Bryan Hess
Kurt Strosahl
Wesley Moore

CI integration of JLAB and the OSG Fabric of services in support of HEP collaborative research

Pascal Paschos
Jason Patton

HEPiX 2022 Online Workshop 2022

Partnership between JLAB and OSG

- OSG Collaboration Support Services and JLAB have long standing partnership in enabling HEP lab science on the OSG Fabric (Grid).
- Lab managed infrastructure is connected to OSG administered resources in providing managed access to dedicated pool of computing resources - which includes the JLAB computing farm from other institutions.
- Effort is coordinated via weekly meetings for strategy or planning and daily communication - as needed - over slack.
- OSG maintains a different communication channel with JLAB experiments in regards to workflow performance & distributed computing infrastructure access to sites in the US & Europe.

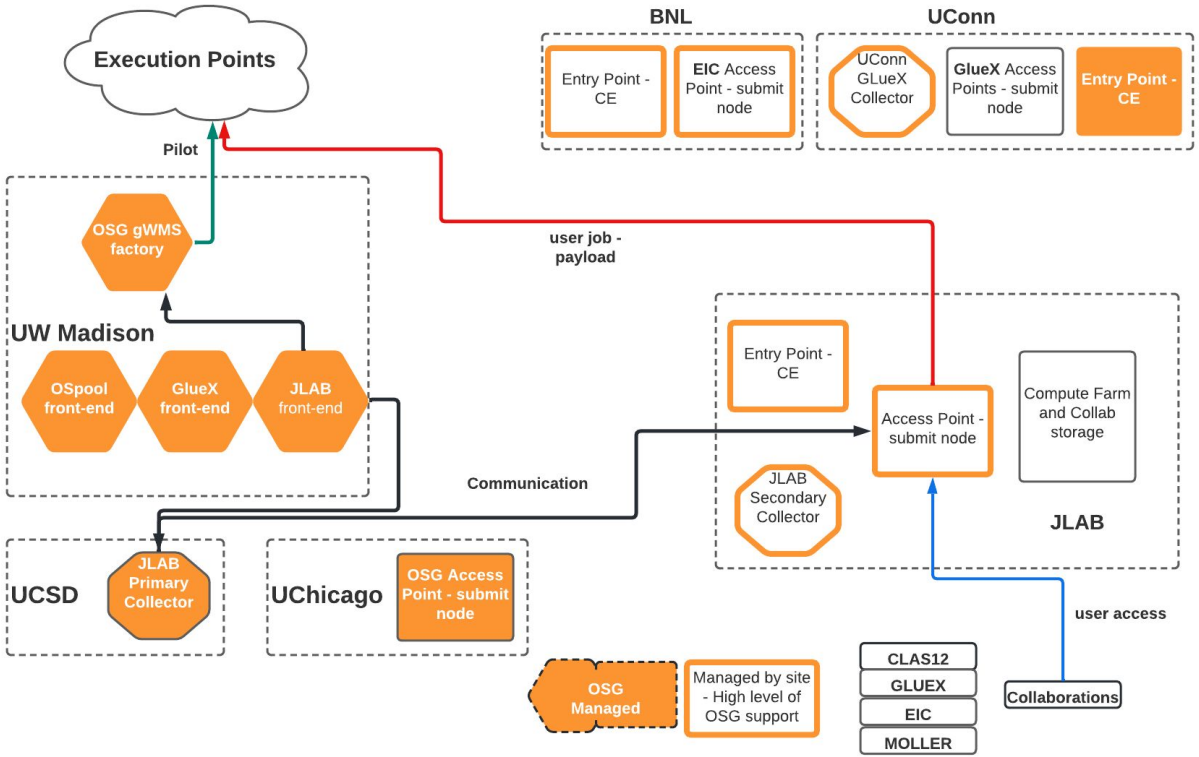
Infrastructure at a glance

- One production submit node, one for development and one upcoming
- Two collectors (Primary at UCSD, secondary at JLAB in HA - High availability mode)
- One Glidein-WMS Front-End (FE) and one FE for Development (UW Madison)
- One entry Point at JLAB (CE) for incoming jobs from the gWMS factory enabled for CLAS12, GLUEX, MOLLER and EIC projects
- 30 TB of staging storage at JLAB for data egress/ingress to/from OSG pool and 7 PB of backend storage for the experiments
- Vault Scitoken server

