A platform for heterogeneous data processing: how IAM supports the PLANET project

Diego Ciangottini
On behalf of the PLANET team

PLANET: **P**ollution **L**ake **AN**alysis for **E**ffective **T**herapy

- **Multidisciplinary project at INFN** which focus on the comprehension of the **effective role of pollution on the Covid-19 pandemic**
- PLANET is **looking for a statistic correlation** between epidemiologic data and pollutants
- Main characterizing traits:
  - Include a **variety of confounds**: clinical data, social and economic metrics, environment data and census data
  - Push the **geo-spatial granularity of the information to the limit**

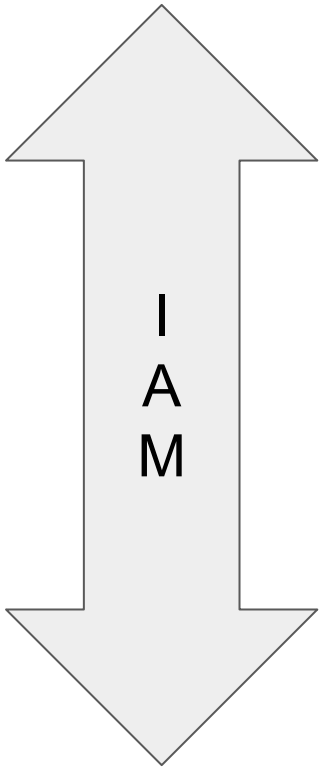
PLANET: the technological challenge

We need a platform for integration, management and analysis of heterogeneous datasets. More in detail:

- **A cloud storage which supports policy based access to data**
  - Need to grant security, trac operations etc
- **Remote data access** via REST and, of course, the possibility to see the cloud storage as a local file system in the remote working station
  - Supporting all OSes
- **A scalable solution for data processing**, fully integrated with cloud storage
  - an authenticated access to a computing platform which should automatically access cloud storage
- **Interactive and batch** processing
  - Web based approach and training pipelines
- **Data visualization**

Matching requirements with technologies

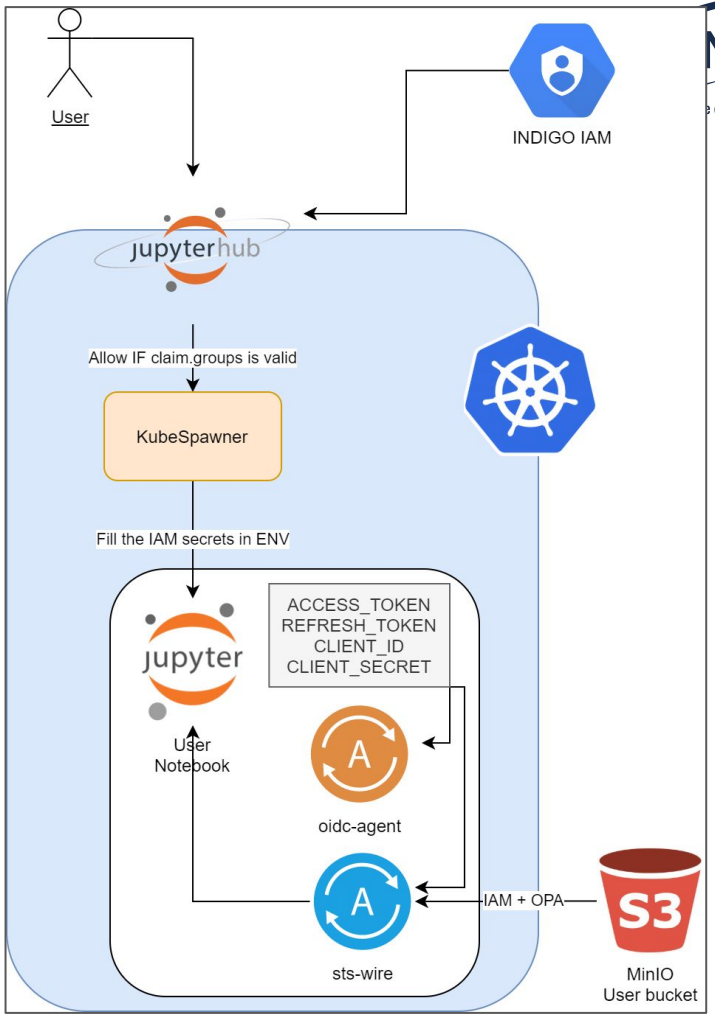
- **Compute resources:**
  - **JupyterHUB** on **k8s**
    - Same available also on bare metal with docker
  - AuthZ based on **IAM groups**
    - Only members of one or more IAM groups can “spawn” a server
  - Notebooks must be able to see a persistent user space
  - Notebooks must integrate cloud storage
- **Data storage:**
  - **MINIO** object storage integrated with **IAM** (through AWS STS standard)
  - Fine-grained AuthZ achieved through **OpenPolicy agent**
- **Access to data:**
  - Posix through **sts-wire, authN/Z via IAM token**
  - API -> aws s3 compatible clients
  - WebUI -> Minio WebUI



