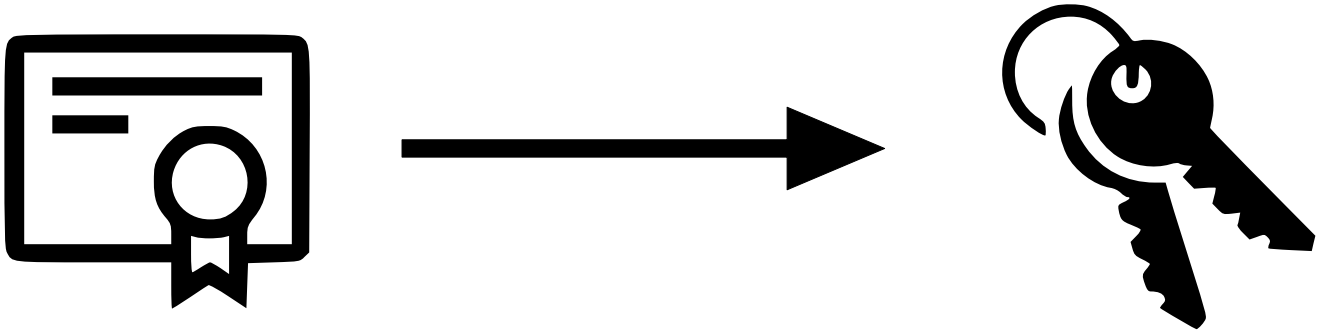
WLCG AuthZ update

Tokens & Storage

Grid Deployment Board

14th July 2021



Rucio DOMA testbed & tokens

destination / source	WLCG_BONN-XRD_H	WLCG_CERN-EOS_H	WLCG_CNAF-STO_H	WLCG_DESY-PROM-DCA_H	WLCG_INFN-T1-STO_H	WLCG_PRAGUELCG2-DPM_H	WLCG_RAL-ECH_H
WLCG_RAL-ECH_H	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
WLCG_PRAGUELCG2-DPM_H	0.0%	0.0%	0.0%	0.0%	0.0%		0.0%
WLCG_INFN-T1-STO_H	13.3%	0.0%	0.0%	0.0%		0.0%	0.0%
WLCG_DESY-PROM-DCA_H	0.0%	0.0%	0.0%		0.0%	0.0%	0.0%
WLCG_CNAF-STO_H	43.5%	50.0%		0.0%	0.0%	0.0%	0.0%
WLCG_CERN-EOS_H	0.0%		0.0%	0.0%	0.0%	0.0%	0.0%
WLCG_BONN-XRD_H		0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

echo
DPM Disk Pool Manager
StoRM
dCache
StoRM
EOS
XRootD

successful transfers with all grid storages

Functional Tests with WLCG JWT Tokens (Rucio → FTS → gfal2 → HTTP-TPC pull)

WLCG JWT compliance tests

Statistics by Tag	Total	Pass	Fail	Elapsed	Pass / Fail
audience	32	16	16	00:00:30	
basic-authz-checks	112	56	56	00:02:20	
cern-eos	18	6	12	00:00:34	
cerntrunk-dpm	18	6	12	00:00:18	
cnaf-amnesiac-StoRM	18	18	0	00:00:20	
infn-t1-xfer-StoRM	18	18	0	00:00:20	
nebraska-xrootd	18	4	14	00:00:27	
prague-dpm	18	8	10	00:00:19	
prometheus-dCache	18	12	6	00:00:32	

- SE implementation not fully compliant with WLCG JWT profile
- Hackathons focused on storage and transfer
  - January 2020 (indico)
  - September 2020 (indico)
- WLCG Token Transition Timeline
  - March 2022: “All storage services provide support for tokens”
  - Development necessary to make compliance table also green

