On Authentication, Authorization and Single Sign On

A.Lytovchenko, A.Tsaregorodtsev,
CPPM-IN2P3-CNRS, Marseille,
DIRAC Users' Workshop
9 May 2022

DIRAC
THE INTERWARE

DIRAC
THE INTERWARE

Outline

- Introduction
- Token based client/server framework
- User Management
- Token Management
- Accessing third party resources
- Summary

Introduction

- So far, DIRAC uses X509 certificates for users' authentication/authorization
  - Mature technology, works well for users in HEP and some other domains
  - Complex certificate issuing procedures, many communities do not have access to recognized CA's
- The new generation AAI frameworks are based on the OAuth2/OIDC industry standards
  - Many Identity Providers (IdP) offer service for user Authentication and Management

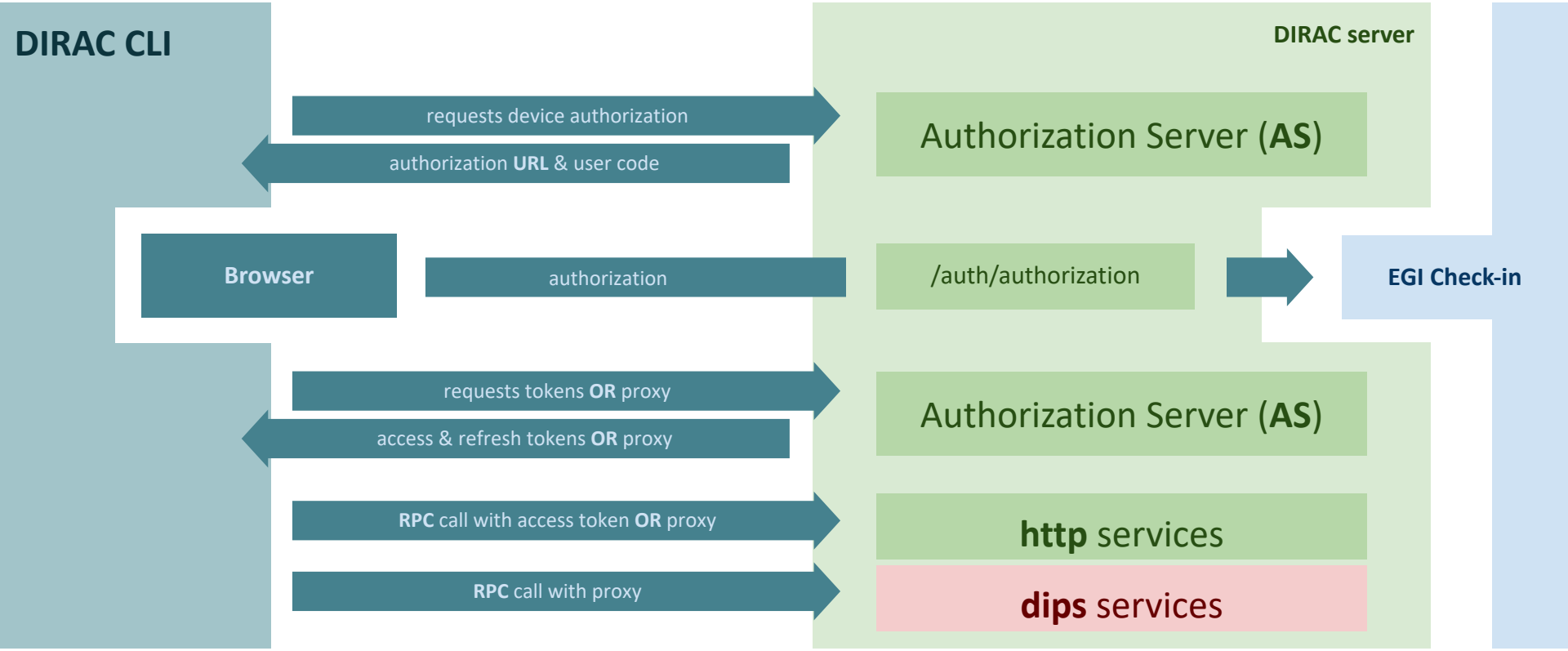

Introduction

- The DIRAC Project is developing its AAI subsystem
  - Compatible with both X509 certificates and OAuth tokens
  - Compatible with multiple IdP
- Main components
  - The DIRAC client/service protocol with AAI based on tokens
  - Token Management to provide valid tokens for asynchronous operations
  - User Management based on the information from IdPs
  - Connectors to external resources/services using tokens based AAI

