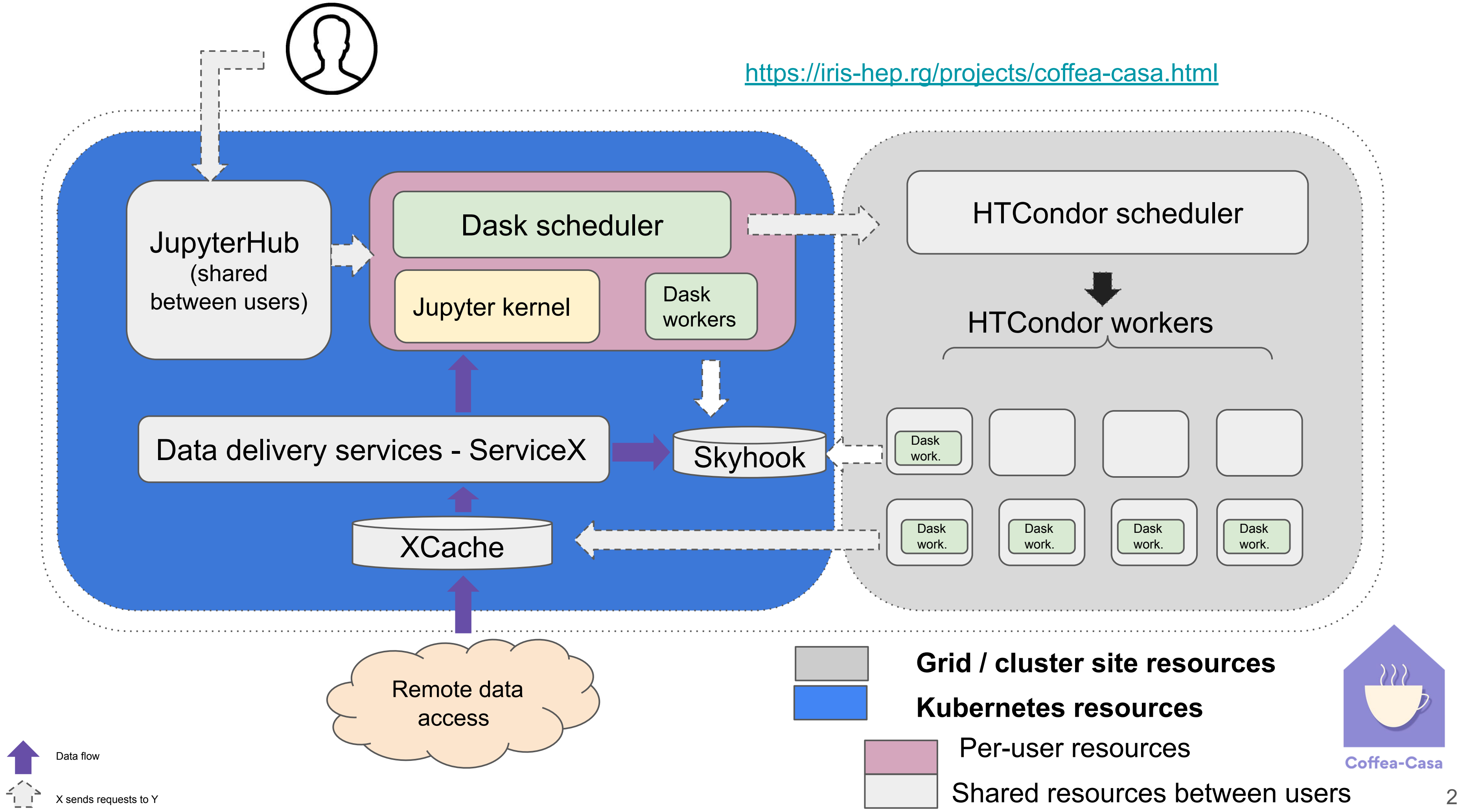
End-to-end AGC walkthrough with facility focus

https://iris-hep.rg/projects/coffea-casa.html



User requested features

CVMFS enabled @ coffea-casa
(we are limited using Ubuntu image, need to support more flavours)