WLCG JWT profile – storage

- WLCG JWT token content based on RFC7519
  - **`sub`** + **`iss`** claim – unique identifier for multi-IAM services
  - **`aud`** can be used to restrict token usage, e.g. https://fqdn:port
  - WLCG JWT extensions used by storage implementations
    - claims starting with **`wlcg.groups`** prefix
    - scopes with **`storage.`** prefix + **`wlcg`** and **`wlcg.groups[:name]`**
- Storage compliant with WLCG JWT profile supports
  - scope based authorization
    - capability
  - group based authorization
    - default groups – present
    - optional groups – on request
  - server should grant union


```
{
  "wlcg.ver": "1.0",
  "sub": "58280cfd-ed7f-4954-90c7-cfde610cb963",
  "aud": "https://wlcg.cern.ch/jwt/v1/any",
  "nbf": 1626228002,
  "scope": "openid profile storage.read:/ storage.create:/
 storage.modify:/ wlcg wlcg.groups",
  "iss": "https://wlcg.cloud.cnaf.infn.it/",
  "exp": 1626231602,
  "iat": 1626228002,
  "jti": "a504bdb2-73c5-496b-a6c5-c58e1a457b13",
  "client_id": "6a7c5c81-f1ee-4f0e-9c2e-7c5280aa5c78",
  "wlcg.groups": [
    "/wlcg",
    "/wlcg/pilots",
    "/wlcg/xfers"
  ]
}
```


WLCG JWT Token example

Storage scopes

- `storage.read:/[path]` – read online data
- `storage.create:/[path]` – allow write but not overwrite
- `storage.modify:/[path]` – create with overwrite and delete
- `storage.stage:/[path]` – reading that can trigger staging
- Path is optional and restrict access to specific directory
  - relative to the base path for given token issuer (“VO”)
  - same `storage.*` scope name can multiple times with different path
  - IAM can drop scope that is not available to the client, e.g. “/”
- Capability based authZ – IAM has full control / define policy
  - can be tricky to get it right together with group based authZ
    - storage administrator defines identity mapping and ACLs
  - IAM shared by several different groups
    - tricky with more resource providers
  - with current WLCG JWT profile groups can’t provision capability

Identity mapping

- Relatively straightforward and supported at VO level
- Sites supporting individual users
  - Linking various user identities (krb, X509, token) to same uid
    - accessing all private data regardless of authZ method
  - VOMS Admin provides DN for VO, used e.g. for gridmap files
  - IAM with improved privacy measures don’t allow anonymous access
    - SCIM API with user / group details
      - special privileges (scope) required, not available by default
      - assigned by IAM admin
        - doesn’t scale for large number of services / hosts
        - we would need better interface if SCIM mapping data required by majority of WLCG sites
      - used e.g. by Rucio account import from IAM
  - Hybrid model without single IdP (e.g. DUNE with CERN and FNAL)

Storage status

- It is not sufficient when transfers with tokens work … 2
  … they must fail as defined in the standard, WLCG JWT profile 3
  - configuration issues
  - implementation issues
- Implementations
  - WLCG tokens
    - StoRM (HTTPS), DPM (HTTPS)
  - WLCG+SciTokens
    - dCache (HTTPS + xroot)
    - XRootD / SciTokens library – Echo, EOS, native (HTTPS + xroot)
- `storage.create` mapped internally to “write” privilege
  - DPM
  - SciToken library
- `storage.stage` not really implemented

- StoRM – perfect with tests designed by same group of developers
- dCache – only minor issues (different HTTP errors 40X vs. 40Y)
- others with more important weaknesses

most probably REST replacement for SRM first and only later tape used with tokens

Storage status

- Only global configuration of accepted audiences
  - avoid using `https://wlcg.cern.ch/jwt/v1/any` in production
- User mapping implementation dependent
  - can't make assumptions based on one storage behavior
  - uid / gid for directories / files stored with scope based authZ
  - DPM use directly user identity – no mapping
    - internally use just **`sub`** as user identity without **`iss`**
    - for scope based access uid / gid is inherited from parent directories
  - dCache provides two gplazma modules – plans to merge&improve
    - `oidc` – general mapping based on **`sub`** and other claims
    - `scitoken` – mapping to one user identity
  - SciTokens
    - simple mapping to one identity
    - more complex using mapfile

mapping individual users with account details from IAM accessible via SCIM (CERN IAM instances provides nickname, personal id, cert. DN, ...)

