CS3 MESH4 EOSC

Connecting European Data

Federated collaborative workflows for Jupyter

Diogo Castro (CERN), Marcin Sieprawski (Software Mind)

diogo.castro@cern.ch, marcin.sieprawski@softwaremind.com

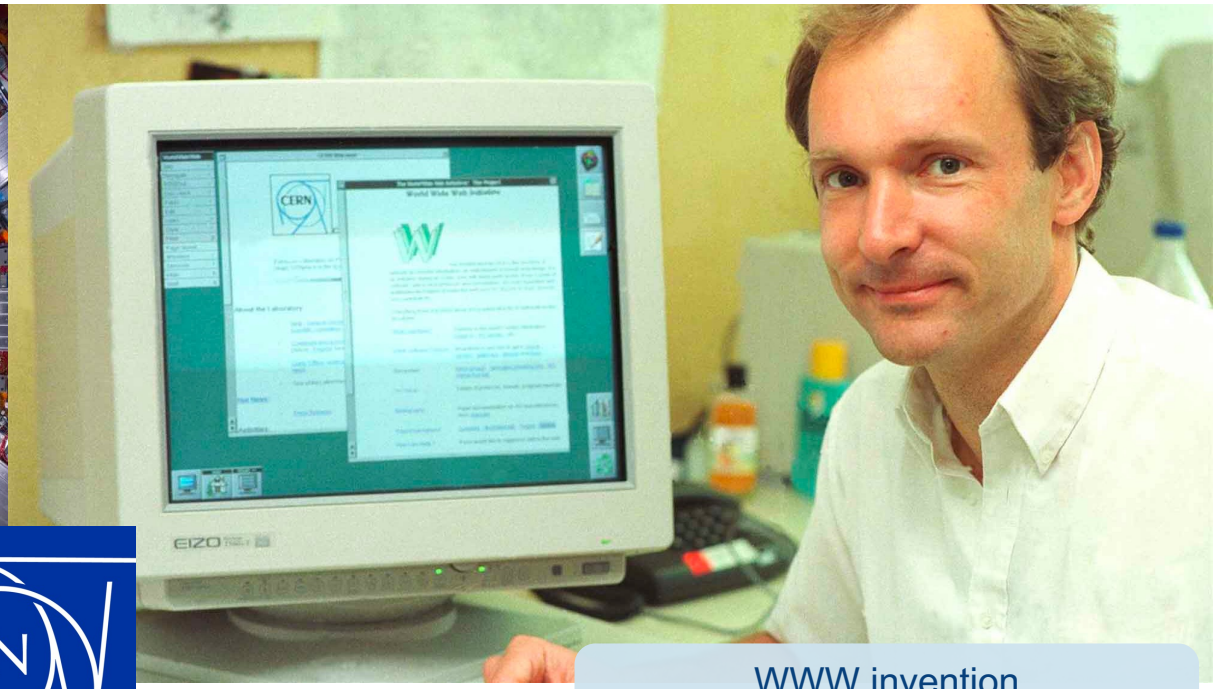
CS3MESH4EOSC has received funding from the European Union's Horizon 2020 Research and Innovation programme under **Grant Agreement No. 863353**.

Jupytercon 2023
10/05/2023

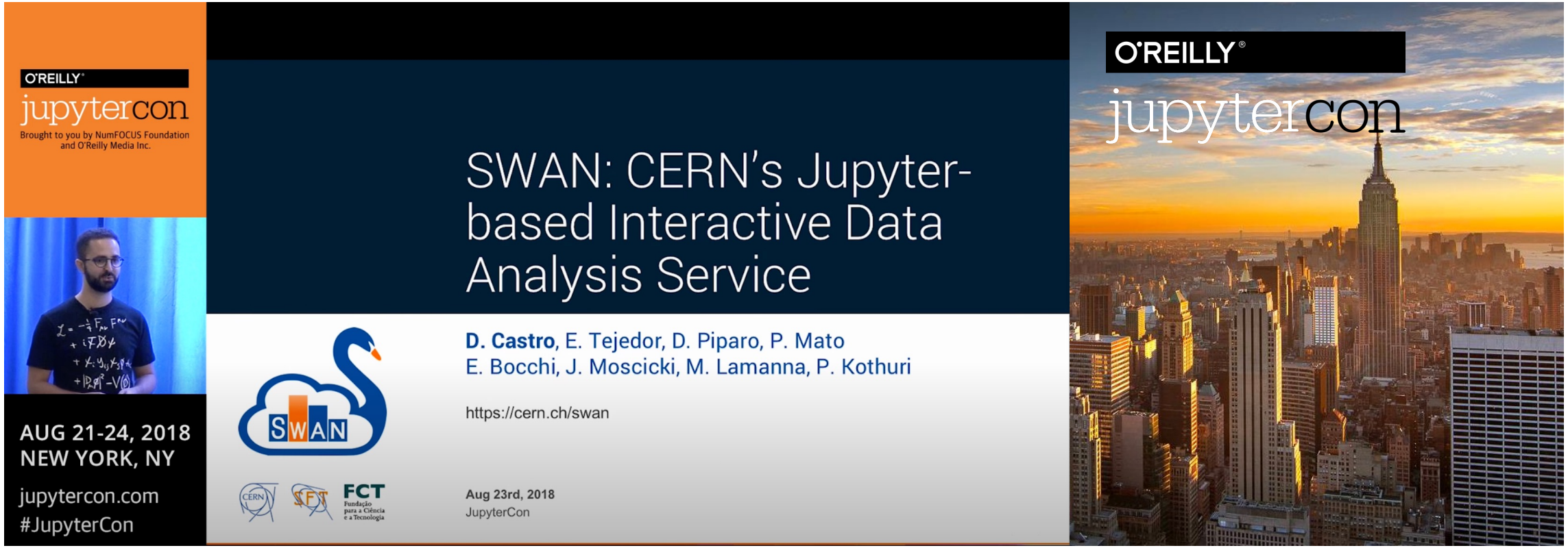
Introduction

CERN

The Large Hadron Collider



WWW invention



- https://conferences.oreilly.com/jupyter/jup-ny/public/schedule/detail/68359.html
- https://www.youtube.com/watch?v=TDp_XlgtpDA