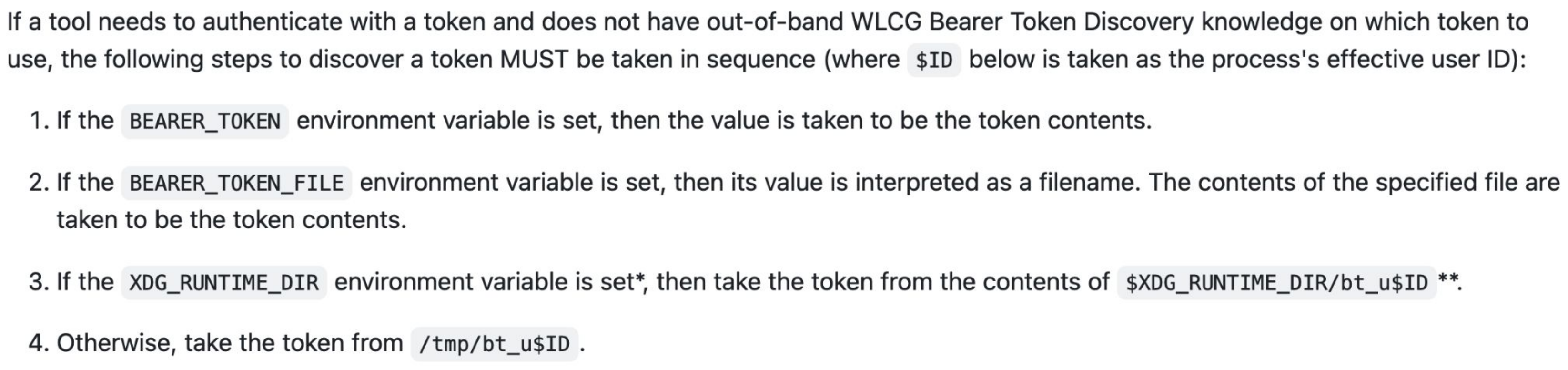
Still not enabled
(we are limited still by using Ubuntu image, WE NEED to move to CC7/ALMA)

Dasgoclient (CMS)

Still not enabled
(working to test if it works with tokens)

Bearer tokens

- The token discovery procedure https://github.com/WLCG-AuthZ-WG/bearer-token-discovery/blob/master/specification.md
- More details about WLCG JWT profile: https://github.com/WLCG-AuthZ-WG/common-jwt-profile/blob/master/profile.md



If a tool needs to authenticate with a token and does not have out-of-band WLCG Bearer Token Discovery knowledge on which token to use, the following steps to discover a token MUST be taken in sequence (where `$ID` below is taken as the process's effective user ID):

