
## Project acronym: CS3MESH4EOSC

Deliverable D3.4: **Report on integration with peer-to-peer storage**

| | |
|---|---|
| Contractual delivery date | 30/06/2022 |
| Actual delivery date | 26/01/2023 |
| Grant Agreement no. | 863353 |
| Work Package | WP3 |
| Nature of Deliverable | R (Report) |
| Dissemination Level | PU (Public) |
| Lead Partner | Cubbit |
| Document ID | CS3MESH4EOSC-22-018 |
| Authors | Stefano Baldi |



*This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 863353*

# Versioning and Contributions History

| *Version* | *Date* | *Authors* | *Notes* |
|---|---|---|---|
| 0.1 | 09.05.2022 | Stefano Baldi (Cubbit SrL) | First Release |
| 0.2 | 8.6.2022 | Hugo Gonzalez (CERN) | Second Release |
| 0.3 | 12.08.2022 | Hugo Gonzalez (CERN) | Third Release |
| 0.4 | 02.09.2022 | Stafano Baldi (Cubbit SrL) | Fourth Release |
| 0.5 | 05.09.2022 | Hugo Gonzalez (CERN) | Review |
| 0.6 | 17.06.2022 | Stefano Baldi (Cubbit SrL) | Fifth Release |
| 0.7 | 17.01.2023 | Stefano Baldi (Cubbit SrL) | Major text rework |
| 0.8 | 24.01.2023 | Jakub Moscicki (CERN) | Final Review |

# Index



# Index of Figures

# 1 Introduction

Cubbit is developing distributed technology that allows customers to share with each other commonly available resources such as powerline and internet access, in order to implement high availability and high resilient services.

The basic idea is that a swarm of simple computational nodes highly scattered over the territory is able to achieve a high reliability rate requiring less economic and environmental resources than a traditional data center does.

Cubbit is currently offering storage based services: sync-and-share and object storage.

Traditional object storage platforms rely on a client-server model where the data is safeguarded centrally by storage servers. Cubbit challenged this model by providing a security-focused peer to peer distributed storage solution.

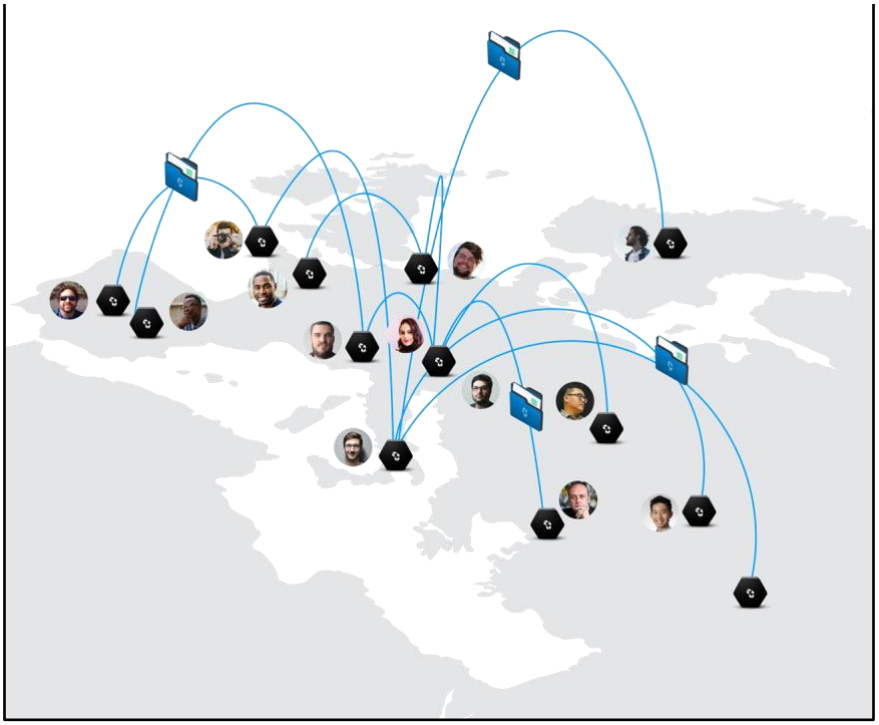

*Figure 1: Cubbit devices distributed across Europe*

## 1.1 Cubbit infrastructure

Most of the Cubbit project is at infrastructure level, easiness of installation and configuration, monitoring of the swarm status, predictive maintenance algorithm, etc.