SWAN – integrating (CERN) services

UI/Core

Software

Analysis platforms

Storage

Compute

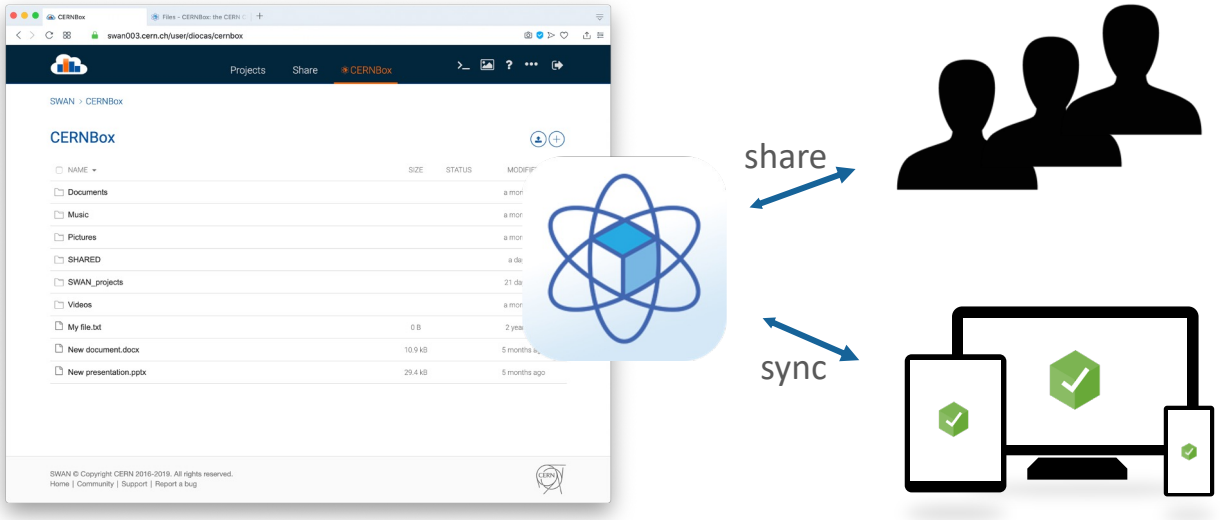
Infrastructure

Service for Web-based Analysis
CERN's Jupyter service
Created in 2016

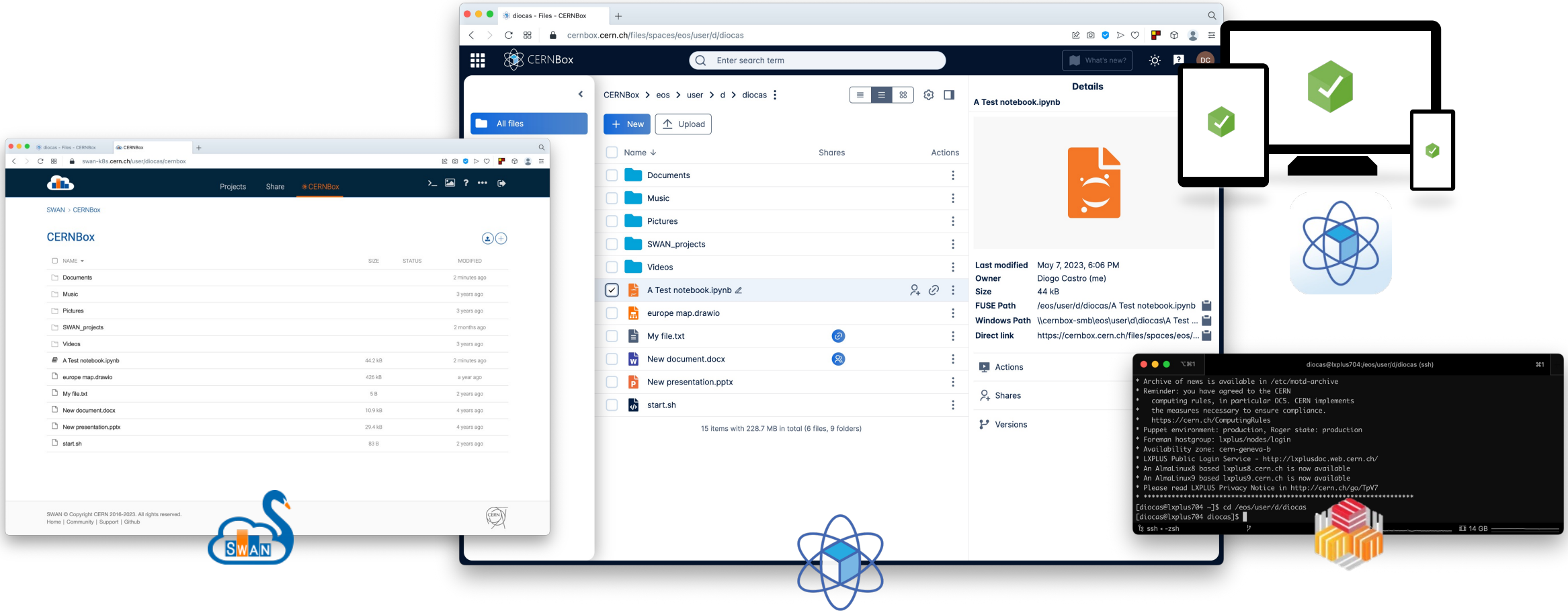
Storage

CERNBox EOS

- All the data our users need for their analysis
  - CERNBox as home directory
  - Experiment repositories, projects, open data, …
- Sync&Share
  - Files synced across devices and the Cloud
  - Simple collaborative analysis