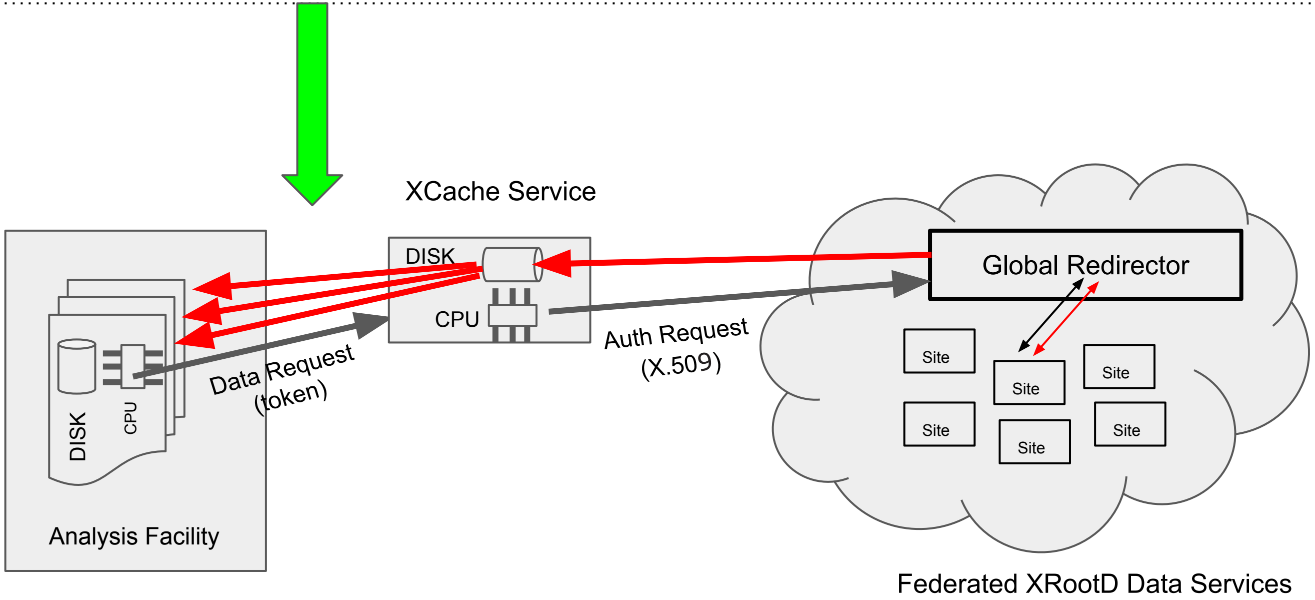
1. If the `BEARER_TOKEN` environment variable is set, then the value is taken to be the token contents.
2. If the `BEARER_TOKEN_FILE` environment variable is set, then its value is interpreted as a filename. The contents of the specified file are taken to be the token contents.
3. If the `XDG_RUNTIME_DIR` environment variable is set*, then take the token from the contents of `$XDG_RUNTIME_DIR/bt_u$ID` **.
4. Otherwise, take the token from `/tmp/bt_u$ID` .

Tokens at coffea-casa

- Pregenerated token available directly in user session
  - For CMS coffea-casa instance we use token issuer `https://cms-auth.web.cern.ch/`
  - At UChicago it is using ATLAS IAM instance
- The same token is used for multiple services:
  - XCache
  - ServiceX

