EPJ Web of Conferences **251**, 02028 (2021) https://doi.org/10.1051/epjconf/202125102028
CHEP 2021

WLCG Token Usage and Discovery

Brian Bockelman[6], *Andrea* Ceccanti[3], *Thomas* Dack[4], *Dave* Dykstra[7], *Maarten* Litmaath[1], *Mischa* Sallé[5], and *Hannah* Short[1,*]

[1]European Organization for Nuclear Research (CERN), Switzerland
[2]Deutsches Elektronen-Synchrotron (DESY), Germany
[3]Istituto Nazionale di Fisica Nucleare (INFN), Italy
[4]Science and Technology Facilities Council (UKRI-STFC), United Kingdom
[5]Nationaal Instituut voor Subatomaire Fysica (Nikhef), Netherlands
[6]Morgridge Institute for Research, United States
[7]Fermi National Accelerator Laboratory, United States

Abstract. Since 2017, the Worldwide LHC Computing Grid (WLCG) has been working towards enabling token based authentication and authorisation throughout its entire middleware stack. Following the publication of the WLCG Common JSON Web Token (JWT) Schema v1.0 [1] in 2019, middleware developers have been able to enhance their services to consume and validate the JWT-based [2] OAuth2.0 [3] tokens and process the authorization information they convey. Complex scenarios, involving multiple delegation steps and command line flows, are a key challenge to be addressed in order for the system to be fully operational. This paper expands on the anticipated token based workflows, with a particular focus on local storage of tokens and their discovery by services. The authors include a walk-through of this token flow in the RUCIO managed data-transfer scenario, including delegation to FTS and authorised access to storage elements. Next steps are presented, including the current target of submitting production jobs authorised by Tokens within 2021.

1 Introduction

Over the past few years there has been significant progress made towards making token based authentication and authorisation a realistic goal for the WLCG. OAuth2.0-based workflows for physics analysis have been prototyped thanks to technical developments, made by both industry and the wider academic community, and to time dedicated by many members of the WLCG collaboration to address WLCG-specific challenges. The current objective is to be able to submit experiment production jobs within 2021.

1.1 Contributing Groups

The WLCG Authorisation Working Group was formed in 2017, at a time when multiple activities were independently beginning to seriously consider token based authorisation. Experts from multiple domains and projects - including SciTokens [4], the INDIGO DataCloud project [5] and EGI [6] - came together to chart a path towards token based authorisation

*e-mail: hannah.short@cern.ch

for WLCG [7]. Work to enhance software was supported by several European Commission Projects: EOSC-Hub [8], EOSC Pilot [9] and AARC2 [10]. The group focuses on the technical and policy challenges affecting WLCG's transition to OAuth2.0-based authorisation.

The Data Organization, Management and Access (DOMA) Third-Party-Copy (TPC) Working Group [11] has been instrumental in getting WLCG token support tested in data handling workflows that will be vital for LHC Run 3 and beyond.

2 Progress towards Token Based Workflows

2.1 WLCG Token Schema

The WLCG Common JSON Web Token (JWT) Schema v1.0 was published in September 2019 [1] and defined the semantics for JWT use within the WLCG. It was largely inspired by the SciTokens schema but, importantly, addressed some WLCG-specific requirements such as the need for including group information in tokens. The schema document defines recommended lifetimes for different tokens in the ecosystem and a mechanism for requesting tokens conforming to a given version of the specification. Many of the areas that required extended discussion have since been addressed by an Internet Draft that defines the content of Access Tokens [12]. A future addition of the WLCG specification will take advantage of this new document and focus on defining only those aspects which fall outside the scope of the draft.

The publication of the WLCG Schema allowed middleware developers to implement support for OAuth2.0 Tokens, which they could test using a WLCG Token Issuer deployed at INFN. This Token Issuer is a deployment of INDIGO IAM [13], the software chosen by the WLCG for this purpose, following a thorough analysis of several very viable options.

