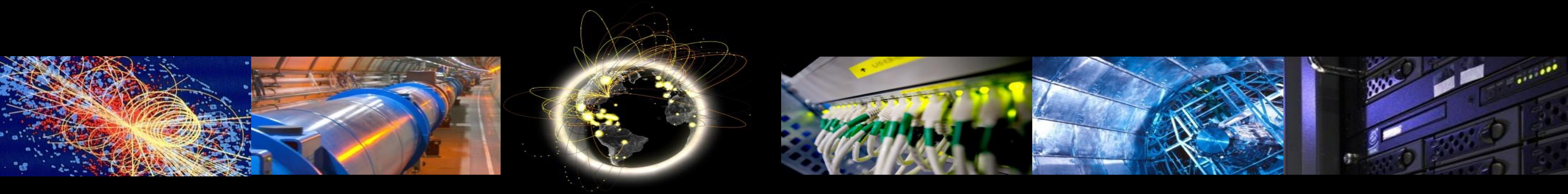
Authored by the WLCG AuthZ WG

WLCG Authorization

HEPiX Spring

March 16th 2021



WLCG Authorization (AuthZ) WG

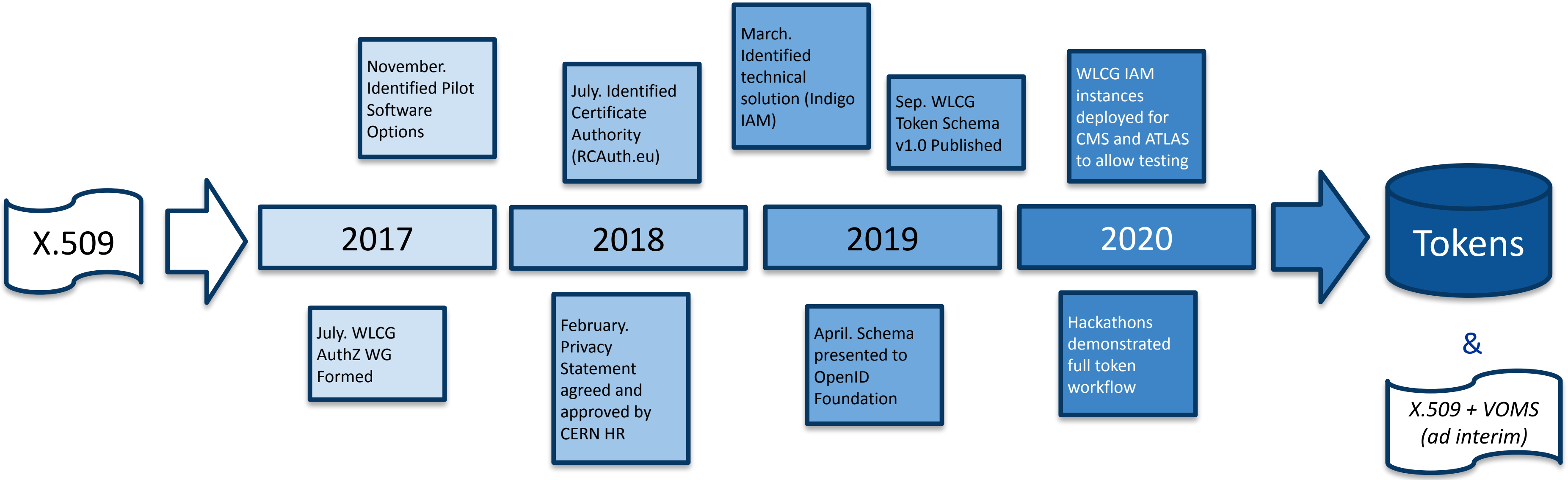
- Membership includes current major users of tokens in High Energy Physics
  - INDIGO IAM
  - EGI Check-in
  - SciTokens
  - dCache
  - ALICE
- Development work of pilot projects supported by:

AARC EOSC-hub EOSCpilot The European Open Science Cloud for Research Pilot Project

- Priority to stick to industry and R&E standards wherever possible

WLCG Worldwide LHC Computing Grid

Towards Tokens



November. Identified Pilot Software Options

July. WLCG AuthZ WG Formed

July. Identified Certificate Authority (RCAuth.eu)

February. Privacy Statement agreed and approved by CERN HR

March. Identified technical solution (Indigo IAM)