Client/Service framework

- The client/service framework is updated to support both tokens and X509 proxies
  - Only HTTPS based DIRAC client service protocol is supporting tokens
    - The custom DISET protocol will stay with X509 proxies only
  - The updated framework is included in the 8.0 prereleases certification
- The user should first obtain a token before connecting to the DIRAC service
  - Using the OAuth standard protocols

User login protocol



CLI login

The login with **DIRAC CLI** is as follows:

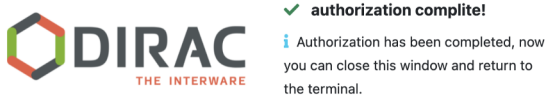
1 **Initialize authorization flow**

```
$ dirac-login --token
Authorization pending.. (use CNTL + C to stop)
```

2 **Authenticate via EGI Check-in in a browser**

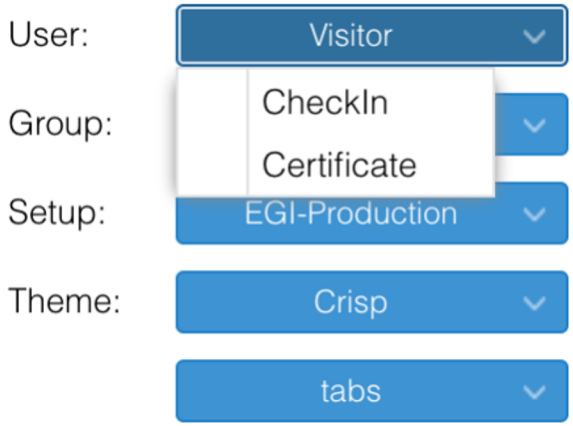
3 **Save the received token**

```
New token is saved to /tmp/bt_u504.
subject       : 22bca818-acea-46bd-b290-c7536c56f962
issuer        : https://wlcg.cloud.cnaf.infn.it/
timeleft      : 01:59:56
username      : alytov
DIRAC group   : wlcg_user
properties    : NormalUser
```



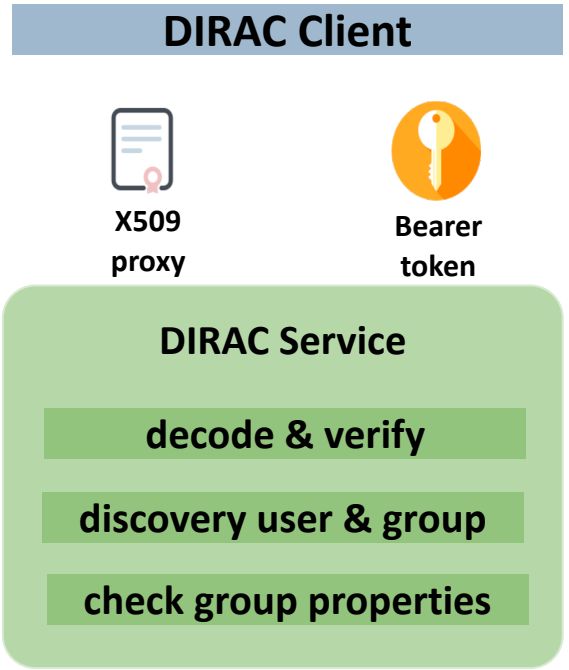
Web Portal login

- WebApp portal users will be given a choice of authorization method by selecting a certificate or identity provider