2.2 Command Line Tools

Whilst other scientific domains are seeing their researchers moving to web based analysis, a large proportion of physicists' work is still performed on the command line. Tools to provision OAuth2.0 tokens into a user's local environment must be both convenient and secure, minimising any requirements for the user e.g. to perform operations on a web portal. Tokens must be discoverable by command line clients, which led to the definition of the Bearer Token Discoverability specification [14] described in Section 3. Further details on Command Line tools for WLCG Token Based workflows can be found in vCHEP 2021 contribution "Secure Command Line Solution for Token-based Authentication" [15].

2.3 Token authorization flows

To make token based authentication and authorization a reality in WLCG we need to define how tokens are obtained from the Virtual Organisation (VO) token issuer (e.g., IAM) and sent across services to drive authentication and authorization in the infrastructure. We will rely only on standard OAuth/OpenID connect authorization flows for this purpose.

It is expected that most services will act as OAuth resource servers (which do not need registration at the token issuer), while services acting as entry points to the infrastructure (e.g., experiment frameworks, UIs, etc.) or that need to exchange tokens will have to be registered in IAM as clients. Table 1 contains an overview of anticipated OAuth flows for different authorization scenarios.

User authentication is requested by including the `openid` scope in the authorization requests. Audience restrictions are requested using the `audience` parameter, as standardized

Authorization Scenario	**OAuth Flow**
User authorization to a registered server-side application.	OpenID Connect authorization flow [16].
Delegation of user authorization across services, to implement audience and scope restrictions and the ability to delegate offline access privileges across the chain of services.	OAuth token exchange flow [17].
User authorization for a registered CLI application, to implement authentication flows on the terminal that can support federated identity providers like the CERN SSO [18]. Authentication on a web browser is required to validate the CLI session.	Device code flow [19].
Registered services acting as their own identity, i.e. not on behalf of a user.	OAuth client credentials flow [3].

Table 1. OAuth flows for authorization scenarios in WLCG.

in the OAuth token exchange standard. IAM honours what is a requested by a client, and includes the requested audience in issued access tokens. When no audience is explicitly requested, the generic audience string is included in access tokens, as required by the WLCG profile.

2.3.1 The RUCIO-FTS-SEs flow

To support DOMA activities, the first scenario we focused on is the RUCIO managed data transfer, which is common to several LHC experiments (as depicted in Figure 1).

In this scenario, RUCIO [20] delegates its identity to FTS [21] to manage a third-party transfer between two storage elements. RUCIO requests a token from IAM using the client credentials flow, since it is acting as its own service identity. In this request (step 1 in Figure 1), RUCIO requests that the token audience is restricted to the FTS service. RUCIO then submits a transfer job to FTS including the token in the request. FTS creates the transfer job, but cannot use the received token to manage the transfer as the token audience is specific to FTS and may not provide the privileges needed for reading and writing data at storage elements. To acquire the privileges needed for the storage elements, FTS starts a token exchange flow with IAM to exchange the received token (step 3.) with two new tokens; an access token and a refresh token that will be used to manage the transfer. In this flow, FTS requests the scopes needed to access the data and restricts the audience of the issued tokens to the target storage elements. The access token obtained in the flow is then used to submit the third-party transfer to one of the SE (step 4.) and for the actual data transfer among the SEs (step 5.). The refresh token can be used by FTS to get fresh access token from IAM (using the standard OAuth refresh token flow) if needed.

2.4 WLCG IAM Operational Readiness

In order to transition to production use of the WLCG IAM Token Issuers, the new Token Based infrastructure must be operated with a level of service at least equivalent to that of the