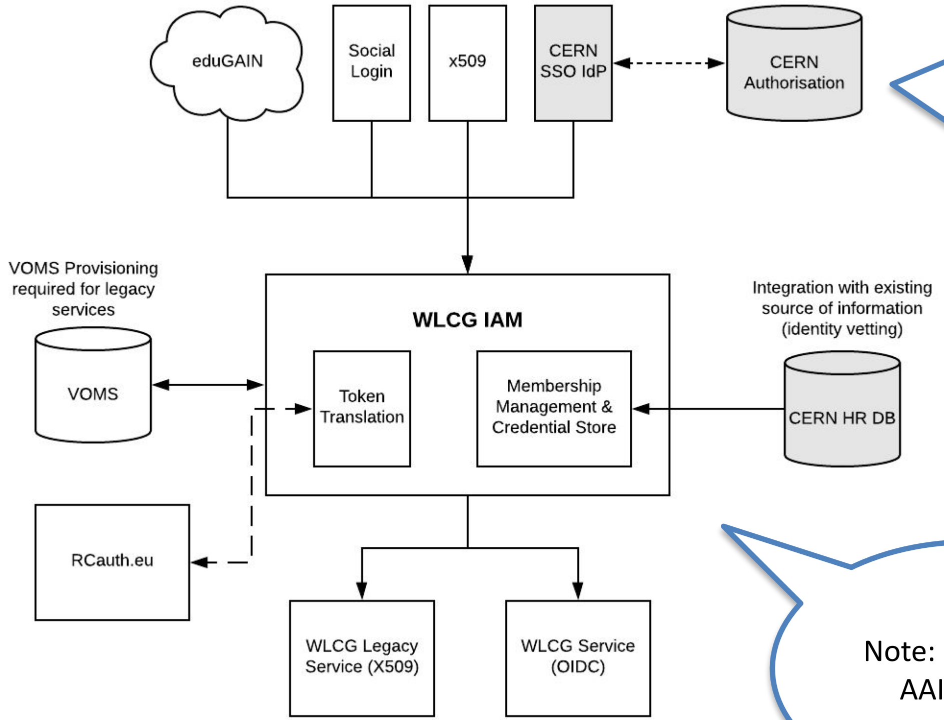
Sep. WLCG Token Schema v1.0 Published

April. Schema presented to OpenID Foundation

WLCG IAM instances deployed for CMS and ATLAS to allow testing

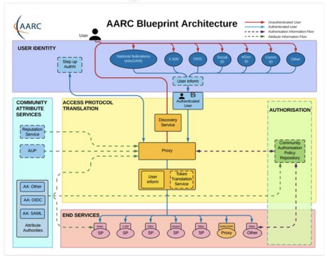
Hackathons demonstrated full token workflow

Infrastructure Design


eduGAIN
Social Login
x509
CERN SSO IdP
CERN Authorisation
VOMS Provisioning required for legacy services
VOMS
WLCG IAM
Token Translation
Membership Management & Credential Store
Integration with existing source of information (identity vetting)
CERN HR DB
RCauth.eu
WLCG Legacy Service (X509)
WLCG Service (OIDC)

CERN SSO configured as sole Identity Provider, enables identity verification via HR DB (match CERN PersonID)

Follows the AARC Blueprint
https://aarc-community.org/architecture/


AARC Blueprint Architecture
USER IDENTITY
COMMUNITY ATTRIBUTE SERVICES
ACCESS PROTOCOL TRANSLATION
AUTHORISATION
END SERVICES

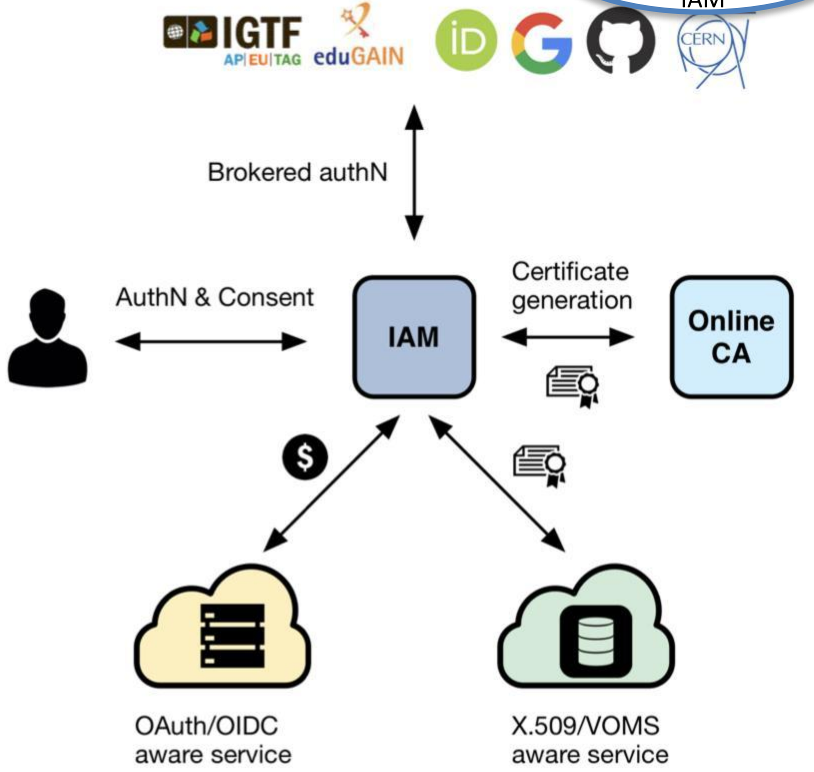
Note: one WLCG AAI per VO

INDIGO Identity and Access Management Service

Identity and Access Management = IAM

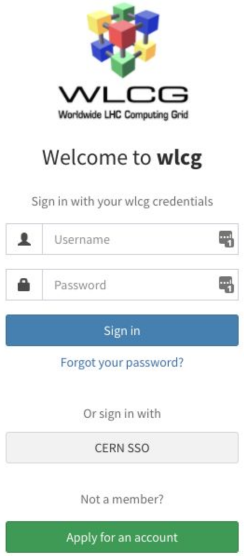
A **VO-scoped** authentication and authorization service that

- supports **multiple authentication mechanisms**
- provides users with a **persistent, VO-scoped** identifier
- exposes **identity information**, **attributes** and **capabilities** to services via **JWT** tokens and standard **OAuth & OpenID Connect** protocols
- can integrate existing **VOMS**-aware services
- supports **Web** and **non-Web access**, **delegation** and **token renewal**



Deployments

The following token issuers have been deployed. The ATLAS and CMS instances are available for testing and integration, with the expectation that they will become the future production token issuers.



Welcome to **wlcg**

Sign in with your wlcg credentials

Username

Password

Sign in

Forgot your password?