The same view everywhere



Integrations

- SWAN acts as a client to other resources
  - Allows support for both *single-node* and *distributed* analysis
  - Run lightweight analysis interactively, offload heavy computations
  - Storage as a shared layer across multiple services

Analysis Facility: why?

- A major upgrade to LHC is coming
  - Called Hi Luminosity (HL-LHC), planned for 2029
  - 30x more data throughput
- Major upgrade to analysis infrastructure is also required
- HL-LHC needs are pushing us to build modern "Analysis Facilities"
  - R&D effort across the whole physics community around the world (WLCG – worldwide LHC computing grid)

Computing

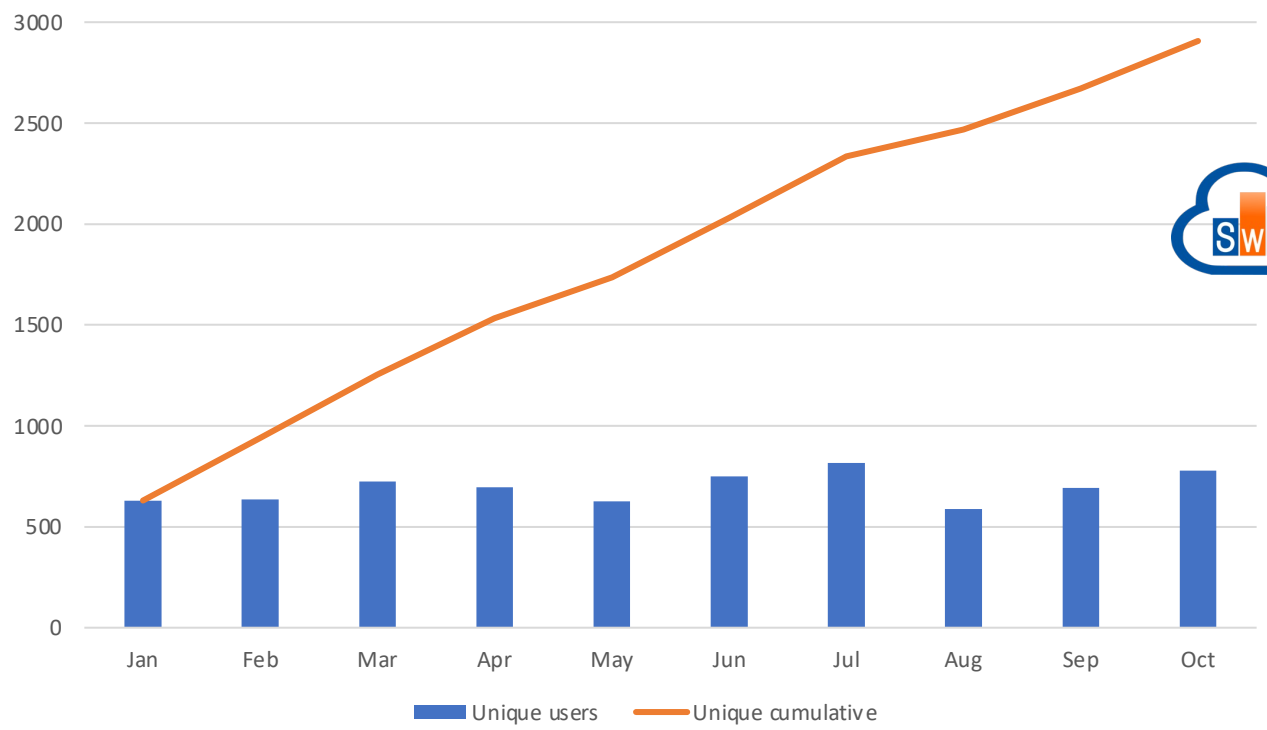
2x	10x
Servers	Compute power

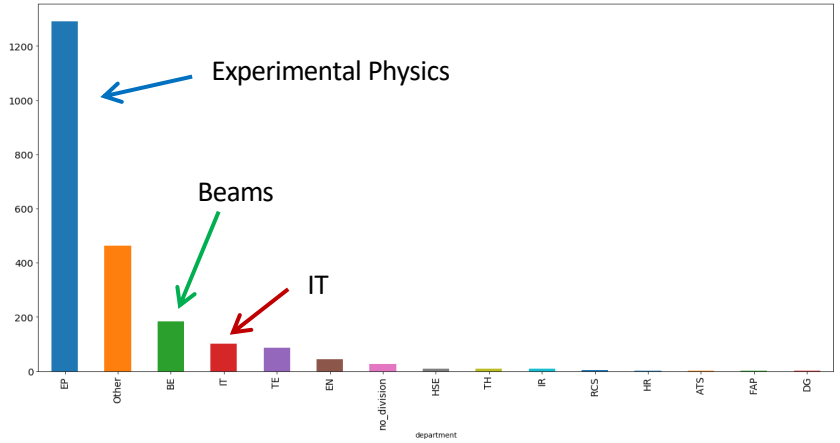
Storage

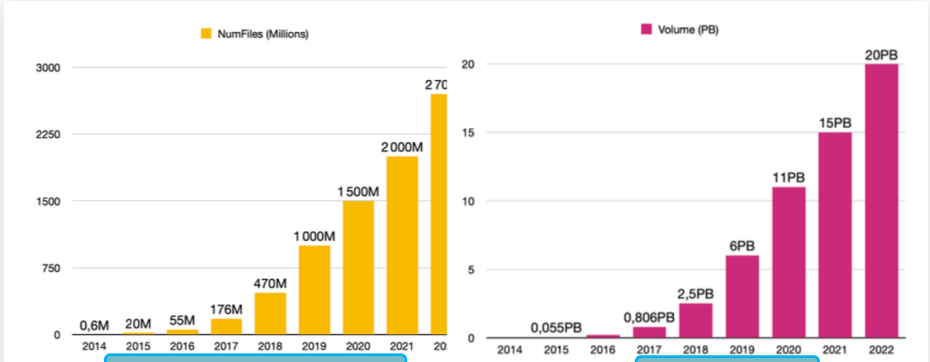
1.5EB	2.5EB
Disk	Tape

WLCG
Worldwide LHC Computing Grid

In numbers







+3 Billion files

20 PB

Analysis Facility: requirements

- Ability to perform fast research iterations on large datasets
- Ability to convert interactive to batch-schedulable workloads
- Ability to scale outside of the facility on occasion
- Ability to reproducibly instantiate desired software stack
- Ability to collaborate in a multi-organisational team on a single resource
- …

"In general, JupyterLab is the main choice when it comes to interactive analysis."

From (work in progress) Analysis Facilities White Paper

CS3MESH4EOSC

EFSS* ecosystem

Commercial cloud providers

