



# Progress in the WLCG Transition to Tokens

The 9<sup>th</sup> Asian Tier Center Forum

**University of Tsukuba**, **Center for Computational Sciences**

24 September 2025

Maarten Litmaath

# Introduction

- WLCG has been working on a transition from X509 + VOMS towards JSON Web Tokens (JWT) since July 2017
- WLCG Common JWT Profiles v1.0 was [published](#) on 25 Sep 2019
- The first token timeline [document](#) with tentative milestones was published on 22 Aug 2022
  - Most milestones have eventually been reached
  - The last milestones provide little detail for what still needs to be done
- WLCG token usage today is still limited, but on the rise
  - FTS production traffic for ATLAS at O(50%) and CMS at O(5%)
  - Job submissions to HTCondor CEs and, to varying extents, ARC CEs

# General considerations

- Going from **multi-day** proxy certificates to access tokens with **O(hour)** lifetimes significantly changes the dependency on the authorization infrastructure
  - **Instead of using the VOMS proxy, now which token for which operation(s)?**
- Token lifetimes should be as short as possible while being compatible with operations
  - **Longer duration OK for tokens with narrow scopes and/or audiences and/or being in safe places**
- It is particularly important to limit the extent of tokens having the **storage.modify** scope, as it allows corresponding data to be deleted
- Beware of tokens ending up in **logfiles** that might get exposed

# Relevant technologies affecting sites

- IAM – token issuer service, INDIGO IAM
- Storage services
- FTS
- Rucio
- DIRAC
- JAliEn
- PanDA, Harvester
- GlideinWMS
- HTCondor CE
- ARC CE

# IAM service concerns

- The IAM service of an experiment may be a bottleneck for certain workflows
  - Access token rates of ~1 kHz can be sustained
- The database of each IAM instance is a single point of failure
  - As of IAM v1.13, access tokens will no longer be stored in the DB  
→ will reduce pressure on the DB by a lot
- Downtime of an IAM service may cause immediate fallout
  - The IAM service layer itself is HA
  - The group running IAM is well versed in providing highly critical services (e.g. SSO)
- Mismatch between expectations and feasible Service Level
  - Common to all WLCG services supported by CERN IT

# IAM service mitigation options

- Improvements to the INDIGO IAM code to increase performance
  - In progress
- Host static content (e.g. public keys) separately to decrease load on system and mitigate downtime risks
  - Only helps for tokens already issued
- Replicate IAM instances off-site
  - Would make the services more complex, with new failure modes
- Tokens could be made more generic and longer-lived
  - Goes against our aims to improve the security architecture
- Offload *time-critical*, *high-rate* token use cases to issuers run by the experiments themselves
  - As done by ALICE for two decades already

# Storage services

- Token support generally in good shape, with remaining issues to be addressed
- Exception: no support yet for tokens in *tape* operations
  - Agreement has been reached on most of the details, though

# File Transfer Service considerations

- Since file transfer requests may be queued for up to many days, there are two options:
  - FTS exchanges tokens and refreshes them as needed
  - FTS is given long-lived tokens that are used directly
- The first option has various drawbacks
  - The token-exchange workflow requires a privileged client
  - Scalability concerns related to the token-exchange workflow
  - Complexity concerns of the token-refresh workflow
  - The time restriction on the token-exchange workflow
  - **Introduction of a third party dependency in the transfer lifecycle**
- Heading towards long-lived (, per-file) tokens for data-intensive communities, FTS-managed tokens for less intensive communities

# Rucio considerations

- File-specific tokens, audience-restricted to a single storage, offer the best level of security
  - In WLCG, however, read tokens can be given the capability to read anywhere in the namespace
- Two challenges in the current token ecosystem
  - Scalability concerns of the token issuer
  - Full dependency on the availability of the token issuer, and its SLA
- Both are overcome if Rucio mints data access tokens itself
  - As ALICE has been doing for two decades already
  - Proof of concept has been tested by ATLAS

# DIRAC considerations

- LHCb will stick to per-file tokens, but with longer lifetimes to avoid the FTS exchange and refresh workflows
- Letting DIRAC mint those tokens avoids an extra dependency, point of failure, and latency
- Interested in moving towards a *pre-signed URL* model instead
  - Would imply significant development for all affected middleware → will not happen soon → we should make tokens work *sufficiently*, as there are some dark clouds surrounding *certificates* already...
- Progress for jobs and users currently blocked by an issue with the WLCG Token Profiles