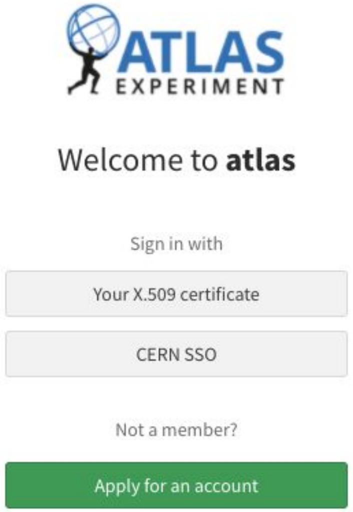
Or sign in with

CERN SSO

Not a member?

Apply for an account

https://wlcg.cloud.cnaf.infn.it



Welcome to **atlas**

Sign in with

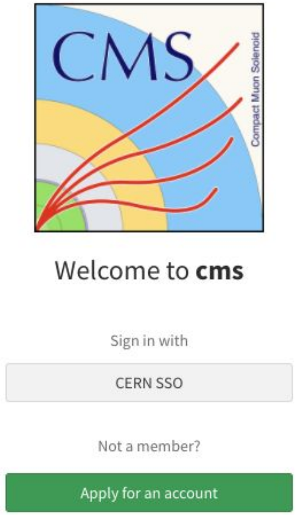
Your X.509 certificate

CERN SSO

Not a member?

Apply for an account

https://atlas-auth.web.cern.ch



Welcome to **cms**

Sign in with

CERN SSO

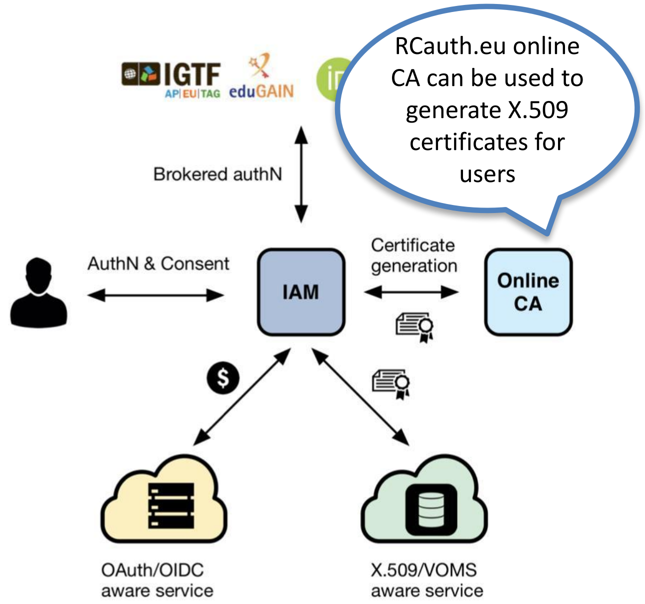
Not a member?

Apply for an account

https://cms-auth.web.cern.ch

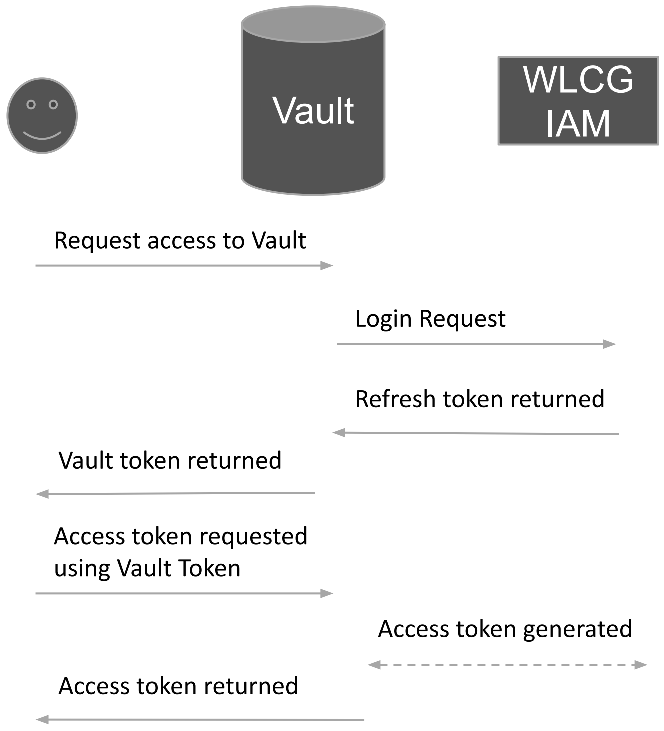
X.509 Backwards Compatibility

- IAM can act as a backwards compatible VOMS server supporting voms-proxy-init etc.
- For users without an end user certificate, RCauth.eu can be used to generate X.509 certificates which are stored in IAM and returned on demand
- Users can be imported from a VOMS server