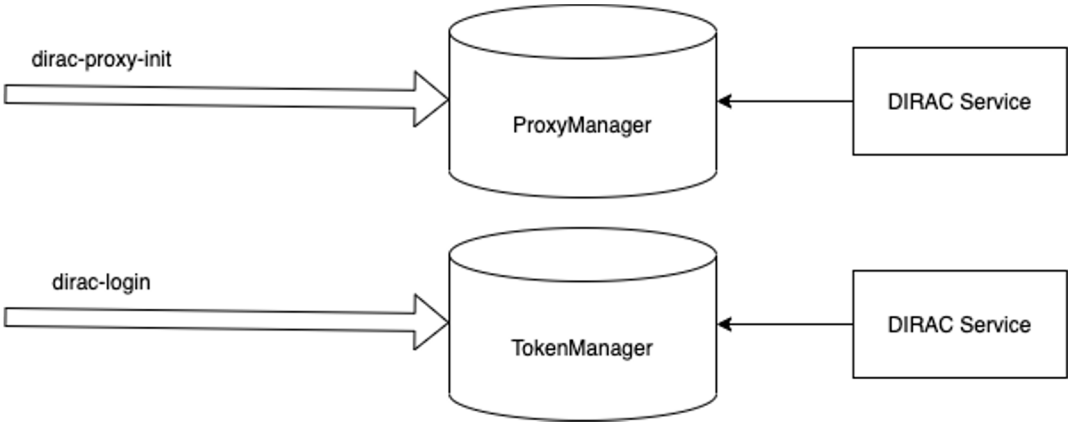
Access to DIRAC services

- The proxy or token received from the client is first checked and validated. In the case of a token, the signature with the public key of the corresponding **IdP** is checked.
- Using the DIRAC **Registry** and the information obtained from the proxy or token, the corresponding group is determined
- Finally, the **AuthManager** checks access rights of the received user group to the requested resource.



Token Management

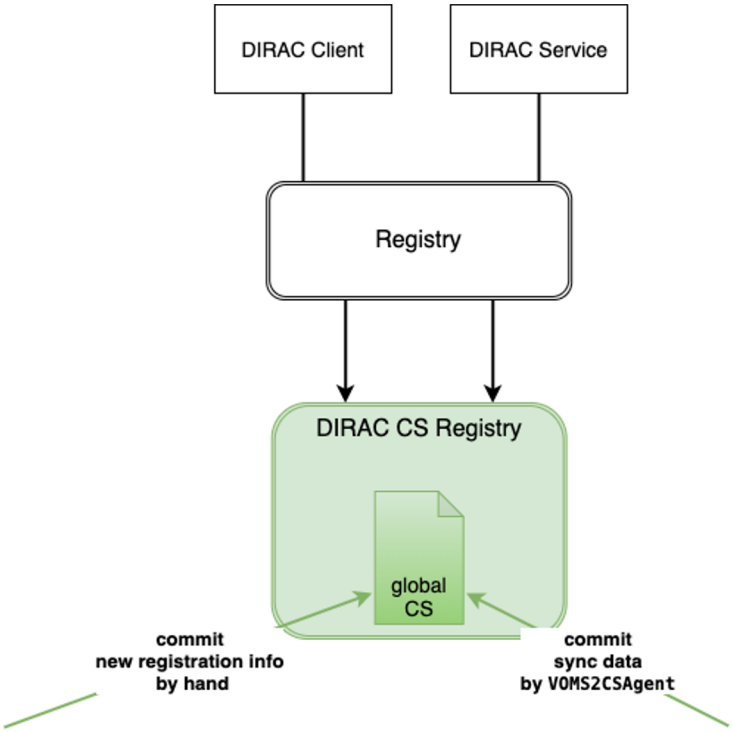
- Users delegate their rights to DIRAC by uploading their long proxy certificates. Similar mechanism for OAuth2 access tokens is necessary