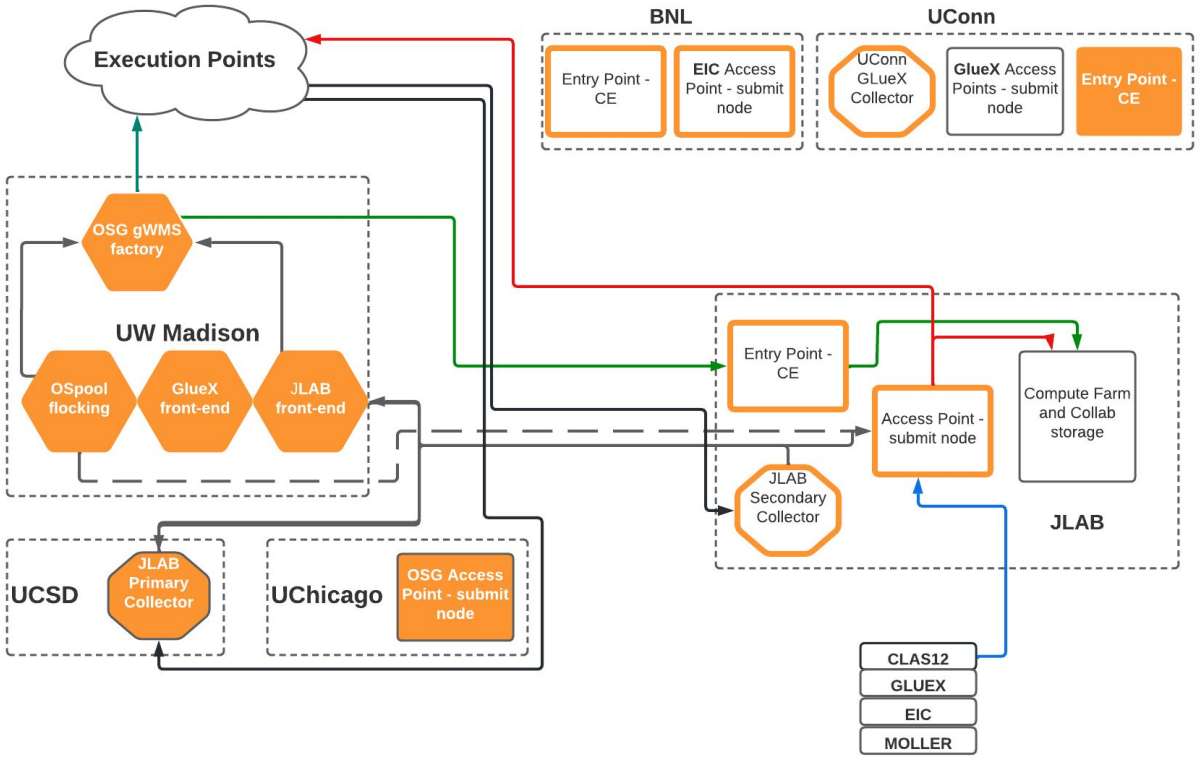
Updates this past year

- Deployed a Science DMZ network for data transfer nodes for and a science portals network for less data-intensive services.
- The Lab's existing 2x10Gbit ESNet connections are being upgraded to 2x100Gbit in 2022 which will enable reconstruction workflows besides simulations
- All JLAB infrastructure has been updated to latest OSG/HTCondor software stack and authentication methods
- Previous entry point (submit node) was decommissioned and rebuilt as a development node to test new deployments before operationalizing
- New entry point has been put in production to support all 4 collaborations. Additional nodes are being deployed in order to split the load and simplify configurations.