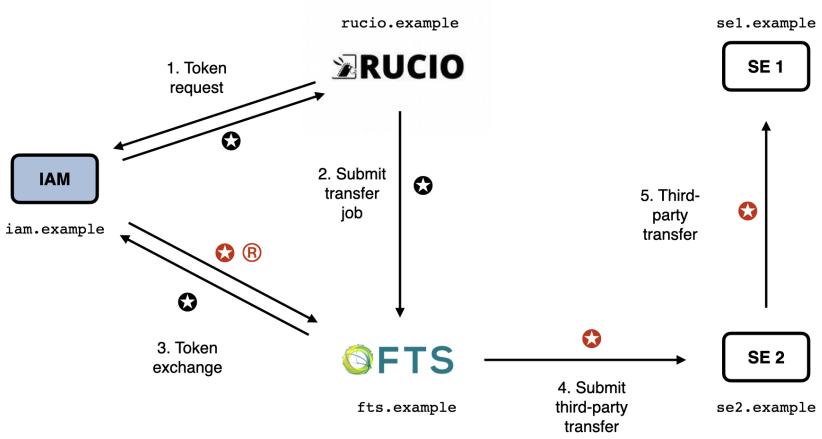

Figure 1. Token flows used in the RUCIO managed data-transfer scenario. Circles with stars identify OAuth Access tokens. Circles with the letter R identify OAuth Refresh tokens. Black indicates the initial tokens requested by Rucio, and red indicates the subsequent tokens requested by FTS following token exchange with IAM.

current infrastructure. This implies running highly available WLCG IAM instances, offering user support, and ensuring that service incidents can be addressed within an acceptable timeframe. In addition, the Token Issuers must demonstrate their trustworthiness by conforming with the EUGridPMA's Guidelines for Attribute Authority Service Provider Operations [22].

WLCG IAM Instances for CMS and ATLAS have been successfully deployed on CERN's Openshift Infrastructure for several months, allowing them to be highly scalable and leveraging central CERN IT services wherever possible. IAM instances for the remaining VOs will be set up similarly. Each instance will be integrated behind CERN's Single-Sign-On [18], improving user experience for the researchers and facilitating the inclusion of such services in the investigation of security incidents. IAM instances must be deployed on CERN infrastructure to enable experiment membership verification against CERN's Human Resources Database, due to data protection requirements. Production level support is expected for the second half of 2021 and load testing with anticipated production volumes will be completed.

3 Token Discovery

Client tools that rely on a bearer token for authenticating themselves need a mechanism for receiving the tokens from their environment. While the browser is a monolithic user agent (and can internally manage tokens), the terminal environment involves a number of independently developed tools; the environment needs a way to communicate the token to be used to Unix processes. As we did not find any existing standard for token discovery, we have defined one to be used by the tools built in our community [14]. The rest of this section is a description of that standard.

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 denotes the process's effective user ID:

1. If the BEARER_TOKEN environment variable is set, then its 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 [1], then take the token from the contents of $XDG_RUNTIME_DIR/bt_u$ID [2].

4. Otherwise, take the token from /tmp/bt_u$ID.

If a potential token is found at a step, then the discovery implementation MUST strip all whitespace on the left and right sides of the string. We define whitespace the same way as the C99 *isspace*() function: space, form-feed (\f), newline (\n), carriage return (\r), horizontal tab (\t), and vertical tab (\v). Upon finding a valid token according to section 2.1 of RFC6750 [23], the discovery procedure MUST terminate and return this token. Upon finding an empty token, the discovery implementation should continue with the next step. Upon finding an invalid token, the implementation SHOULD stop and return an error.

Upon discovery of a valid token, referred to as $TOKEN, if the tool is to use it to authenticate an HTTP request the tool MUST use it in accordance with RFC6750. For example, in the Authorization header as follows:

```
Authorization: Bearer $TOKEN
```

High-level tools that need to simultaneously support bearer tokens for multiple purposes (e.g. multiple VOs) MAY set $BEARER_TOKEN_FILE using the patterns of steps 3 and 4 with filenames having an added hyphen and purpose name appended to the filename. For example, the toolset named *fife*, keeping one token per VO, may choose the following name for user 1221 and VO *xyzzy*:

```
/tmp/bt_u1221-fife-xyzzy
```

The purpose syntax and semantics are deliberately left undefined and intended for use by the tool implementer. These high-level tools SHOULD consider potential filename collisions with other tools when implementing a naming scheme. When executing lower-level tools, the high-level tool SHOULD set the $BEARER_TOKEN_FILE to the desired file. Tools SHOULD NOT inspect multiple tokens to try to determine which one to use based on content.