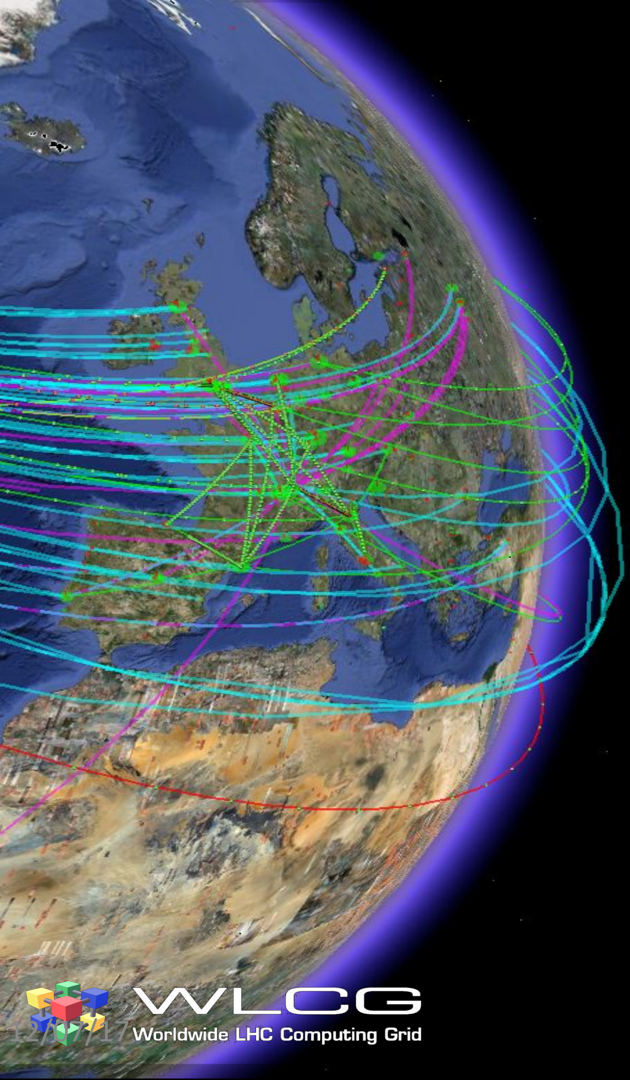


Command Line Tools

- Since many workflows are performed on the command line, we need users to be able to get Tokens in their local environment
  - Must be user friendly
  - Must be secure (i.e. refresh tokens protected)
- Solution identified: **use Hashicorp Vault as the registered client and manager of refresh tokens**



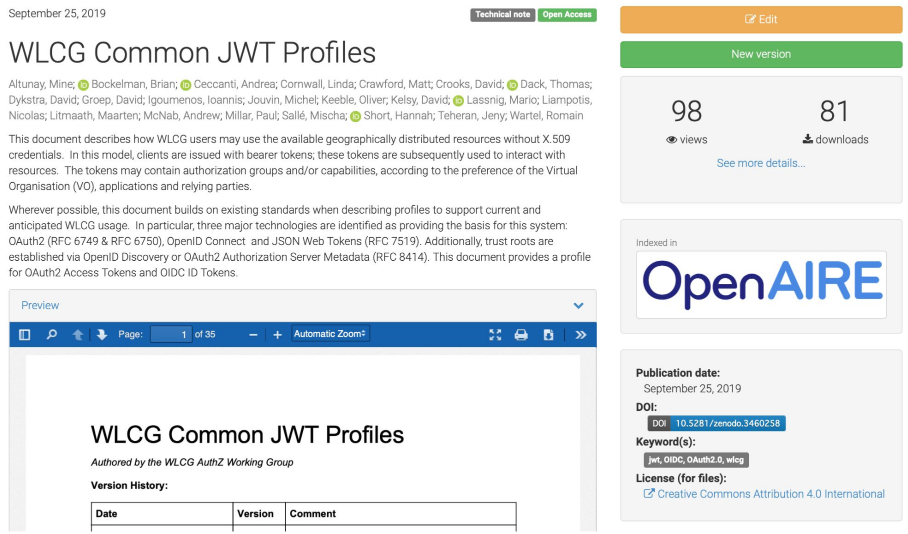
See https://github.com/fermitools/htvault-config



WLCG Token Schema

Schema V1.0

- Published on Zenodo, September 25th 2019
- Allows middleware developers to enable token based authorization to an agreed schema
- Working document at https://github.com/WLCG-AuthZ-WG/common-jwt-profile



September 25, 2019

Technical note Open Access

WLCG Common JWT Profiles

Altunay, Mine; Bockelman, Brian; Ceccanti, Andrea; Cornwall, Linda; Crawford, Matt; Crooks, David; Dack, Thomas; Dykstra, David; Groep, David; Igoumenos, Ioannis; Jouvin, Michel; Keeble, Oliver; Kelsy, David; Lassnig, Mario; Liampotis, Nicolas; Litmaath, Maarten; McNab, Andrew; Millar, Paul; Sallé, Mischa; Short, Hannah; Teheran, Jeny; Wartel, Romain

This document describes how WLCG users may use the available geographically distributed resources without X.509 credentials. In this model, clients are issued with bearer tokens; these tokens are subsequently used to interact with resources. The tokens may contain authorization groups and/or capabilities, according to the preference of the Virtual Organisation (VO), applications and relying parties.

Wherever possible, this document builds on existing standards when describing profiles to support current and anticipated WLCG usage. In particular, three major technologies are identified as providing the basis for this system: OAuth2 (RFC 6749 & RFC 6750), OpenID Connect and JSON Web Tokens (RFC 7519). Additionally, trust roots are established via OpenID Discovery or OAuth2 Authorization Server Metadata (RFC 8414). This document provides a profile for OAuth2 Access Tokens and OIDC ID Tokens.

WLCG Common JWT Profiles

Authored by the WLCG AuthZ Working Group

Version History:

Date	Version	Comment

98 views

81 downloads

See more details...

Indexed in OpenAIRE

Publication date: September 25, 2019

DOI: DOI 10.5281/zenodo.3460258

Keyword(s): jwt, OIDC, OAuth2.0, wlcg

License (for files): Creative Commons Attribution 4.0 International

https://zenodo.org/record/3460258

Token Claims

Common Claims

- sub
- exp
- iss
- acr
- aud
- iat
- nbf
- jti
- eduperson_assurance (REFEDS)
- wlcg.ver (WLCG)
- wlcg.groups (WLCG)

iss+sub used to uniquely identify a user, e.g. for blocking

wlcg prefix added to avoid collisions with other schemas

ID Tokens

- auth_time
- general OIDC Claims

Access Tokens

- scope (RFC8693)

Access tokens should include at least scope (capabilities) or group for authorization