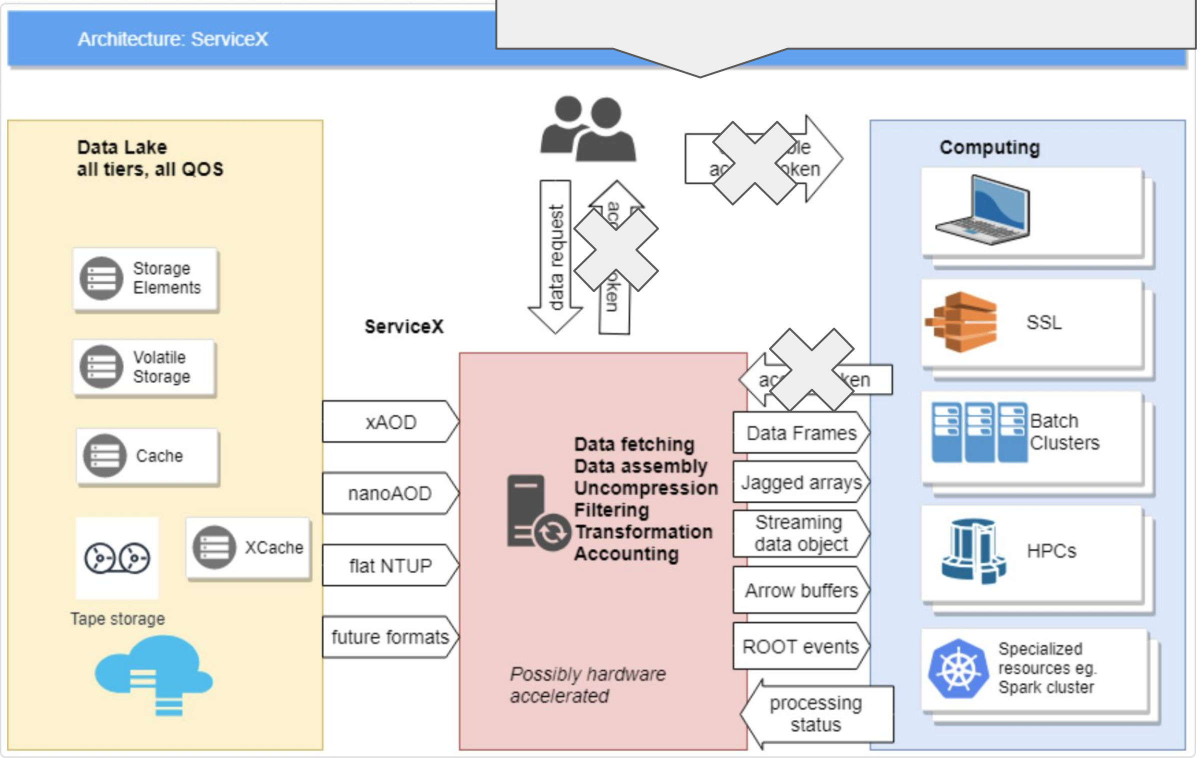
Xcache and bearer token

no GSI credential within the facility, **the auto-generated data access token can be used to authenticate with an proxy service based on XRootD/XCache**



ServiceX and tokens

ServiceX now support WLCG token discovery procedure



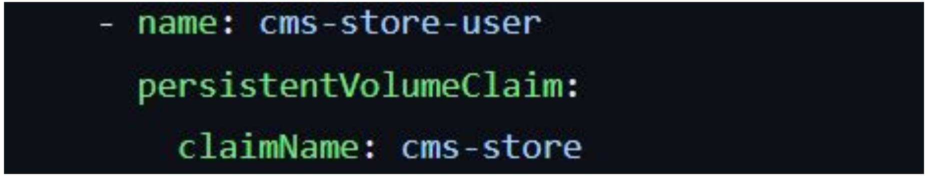
Shared Filesystem Coffea-jupyterhub

- Using BindFS the shared Ceph directory */tier2/cms/store/user* is mapped with appropriate permissions to */mnt/cms/store/user* on the workers in the flatiron cluster

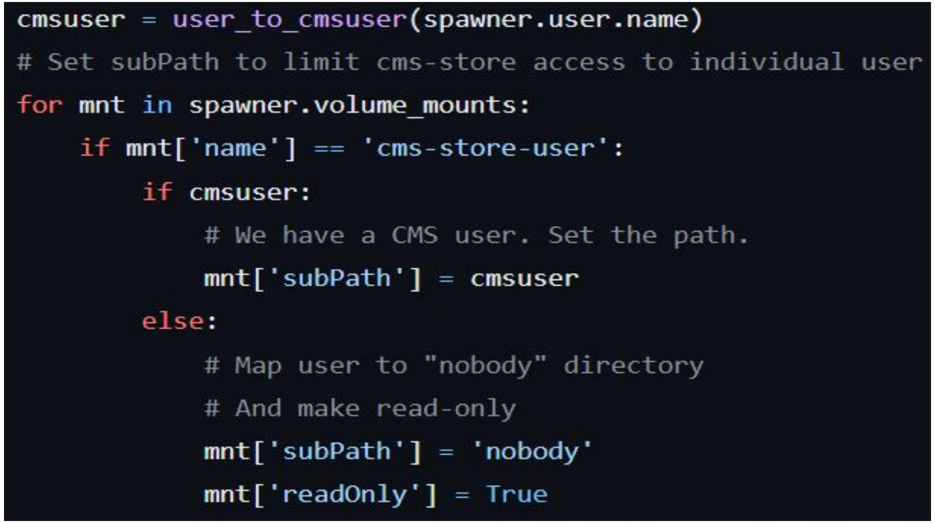
- A cms-store PV/PVC is created for the new */mnt/cms/store/user* directory

- The jupyterhub helm chart is configured to mount the newly created *cms-store* directory into every user's jupyterhub pod upon start.

