



Token Transition

overview

[WLCG / HSF Workshop](#), 6 May 2025

WLCG Token TF

v1.1

Introduction

- The first token timeline [document](#) with tentative milestones was published on August 22, 2022, in Zenodo
 - Most milestones have eventually been reached
 - The last milestones provide little detail for what still needs to be done
- Though tokens are already being used in FTS production traffic for ATLAS at $O(25\%)$ and CMS at $O(5\%)$, concerns exist about the sustainability of certain underlying assumptions
- Though jobs are submitted with tokens to many CEs since over two years, some job submission remains X509 based even today (waiting on middleware updates)

General considerations

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

Relevant technologies affecting sites

- IAM (Token issuer software, 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
- The database of each IAM instance (CERN IT DBOD) is a single point of failure
 - Note: It is frequently backed up and DBOD instances are well used by other WLCG services
- Downtime of an IAM service may cause immediate fallout
 - Note: the team that provides the IAM service is well versed in providing highly critical services
- Mismatch between expectations and actual SLA
 - Note: this is common to all WLCG services supported by CERN IT

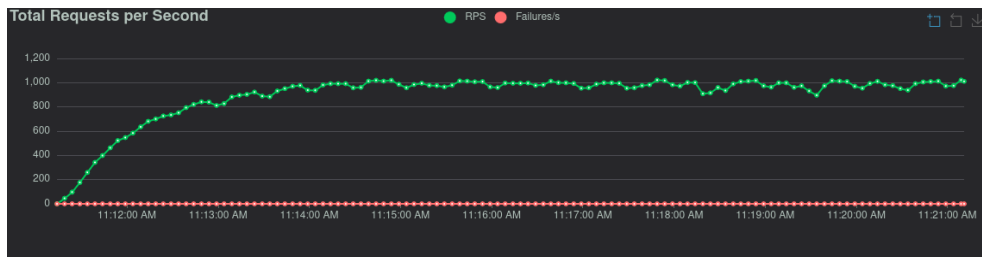
IAM service mitigation options

- 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
 - In progress
 - Only helps for tokens already issued
- Replicate IAM instances off-site
 - Would make the services more complex, with new failure modes
 - Investigate distributed DB options e.g. those planned by SKA
- 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

IAM performance measurements

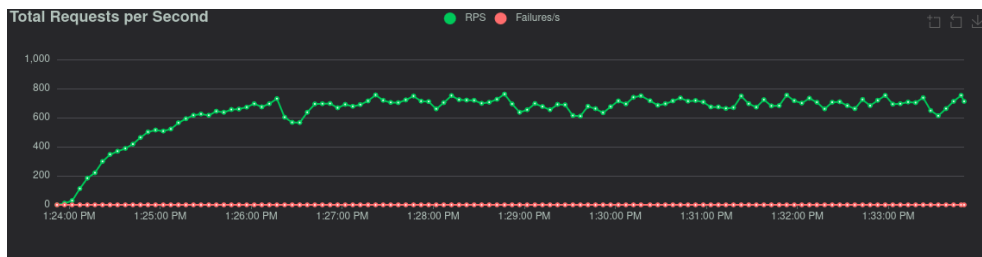
Get Access Tokens (client_credentials)

Roughly 1000 requests per second



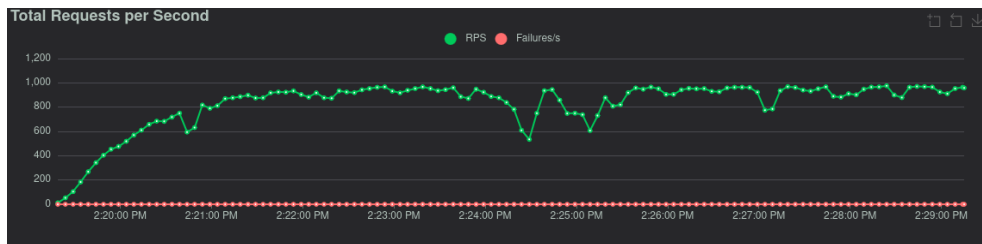
Token exchange to get a refresh token

Roughly 700 requests per second



Get Access Tokens (using a refresh token)

Roughly 900 requests per second



Storage services

- Token support generally in good shape, with remaining issues to be addressed
- Exception: no support yet for tokens in tape operations
 - Please see the [next talk](#) for a proposal from the FTS Team
- Improvements to be discussed and agreed in DOMA BDT WG

FTS considerations

- Since FTS 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
- Progress for jobs and users currently blocked by an [issue](#) with the WLCG Token Profile

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
- Pilot jobs use JAliEn custom tokens to obtain payloads with corresponding data access tokens from the central services

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

GlideinWMS

- CMS jobs are already set up to use tokens if present
- Read scopes cover the whole namespace, uploads are 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 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
 - Short-term workaround(s) needed in APEL and/or HTCondor CE
 - HTCondor developers have agreed to mechanisms resembling what ARC v7 supports → implemented and tested OK by Petr Vokac!
 - AUDITOR framework: similar approach already in use at KIT
 - More in the [WLCG Operations](#) session tomorrow

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 envisaged by CMS, building on successful usage by other experiments at FNAL*

Analysis of several identified risks

- Complete reliance on IAM instances for tokens
 - New single point of failure or, if IAM services are replicated, a more complex architecture to operate and maintain with limited effort
- Reliance on experiment data management frameworks for high-rate token issuance
 - Multiple implementations
 - Multiple issuers for sites to configure per experiment
 - More complexity in the containment and followup of security incidents?
- Reliance on FTS token exchange and refresh workflows
 - More complexity in the FTS
 - More reliance on reliability and scalability of IAM services
- Unresolved issues in the WLCG token profile
 - Different experiments may need to take different approaches, leading to a more complex ecosystem

Technical roadmap – tentative milestones

- **2025-Q2** Specification of token usage for tape operations
- **2025-Q3** Release of WLCG Token Profile v2.0
- **2025-Q4** First production usage of Rucio tokens in ATLAS
- **2025-Q4** First usage of DIRAC tokens in LHCb
- **2025-Q4** First usage of tokens in ATLAS jobs
- **2025-Q4** Operational Risks: Service risk document detailing the effects of outages – possibly with actual downtime tests
- **2026-Q3** CMS grid jobs with only tokens
- **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.
- We are 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 Profile and the IAM services
 - [Token Trust and Traceability WG](#) – best practices and policies for all parties