Note: Where unspecified, the origin is RFC7519 or OpenID Connect core

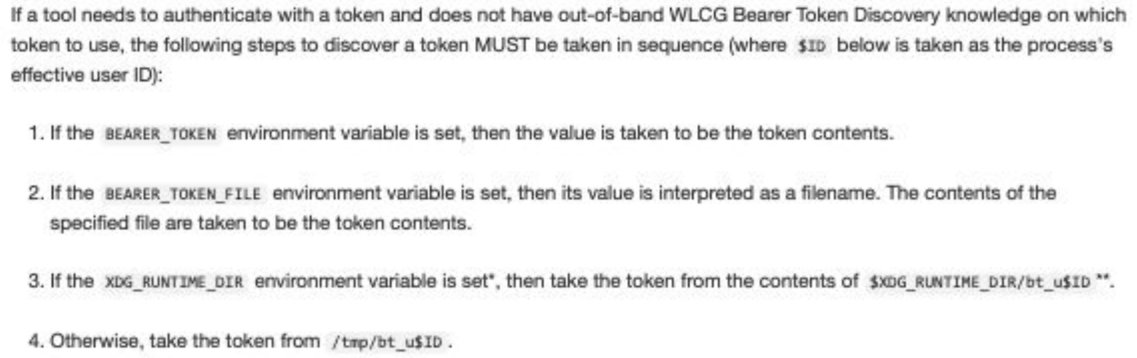
Lifetimes

Token Type	Recommended Lifetime	Minimum Lifetime	Maximum Lifetime	Justification
Access Token & ID Token	20 minutes	5 minutes	6 hours	Access token lifetime should be short as there is no revocation mechanism. The granted lifetime has implications for the maximum allowable downtime of the Access Token server.
Refresh Token	10 days	1 day	30 days	Refresh token lifetimes should be kept bounded, but can be longer-lived as they are revocable. Meant to be long-lived enough to be on a "human timescale".
Issuer Public Key Cache	6 hours	1 hour	1 day	The public key cache lifetime defines the minimum revocation time of the public key. The actual lifetime is the maximum allowable downtime of the public key server
Issuer Public Key	6 months	2 days	12 months	JWT has built-in mechanisms for key rotation; these do not need to live as long as CAs. This may evolve following operational experience, provision should be made for flexible lifetimes.

Token Discovery

- Many tools will rely on tokens being stored in the local environment
- Token discoverability specification v1.0 published https://zenodo.org/record/3937438

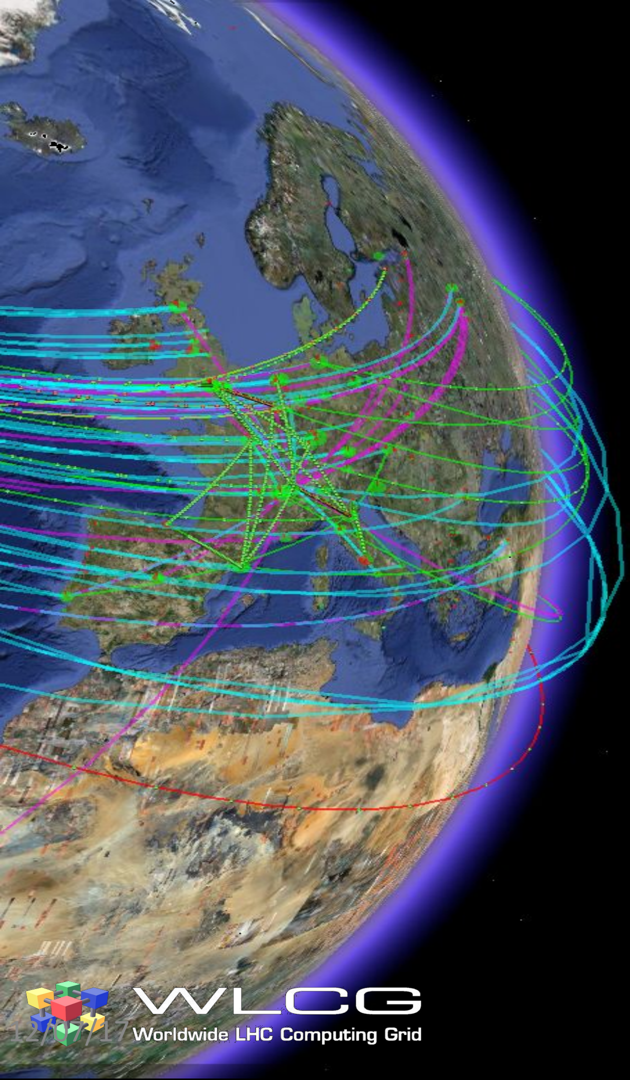
Logic of where to search for (or place) tokens locally



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` .

WLCG
Worldwide LHC Computing Grid



Authorization

Authorization

- Two models
  - Groups e.g. /atlas/production
  - Capabilities e.g. storage.read/atlas

"Access tokens may convey authorization information as both groups and capabilities. If both group membership and capabilities are asserted, then the resource server should grant the union of all authorizations for the groups and capabilities that it understands." From the WLCG Token Schema

WLCG Worldwide LHC Computing Grid

Groups (in wlcg.groups claim)