Compute Resources: “All you need is JupyterHUB”

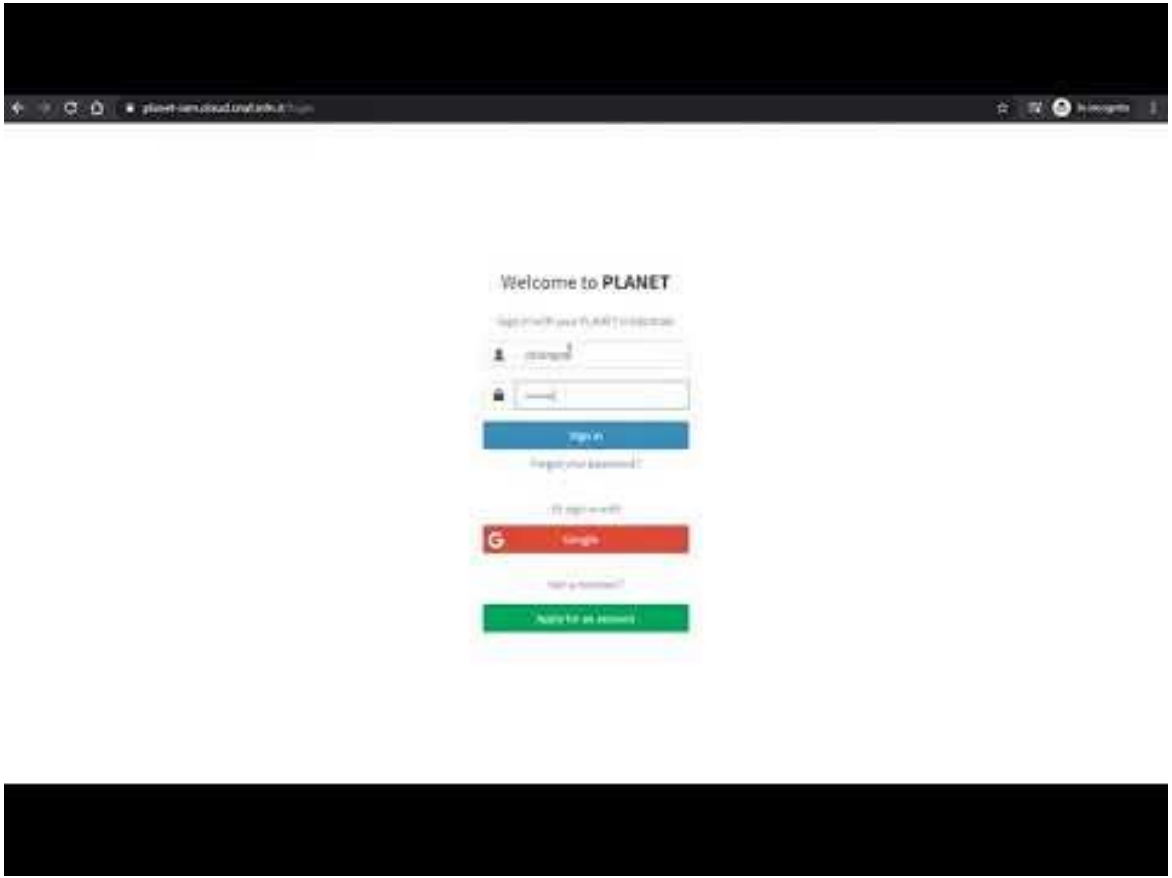
IAM-based AuthN/Z

- **Auto/self registration of IAM client at deployment time**
- customized **kubespawner** for:
  - **extraction of groups claims**
  - authorize notebook spawn **only for members of certain groups**
- **user creds are passed to spawned notebooks**
  - used to setup oidc-agent
  - used by sts-wire to mount s3 buckets at notebook startup - see later