Such an effort is transparent to Cubbit users, they get a SaaS/PaaS product, even when hosting a Cubbit Device (Cubbit Cell see picture).

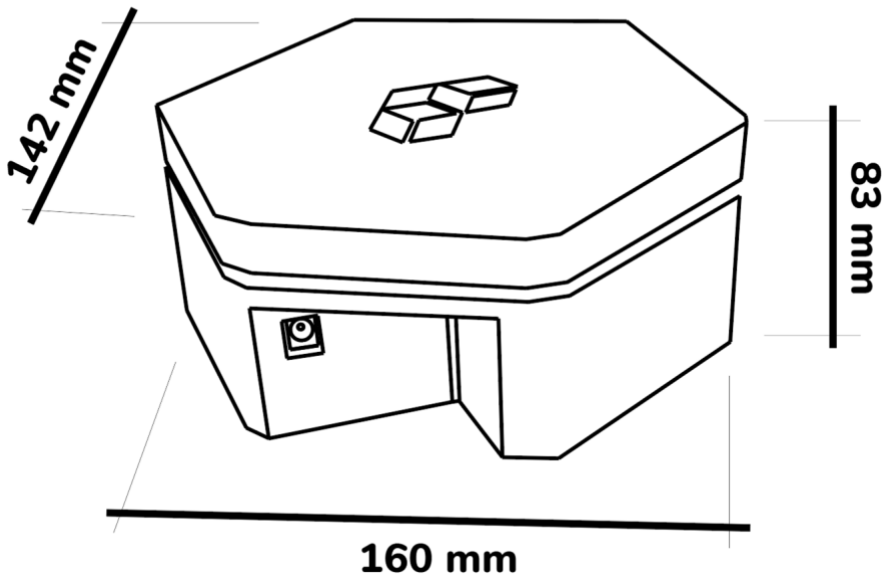


*Figure 2: Cubbit devices (Cubbit cell) dimension*

That's why our efforts are focused on finding UI, protocols and use cases that let users integrate our solution, without having consciousness of the complexity of the algorithms that guarantee data integrity, availability and durability.

This document details the research work and the possibilities of integrating a peer-to-peer decentralised cloud storage system in the ScienceMesh federation.

During the development of the present deliverable Cubbit Infrastructure evolved significantly: from a peer-to-peer network where node were hosted by private (consumer level) to peer to peer network where nodes are hosted by companies (business level). This increased the service level and availability but also configuration complexity of the nodes.

In the next paragraph we report:

- Cubbit as a consumer of service from Science Mesh node.
- Cubbit as a provider of service for Science Mesh
    - Description of cloud (NGC) where Cubbit server relies on
    - Description of S3 API that Science Mesh can consume from Cubbit Cloud.

# 2 Cubbit as a consumer of service from Science Mesh

As mentioned before Cubbit relies on a hybrid cloud infrastructure consisting of a network of P2P devices (the "Swarm") coordinated by a central optimization layer of microservices.