OneDrive

Google Drive

Dropbox

Open source (self deployed) alternatives

ownCloud

Nextcloud

Seafile

Pydio
PUT YOUR DATA IN ORBIT

CERNBox

sciebo

cloudstor

SURF DRIVE

SWITCHdrive

polybox

Universitäten und Hochschulen von Rheinland-Pfalz

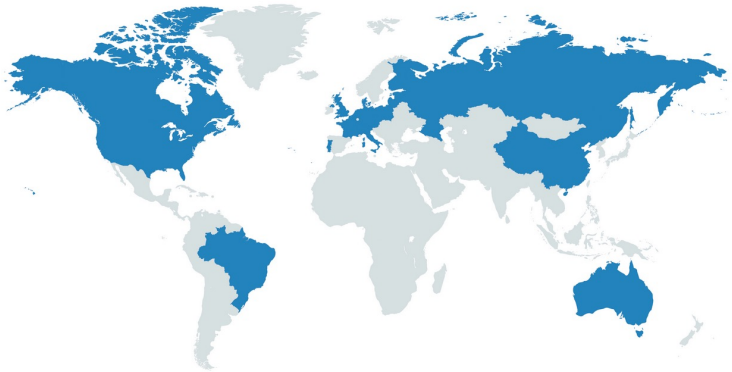
CS³

* Enterprise File Sync and Share

www.cs3community.org

Status quo

- Many EFSS nodes, providing valuable **services** to the community
  - **Mostly Sync and Share**, but not limited to that
- User environments, higher level **applications**
  - (e.g. editors, Jupyter data analysis…)
- Basic **file sharing** possible

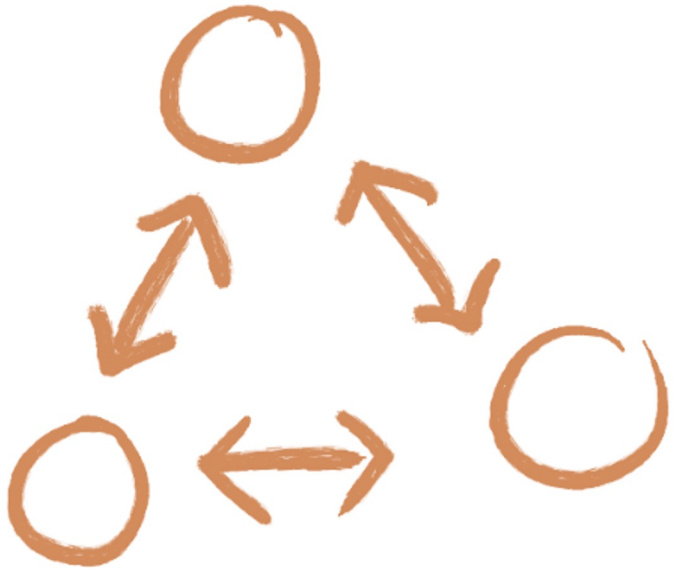


CS3 Site Reports:

- **~16 PB** of data
- **> 20 nodes**
- **> 400.000 users**
- **> 3.5 billion files/dirs**

But…

- Researchers remain **isolated on data islands** because these services aren't interconnected
- **No common, ratified API**
  - Hard to share add-ons between sites
  - Hard to get traction with eScience community
- **Suboptimal knowledge transfer** back to commercial and business environments.
  - We can't make a joined-up front this way



- **3-year** Project
  - Ending in June 2023
  - Led by CERN
- **Objectives**
  - Delivering a **Global Collaboration Service** for researchers, educators, data curators, analysts…
  - Providing an **interoperable platform** to easily share and deploy applications and software components
  - Leveraging the potential of the **CS3 Community** and expanding it