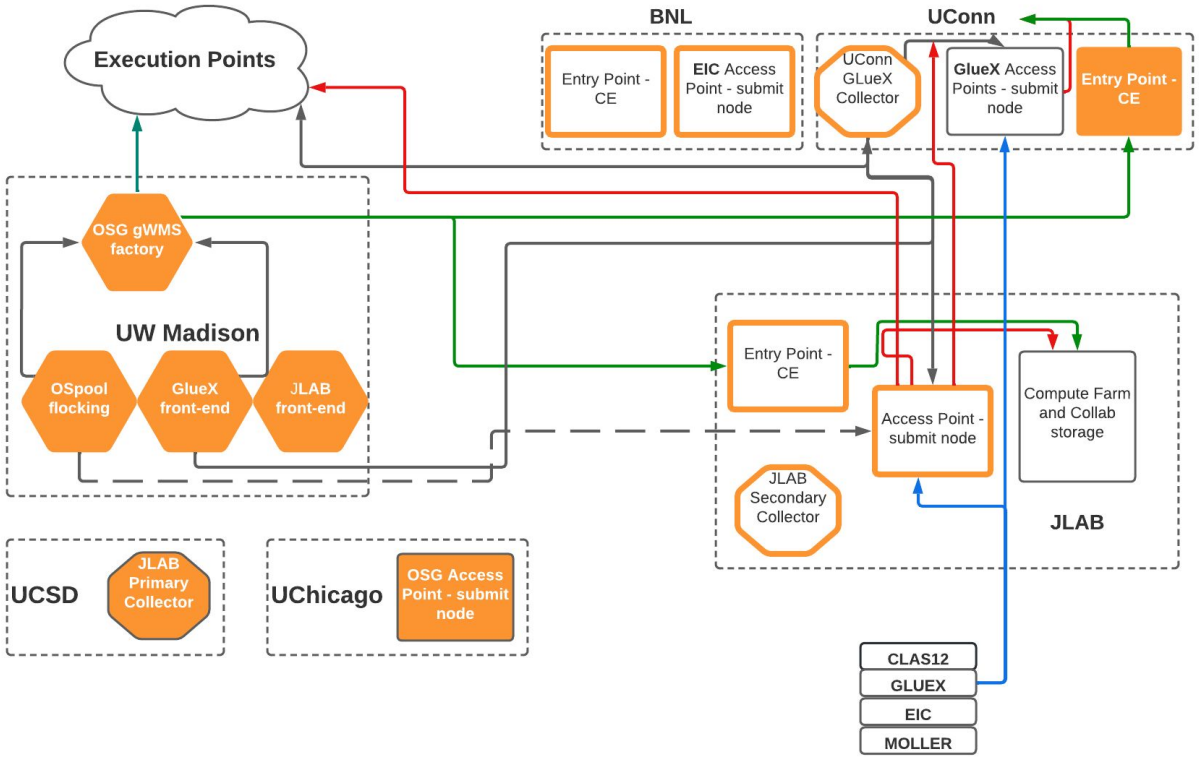
JLab Experiments on the OSG



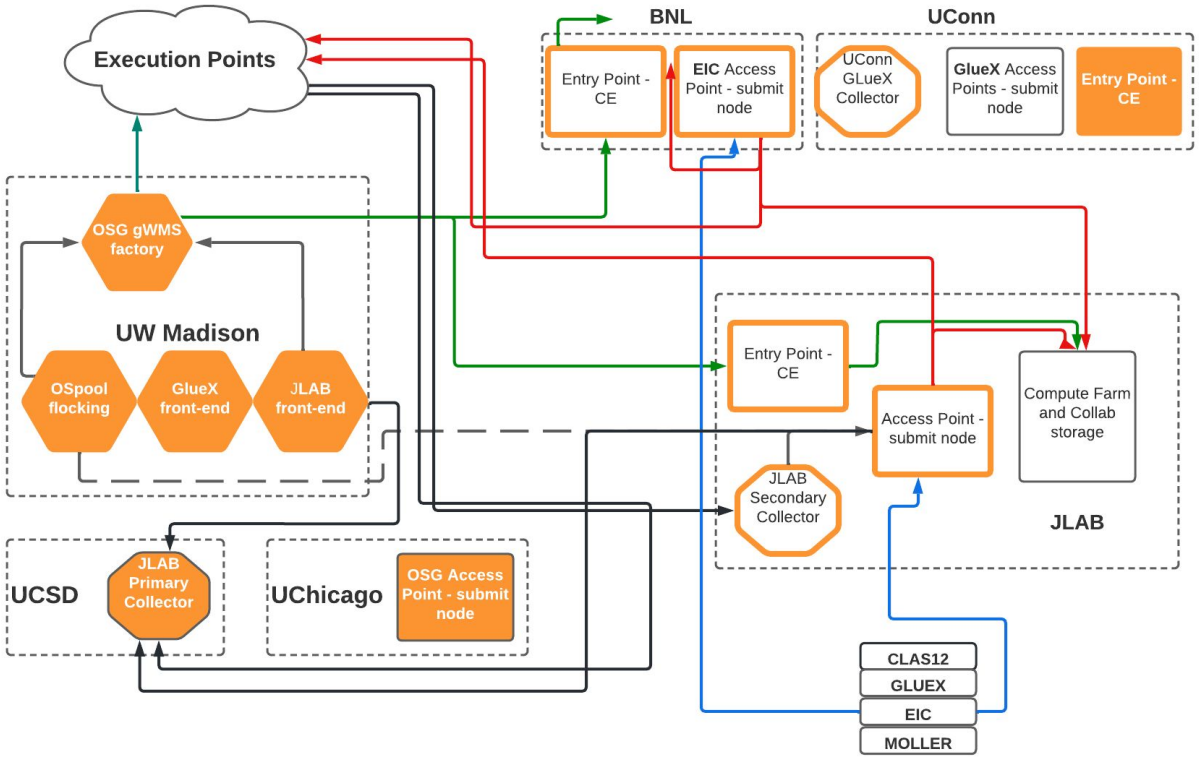
JLab Experiments on the OSG



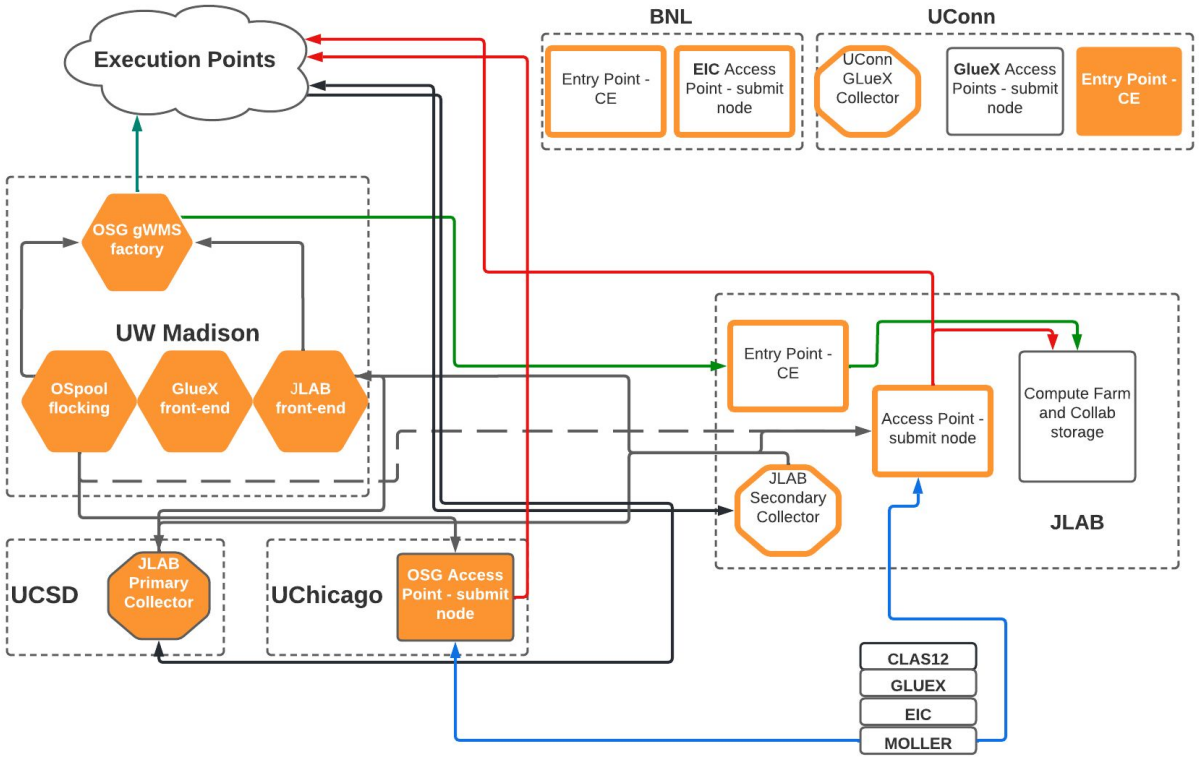
JLab Experiments on the OSG



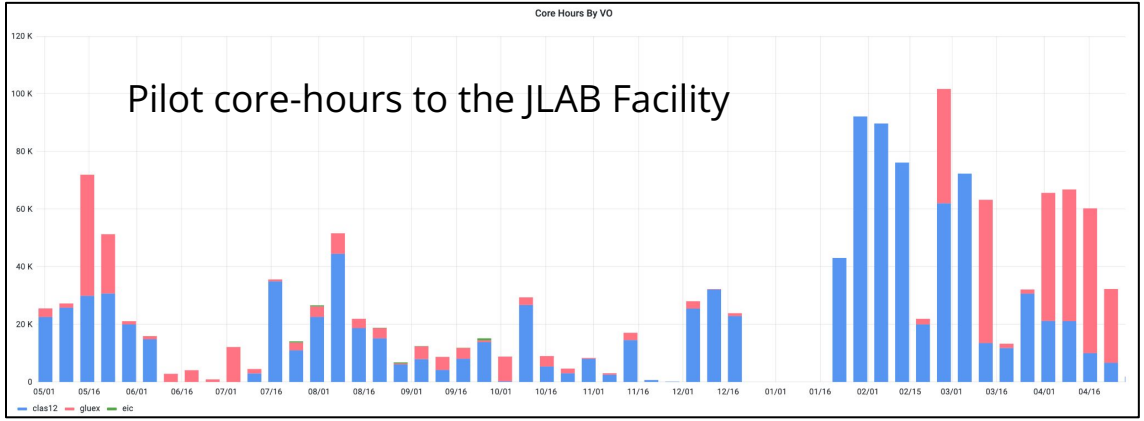
JLab Experiments on the OSG