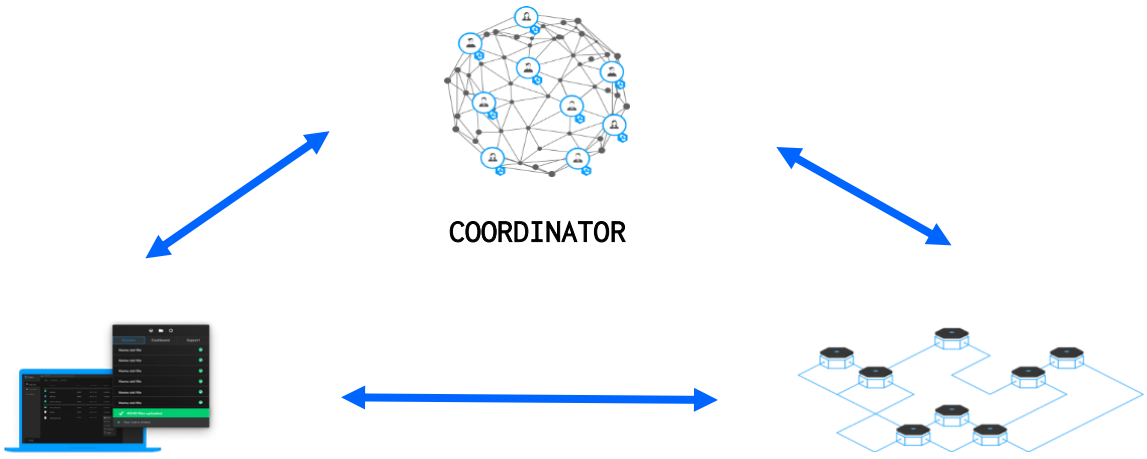


*Figure 3: Cubbit coordinators swarm and clients.*

Differently from a centralized cloud provider architecture, a file uploaded to Cubbit is not stored on a central server infrastructure, but it is encrypted and split up into several shards that are distributed over the swarm. In the same way, when a file is downloaded from Cubbit, all the shards are recollected, decrypted, and joined back together to retrieve the original payload.

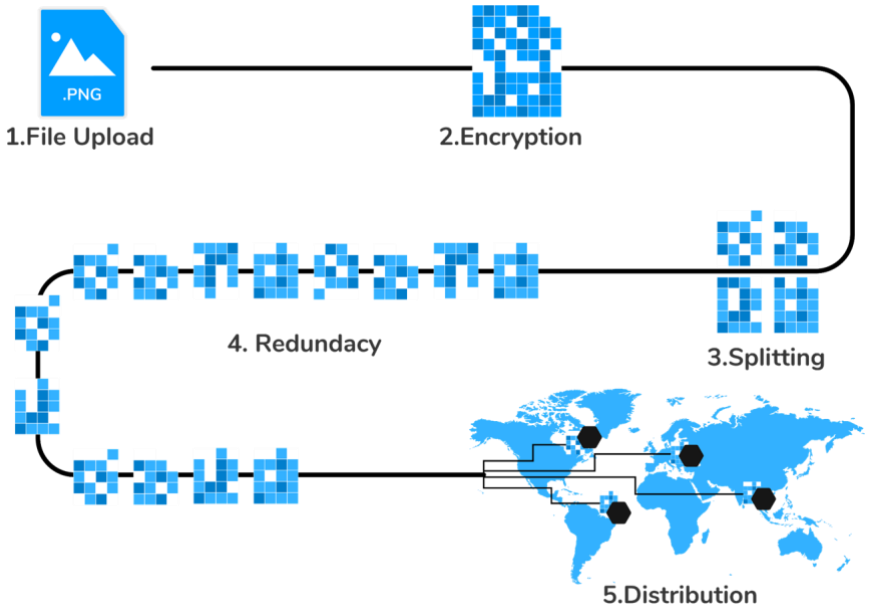


*Figure 4: Cubbit upload/download protocol.*

Due to a different architecture, a seamless federation between the Cubbit service and other ScienceMesh nodes requires a specific layer that acts as a "translator" between Cubbit's P2P architecture and more centralized storage architectures.

An internal integration strategy was developed by Cubbit based on the API layer to assure transparent and portable integration. The first set of endpoints was developed and deployed to allow basic file upload and download:

- **/files (POST):** This endpoint allows a user to upload a file to the Cubbit cloud. Since Cubbit allows for encryption and redundancy, a user can specify them as options in the request body. If encryption is enabled, an encryption key has to be passed as a mandatory parameter.

- **/files/{file_id} (GET):** This endpoint allows a user to download a file with the specified file_id from the Cubbit network. If the uploaded file had been encrypted during the upload, the user must give the file key as a body parameter for this request.

Next, the IOP connector for Cubbit was implemented using the simple endpoints described above. The connector allows the Cubbit service to consume OCM shares from ScienceMesh node.