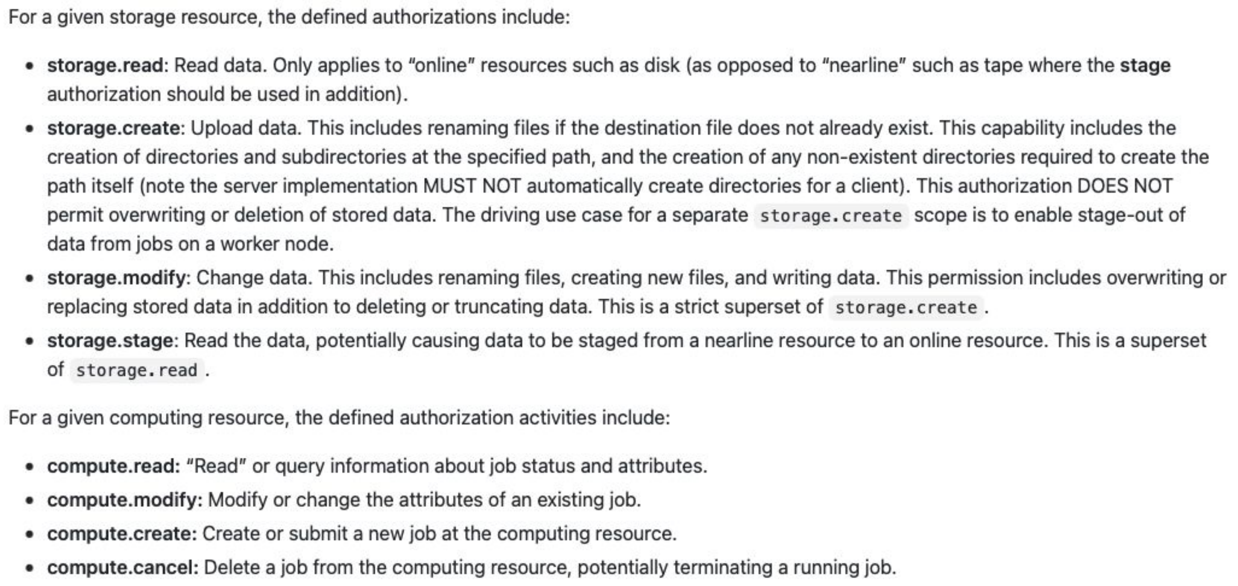
- Authorization may be based on the **wlcg.groups** claim.
- **wlcg.groups** semantics are equivalent to existing VOMS groups. VOMS roles should be considered as optional (i.e. returned only if requested) **wlcg.groups**
- Requesting the **wlcg.groups** scope returns all default groups

WLCG Worldwide LHC Computing Grid

Capabilities (in scope claim)

- Authorization may be based on the **scope** claim.
- Format **$AUTHZ:$PATH** where **$PATH**is mandatory (may be '/' for *)

Scopes proposed for storage and compute



For a given storage resource, the defined authorizations include:

- **storage.read**: Read data. Only applies to "online" resources such as disk (as opposed to "nearline" such as tape where the **stage** authorization should be used in addition).
- **storage.create**: Upload data. This includes renaming files if the destination file does not already exist. This capability includes the creation of directories and subdirectories at the specified path, and the creation of any non-existent directories required to create the path itself (note the server implementation MUST NOT automatically create directories for a client). This authorization DOES NOT permit overwriting or deletion of stored data. The driving use case for a separate `storage.create` scope is to enable stage-out of data from jobs on a worker node.
- **storage.modify**: Change data. This includes renaming files, creating new files, and writing data. This permission includes overwriting or replacing stored data in addition to deleting or truncating data. This is a strict superset of `storage.create`.
- **storage.stage**: Read the data, potentially causing data to be staged from a nearline resource to an online resource. This is a superset of `storage.read`.

For a given computing resource, the defined authorization activities include:

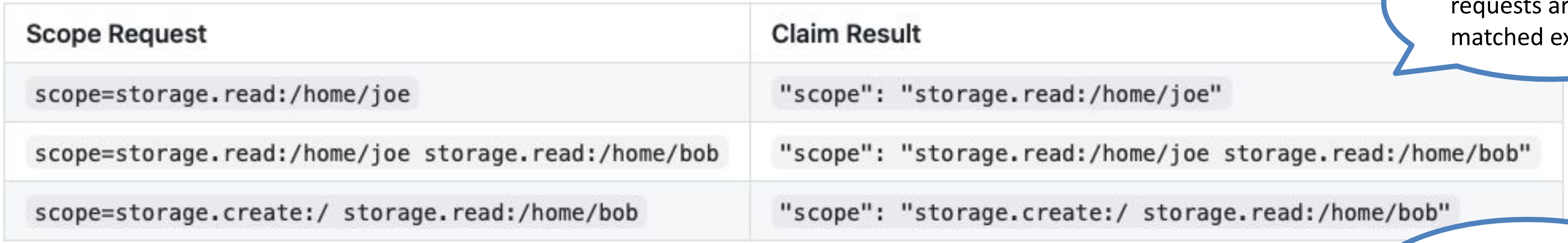
- **compute.read:** "Read" or query information about job status and attributes.
- **compute.modify:** Modify or change the attributes of an existing job.
- **compute.create:** Create or submit a new job at the computing resource.
- **compute.cancel:** Delete a job from the computing resource, potentially terminating a running job.

Scope Based Attribute Selection

- We propose to use scopes to implement an attribute selection mechanism for **both groups and capabilities** following the approach outlined in the OpenID Connect standard:
  - https://openid.net/specs/openid-connect-core-1_0.html#ScopeClaims
- Authorizations are requested using scopes and returned by the token issuer if the client and user are entitled