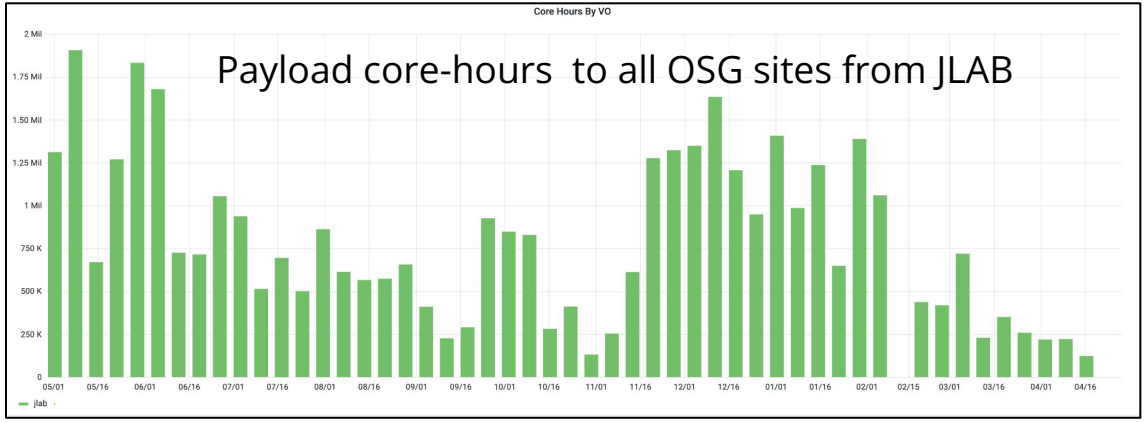


JLab Experiments on the OSG



JLAB VO Jobs on OSG





- During the past year, JLAB run 1.4 million core hours of pilots jobs from the OSG factory. Dominated by GlueX and CLAS12.
- During the past year, JLAB submitted and run 40 million core-hours at the OSG