4 Conclusion and Next Steps

Significant progress has been made towards the current goal of submitting jobs authorised with WLCG Tokens within 2021; a first draft of the Token Discoverability specification has been published, Token Based Workflows are being planned and tested, a viable solution for the command line is taking shape and production level Token Issuer support is being set up at CERN IT. Much work remains to align the community on a realistic timeline for the full transition to Tokens, away from end entity X.509 certificates. This will be a focus of the coming months whilst, in parallel, development and testing of token based workflows will continue.

[1] https://specifications.freedesktop.org/basedir-spec/basedir-spec-latest.html

[2] This additional location is intended to provide improved security for shared login environments as $XDG_RUNTIME_DIR is defined to be user-specific as opposed to a system-wide directory.

References

[1] M. Altunay, B. Bockelman, A. Ceccanti, L. Cornwall, M. Crawford, D. Crooks, T. Dack, D. Dykstra, D. Groep, I. Igoumenos et al., *WLCG Common JWT Profiles* (2019), https://doi.org/10.5281/zenodo.3460258

[2] M. Jones, J. Bradley, N. Sakimura, *JSON Web Token (JWT)*, RFC 7519 (2015), https://rfc-editor.org/rfc/rfc7519.txt

[3] D. Hardt, *The OAuth 2.0 Authorization Framework*, RFC 6749 (2012), https://rfc-editor.org/rfc/rfc6749.txt

[4] *INDIGO Identity and Access Management (IAM)*, https://indigo-iam.github.io/docs

[5] *Indigo Data Cloud*, https://www.indigo-datacloud.eu

[6] *EGI*, https://www.egi.eu

[7] Bockelman, Brian, Ceccanti, Andrea, Collier, Ian, Cornwall, Linda, Dack, Thomas, Guenther, Jaroslav, Lassnig, Mario, Litmaath, Maarten, Millar, Paul, Sallé, Mischa et al., EPJ Web Conf. **245**, 03001 (2020)

[8] *EOSC Hub*, https://www.eosc-hub.eu

[9] *EOSC Pilot*, https://eoscpilot.eu

[10] *Authentication and Authorisation for Research and Collaboration (AARC)*, https://aarc-project.eu

[11] *Data Organization, Management and Access (DOMA) Working Group Twiki* , https://twiki.cern.ch/twiki/bin/view/LCG/DomaActivities

[12] *JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens*, https://tools.ietf.org/html/draft-ietf-oauth-access-token-jwt-11

[13] *INDIGO Identity and Access Management (IAM)*, https://indigo-iam.github.io/docs

[14] WLCG Authorization Working Group, *WLCG Bearer Token Discovery* (2020), https://doi.org/10.5281/zenodo.3937438

[15] D. Dykstra, *Secure Command Line Solution for Token-based Authentication* (????), https://indico.cern.ch/event/948465/contributions/4323985/

[16] Nat Sakimura and John Bradley and Michael B. Jones, *The OpenID Connect discovery specification*, https://openid.net/specs/openid-connect-core-1_0.html (2014)

[17] M. Jones, A. Nadalin, B. Campbell, J. Bradley, C. Mortimore, *OAuth 2.0 Token Exchange*, RFC 8693 (2020), https://rfc-editor.org/rfc/rfc8693.txt

[18] Aguado Corman, Asier, Fernández Rodríguez, Daniel, Georgiou, Maria V., Rische, Julien, Schuszter, Ioan Cristian, Short, Hannah, Tedesco, Paolo, EPJ Web Conf. **245**, 03012 (2020)

[19] W. Denniss, J. Bradley, M. Jones, H. Tschofenig, *OAuth 2.0 Device Authorization Grant*, RFC 8628 (2019), https://rfc-editor.org/rfc/rfc8628.txt

[20] *RUCIO: Scientific Data Management*, http://rucio.cern.ch/

[21] *The CERN File Transfer Service*, https://fts.web.cern.ch

[22] *EUGridPMA Guidelines for Attribute Authority Service Provider Operations*, https://www.eugridpma.org/guidelines/aaops/

[23] M. Jones, D. Hardt, *The OAuth 2.0 Authorization Framework: Bearer Token Usage*, RFC 6750 (2012), https://rfc-editor.org/rfc/rfc6750.txt