- Proxies are stored and served by the ProxyManager service
- Tokens are also deposited in the database in the TokenManager service:
  - after successful authorization through the **IdP**
  - Both access and refresh token
  - DIRAC service being a registered client of the **IdP**, is able to maintain access tokens valid
- TokenManager service is developed and will be included in 8.0 release

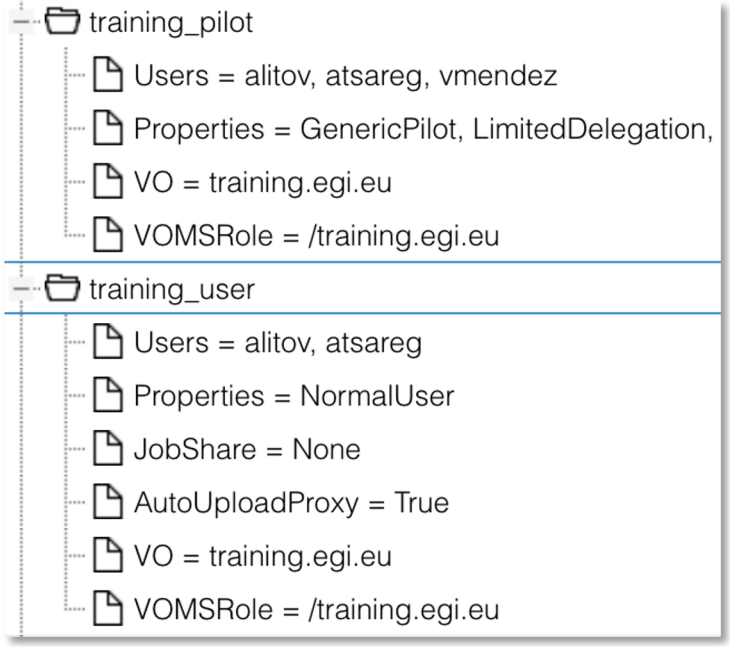
User Management

- Currently, all users are described in the DIRAC configuration in the Registry section
- The user's information is synchronized with the VOMS servers using VOMS2CS agent
- User's IdP information can be added by hand by administrator



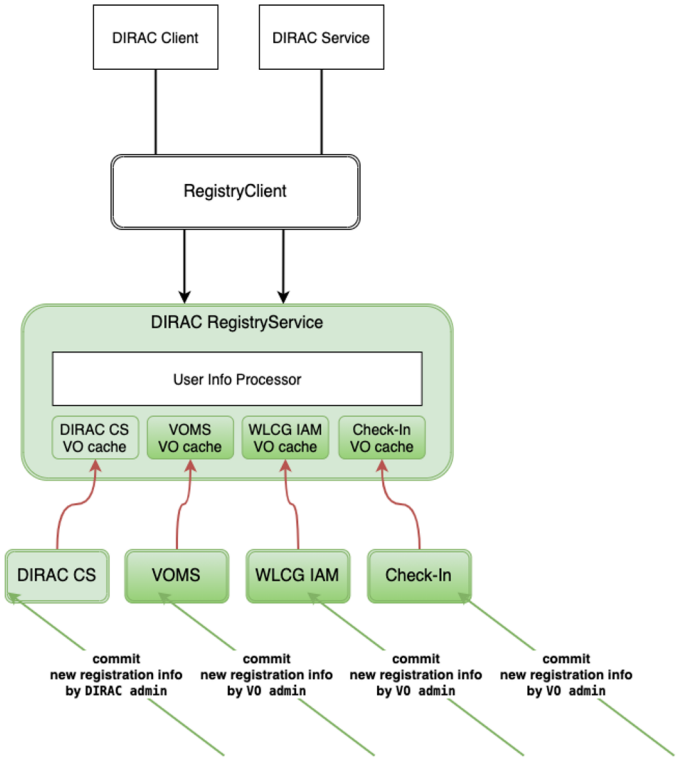
User Management

- User rights with respect to DIRAC services are determined by DIRAC groups
- Currently, DIRAC groups correspond to VOMS roles
  - One-to-many relation
  - DIRAC defines which users belong to which groups
- With IdPs DIRAC groups will correspond to unique scopes
  - One-to-one relation
- IdP scopes will completely defined user rights in DIRAC
  - User management is moved completely to IdPs