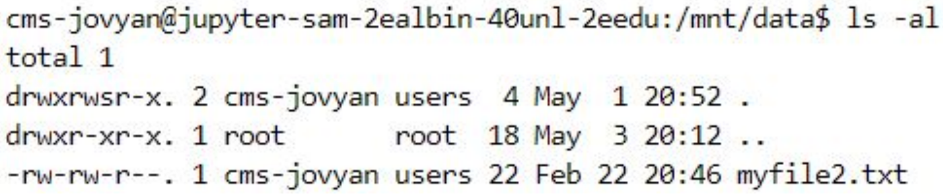
- A creation pre-hook python script then reads the users email and then configures the cms-store volume to mount only the subpath belonging to the user. If the user does not have a folder they're given an empty read-only data directory inside their pod under */mnt/data*.



```
    - name: cms-store-user
      persistentVolumeClaim:
        claimName: cms-store
```

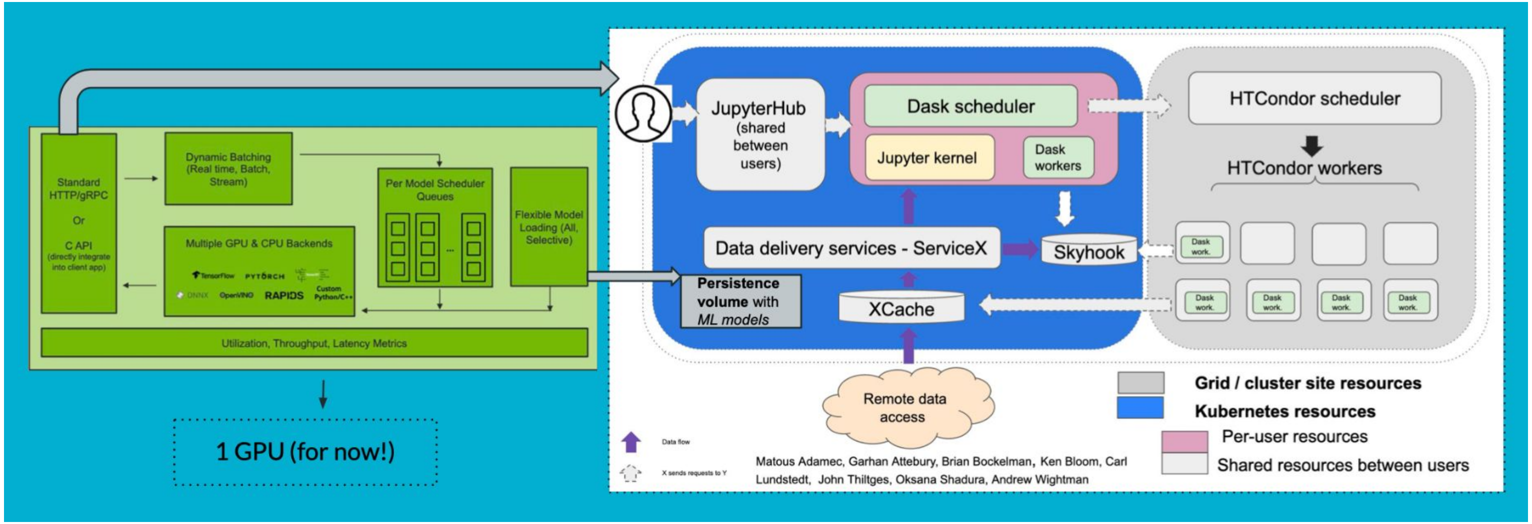


```
cmsuser = user_to_cmsuser(spawner.user.name)
# Set subPath to limit cms-store access to individual user
for mnt in spawner.volume_mounts:
    if mnt['name'] == 'cms-store-user':
        if cmsuser:
            # We have a CMS user. Set the path.
            mnt['subPath'] = cmsuser
        else:
            # Map user to "nobody" directory
            # And make read-only
            mnt['subPath'] = 'nobody'
            mnt['readOnly'] = True
```