From user perspective...

https://youtu.be/AYYH_014_HY

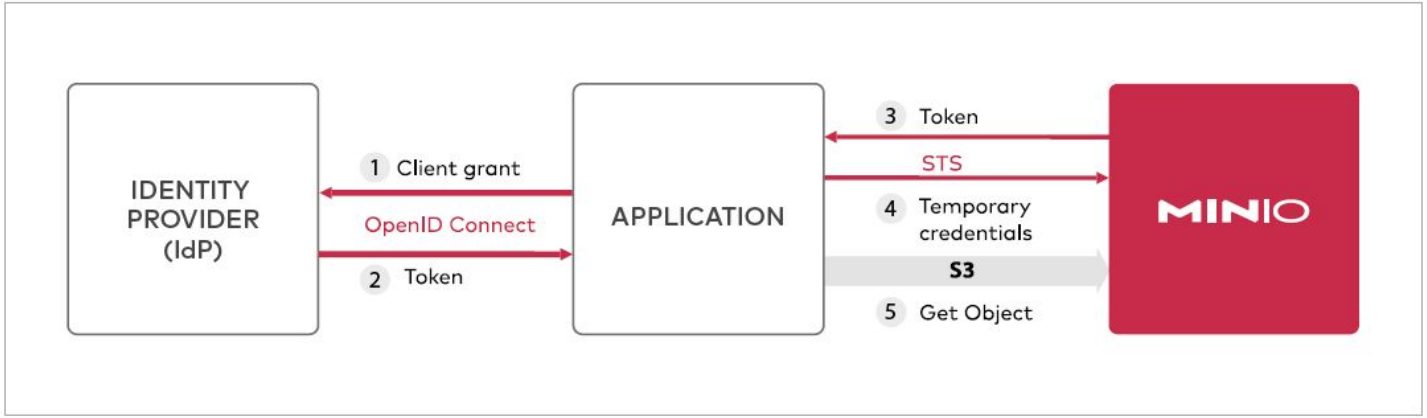


Data Storage: MINIO S3

- Ephemeral: **thanks to IAM based data access there is no need to pre-create any account**
  - **AuthN:** MINIO temporary STS credentials through AWS “AssumeRoleWithWebIdentity” flow
  - **AuthZ:** MINIO native integration with OpenPolicyAgent instance
    - adopted to introduce **fine grained authZ policies based on IAM access token information**