Storage status

- Majority of implementations provides web interface
  - data access via web browser with OIDC login
- DPM example configuration with just one issuer
  - different apache module necessary to support for multiple issuers
- Tokens with xroot protocol
  - supported since XRootD 5 and dCache 6.2
  - require xroot-over-TLS to protect tokens
    - XRootD 5 can encrypt only specific messages, e.g. exclude data
  - recent client libraries vs. old software releases linked with XRootD 4
    - can’t use directly directio
    - copy2scratch by pilot or local XCache proxy
  - important to get experience with xroot-over-TLS with tokens
    - xroot preferred protocol for job stage-in / stage-out
    - no large scale test

tokens => encryption
(except for XRootD 5)

Client tools

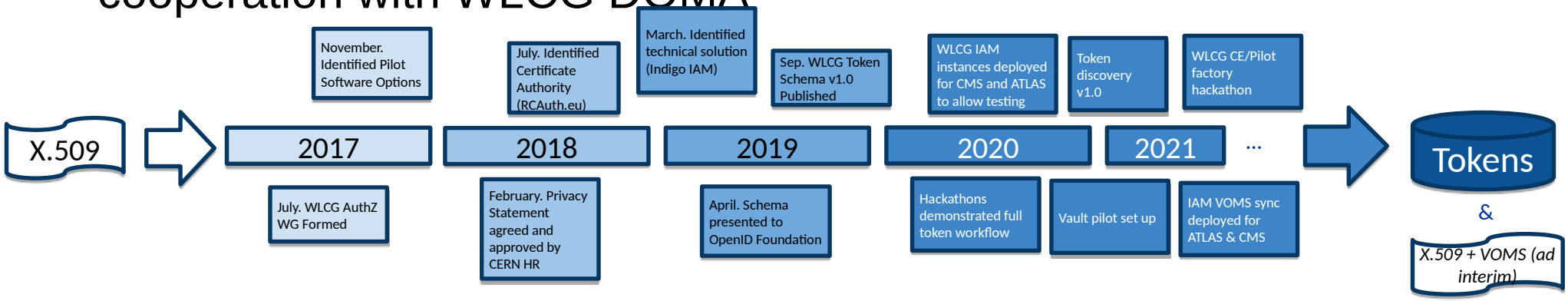
- Rucio
  - necessary to use tokens with different **`scope`** and **`aud`**
  - token not stored in a location defined by WLCG Token Discovery
  - no group based authorization
- FTS
  - don’t use specific audience for each transfer party
  - no WLCG Token Discovery support
- gfal2, davix
  - no WLCG Token Discovery support (two tokens for TPC)
  - possible to use directly gfal2 python API
  - unable to pass token for TURL in SRM requests
- xrdcp
  - Bearer token can be passed as argument
  - no WLCG Token Discovery support

using transfer clients with tokens not yet user friendly / more complex than X509

Questions?

WLCG Authorization WG

- Authorization standards used by industry
- Shift towards federated identities, new data protection requirements
- Adopted by Research & Education sector
- => WLCG Authorization WG
  - transition from X.509 to tokens
  - technical solutions, software, standards
    - WLCG JWT profile, token discovery
  - define authentication schema
  - development and token integration
  - cooperation with WLCG DOMA



Identity and Access Management

- Indigo IAM – OAuth / OpenID Connect
  - support both tokens and X.509
- New security model based on tokens
  - opportunity to improve security
  - reduce impact of compromised (job) credentials
  - more granular (scope, aud), capability, lifetime (access vs. refresh)