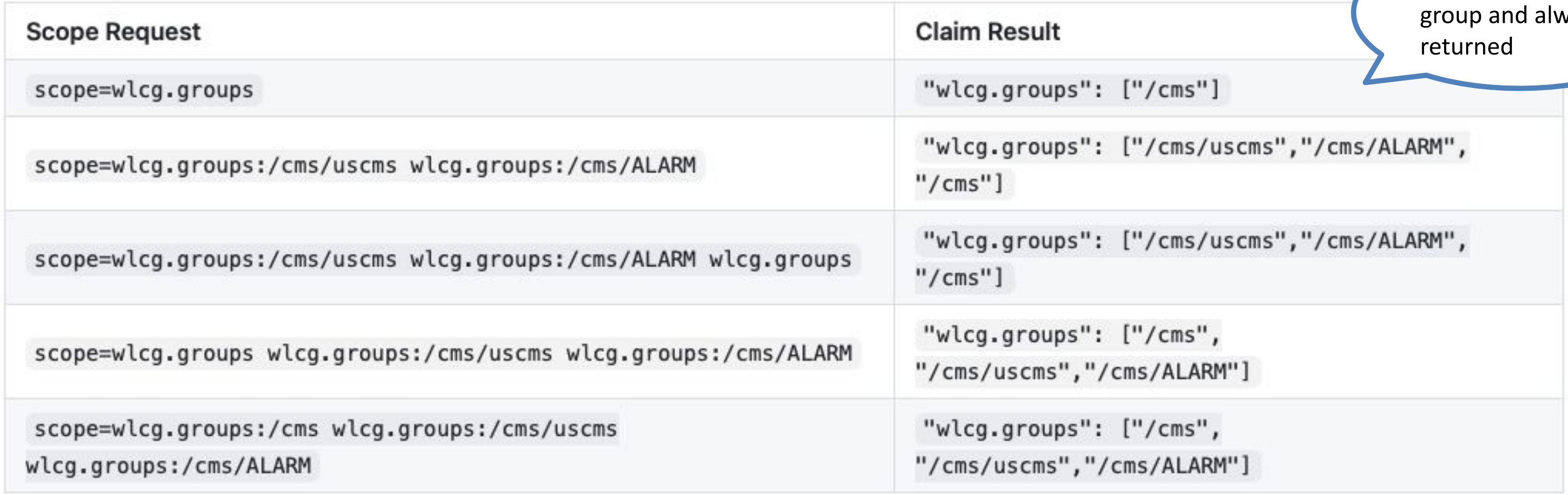
WLCG Worldwide LHC Computing Grid

Scope Based Attribute Selection



Scope Request	Claim Result
`scope=storage.read:/home/joe`	`"scope": "storage.read:/home/joe"`
`scope=storage.read:/home/joe storage.read:/home/bob`	`"scope": "storage.read:/home/joe storage.read:/home/bob"`
`scope=storage.create:/ storage.read:/home/bob`	`"scope": "storage.create:/ storage.read:/home/bob"`

Capability requests are matched exactly



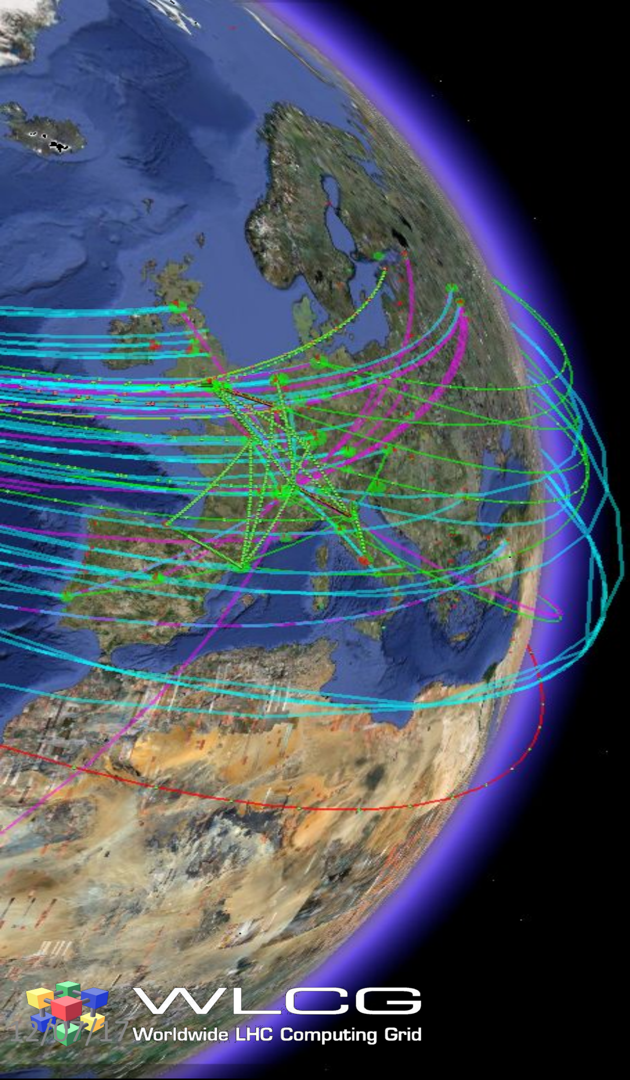
Scope Request	Claim Result
`scope=wlcg.groups`	`"wlcg.groups": ["/cms"]`
`scope=wlcg.groups:/cms/uscms wlcg.groups:/cms/ALARM`	`"wlcg.groups": ["/cms/uscms","/cms/ALARM", "/cms"]`
`scope=wlcg.groups:/cms/uscms wlcg.groups:/cms/ALARM wlcg.groups`	`"wlcg.groups": ["/cms/uscms","/cms/ALARM", "/cms"]`
`scope=wlcg.groups wlcg.groups:/cms/uscms wlcg.groups:/cms/ALARM`	`"wlcg.groups": ["/cms", "/cms/uscms","/cms/ALARM"]`
`scope=wlcg.groups:/cms wlcg.groups:/cms/uscms wlcg.groups:/cms/ALARM`	`"wlcg.groups": ["/cms", "/cms/uscms","/cms/ALARM"]`