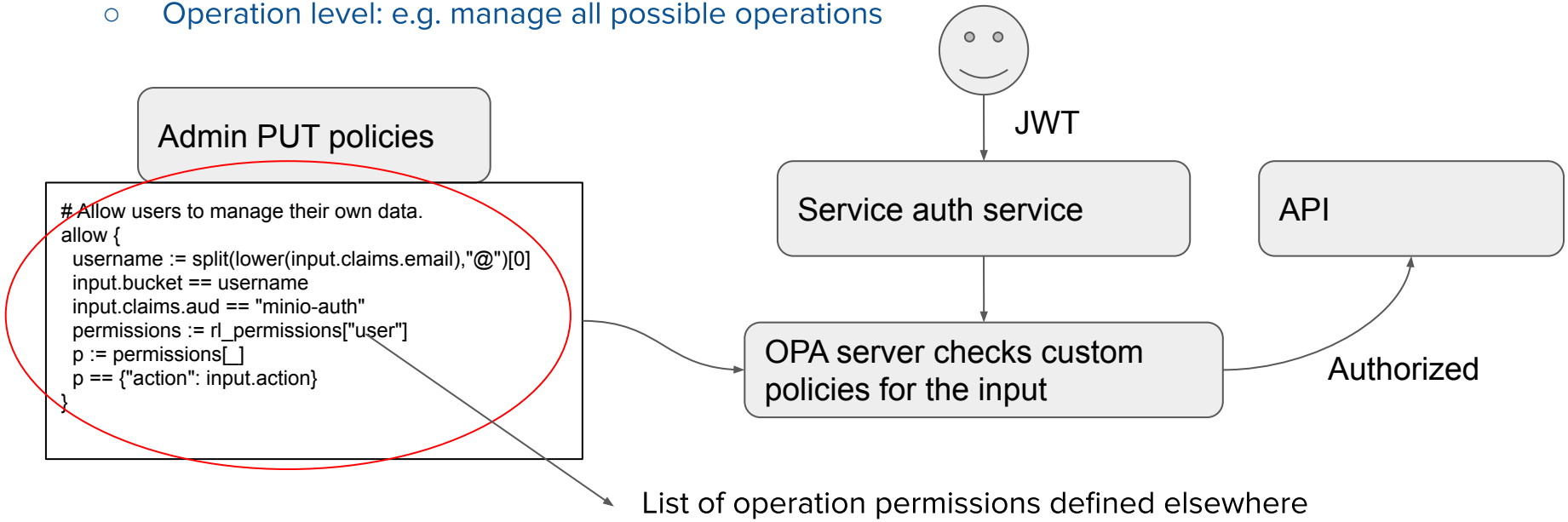
AuthN: AWS STS temporary credentials

- MinIO supports AWS temporary credentials
- **We leveraged this to integrate the AuthN with IAM**:
  - Using assume-role-with-web-identity flow
- Native support for this ➔ 0 code required, only configuration and a IAM client



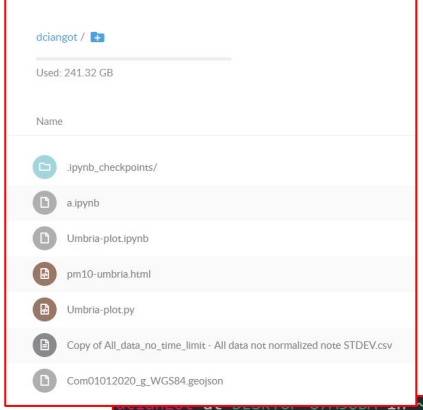
AuthZ: Open Policy Agent

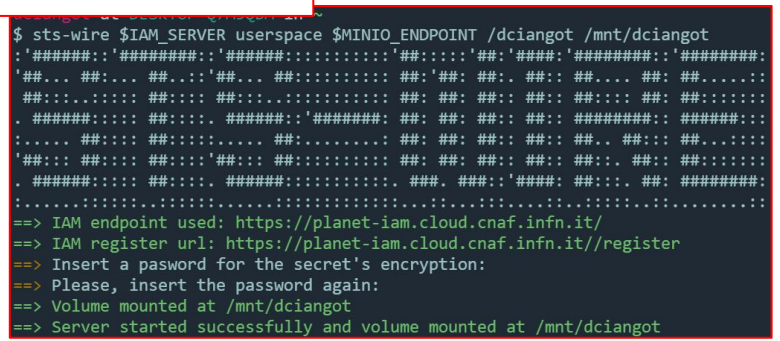
- MinIO authZ is capable of **delegating the decision to** OpenPolicyAgent
- **Easy to extend and different layer of granularity:**
  - User level: e.g. exclusive rw access to a user bucket based on IAM preferred username
  - Group level: e.g. rw or ro access based on IAM group claim
  - Operation level: e.g. manage all possible operations