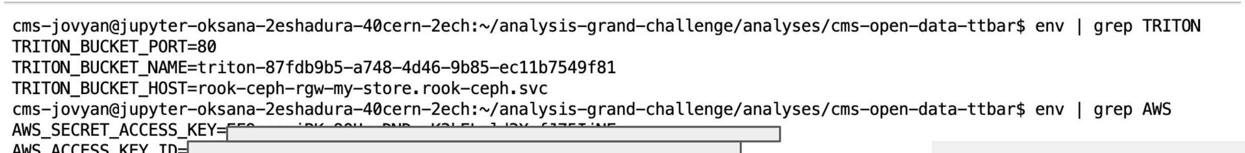
```
cms-jovyan@jupyter-sam-2ealbin-40unl-2eedu:/mnt/data$ ls -al
total 1
drwxrwsr-x. 2 cms-jovyan users   4 May  1 20:52 .
drwxr-xr-x. 1 root       root   18 May  3 20:12 ..
-rw-rw-r--. 1 cms-jovyan users 22 Feb 22 20:46 myfile2.txt
```

Triton



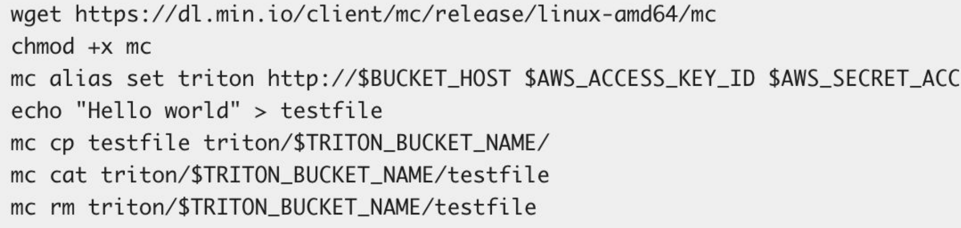
Standard HTTP/gRPC
Or
C API
(directly integrate into client app)
Dynamic Batching (Real time, Batch, Stream)
Multiple GPU & CPU Backends
TensorFlow
PYTORCH
ONNX
OpenVINO
RAPIDS
Custom Python/C++
Per Model Scheduler Queues
Flexible Model Loading (All, Selective)
Utilization, Throughput, Latency Metrics
1 GPU (for now!)
JupyterHub (shared between users)
Dask scheduler
Jupyter kernel
Dask workers
HTCondor scheduler
HTCondor workers
Dask work.
Data delivery services - ServiceX
Skyhook
Persistence volume with *ML models*
XCache
Remote data access
Data flow
X sends requests to Y
Grid / cluster site resources
Kubernetes resources
Per-user resources
Shared resources between users

Matous Adamec, Garhan Attebury, Brian Bockelman, Ken Bloom, Carl Lundstedt, John Thiltges, Oksana Shadura, Andrew Wightman



```
cms-jovyan@jupyter-oksana-2eshadura-40cern-2ech:~/analysis-grand-challenge/analyses/cms-open-data-ttbar$ env | grep TRITON
TRITON_BUCKET_PORT=80
TRITON_BUCKET_NAME=triton-87fdb9b5-a748-4d46-9b85-ec11b7549f81
TRITON_BUCKET_HOST=rook-ceph-rgw-my-store.rook-ceph.svc
cms-jovyan@jupyter-oksana-2eshadura-40cern-2ech:~/analysis-grand-challenge/analyses/cms-open-data-ttbar$ env | grep AWS
AWS_SECRET_ACCESS_KEY=[redacted]
AWS_ACCESS_KEY_ID=[redacted]
```