The connector leverages on the following CS3APIs:

**OcmCoreAPI.CreateOcmShare:** this endpoint allows other partners to create a Share with Cubbit users
**OcmAPI.ListReceivedOCMShares:** this endpoint allows listing all the received shares

The implementation is based on a basic NodeJS server that is responsible for accepting the OCM requests forwarded by the Reva gateway. These requests are then validated and processed by a Coordinator Cubbit microservice which redirects as appropriate.

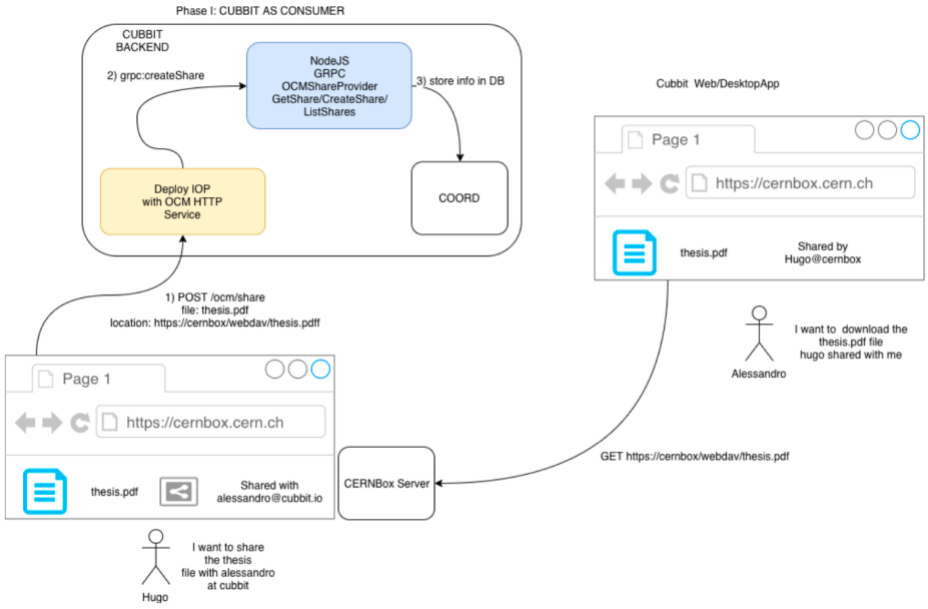


*Figure 5: Cubbit as a consumer for Sciene Mesh Services.*

The CS3APIs repository has been integrated as a submodule of the Cubbit software stack — this helps the interface (proto-definitions) to stay up-to-date with the CS3APIs upstream. Further down the line the CS3APIs are integrated with the EFSS products and thus made available in the Science Mesh nodes. The CS3APIs integrate either natively (Owncloud Infinite Scale) or as a Science Mesh application plugin (for Owncloud 10 and Nextcloud) and are described in other Deliverables of the CS3MESH4EOSC project.

Implemented server and endpoint was provided and tested in 2021.

# 3 Cubbit as a provider of service for Science Mesh

The experience of integrating services with Science Mesh, made Cubbit understand the importance of adopting industrial/commercial standards.

At the same time Cubbit experimented with the limitations in level of service that can be achieved by relying on privately hosted devices only. Private hosts are not reliable enough: thet turn-off devices, network access, or they simply lose interest in the device after receiving it. A swarm of private hosted devices risks to cost too much in terms of redundancies needed to overcome the limitation of losing too many peers.

For these reasons Cubbit decided to deploy a new cooperative cloud on the following bases:

1. rely on a new swarm of company hosted devices: Cubbit NGC (Next Generation Cloud). Professional level customers are more reliable then private, they are interested in preserving the value of what they purchase. Moreover, SLAs have been introduced for public hosts. The SLAs commit hosts to keep Cubbit devices available to the swarm 99.95% of the time.

2. Professional hosts are also interested in consuming service from Cubbit NGC as users. In order to facilitate the adoption of Cubbit NGC, Cubbit introduced in 2021 the object-storage service and also adopted the de-facto standard API for cloud object storage utilization: **Amazon S3**. Introduction of Amazon S3 APIs allows users to adopt any compatible third-party client to consume space available in Cubbit Cloud.