Jupyter is one of the main applications in the ecosystem

www.cs3mesh4eosc.eu

Partners

CERN

Joint Research Centre
JRC

aarnet
Australia's Academic and Research Network

Cubbit

Trust-IT Services
Communicating ICT to markets

cesnet

esade

SURF SARA

PSNC

SWITCH

ailleron

WWU MÜNSTER

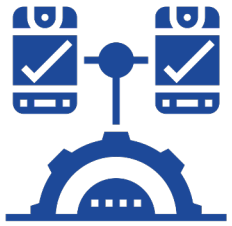
DeiC

Science Mesh

www.sciencemesh.io

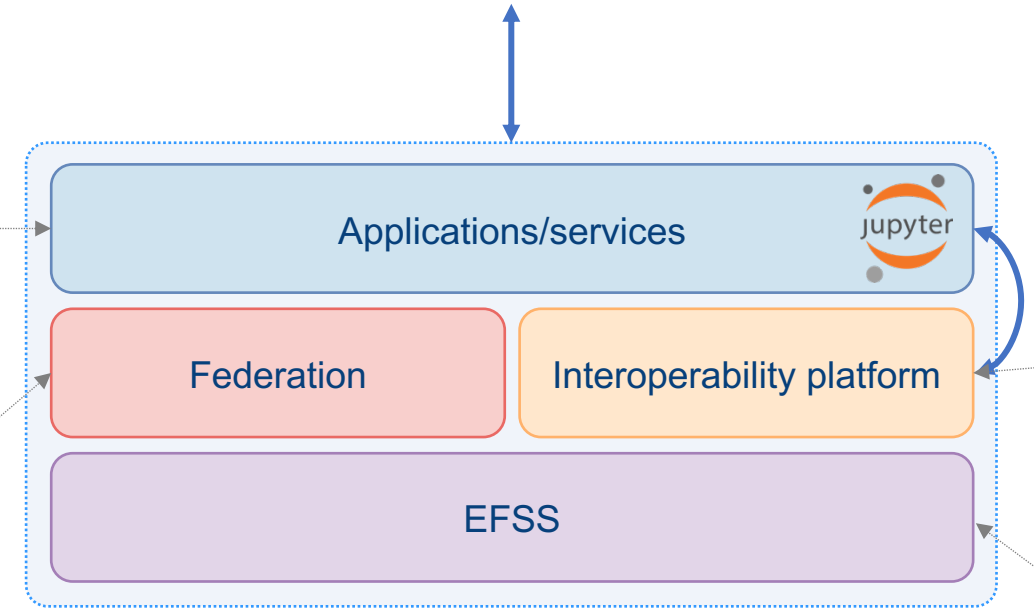
What is ScienceMesh?

- Decentralized **Mesh of EFSS nodes**
  - **Trusted federation of nodes**
- Based on **Open Standards** and **Open Source Software**
- **Federated** environment where researchers can collaborate
  - Usually on the the same UX as if they were local users
- **Application Platform** for distributed collaborative tools



New domain-specific applications developed in the community

Build upon existing infrastructure and long-term efforts
eduGAIN, …



Some applications might need a connector
ScienceMesh plugin for Jupyter

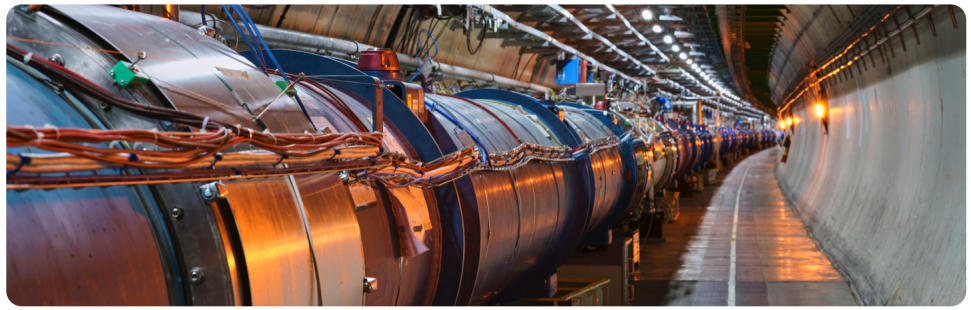
Lightweight add-on
Easy to deploy and install new functionality
OCM, CS3APIs, Reva

Connect to already deployed and commercially supported products
ownCloud, Nextcloud, seafile, …

Scientific Use Cases

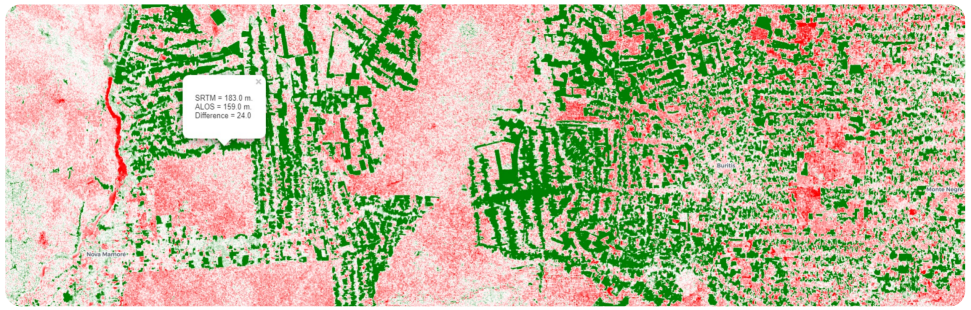
Physics

- High Energy Physics analysis
- Collaborative Analysis
- LHC Accelerator logging and monitoring
- Statistical and operational studies
- Open data access
- Education and outreach

