SciTokens and WLCG JWT Profile

MORGRIDGE
INSTITUTE FOR RESEARCH
CORE COMPUTATION

FEARLESS SCIENCE

SciTokens: Federated Authorization Ecosystem for Distributed Scientific Computing

The SciTokens project, funded by NSF and started July 2017, aims to:

1. Introduce a capabilities-based authorization infrastructure for distributed scientific computing,
2. provide a reference platform, combining CILogon, HTCondor, CVMFS, and Xrootd, AND
3. Implement an instance to help our science stakeholders (LIGO and LSST) better achieve their scientific aims.

When we wrote the proposal (early 2017), the Globus Toolkit was supported and the WLCG AAI group didn't exist.

- We are excited and proud of how the community has moved forward since!

SciTokens: Reference Platform

The motivating idea behind SciTokens is introducing capability-based token authorization. The reference platform shows one way to get there!

- The SciTokens profile provides a specification of the token format.
- The verification document says how to validate a token.
- We maintain Python and C clients to help parse, validate, and utilize tokens.

Building on these libraries, we have several applications/plugins to help manage and utilize tokens:

- The XRootD plugin allows XRootD to honor capabilities in tokens.
- The Apache module allows websites to require SciToken-based auth

Client Library Design

The API design has three important concepts:

- The SciToken, which represents the token in memory.
  - Typically created by the scitoken_deserialize function, which also retrieves the public key (possibly cached) and verifies the signature matches the token contents.
- The Validator, which determines the token's validity (e.g., is it expired?).
  - The user can provide additional callbacks for domain-specific claim validation checks.
- The Enforcer, which generates the ACLs associated with the token.
  - This provides an easier way to handle the capability-based scopes, especially those that are path based (path normalization is quite tricky!).

The intent of the library is to keep developers from using the JSON payload directly – keep them at the higher-level semantics and away from the specific token format.

SciTokens and WLCG JWT

SciToken Example


```
{
   "sub": "e1eb758b-b73c-4761-bfff-adc793da409c",
   "iss": "https://demo.scitokens.org",
   "nbf": 1555059791,
   "ver": ”scitoken:1.0",
   "aud": "https://dteam-test-client.example.com",
   "exp": 1555060391,
   "iat": 1555059791,
   "jti": "aef94c8c-0fea-490f-9027-ff444dd66d8c",
   "scope": "read:/store storage.create:/store/mc/datasetA",
}
```


Notes:

- The WLCG example here is a capability-centric authorization.
- The two tokens are strikingly similar.

WLCG JWT Example


```
{
   "sub": "e1eb758b-b73c-4761-bfff-adc793da409c",
   "iss": "https://demo.scitokens.org",
   "nbf": 1555059791,
   "wlcg.ver": "1.0",
   "aud": "https://dteam-test-client.example.com",
   "exp": 1555060391,
   "iat": 1555059791,
   "jti": "aef94c8c-0fea-490f-9027-ff444dd66d8c",
   "scope": "storage.read:/store
storage.create:/store/mc/datasetA",
   "eduperson_assurance":
["https://refeds.org/assurance/profile/espresso"],
   "acr": "https://igtf.net/ap/authn-assurance/cedar"
}
```

Key Observation: Interoperability

A subset of the WLCG JWT format can be mapped cleanly to SciTokens.

SciTokens and WLCG JWT Interoperability

I aim to have the SciTokens libraries to parse and validate WLCG JWTs.

- Will start to work toward this goal once the WLCG JWT
- For capability-based authorizations, this will be a drop-in upgrade.
  - **Any software using the SciTokens library today should, unmodified, be able to use both SciTokens and WLCG JWTs once the library is upgraded.**
- WLCG JWTs provide additional authorizations not in SciTokens.
  - I am not planning to write the patches needed to expose high-level APIs for identity / group-based authorization…
    - … but these will be available by accessing the claims directly.
    - … and certainly patches are accepted!

SciTokens and IAM Interoperability

There is a strong separation between the token contents (SciTokens, WLCG JWT) and the workflow to obtain the token (OAuth2).

- I don't expect any interoperability issues here, although I've had some technical difficulties getting the "end-to-end" working on my laptop (maybe something for today?).

The more pertinent question is "when will IAM support WLCG JWT?"

- Like the SciTokens client, I suspect they are in a bit of a holding pattern.

As we are working with more software providers to prepare the way for tokens, my recommendation is assume the IAM piece will issue the needed JWTs soon and proceed as "acquiring the token" can be completely separated from "using the token".

Next Hill To Climb

Where to next?

- Both the storage (XRootD) and compute (HTCondor-CE) have initial support for SciTokens; we need to demonstrate the end-to-end.
- I propose four challenges that, theoretically, require almost no development - but would demonstrate broad coordination!
  - **Challenge 1**: Acquire a token from IAM via OAuth2 and use it to upload files to dCache and XRootD.
  - **Challenge 2**: Acquire a token from IAM via OAuth2 and use it to submit a pilot job.
  - **Challenge 3**: Have the HTCondor "credmon" acting as an OAuth2 client acquire a token from IAM, send the token along with a job, and have the job stage out to dCache.
  - **Challenge 4**: Author a whitepaper describing how our community plans to use tokens for data management – including Rucio, FTS, IAM, XRootD, dCache, and others.

I suspect (1) is nearly done if you trawl through private email threads. The important aspects of an interoperability challenge would be to create knowledge and documentation for others to follow!

MORGRIDGE

INSTITUTE FOR RESEARCH

CORE COMPUTATION

morgridge.org

This material is based upon work supported by the National Science Foundation under Grant No. 1738962. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

FEARLESS SCIENCE