
DIRAC User's Workshop Summary



Federico Stagni, Andrei Tsaregorodtsev

- Format: fully online, 9-10 May
 - Developers' reports: review of subsystems with the focus on novelties
 - Users' reports: usage examples and requests to developers
- Participation
 - 63 registered users
 - effectively max 46 attending at the same time
 - few more participants on Monday, especially on the first part
 - "the usual suspects", and new faces
 - a few less participants than last year's, but tbh we did very little publicity
 - participation from the Americas of course problematic for the timezone

<https://indico.cern.ch/event/1107386/>

Installations and communities

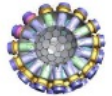


Shared by multiple experiments/projects, both inside
HEP, astronomy, and life science

Experiment agnostic

Extensible

Flexible



Developers' reports

Production

- **7.2 (March 2021)**
 - Code structure changed
 - PyPi deployment compatible
 - https services (few)
 - Python3 clients (optional)
- **7.3 (Sept 2021)**
 - https services (some more)
 - Python3 clients (default)
 - Python3 server (optional, but stable)

Upcoming

- **8.0 (May -> June 2022)**
 - https services (more)
 - Python3 only
 - Client and server
 - Initial support for tokens/OAuth2/IdPs
- **8.1**
 - All services via https
 - OAuth2: full support
- **8.2**
 - Abandoning DISET (?)

- DIRAC AAI subsystem
 - Compatible with both X509 certificates and OAuth tokens
 - Compatible with multiple IdP's
- 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 protocol
 - 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
- User login

CLI: `dirac-login -token`

Web Portal login

Authenticate via EGI Check-in or WLCG
IAM in a browser



Identity Provider selection..
Dirac itself is not an Identity Provider.
You will need to select one to continue.



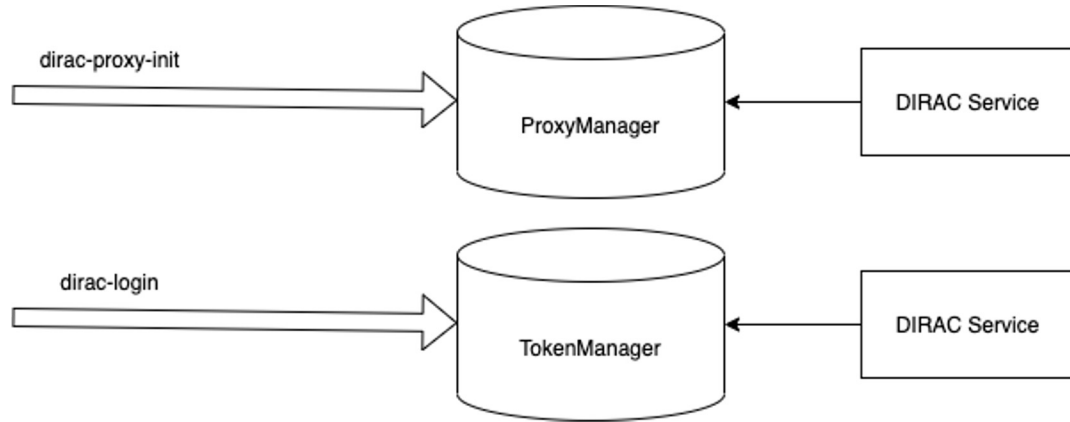
User:

Group:

Setup:

Theme:

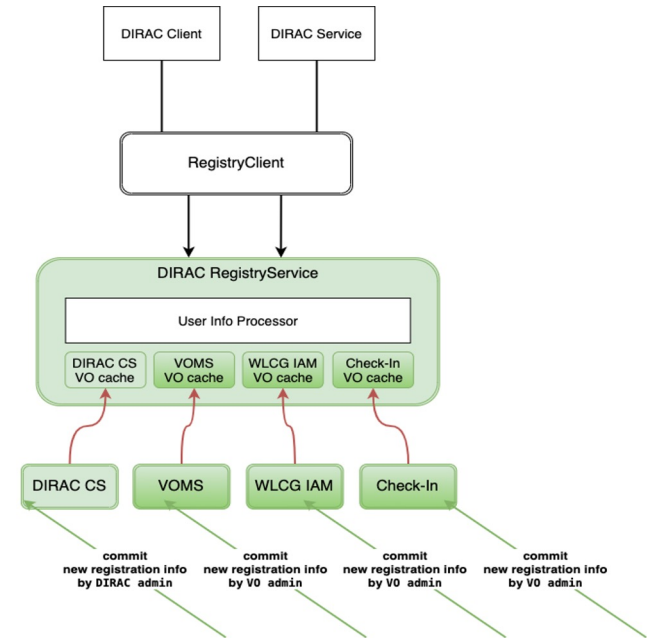
Token Management



- 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
 - Certification pending

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
 - **HTCondorCE** connector is upgraded – being tested
 - **ARC CE** connector using **AREX** REST interface will be instrumented with tokens support
 - The Pilot framework (pilot/server communications) will continue to use proxies
 - Access to Cloud resources with tokens was demonstrated (EGI Fedcloud sites)
 - Openstack and apache-libcloud based connectors
 - Using personal client for token generation (oidc-agent)
 - Integration with the TokenManager, Cloud/SiteDirectors is to be developed
- Developing access to Storage Elements with tokens will start soon

WebApp framework

- Major upgrade: Python3, deployment in the PyPi repository
 - Ready for the DIRAC 8.0 release
- Sharing the same framework as for the HTTPS based DIRAC service handlers
- Ready for OAuth tokens based authentication
 - E.g. running in production in dirac.egi.eu portal

REST interface

- Used by several communities in the EGI DIRAC service
- Should be upgraded to use the new WebApp framework

COMDIRAC

- A command line interface close in syntax to shell and batch system commands
 - **dsub, dstat, doutput**
 - **dput, dget, drepl, dls, dcd, dpwd, dfind, dsize**, etc
 - Heavily used by several communities
- Updated to python3, PyPi deployment

- Continuously evolving landscape, adaptation is key
 - HTCondor: support for MP slots
 - ARC: moving to use ARES REST interface
 - CloudComputingElement: uniform interface with other CEs
 - Allow to drop Cloud (VMDIRAC) subsystem eventually
- Several developments ongoing for supporting HPCs, moving in different directions
 - PushJobAgent for sites without outgoing connectivity
 - PoolCE – inner CE to exploit efficiently multi-core slots
 - Based on ProcessPool

- Monitoring of pilots being improved
 - common request from the previous workshop
 - pilot logs stored in a SE
 - Common mechanism for all the computing resources
- DB12 benchmark
 - fully ported to Python 3
 - Deployed in PyPi as independent package
 - Analysed results:
 - found some discrepancies (python2/python3, AMD/Intel)

- **DMS**
 - One of the focus is keep supporting protocols and storages
 - https, CTA, etc
 - (limited) multihop implemented
 - if anyone needs it
 - several building blocks
 - Most of functionalities are in plugins
 - pragmatic approach
- **DIRAC → Rucio**
 - RucioFileCatalog developed for Belle II
 - An extended interoperability (not only a RucioFileCatalog) is added in DIRAC v7.3
 - Syncing users and resources
 - Checked against the multi-VO Rucio instance of GridPP

- Several new developments presented, but lots of work is going on under the hood
 - Often “just better code”
- Trying to profit of the “on the shelf” tools, and in-house developments kept only when really necessary
 - ES/Opensearch, Kibana/Grafana, Tornado, MQ, celery, etc
 - this has the obvious drawback of needing support for other tools than DIRAC
- VOMS → CheckIn/IaM/ ?
 - drive for re-thinking of DIRAC AuthN/Z, but also the protocol

Experiments' reports

- Single community installations
 - LHCb, Belle II, CTA
- Multi-community services
 - CLIC/ILC/FCC/Calice, GridPP, EGI, IHEP, JINR
- Everyone's using WMS and Accounting
- Everyone's using at least partially the DMS
 - everyone (but Belle2) use the DFC
 - few more installations needing to use Rucio
- Several using RMS + TS
 - ProductionSystem also available, but not widely used yet
 - Used by CTA
 - if RMS+TS not yet used, the suggestion is to use RMS+TS+PS

- BDII cited in few reports, DIRAC users still rather depending on it, but no obvious solution for a single definite source of resources information
- Support for central DIRACOS2 CVMFS installation that can be reused by different pilots can be very useful
 - Clients are available from `/cvmfs/dirac.egi.eu`
 - Making pilots available from `cvmfs` needs some developement