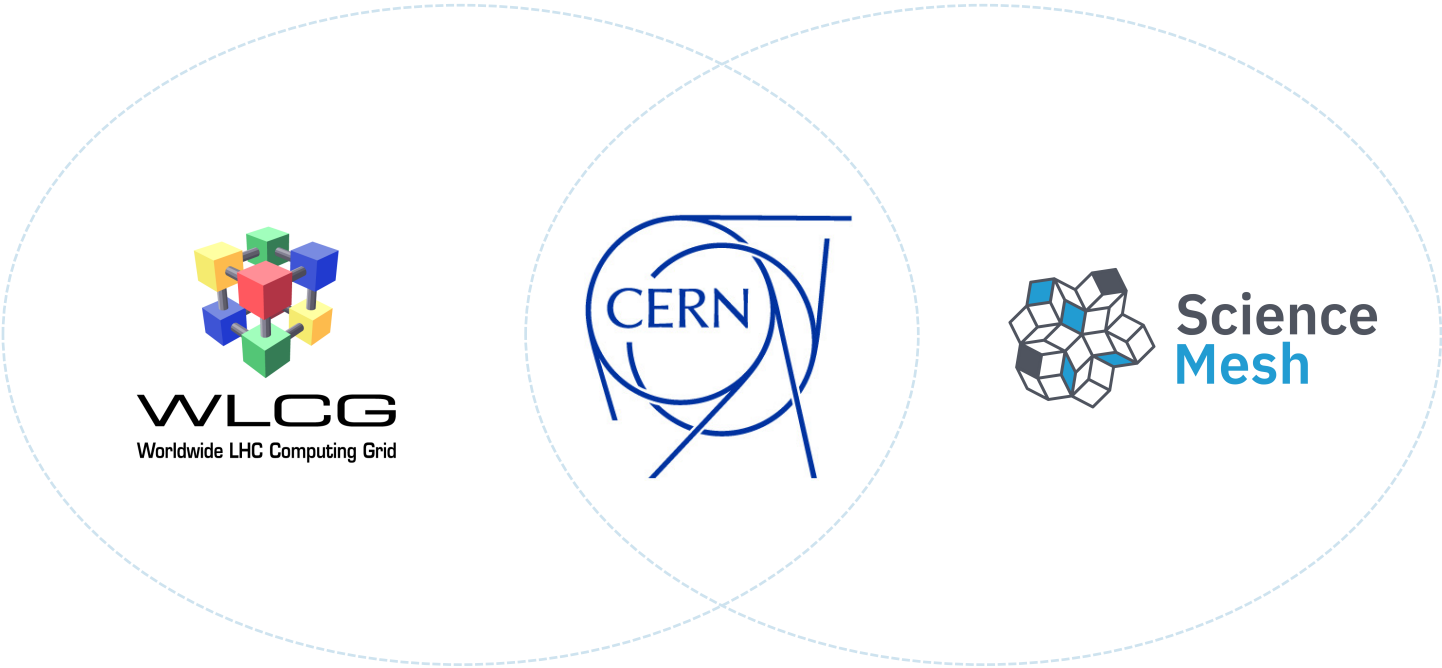


Earth Observation

- Copernicus Earth Observation
- Geo Visualization and Data Exploration
- Interactive dashboards
- Sustainable natural resources and water management (decision-making support); Land degradation (monitoring and assessment); …



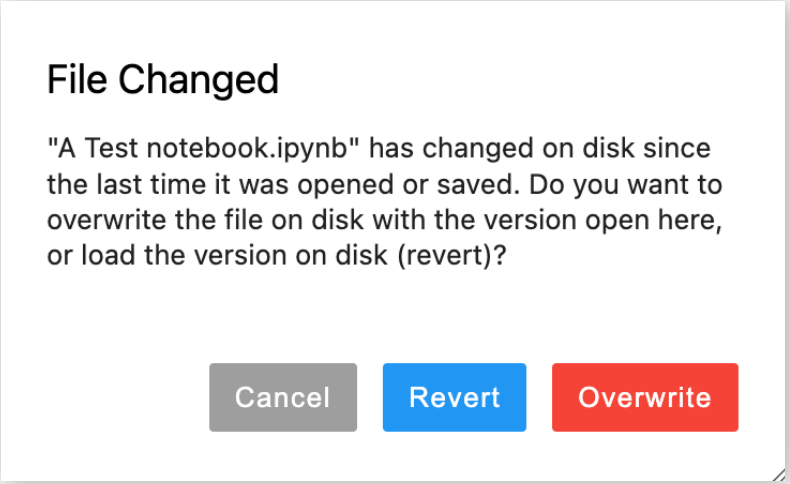
- CERN sits at the intersection of these two communities
  - It has been trying to put it to work for the physics community
  - But this technology also has interest for long tail science or even education use cases outside of it



ScienceMesh Plugin for Jupyter

Current collaboration model for Jupyterlab

- In the beginning, notebooks could not be open in parallel
  - Conflicts would happen, especially on shared filesystems
- Now they can, and their data structures are synchronized
  - This looks awesome!
  - But optimal usage requires sharing the same Jupyter server and kernel (?)



