data filtering by applying cuts and producing of histograms. The typical offline data analysis scenario for TOTEM experiment at LHC, CERN involves processing of 100s of ROOT

ntuples of 1-2GB size, which gives up to a 1TB of data per analysis. The event size is relatively small (1KB-1MB) with most events of 1KB in size.

The goal of our work is to investigate the usability of one of modern big data toolkits, namely Apache Spark, to provide an interactive environment for parallel data analysis. As a proof of concept solution we employed Apache Spark 2.0, combined with Spark ROOT for accessing ROOT files. To provide interactive environment, we coupled it with Zeppelin, which provides a Web-based notebook environment, which allows combining analysis code in Scala to access Spark API and in Python for creating plots. This environment was deployed on Prometheus cluster at Academic Computer Centre Cyfronet AGH and integrated with SLURM resource management system. We developed scripts for combining these tools in a user friendly way and a set of notebooks showing sample analysis.

Our plans include the implementation of selected data analysis pipelines, performance analysis, integration with the Jupyter notebooks and SWAN (Service for Web based ANalysis) from CERN, and the development of high-level user friendly data analysis tools and libraries dedicated to high energy physics.

This research was supported in part by PLGrid Infrastructure.

References:

- <https://spark.apache.org/>
- <https://github.com/diana-hep/spark-root>
- <https://zeppelin.apache.org/>
- <http://www.plgrid.pl/en>
- <https://swan.web.cern.ch/>

**Welcome / 50**

## **Keynote: multiwinner voting: theory, experiment, and data**

Piotr Faliszewski<sup>1</sup>

<sup>1</sup> *AGH Kraków*

**Corresponding Author(s):**

**Welcome / 49**

## **CS3 2018 Welcome**

**Corresponding Author(s):** jakub.moscicki@cern.ch

64

## **Visit to CYFRONET Computing Center**