Token support in Rucio

Jaroslav Günther

(on behalf of the Rucio team)

DOMA / TPC Meeting

20th May 2020

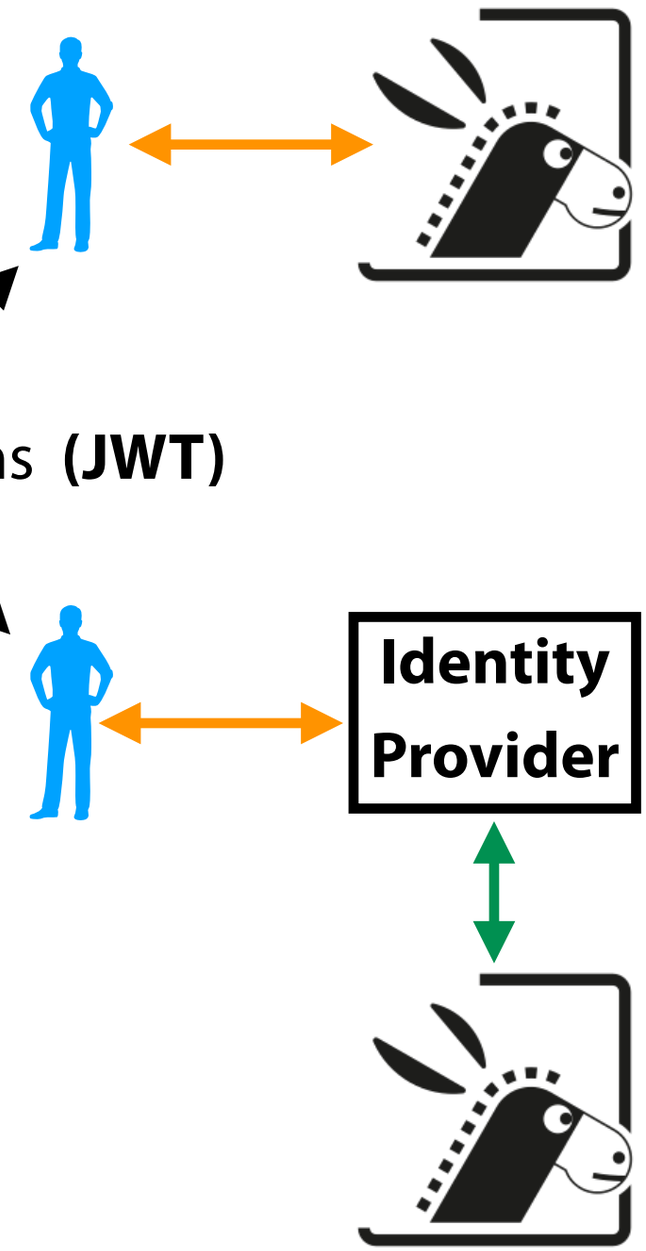
User Identity

User Identity in Rucio

✓ Rucio user = "account" + "identity" (N:M mapping)

✓ “account" = nickname in VOMS (CERN LDAP username)

✓ "identity" = specification of authentication type
+ user identifier



✓ supported identities:

- username/password, X.509, SSH public keys, GSS/Kerberos
- Open ID Connect (AuthN), OAuth 2.0 (AuthZ), JSON Web Tokens **(JWT)**

✓ User Registration **Not Free:**

- only carefully ‘pre-provisioned’ users are allowed
- Rucio daemon syncs accounts & identities with VOMS + CERN LDAP

Rucio Authentication Workflows 1/2

User requests JWT from Rucio (authorization code flow)

✓ 3 Rucio Client (**RC**) CLI authentication methods **(authorization code flow)**
- RC polling RS for an access token (**AT**) after authentication
- RC waiting for fetch code (copy-pasting fetch code from Internet Browser)
- automatic, RC trusted with user's IdP credentials **(strongly discouraged)**

✓ oauth_manager daemon (Rucio Auth Server - **RS**)
- manages tokens saved in Rucio DB (delete/refresh)

✓ RC stores locally user's AT in a file
- AT file location can be configured (rucio.cfg)
- no need to re-authenticate: RC can get a new AT from RS regularly