in a mix of opportunistic and dedicated sites (which includes JLAB itself).

OSG Token Transition

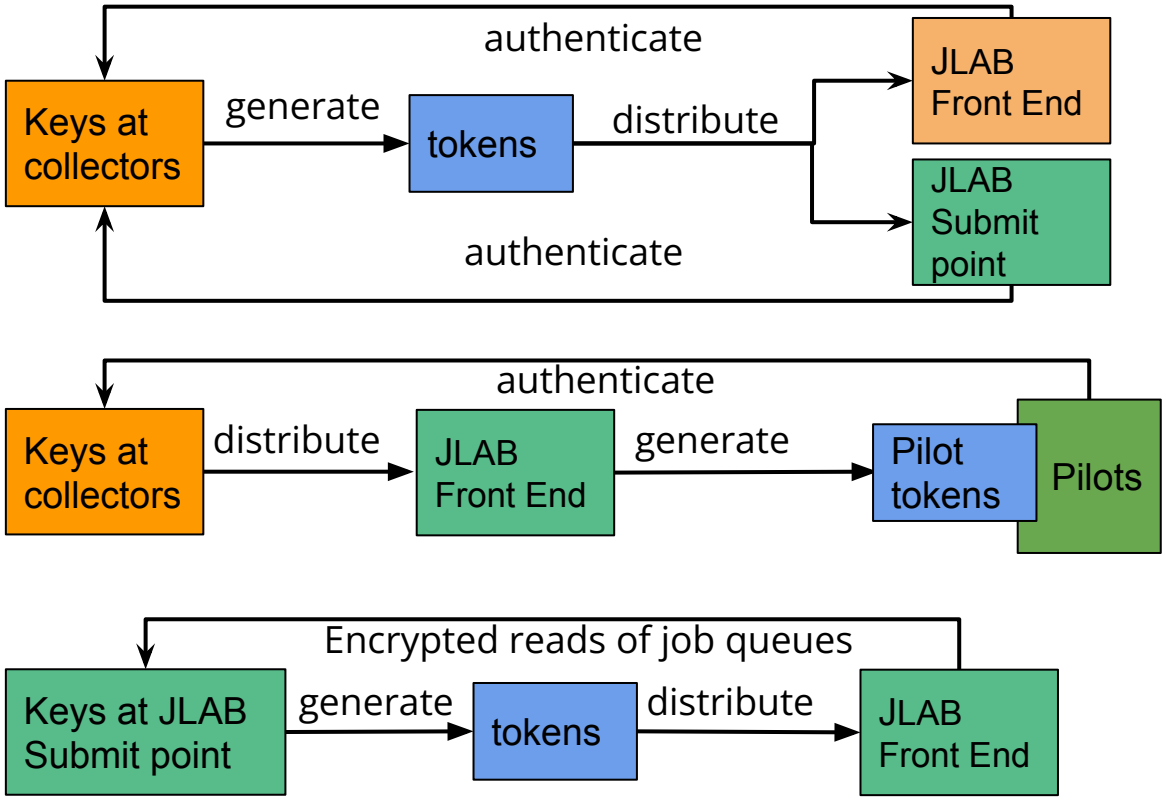
- OSG 3.5 support ends May 1, OSG 3.6+ will not support GSI authentication (i.e. daemon-to-daemon authentication with X509) and GridFTP based transfers