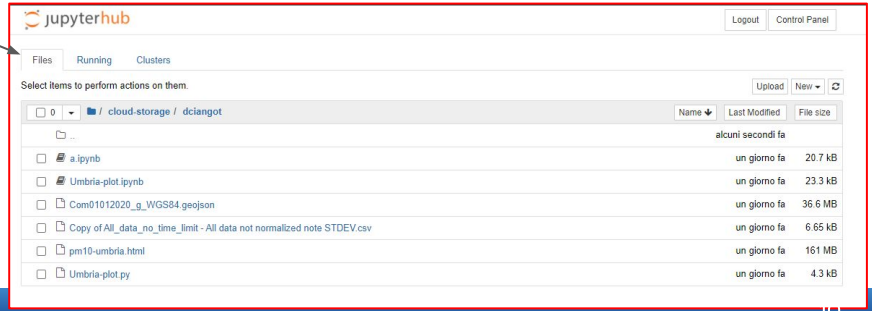


Access to data

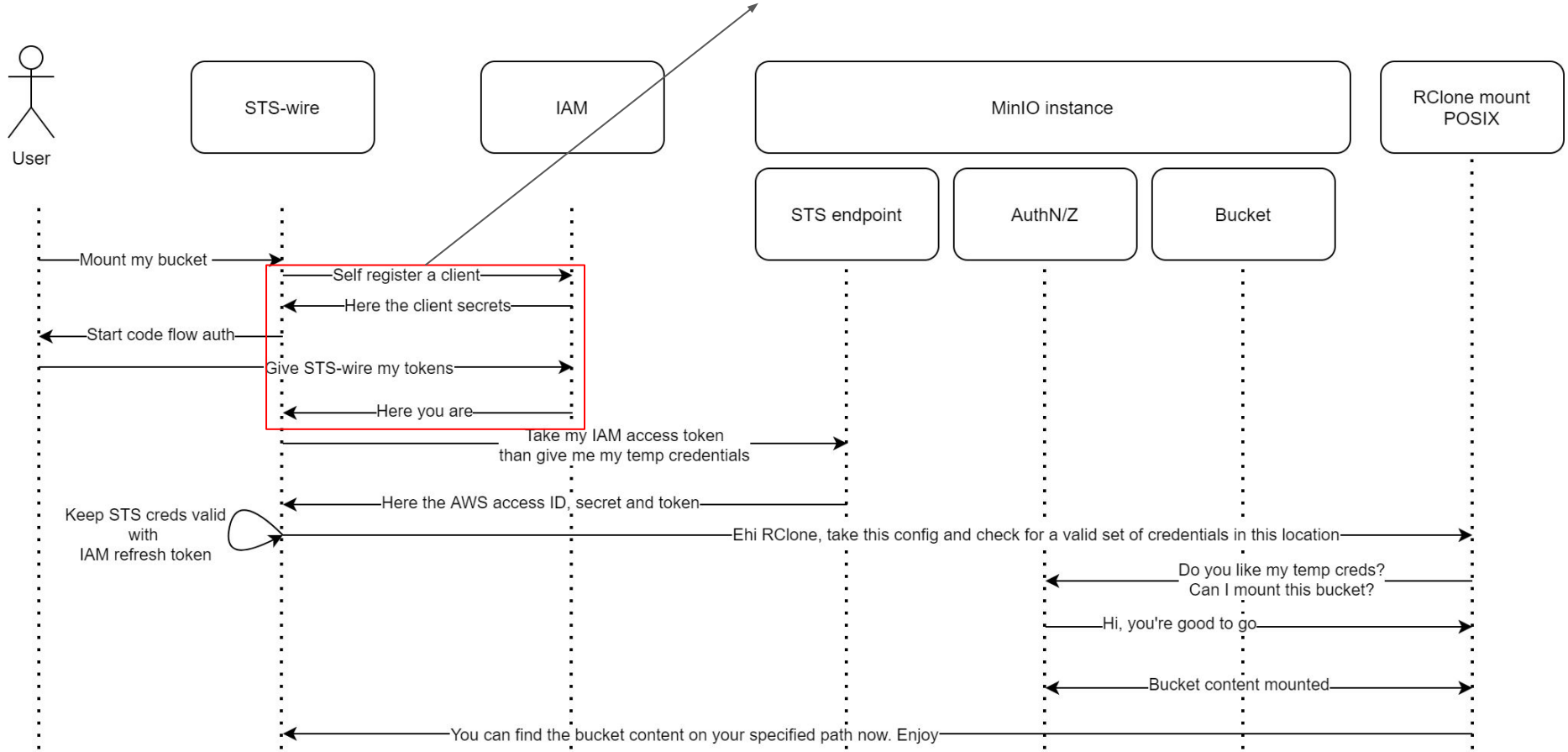
- **Access to data** in this ephemeral integration is possible via:
  - **MINIO WebUI**
  - STS-wire for **posix like access**. A CLI able to:
    - exchange a IAM access token (got through auth code flow) with MINIO temporary STS credentials through AWS “AssumeRoleWithWebIdentity” flow
    - refresh and use those credentials to mount locally a bucket using RClone mount
  - **Any S3 compatible client** (available for any language in practice)