During the first half of 2021, Cubbit NGC has been rolled out and tested. Since the second half of 2021 Cubbit NGC is in service, and can be used by any user even if they do not host a Cubbit device.

Science Mesh is capable of connecting to an S3 compatible object storage, which means it can consume storage space that Cubbit NGC provides.

The aim of this part of the deliverable is to demonstrate that Science Mesh can effectively consume storage from a Cubbit architecture. The adoption of the S3 commercial standard, will make it easy to connect and integrate with a storage service NGC provides, also as a way to provision additional storage space within the Science Mesh federation.

In line of principle we will demonstrate that Science Mesh can consume storage space from a dedicated swarm of Cubbit devices hosted by a restricted subset of trusted users/entity and even implement its own ‘private’ Cubbit peer-to-peer network and easily consume cloud object storage service from it.

## 3.1 NGC infrastructure

The infrastructure Cubbit relies on, doesn’t change so much compared to section 2: Cubbit NGC is still a hybrid cloud infrastructure consisting of a network of P2P devices coordinated by a set of microservices.

Companies that join the NGC project accept to host a set of Cubbit devices, and share them with other users via the internet.

The more relevant changes in infrastructure compared to the old one is that:

- A single company hosts more than one Cubbit devices
- Cubbit cells are generally hosted behind a firewall, and their installation requires extra configuration.

Devices involved into Cubbit NGC are totally dedicated to NGC user only. All of them are hosted in Italy.


*Figure 6: Cubbit NGC principle*

## 3.2 S3 API

S3 Protocol by Amazon is the de-facto standard for the cloud object storage solutions. It has existed since 2006 and during these years a wide ecosystem of applications has been developed by third parties.

Adopting this protocol Cubbit allows users to choose an S3 client that satisfies a specific use cases. For example, some users use Cubbit as a second level backup destination for third party applications as Veeam, Nakivo etc.

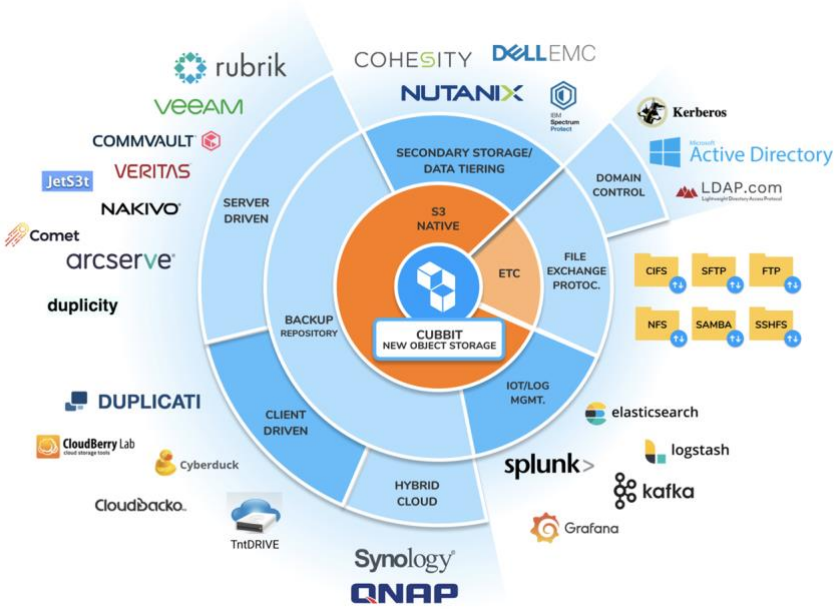


*Figure 7: Cubbit S3 use-cases and third-party applications*

S3 APIs can be supported at various levels. APIs related to object and bucket operation, permission level (ACL), versioning and sharing are the most commonly used. Details for the currently supported API are in section 3.4

### 3.2.1 Cubbit S3 Implementation

S3 is not the native protocol of the Cubbit cells, adding it to Cubbit Cloud implies the deployment of a translation service. This service (named **S3 gateway**) has been implemented as a subset of microservices that can be deployed in public cloud or on-premise cloud.