  Examples:

  - GSI authentication used to validate communication between CI components, e.g. between an Access Point and a Central manager
  - X509 credentials used to validate access by a user to a privileged storage location
- OSG, in coordination with sites - including JLAB - has invested significant effort to transition away from previous authentication model
- Two types of (token) authentication supported:
  a. IDTOKENS - JWTs issued using a key known by the server daemon
  b. SciTokens - JWTs issued via a SciToken issuer trusted by the server daemon

JWT - JSON Web Token: https://jwt.io/

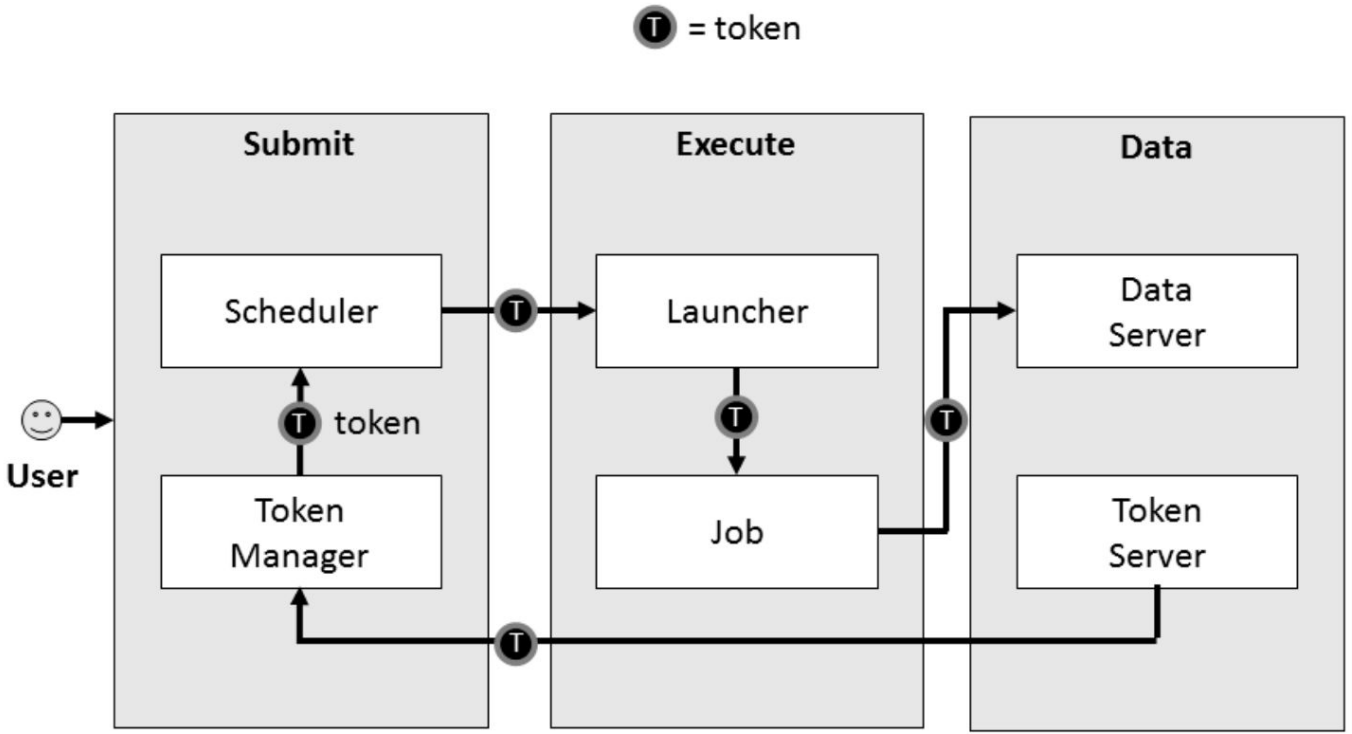
IDTOKENS


authenticate
Keys at collectors
generate
tokens
distribute
JLAB Front End
JLAB Submit point
authenticate
authenticate
Keys at collectors
distribute
JLAB Front End
generate
Pilot tokens
Pilots
Encrypted reads of job queues
Keys at JLAB Submit point
generate
tokens
distribute
JLAB Front End