/cms is a default group and always returned

Capability Sets

- In some use cases (e.g. Vault) a client may not know exactly which capabilities will be required by downstream services
- A capability set can be requested; if granted, multiple capabilities will be returned in the token
- Work ongoing at https://github.com/WLCG-AuthZ-WG/common-jwt-profile/pull/10

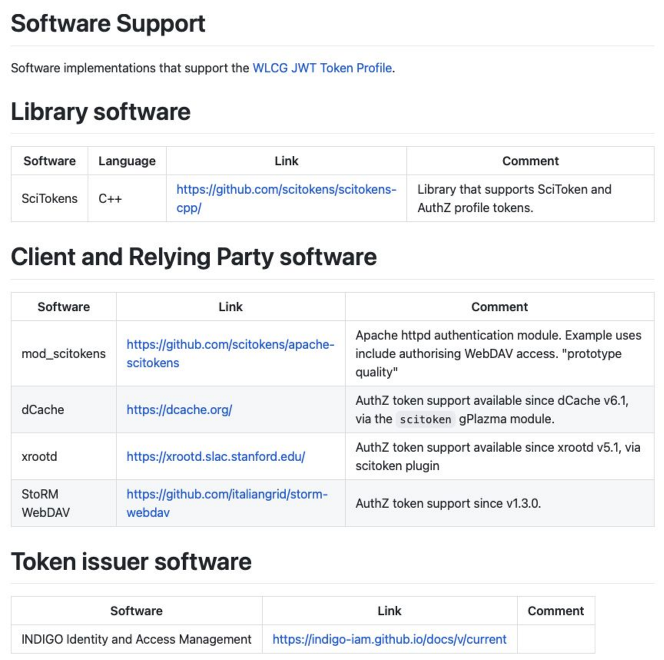
WLCG Worldwide LHC Computing Grid



Next Steps

Schema Adoption

- A catalogue of software that supports the WLCG JWT Schema is being compiled at https://github.com/WLCG-AuthZ-WG/software-support
- Aware that storage components are progressing faster than compute



Software Support

Software implementations that support the WLCG JWT Token Profile.

Library software

Software	Language	Link	Comment
SciTokens	C++	https://github.com/scitokens/scitokens-cpp/	Library that supports SciToken and AuthZ profile tokens.

Client and Relying Party software

Software	Link	Comment
mod_scitokens	https://github.com/scitokens/apache-scitokens	Apache httpd authentication module. Example uses include authorising WebDAV access. "prototype quality"
dCache	https://dcache.org/	AuthZ token support available since dCache v6.1, via the `scitoken` gPlazma module.
xrootd	https://xrootd.slac.stanford.edu/	AuthZ token support available since xrootd v5.1, via scitoken plugin
StoRM WebDAV	https://github.com/italiangrid/storm-webdav	AuthZ token support since v1.3.0.

Token issuer software

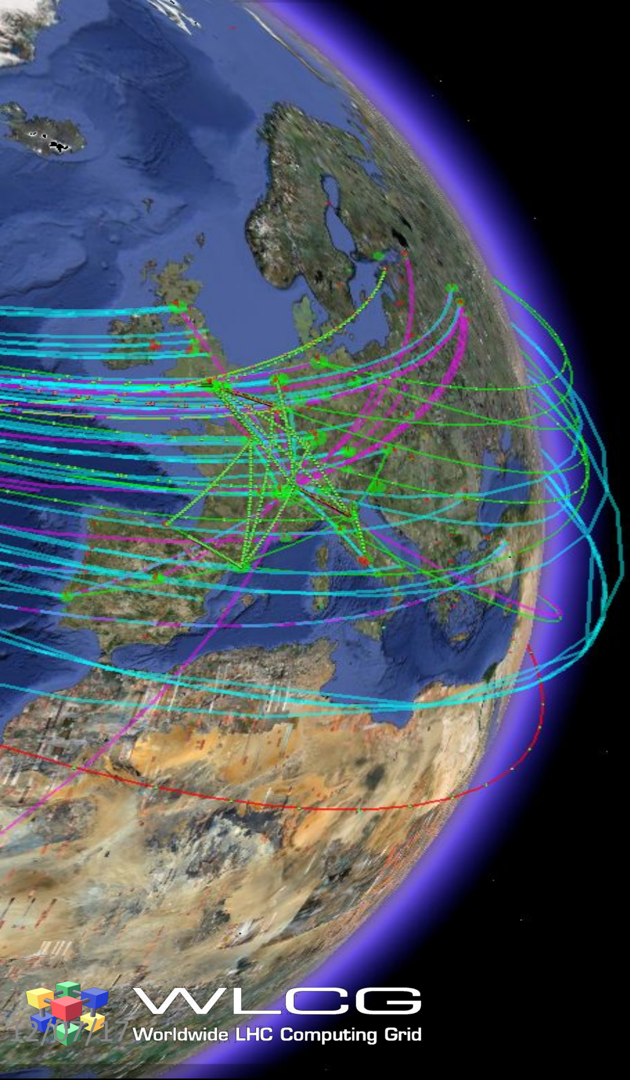
Software	Link	Comment
INDIGO Identity and Access Management	https://indigo-iam.github.io/docs/v/current	

Interoperability

- Participating in AEGIS community to facilitate interoperability between infrastructures e.g. EOSC https://aarc-project.eu/about/aegis/

Timeline

- Timeline under discussion https://twiki.cern.ch/twiki/bin/view/LCG/WLCGTokensGlobusWG
- Summary at https://indico.cern.ch/event/876787/#6-globus-retirement-timeline
- Key points (TBC)
  - Q2 2021 production IAMs available for VOs
  - Q3 2021 pilot jobs may be performed with tokens
  - Q4 2021 VOMS-Admin retired



Questions?