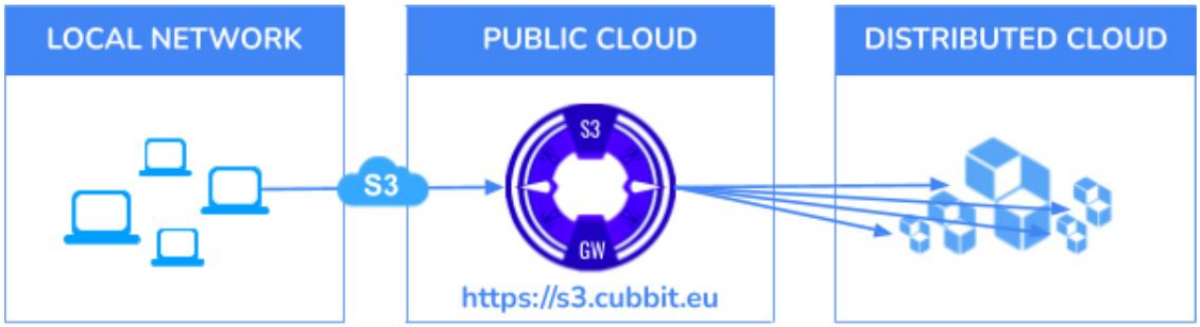


*Figure 8: Cubbit S3 Gateway public implementation*

Cubbit users generally prefer to use it as a public cloud service. But on-premise cloud have several advantages:

1. **reduced latency** - S3 commands are locally converted to native cubbit protocol, that is a multi-connection oriented protocol and results in faster upload/download operations.

2. **Improved security** - S3 relies on asymmetric cryptography, anyway Cubbit hosts part of all the S3 keys when users adopt the public implementation. On-premise installation is totally zero-knowledge: Cubbit doesn't require to know any authentication details about users.

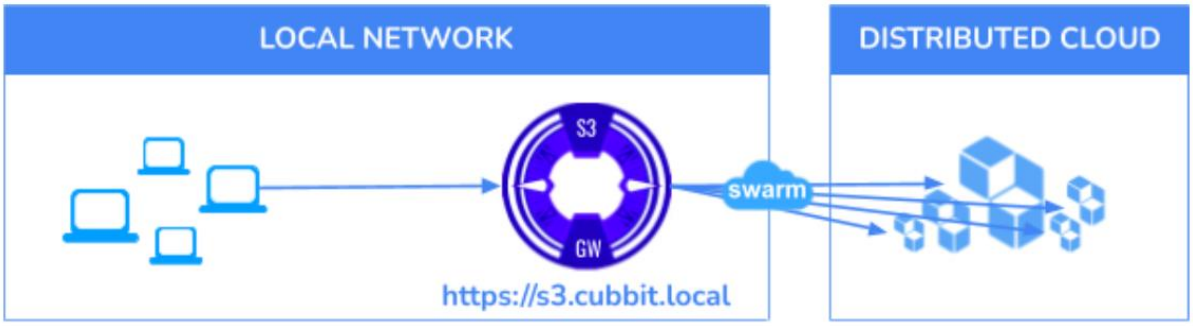


*Figure 9: Cubbit S3 Gateway on-pemise implementation*

## 3.3 Cubbit Console

To gain access to S3 API-based services, any Cubbit user must have a Cubbit account. Cubbit account grant access to the Cubbit Console.