STS-wire workflow

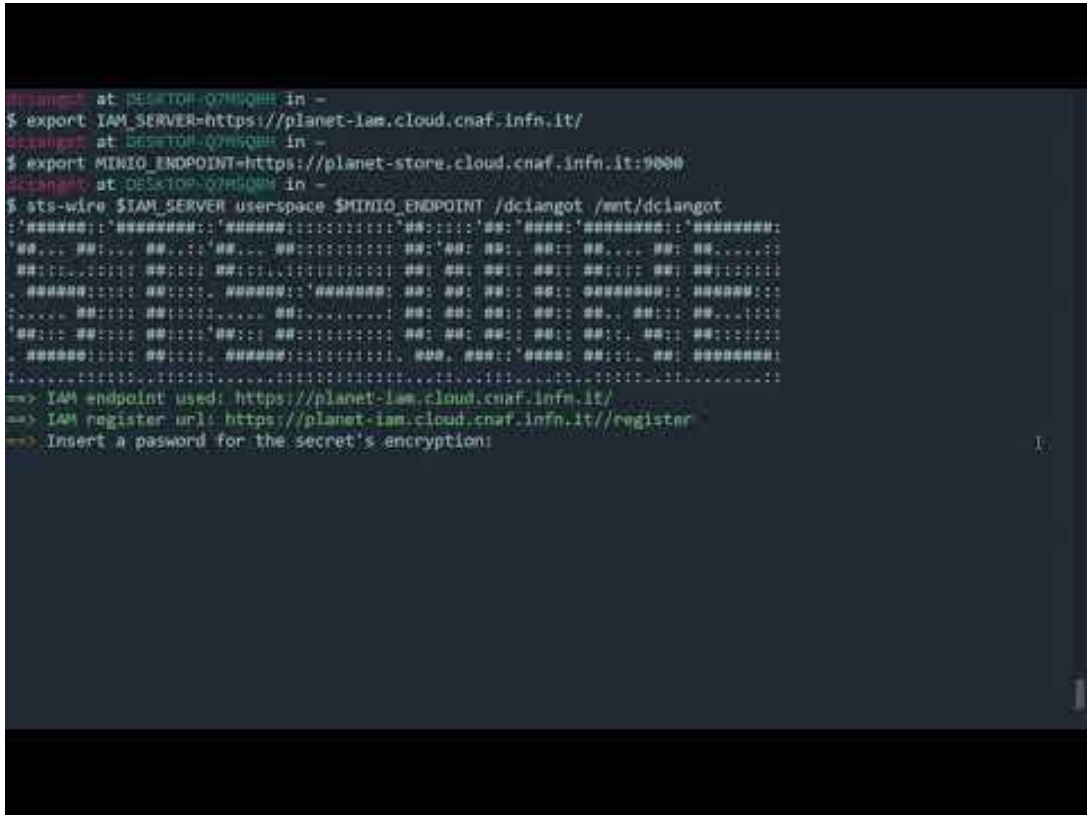


User experience, in a nutshell

Full integration of all the components

- Web UI
- STS-wire:
  - jupyter auto mounted space
  - Local workstation posix folder

https://youtu.be/5ZOKFIdQ1NU



Consideration (toward a full stack with IAM as AAI)

- Thanks to IAM capabilities we can **provide the user with a single entrypoint to work with**
  - This include **data and compute infrastructure**
- With the presented design we allow user to **bring up on-demand a personal UI fully configured to interact with all the data and compute tools**
  - the very same UI environment on user laptop
  - No need to register or create any account other than the IAM one, important for both users and admins

Summary

- **IAM OpenID-Connect support supports flexible integration** with various existing software
  - Often using **standard workflow already natively supported**
- We plan **to expand it further letting batch systems manage tokens on user behalf** to access data
  - HTCondor has a first prototype already
- Nothing of what we presented is PLANET specific, **you can use the same set of tools everywhere with whatever set of data**
  - In fact already exported to other use case at INFN and on CMS analysis facility prototype in ESCAPE project
- **Compliant with INFN cloud service deployment strategy** (see Marica yesterday talk)
- For more information contact : ciangottini@pg.infn.it