# ALICE – JAliEn considerations

- ALICE has been using “access envelope” tokens for data operations for two decades already
- Replacing those with WLCG tokens is being considered
  - Probably not before the middle of LS3
  - Would help make the configuration of storage services uniform
- Pilot jobs use JAliEn custom tokens to obtain payloads with corresponding data access tokens from the central services

# ATLAS – PanDA and Harvester

- PanDA and Harvester can use OIDC tokens at all levels ([doc](#))
  - Job submission to the CE from Harvester
  - Authorization to the monitoring
  - Communication from pilot with PanDA server – fallback on X509
  - Pilot can replace long-lived initial token with short-lived job token
  - Pilot can also get new arbitrary (e.g. storage) tokens from PanDA servers – this is not used in production
- Payload long-lived WLCG access tokens are restricted to talking only with the PanDA service

# CMS – GlideinWMS

- CMS jobs are already set up to use tokens when present
  - Only used in tests for now
- Read scopes cover the whole namespace, uploads are or will be restricted to dataset / user level
- Failed token-based uploads are retried with the VOMS proxy
- HTCondor components are used to maintain valid access tokens in jobs (analogous to VOMS proxies)
- Long-lived refresh tokens are safeguarded in HTVault services (now based on OpenBao) that interact with IAM

# HTCondor CE

- Some EGI sites are still running unsupported versions
  - Will continue to be followed up by EGI Operations
- Concerns about EGI accounting of jobs submitted without VOMS proxies
  - APEL log parser relies on logged VOMS attributes to determine the VO of each job
  - LHC experiments can continue equipping jobs with VOMS proxies for the time being, even when no longer needed by the pilots
  - HTCondor developers have agreed to mechanisms resembling what ARC v7 supports → implemented and tested OK, still to be released
  - AUDITOR framework: similar approach already in use

# ARC CE

- ARC v7 allows the token configuration for a VO to be connected directly to the accounting of its jobs
- Input and output file management needed for certain HPC sites still relies on VOMS proxies for the time being
  - The use of tokens requires further investigations

# Usage of tokens by users

- Users should be exposed to tokens as little as possible
- The frameworks and tools they use should know how to acquire the right tokens at the right times, occasionally prompting the user to (re)authenticate as needed
- Experiments may want to make use of auxiliary services like HTVault to help simplify user workflows
  - [Already planned by CMS, building on successful usage by other experiments at Fermilab](#)

# A new version of the WLCG Token Profiles

- In the past 6 years it has become clear that v1.0 has various deficiencies that needed to be worked around
  - [Experts know what to ignore where](#)
- It was time for a new version that is a better match for what we should be able to make work
  - [For example, more realistic token lifetimes, depending on the use case](#)
  - [Source repository, issue and change tracking are here](#)
- We settled for **v1.1** at this time and not yet v2.0, because the updated profile specifications do not break anything we already have in production today
- A proposed draft was [presented](#) to the WLCG Management Board a week ago and the actual v1.1 is [published](#) since yesterday!

# Tentative milestones – updated since May

- **2025-Q3** Release of WLCG Token Profiles v1.1 → done!
- **2025-Q4** Specification of token usage for tape operations
- **2025-Q4** Risk assessment including the effects of outages – possibly with actual downtime tests
- **2025-Q4** First production tests of *Rucio-minted tokens* in ATLAS?
- **2025-Q4** First usage of tokens in ATLAS jobs?
- **2026-Q3** CMS grid jobs with only tokens
- **2025-Q4** First tests of *DIRAC-minted tokens* in LHCb?
- **2026-Q4** Token Grand Challenge, with use of tokens by jobs
- **2027-Q1** Data Challenge 2027
- **2028-Q1** Completion of the X509 / VOMS phaseout

# Conclusions and outlook

- While great progress with the transition to tokens has been made in the last years, a number of challenges remain to be addressed in the next few years, as detailed on the preceding pages.
- The WLCG Token TF has been created to coordinate common features of the transition.
  - In particular for the LHC experiments, but also taking partner communities into account as needed.
- The TF is working with several WGs dealing with specific aspects
  - DOMA BDT WG – evolution of token usage in services for data operations
  - Authorization WG – evolution of WLCG Token Profiles and the IAM services
  - Token Trust and Traceability WG – best practices and policies for all parties