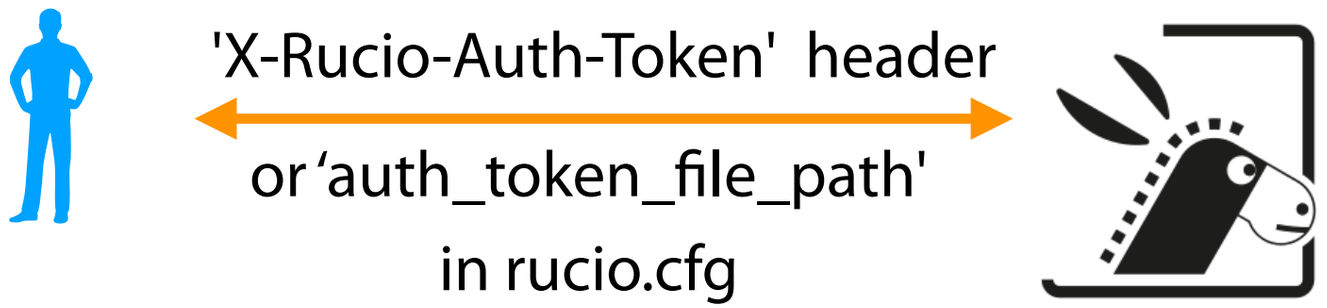
Basic re-auth assumptions:
valid AT + refresh token (RT) in Rucio DB
+ scope, audience as Rucio requires
+ user's identity registered in Rucio DB

Rucio Authentication Workflows 2/2

User has a JWT from external sources

✓ User presents ATs to Rucio in 2 ways:
- REST API
- Rucio Client token file

Basic auth assumptions:
presented valid AT
+ scope, audience as Rucio requires
+ user's identity registered in Rucio DB



✓ RS saves the AT and:
- exchanges AT when passing it down to services (FTS)
- the exchanged AT comes with RT —> triggers automatic refresh (RS oauth_manager)

TO-DO: - refresh lifetime configurable also from REST API

✓ Rucio external JWT management (to be discussed):
- via token exchange (external AT exchanged and return to user Rucio AT)
- user's external tools (oidc-agent)

✓ Should Rucio Client &/ REST API provide external token management tools ?

Rucio DOMA testbed

```
[client]
rucio_host = https://rucio-doma.cern.ch
auth_host = https://rucio-doma-auth.cern.ch
[…]
```

✓ Full chain tests (me, Alberto Brigandi, Marica Antonacci, Paul Millar)

Rucio —> RSE + rules —> FTS —> dCache

- "user identity, capability-based authz" **works !** (XDC IAM + dCache + webdav)
- "service identity, capability-based authz" **to-be-tested**
- XDC & WLCG IAM instances used

  [required scopes, audiences for Rucio and FTS (token exchange)

  configurable on the RS side]

✓ Rucio direct interaction with storage [`rucio upload/download`] **works !**

(XDC dCache + webdav)

- gfal needs testing

Rucio DOMA testbed

- ✓ Rucio Client configuration tests (Thanks to Paul Millar for spotting few bugs !)
- ✓ REST API testing (me, Alberto Brigandi)
- ✓ oauth_manger daemon
  - successful long term token refresh and token deletion
    + deletion of expired oauth sessions

TOKEN	ACCOUNT	IDENTITY	CREATED_AT	EXPIRED_AT	OIDC_SCOPE
root-ddmlab-unknown-33e684	root	ddmlab	2020-05-15 14:23:51	2020-05-15 15:23:51	[NULL]
root-ddmlab-unknown-f42d81	root	ddmlab	2020-05-15 14:23:51	2020-05-15 15:23:51	[NULL]
eyJraWQiOiJyc2ExliwiYWxnIjoi	guenther	SUB=b3127dc7	2020-05-15 13:56:17	2020-05-15 14:56:16	openid offline_access profile
eyJraWQiOiJyc2ExliwiYWxnIjoi	root	SUB=2927e1d8	2020-05-15 14:33:27	2020-05-15 15:33:27	fts:submit-transfer