SciTokens

- Federated Authorization Ecosystem for Distributed Scientific Computing
- The term "SciToken issuer" is a bit overloaded, referring to either/both the service that generates a SciToken or the (HTTPS) service containing the information needed to validate a SciToken.
- The front-end generates SciTokens for pilot jobs, which use the SciTokens to authenticate with execution sites (that list the JLab issuer in their mapfile), ensuring that the pilot jobs originated at the JLab front-end.
- Front-End has a private signing key and the JLAB-issuer stores the public one on the JLAB “issuer”. As long as both parties trust this issuer, a SciToken validated by the public key information presented by the issuer should be trusted by both parties.

The SciTokens Model



https://scitokens.org/

Summary

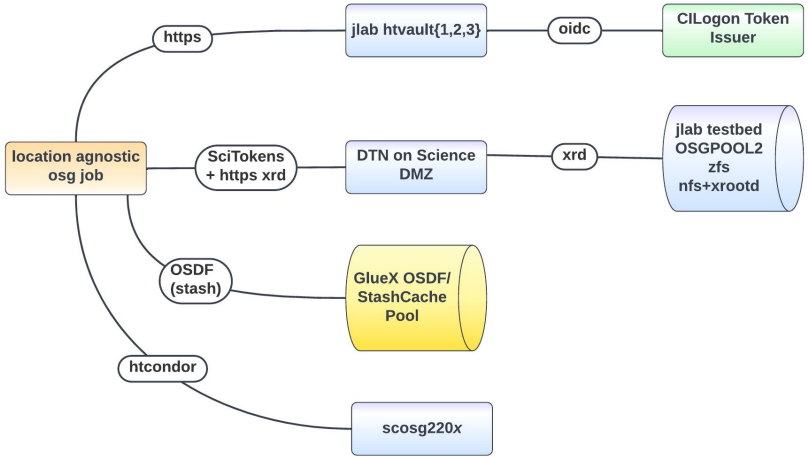
- JLAB, in coordination with OSG, continues to maintain a high availability access point for HEP experiments to run workflows on OSG and contributes resources to the pool of computing resources
- Infrastructure has been updated to latest iteration of software updates while enhancing it with additional features such as
  - A secondary collector managed by JLAB
  - A Scitoken’s issuer
  - Adding additional experiments in their grid-support portfolio (MOLLER)
  - Extra network capacity to enable data heavy workflow such as reconstruction jobs