Dynamic User Registry

- **Registry service** receives information about VO users from **IdP's** or VOMS services. There is no need to store this information in the DIRAC configuration
- User management is completely outside DIRAC, this is done by VO administrators using IdP web interfaces
- DIRAC trusts and relies entirely on the information received from the relevant **IdP**.
- This is the work in progress



Access to third party services

- Using tokens to access computing resources
  - Access to Computing Elements (HTCondorCE, ARC) is in development
    - Need for token enabled services for testing
    - Use AREX ARC CE connector (*see Alexandre's talk*)
  - The Pilot framework will continue to use proxies at the first stage
    - Using tokens for pilot/server communications is to be developed as the next step
  - Access to Cloud resources with tokens was demonstrated (EGI Fedcloud sites)
    - Openstack and libcloud based connectors
    - Integration with the TokenManager, Cloud/SiteDirectors is to be developed
- Developing access to Storage Elements with tokens will start soon

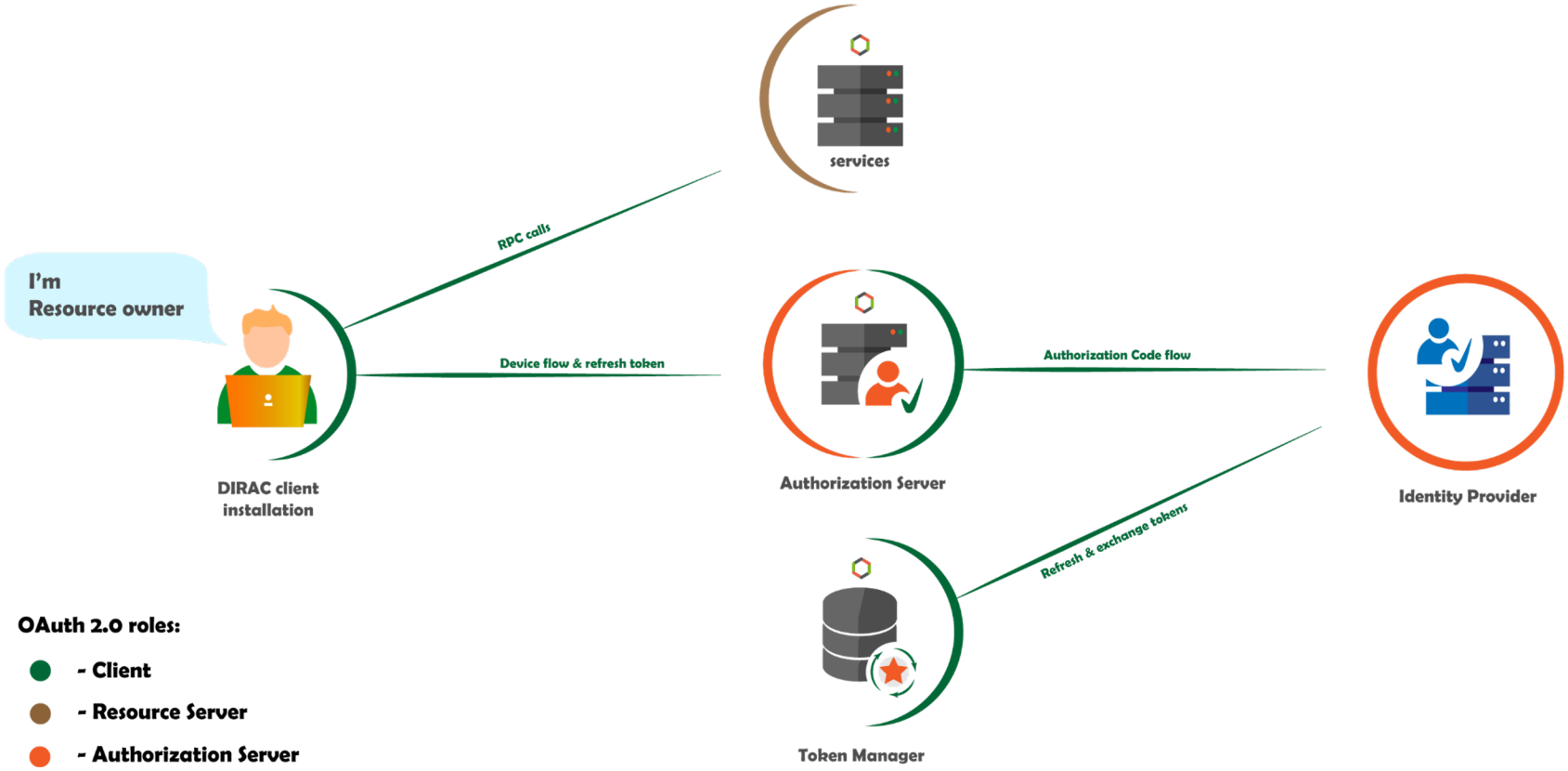
Summary

- The base DIRAC framework with tokens is ready for 8.0 release and will allow user's interaction with DIRAC service – both CLI and Web Portal access
- Basic User Management is developed. Dynamic Registry delegating User Management to IdP's is under development
- TokenManager service being certified – will allow asynchronous access to Computing and Storage Elements
- Token based connections are available for cloud resources. Connection to other resources (HTCondorCE, ARC, Storage) are under development

DIRAC
THE INTERWARE

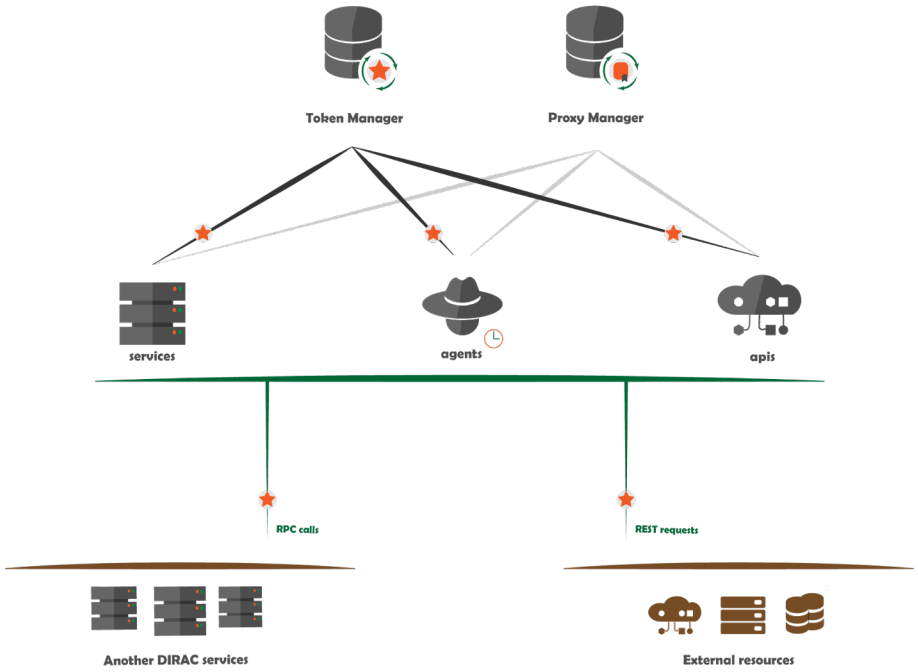
Backup

DIRAC
THE INTERWARE



OAuth 2.0 roles:

- **Client**
- **Resource Server**
- **Authorization Server**



OAuth 2.0 roles:

- **Client**
- **Resource Server**

Access type:

- **access token**