OIDC_SCOPE	REFRESH_TOKEN	REFRESH_START	REFRESH_LIFETIME	REFRESH_EXPIRED_AT
[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
openid offline_access profile	eyJhbGciOiJub25lIn0	2020-05-09 16:06:02	192	2020-05-19 13:56:17
fts:submit-transfer	[NULL]	[NULL]	[NULL]	[NULL]

Summary

Done

✓ full chain transfer tested to work with XDC IAM instance

✓ Rucio OAuthManager for token refresh and deletion

✓ Rucio Client capabilities improved

Requests to consider

✓ Rucio Client token management via env variables (Andrea's proposal)

✓ refresh lifetime management via REST API

✓ Rucio Client & REST API management of "external" user tokens

- is it needed or just a convenience ?
- if needed, token exchange strategy vs tools such as oidc-agent

More Testing needed:

✓ we need more people to test with WLCG IAM instance

✓ we need more RSEs with other storages

Thank you for your attention !

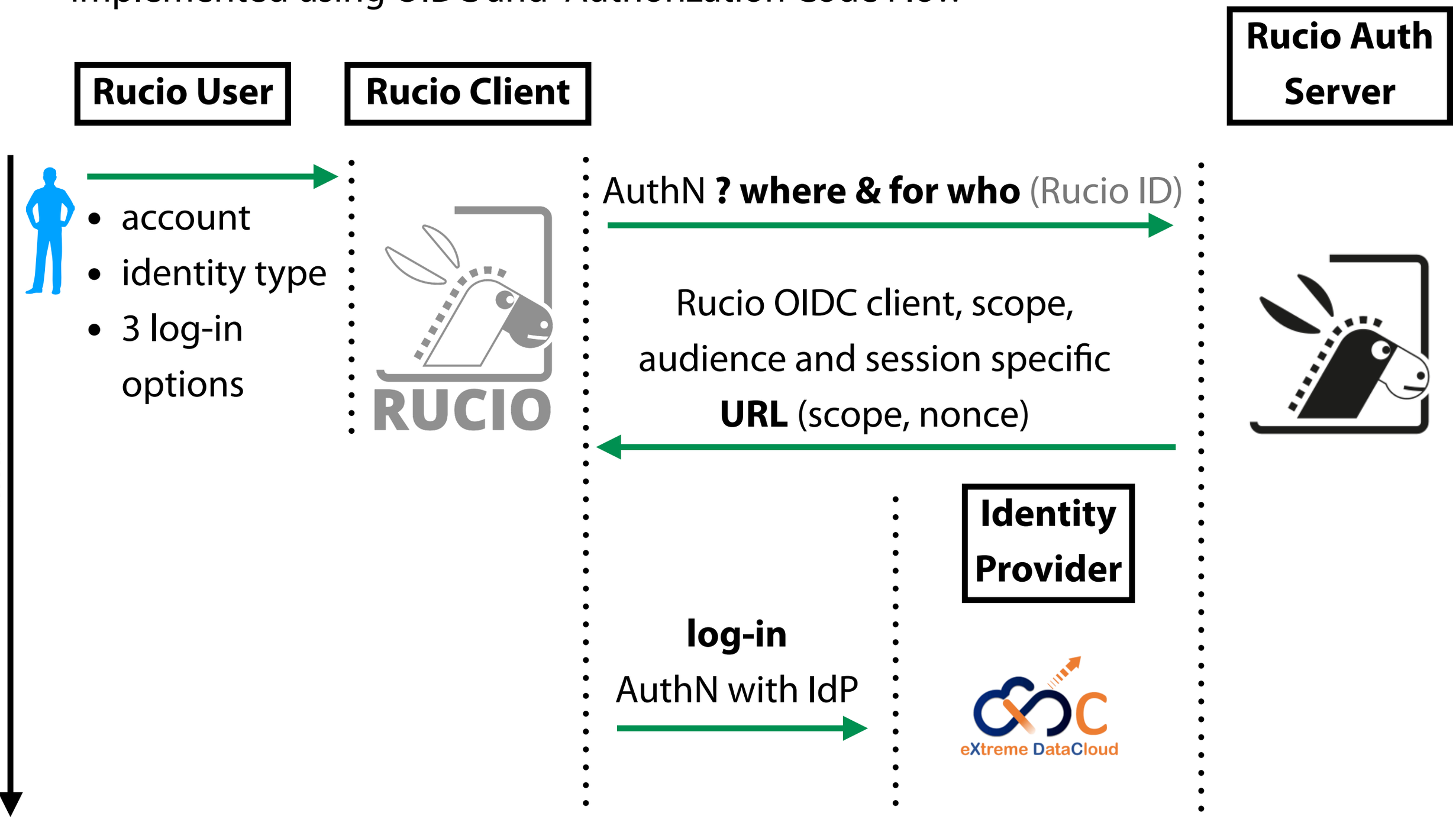
and special thanks to Andrea Ceccanti, Paul Millar, Alberto Brigandi and Marica Antonacci for their help with testing !