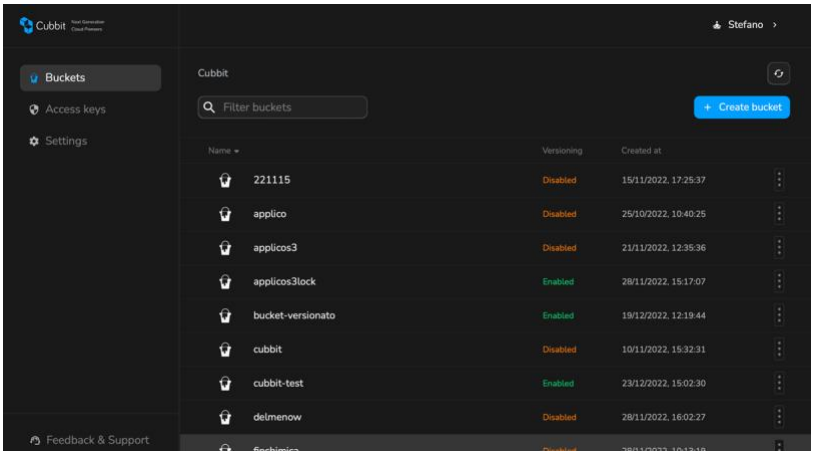


*Figure 10: Cubbit Console - landing view*

Cubbit console allow users to recover S3 credentials

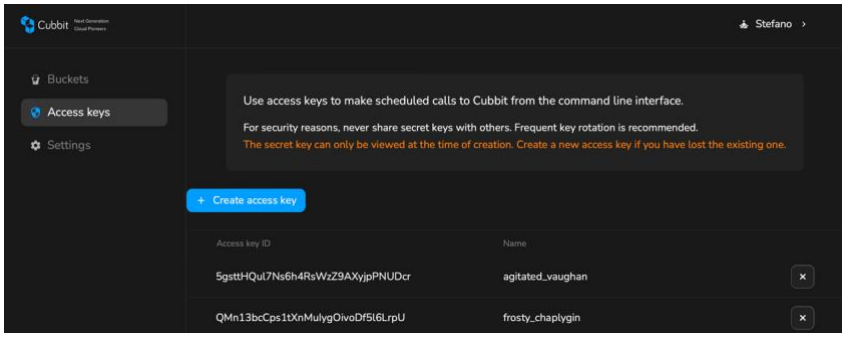

*Figure 11: Cubbit Console - s3 credentials*

Or it allows to create/delete/navigate/configure buckets, create/delete/recover/modify objects:

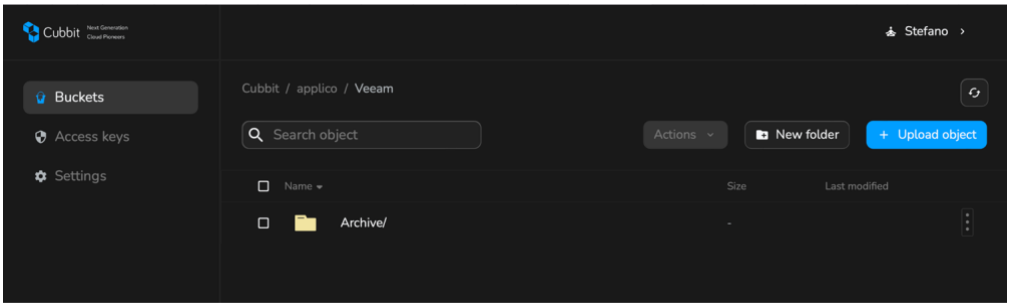


*Figure 12: Cubbit Console - bucket navigation*

Or change user preferences:

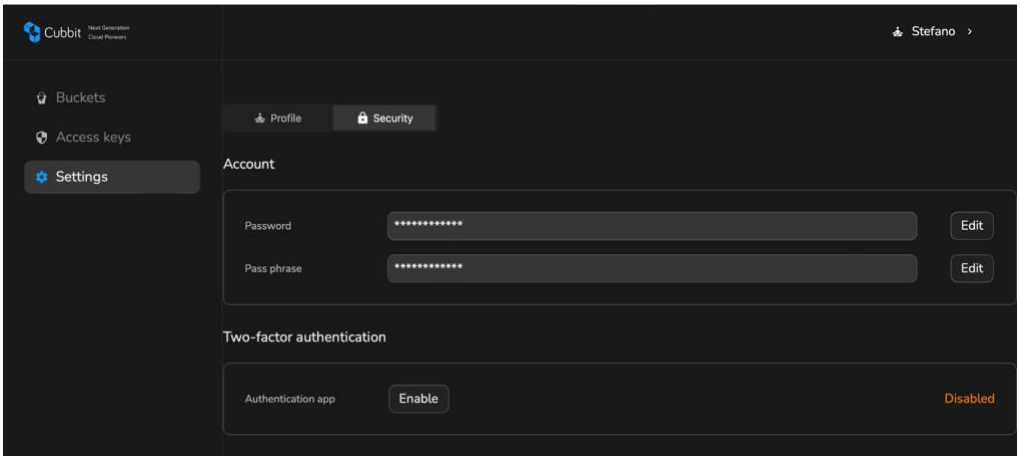


*Figure 13: Cubbit Console - user preferences*

### 3.3.1 Using Cubbit with a generic S3 client

To configure any S3 Compatible client only S3 credentials and an URL are required. User will retrieve s3 credentials from Cubbit console as described in previous section. Public S3 URL for Cubbit NGC is:

**https://s3.cubbit.eu**

In the next paragraph we will show how to use Amazon AWS Command Line Interface to connect to the Cubbit NGC storage space.

### 3.3.2 Using Cubbit with by Amazon AWS

Run `aws configure` command and type credentials in (as shown below):

```
C:> aws configure

AWS Access Key ID [NONE]: 13awQ1J3jBR2d8/im52Gat+Y0kCwFQpb

AWS Secret Access Key ID [NONE]:7e6e************EK1vlw=

Default region name [NONE]:

Default output format [NONE]:
```

Remember to add the Cubbit endpoint (https://s3.cubbit.eu) at the end of each command by `--endpoint-url` option. Some command examples follows.

bucket creation:

```
C:> aws s3 mb s3://test --endpoint-url https://s3.cubbit.eu
```

bucket content listing:

```
C:> aws s3 ls s3://test --endpoint-url https://s3.cubbit.eu
```

bucket content listing:

```
C:>aws s3 cp C:\tst s3://test/tst --endpoint-url https://s3.cubbit.eu
```

## 3.4 Science Mesh Connecting to Cubbit via S3

S3 storage endpoints provided by Cubbit may be readily consumed by Science Mesh nodes using the External Storage plugin for S3 which is offered by popular EFSS solutions such as Nextcloud or Owncloud.

# 4 Outlook

The work presented in this document resulted not only in technical demonstrators and proof-of-concept but also helped Cubbit to understand the importance of adopting industry standards to enable data integration in the rapidly changing and challenging global ICT landscape.

Data integration is today's challenge in every manufacturing sector. Data is becoming larger and more widely available. We consume it, we share it and we pay to store it every day. The digital transformation has reached the manufacturing sectors with the consequence that the relevant part of know-how is moving from the physical world to digital one. A factory that makes floor tiles uses the

same substrate used by the competition to produce them, but applies a different 3D image model to reproduce oak, birch or mogon wood. Image quality makes the difference, and this is where the value of the product lies: it is the digital value to be defended and improved over time.

Document sharing has become object sharing, syncing means often backing up objects, in order to preserve them from physical world disaster or loss.

Actors also changed: we were used to thinking “sync and share” for humans, but today we see routines on servers perform or move huge backups along the networks, or prepare a production textile pipeline with latest changes an artist performed during the night at the other side of the world. Alexa proposes new breakfast recipes for us as soon as we add cart almond milk, while our entire personal genome is lossless digitally stored for years, waiting for a new advancement in science to instruct digital agents how to identify a specific sequence in it, that will help prevent or cure a disease.

Moving from "sync and share" to object-storage, it gives Cubbit a larger scope of application, and it will multiply market opportunity. Moreover it doesn't mean completely renouncing the traditional “sync and share" use case: being good at storing is a very challenging task, that is a fundamental issue for the "sync and share". It just allows us to save development and testing time in order to focus on data integrity, durability and availability, letting a third-party software implement specific and complex UX/UI as "sync and share" is.

In order to be simple to integrate we choose S3 as a communication protocol, as it is the de-facto market standard, it is a market driven choice (and it was also the fastest way to improve the level of compatibility with ScienceMesh), but adding another protocol (i.e. webDAV) is as easy as extending to S3 was, from a technological point of view.

Bi-directional integration with Science Mesh was the first huge federation Cubbit was able to join in, and the changes required to Cubbit infrastructure to achieve this were conceptual and not only technical.

These changes not also drove Cubbit to be more reliable and very simple in being integrated with ScienceMesh but with any other federation we will work in future.