The problems of the current collab model

- A shared filesystem might mean access from different Jupyter servers
  - Or even other applications altogether
  - The concurrent editing does not work fully
- Collaboration requires coordination
  - Which might not always be easy, especially if we don't know who is editing on the other side…
- Sharing the same server + kernel is risky
  - Full access to another user's account, storage and permissions on many resources
- We're not aware of use cases that would benefit from true concurrent editing

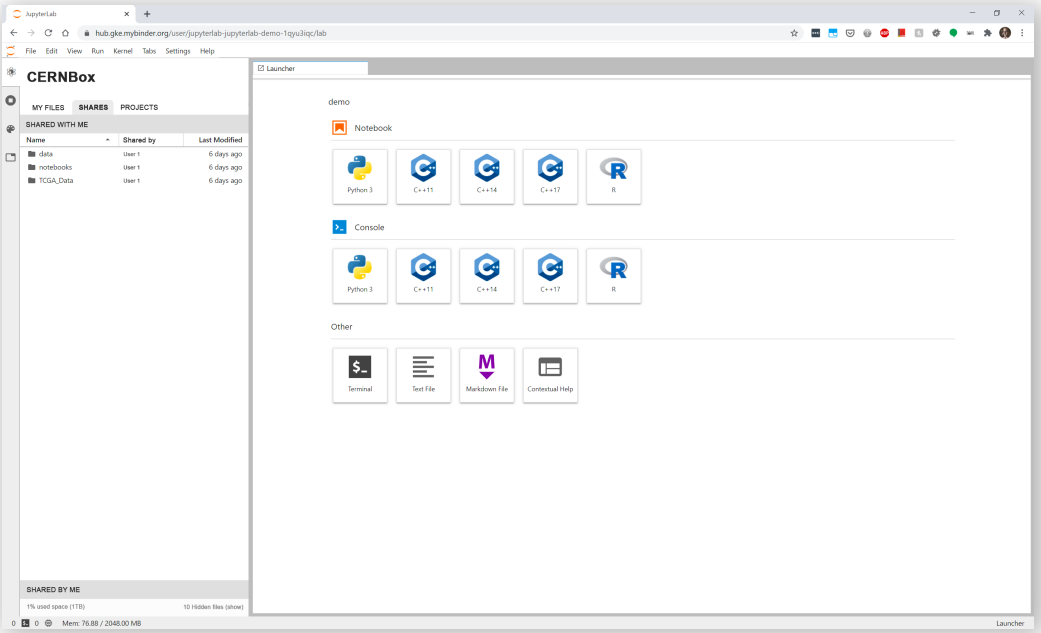
We propose a complementary model better suited for large scale distributed environments

ScienceMesh Plugin for Jupyter

The Science Mesh plugin

- Connect to EFSS (and the Mesh)
  - Using the IOP and CS3APIs
- Generic JupyterLab extension
  - Easy to install and configure
  - Platform/Infrastructure independent