Rucio Authentication

New Industry Standards & WLCG (link):

✓ shift towards federated identities

- implemented using OIDC and "Authorization Code Flow"



Rucio Authentication & Authorization

Rucio User & Client | **Identity Provider** | **Rucio Auth Server**

time

logs in

Do you **really** trust Rucio ?

scope authorization*

session **state** (AuthN) + code (AuthZ)

? user info & token

state check (known AuthN ?)

user info & token

identity + nonce (replay attack) check

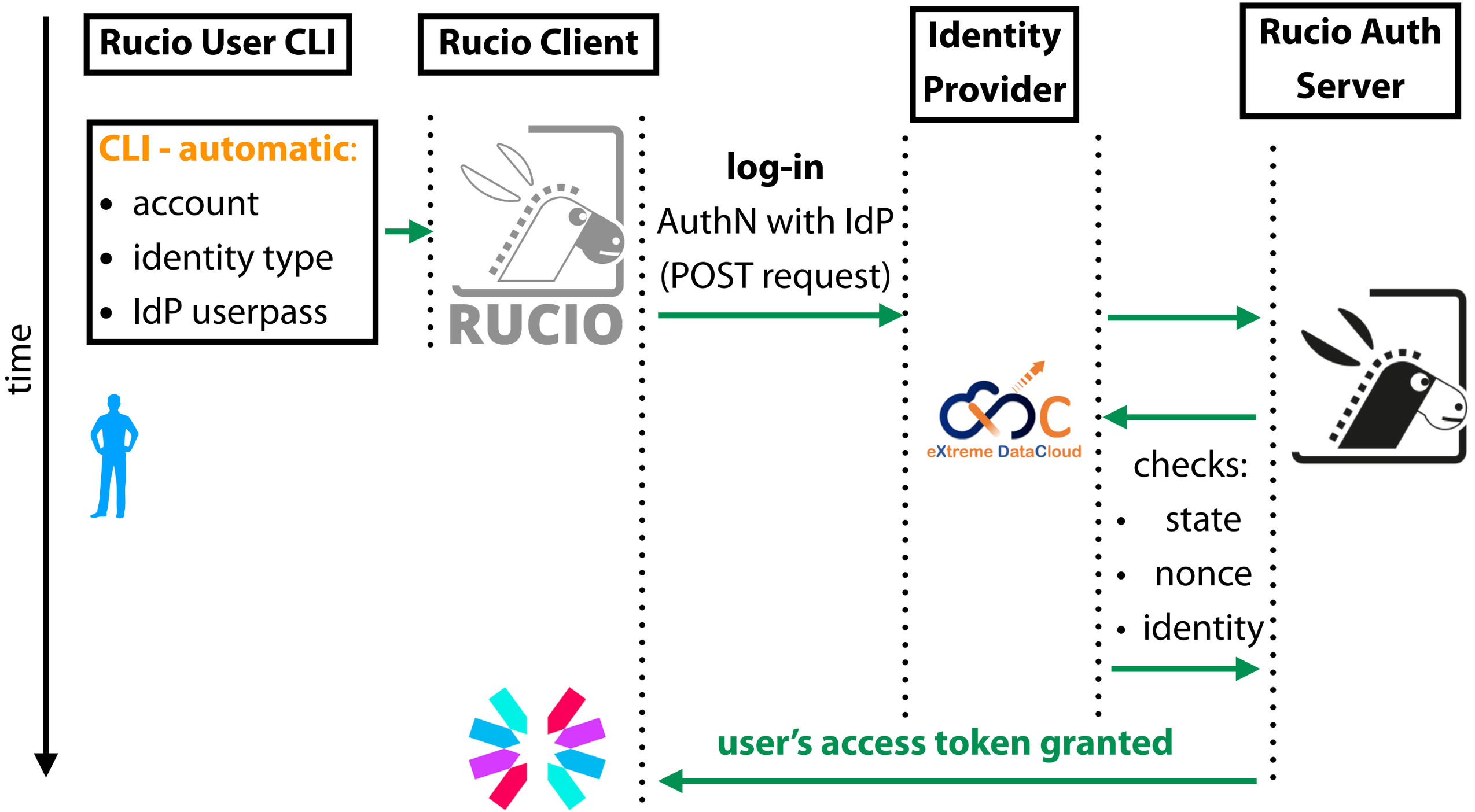
user's access token granted

*Rucio Client can say "yes" for the user automatically here

Rucio User IdP log-in

Rucio IdP Log-in strategy:

✓ Do you trust Rucio Client with your IdP password ?



Rucio User IdP log-in

✓ No, you do not trust → semi-automatic option

Rucio User CLI

Rucio Client

Rucio Auth Server

Web based:

- account
- identity type
- **with polling**

specific **URL to Rucio Auth Server endpoint**

Web Browser

Identity Provider

checks:

- state
- nonce
- identity

gets token

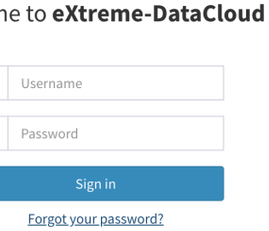
time