Environment integration

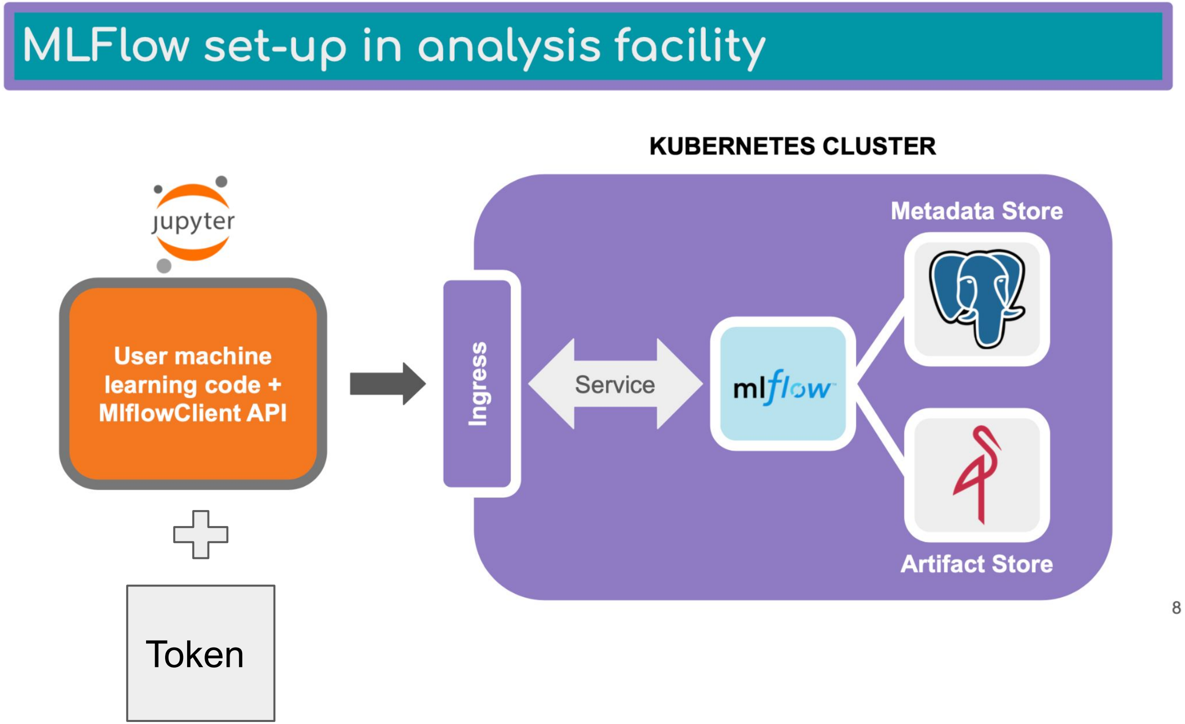


```
wget https://dl.min.io/client/mc/release/linux-amd64/mc
chmod +x mc
mc alias set triton http://$BUCKET_HOST $AWS_ACCESS_KEY_ID $AWS_SECRET_ACC
echo "Hello world" > testfile
mc cp testfile triton/$TRITON_BUCKET_NAME/
mc cat triton/$TRITON_BUCKET_NAME/testfile
mc rm triton/$TRITON_BUCKET_NAME/testfile
```

How to load model in S3

MLFLow

- We are still missing deployment at coffea-casa **(WIP)**
- Many thanks to Ben, we are using MLFlow instance at NSCA
- Need to think how to integrate it in facility
    - ○ Can we use the same token as for Servicex?


MLFlow set-up in analysis facility
KUBERNETES CLUSTER
jupyter
User machine learning code + MlflowClient API
Token
Ingress
Service
mlflow
Metadata Store
Artifact Store