github.com/sciencemesh/cs3api4lab



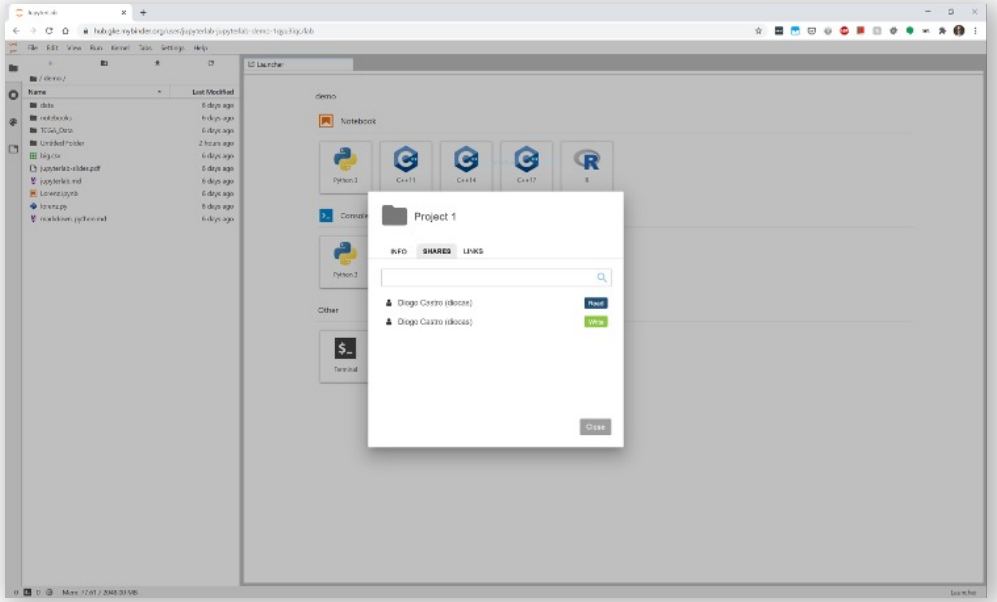
Science Mesh

Software Mind

ailleron

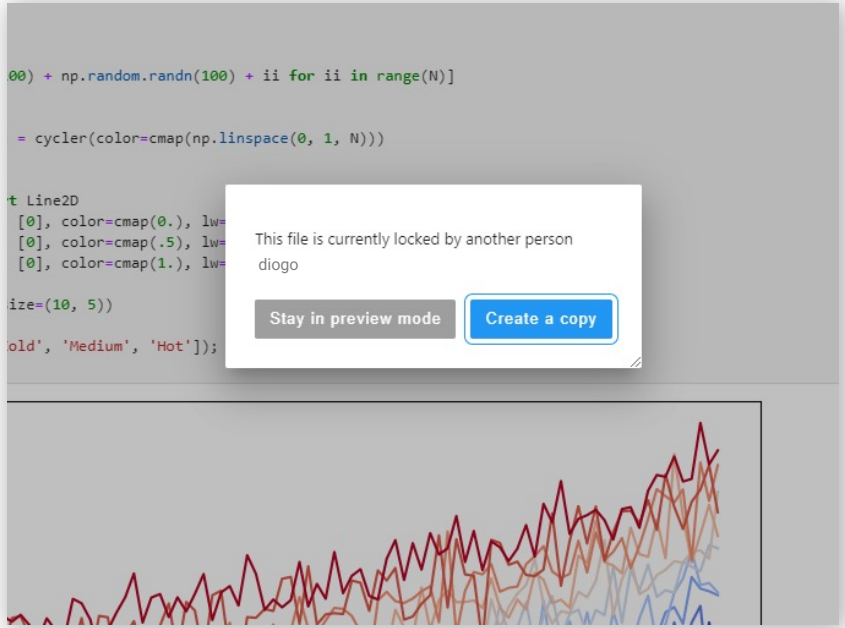
Functionalities

- Same view as EFSS inside Jupyter
  - Access files, different mounts, shares, versions, etc.
- Sharing functionality
  - Share with users or public links
  - Same permissions everywhere
- **Parallel access to notebooks**
  - As alternative to concurrent editing
  - Opening the same notebook without creating conflicts (both locally or remote)
  - Execution environment independence



Parallel access

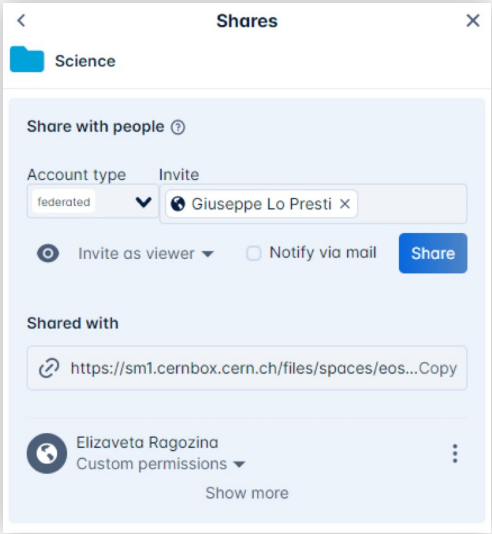
- Uses locks to prevent other users from editing
  - Independent of application/server/etc opening it
    - (they need to support it, but storage backend can enforce it)
  - Keeps track on the user that is holding the lock
- A different approach
  - The first to open locks, the following users can see it in Read Only (or create a copy…)
  - Different but - actually - **complementary** to concurrent editing!



- https://indico.cern.ch/event/1210538/contributions/5317088/

Future work

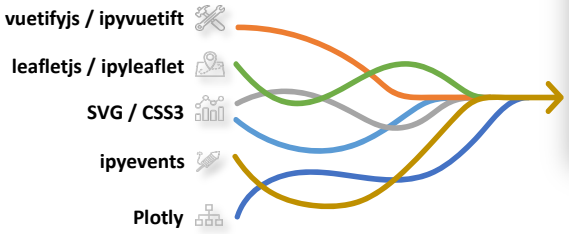
- Clone + merge
  - Visual diff of what changed between original and the copies (nbdime)
  - Allow merge from UI (automated if possible)
- Enable collaboration across Mesh sites
  - Finish the implementation for collaboration between different institutions (federated workflows)
  - Might also allow sharing of other types of resources
    - But doing it with external users might be a security risk
    - So we're just focusing on remote access to files



Other work in ScienceMesh Data Science envs

- **VOIlà Simplification library**
  - Facilitate full exploitation of ipyvuetify/vuetify.js components with less code
    - Consistent usage of widgets variants/colors/ themes
    - Fullscreen apps, responsiveness, multipage, layered popups, etc

vois library



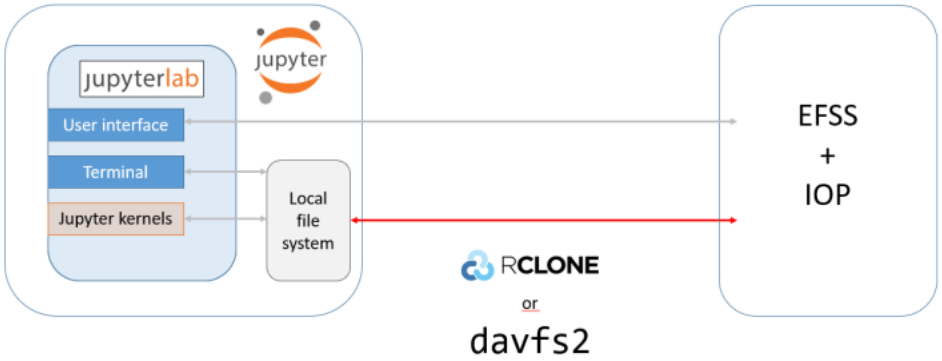
vois.readthedocs.io

Joint Research Centre
JRC

- https://indico.cern.ch/event/1210538/contributions/5207924/

Other work in ScienceMesh Data Science envs

- VaaS - Voilà as a Service
- File system integration (backend)
  - For data access from kernels
  - Tested for compatibility with various EFSS systems
  - Local filesystem access
    - Sync with native EFSS clients
    - Sync with Rclone
  - Online access
    - FUSE mount with davfs2
    - FUSE mount with native clients (i.e EOS client)



CS3 MESH4 EOSC

Connecting European Data

Thank you!

Discover more on...

cs3mesh4eosc.eu

company/cs3mesh4eosc

@cs3mesh4eosc

CS3MESH4EOSC has received funding from the European Union's Horizon 2020 Research and Innovation programme under **Grant Agreement No. 863353**.