Extra Slides

Scitoken Issuer at JLab

Existing Production Workflow

SciTokens Testbed Workflow

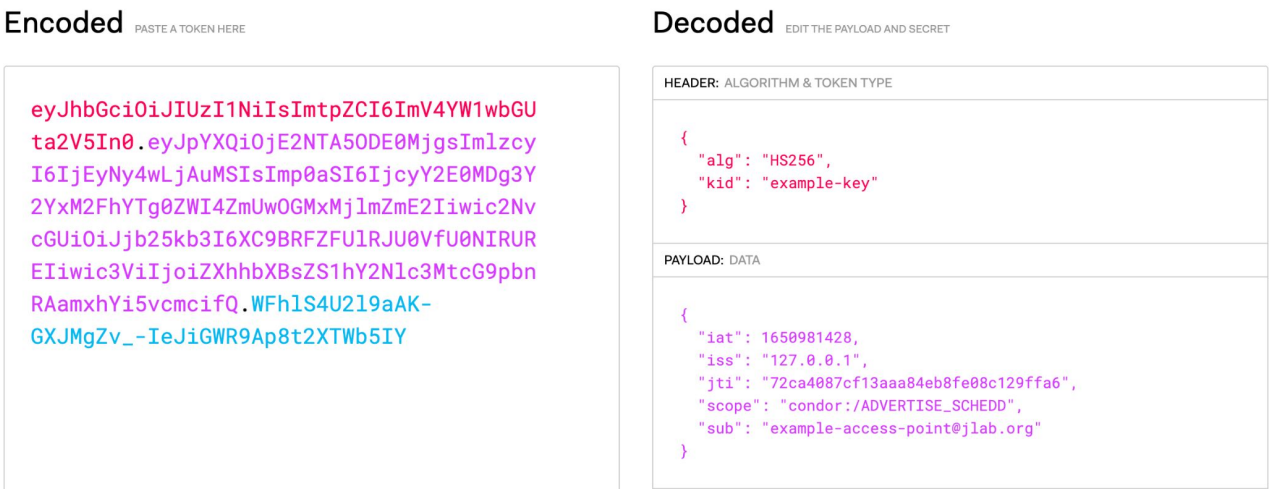


IDTOKENS

- Pool-wide key(s) generated at the collector (using `condor_store_cred`), stored in HTCondor passwords directory (`/etc/condor/passwords.d`).
- Multiple methods for generating tokens from keys, we generated tokens by hand on the collector with `condor_token_create`.
- Tokens generated and handed to access points and front-end so that they can authenticate to the collector.
- Separate front-end specific key generated at collector, stored in the front-end, and used by front-end to generate tokens for pilot jobs. The pilots use these tokens to authenticate with the collector.
- Separate access point specific keys generated at the access points, tokens generated and given to the front-end so that the front-end can do encrypted reads of the access points' job queues.
- Why separate keys for specific daemons? Revoking a key (i.e. deleting it on disk) will invalidate only the tokens generated from that key. No sense in revoking all of the access point tokens if the front-end key was compromised.

IDTOKENS

```
# condor_store_cred -f /etc/condor/passwords.d/example-key -p $(openssl rand -base64 32)
# condor_token_create -key example-key -identity "example-access-point@jlab.org" -authz ADVERTISE_SCHEDD
eyJhbGciOiJIUzI1NiIsImtpZCI6ImV4YW1wbGUta2V5In0.eyJpYXQiOjE2NTA5ODE0MjgsImlzcyI6IjEyNy4wLjAuMSIsImp0aSI6IjcyY2E0MDg3Y2YxM2FhYTg0ZWI4ZmUwOGMxMjlmZmE2Iiwic2NvcGUiOiJjb25kb3I6XC9BRFZFUlRJU0VfU0NIRURFRFIiwic3ViIjoiZXhhbXBsZS1hY2Nlc3MtcG9pbnRAamxhYi5vcmcifQ.WFhlS4U2l9aAK-GXJMgZv_-IeJiGWR9Ap8t2XTWb5IY
```



SciTokens

```
# scitokens-admin-create-key --create-keys --pem-private > example-key.pem
# scitokens-admin-create-token --keyfile example-key.pem --key_id 1234 --issuer
"https://example.com/jlab" --lifetime 3600 sub="example-pilot-job@jlab.org"
```


```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjEyMzQifQ.eyJzdWIiOiJleGFtcGxlLXBpbG90LWpvYkB
qbGFiLm9yZyIsImlzcyI6Imh0dHBzOi8vZXhhbXBsZS5jb20vamxhYiIsImV4cCI6MTY1MDk4NTk1MCwiaWF0IjoxN
jUwOTgyMzUwLCJuYmYiOjE2NTA5ODIzNTAsImp0aSI6IjYxYTNkN2M0LTk1YTAtNDUxNi04MDQ2LTVkZGIxNTFjMmM
5MiJ9.Gs64fDsTvOeC0i8fqRAPihFvY8txd-kl-yMHUX1Z_39Kspg4R1ljbE1bznSGp8Pw8MBvaJNCC-6vA4l5yiki
LOgC37KUzn9Itae38nohU55uhK-9JlV35ZRs0ojF2p1V3tbn1U9k_d0FW87b2V_rXfyq9kWrqaxIb_sRUr8HckfIx5
z_anaNS_CsBoC8RuzBruS-0YPOU0vQoq7i6xh6Ek4dqYlVeH1gO5ayhGQW_LiwVpc9beUaFQg3kxJcXJK_F4VE3YGj
h7k4WgNFQy5nVhIoVTuNX46gb3THU5spYAJmFj3PPlWYr0jbuoNwoPd0ousGCgkpHyzp53dJFWTNGg
```


HEADER: ALGORITHM & TOKEN TYPE


```
{
  "typ": "JWT",
  "alg": "RS256",
  "kid": "1234"
}
```


PAYLOAD: DATA


```
{
  "sub": "example-pilot-job@jlab.org",
  "iss": "https://example.com/jlab",
  "exp": 1650985950,
  "iat": 1650982350,
  "nbf": 1650982350,
  "jti": "61a3d7c4-95a0-4516-8046-5ddb151c2c92"
}
```