URL copy-paste CLI

→ web browser

→ Rucio Auth Server

→ Identity Provider Login Page



Rucio authorization server has been granted access to your information. Rucio Client should now be able to fetch your token automatically.

All OK page

Rucio Client fetches token from Auth Server

Rucio User IdP log-in

✓ safest option available now

Rucio User CLI | **Rucio Client** | **Rucio Auth Server**

time

Web based:
- account
- identity type

specific **URL to Rucio Auth Server endpoint**

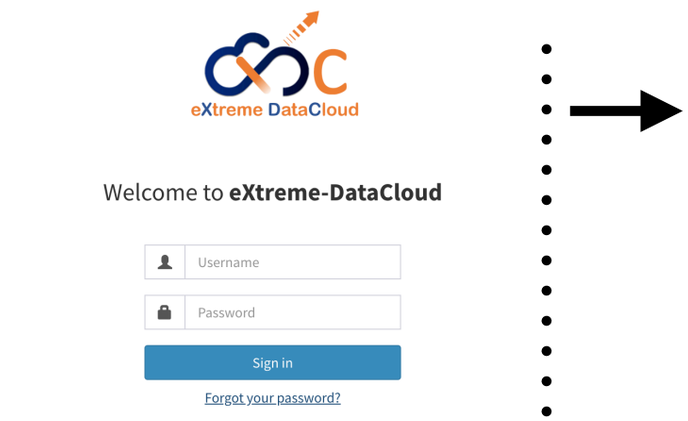
URL copy-paste CLI

→ web browser

→ Rucio Auth Server

→ Identity Provider Login Page

Web Browser



Identity Provider

checks:
- state
- nonce
- identity

gets token

fetch code

fetch code copy-paste

web browser → CLI

RUCIO SCIENTIFIC DATA MANAGEMENT

Rucio authorization server has been granted access to your information.
Please copy-paste the following code to the open terminal session with Rucio Client in order to get your access token:

HdwH1VXkniuA8C6DTOS6wsJJSzwNA9XSEwAG8dMFZXXGfapEyu

Rucio Client fetches token from Auth Server