- Pilots monitoring independent of resources is still requested
 - (cited in few reports) but good to see that this is progressing
- Some in-house developments have the potential of becoming part of vanilla DIRAC:
 - ILCDirac jobs and requests monitoring
 - Rucio–DIRAC FC methods from Belle2
 - Scout jobs (Belle2)
 - Dataset Searcher ? (Belle2)
 - CTADIRAC transformations' datasets
- Not many updated to python 3, but those who did are happy
 - several still on v7r2, if not v7r1
- No-one mentioned using HTTPS DIRAC services (LHCb has a few)
 - Few available also in v7r2, why not tried out?

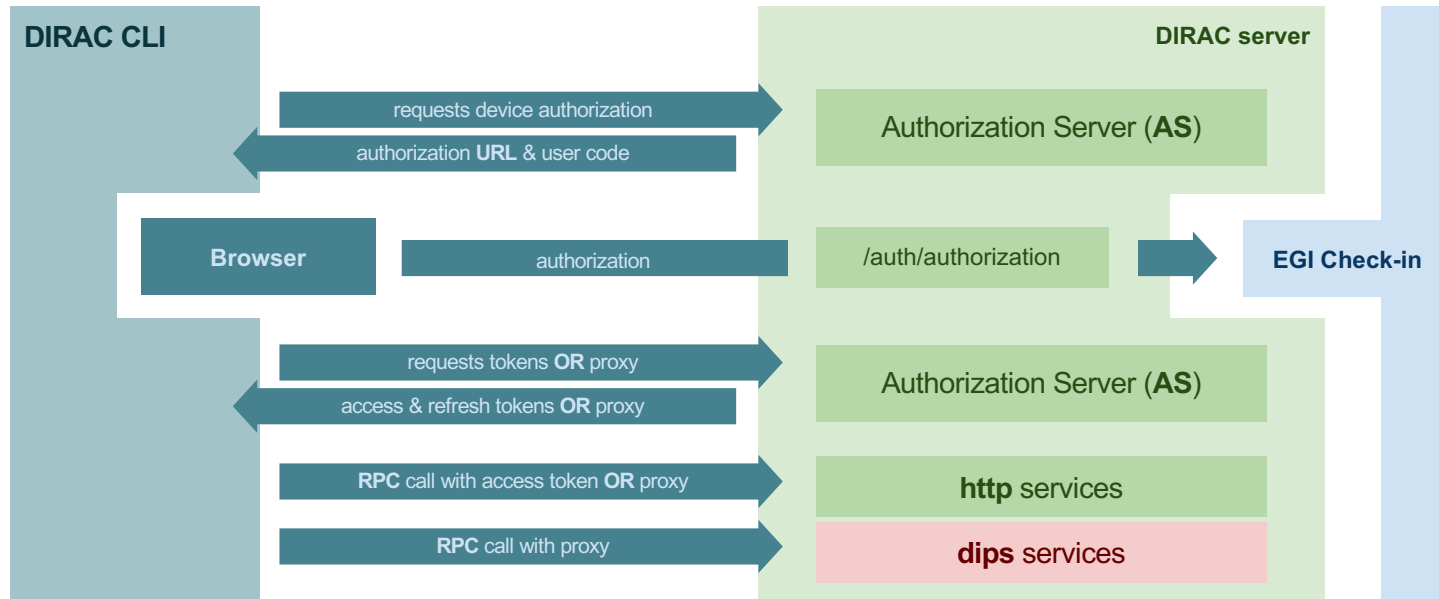
- Current members :
 - CERN, CNRS, IHEP, KEK, IC, UoM
- 2 new official members joined the Consortium in 2021
 - Imperial College
 - University of Montpellier
- Membership of PNNL in the Consortium is stopped
- Extension of the Consortium Agreement for another 5 years till 2027 is being prepared

- Many ongoing developements to follow technology evolution and to improve code quality
- Pragmatic focus on the needs of DIRAC communities
- Multiple extensions developed with a good potential for contributing to the core DIRAC

Workshop recordings are available at
<https://indico.cern.ch/event/1107386>

Backup

User login protocol



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

1 Initialize authorization flow

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

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
```

2 Authenticate via EGI Check-in in a browser



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

User:

Group:

Setup:

Theme: