Status of CERN Authentication and Authorisation

Asier Aguado Corman, Hannah Short, Adeel Ahmad, Maria Fava, Sebastian Lopienski, Antonio Nappi, Paolo Tedesco

HEPiX Spring 2023

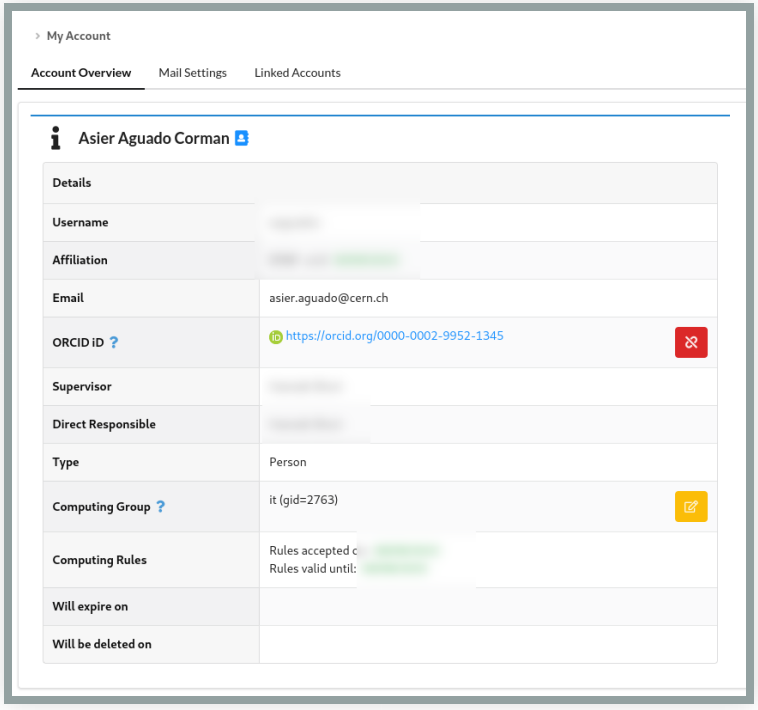
Summary

1. Our service
2. Architecture summary
3. Single Sign-On software and customisations
4. Special authentication use cases
   6.1. Machine to machine authentication
   6.2. Command line access
5. Two-factor authentication
6. Other challenges and future roadmap

What our service offers

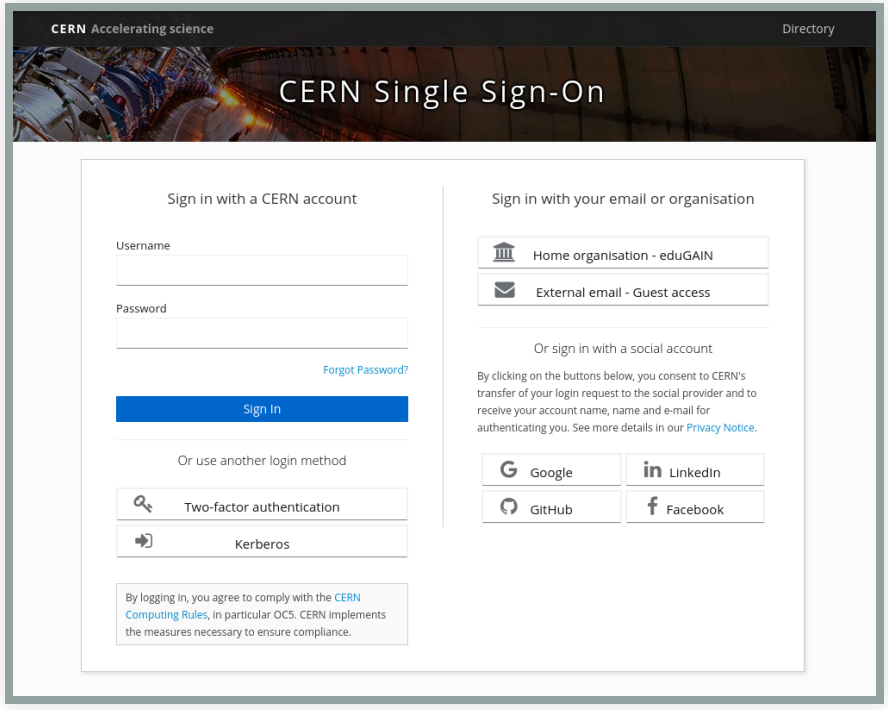
Identity Management

Identity and account lifecycle.



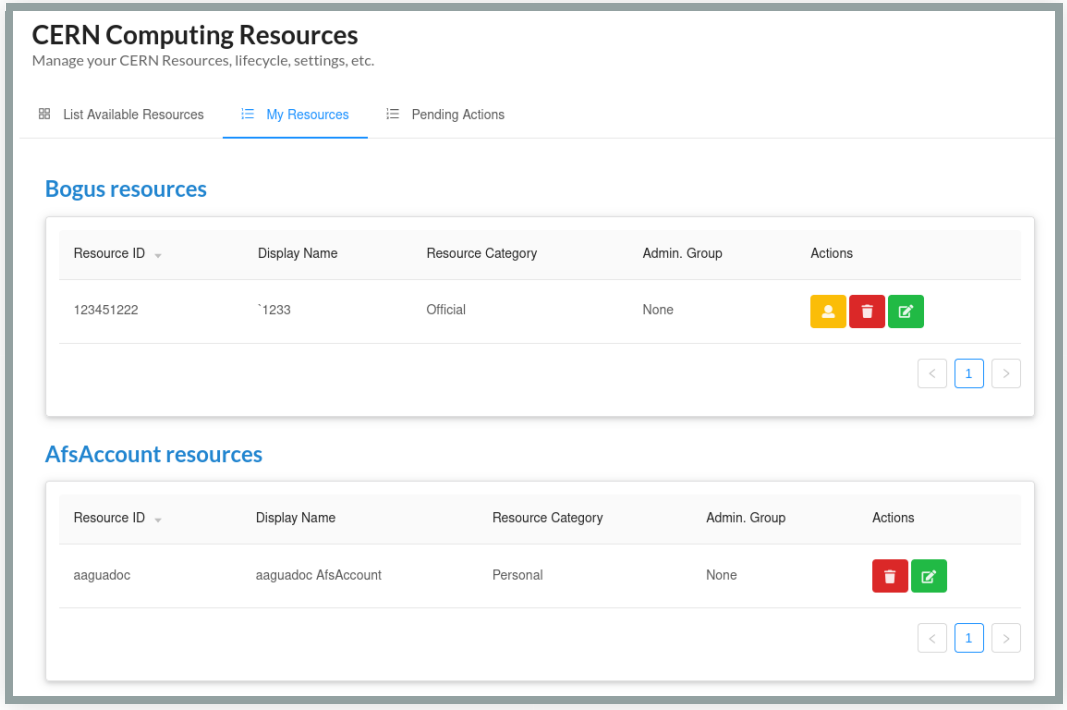
Authentication and Authorisation

- Single Sign-On
  - OAuth, OpenID Connect and SAML.
- Groups, Roles and levels of access/assurance (LoA).
- X.509 Certificate Authorities
  - Not covered in this presentation.



Computing Resource management

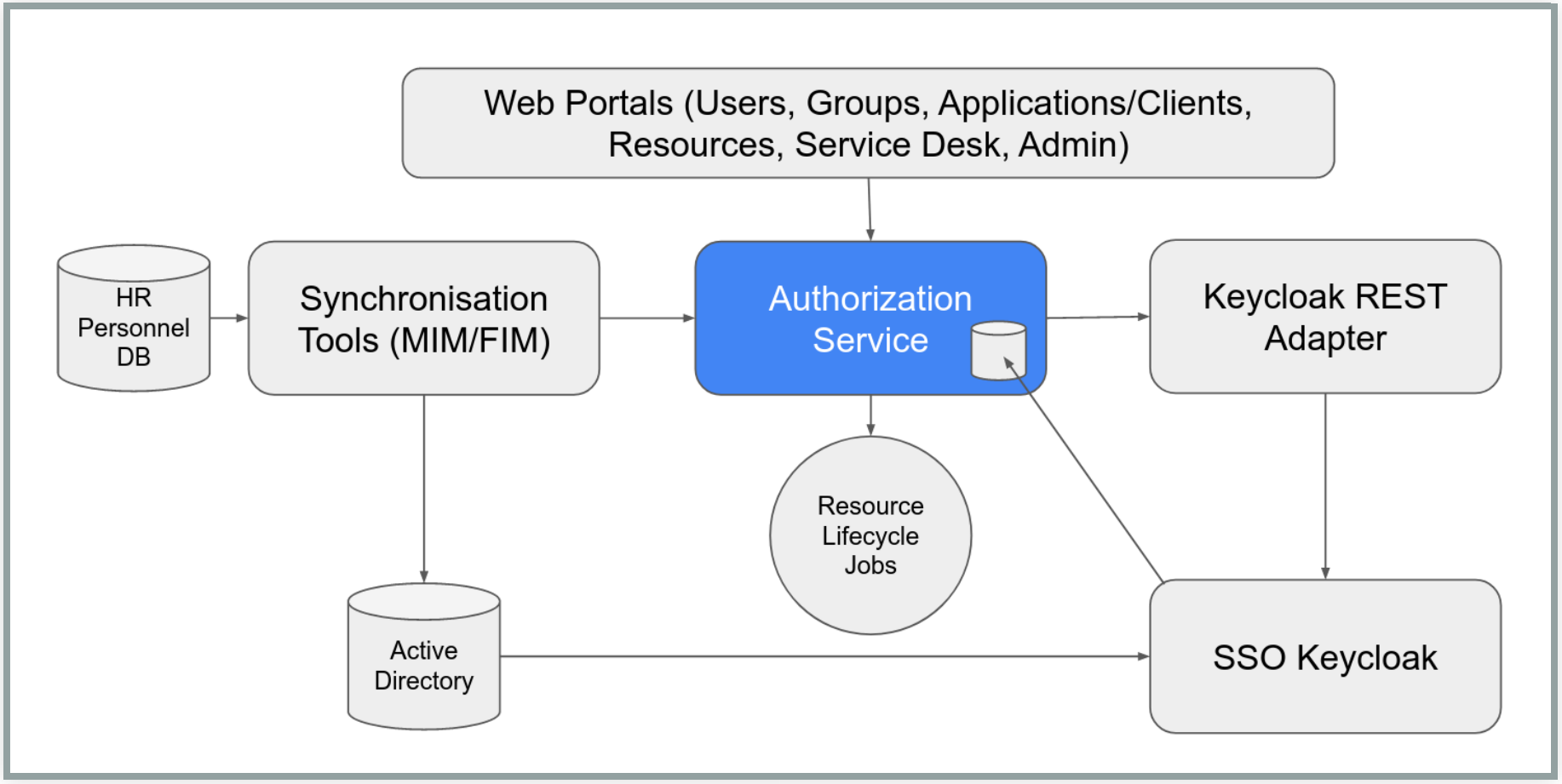
- Manage computing resource ownership
  - Such as websites, database accounts.
- Automatically reassign or delete resources
  - Based on elegibility (e.g. no longer at CERN).



Current usage

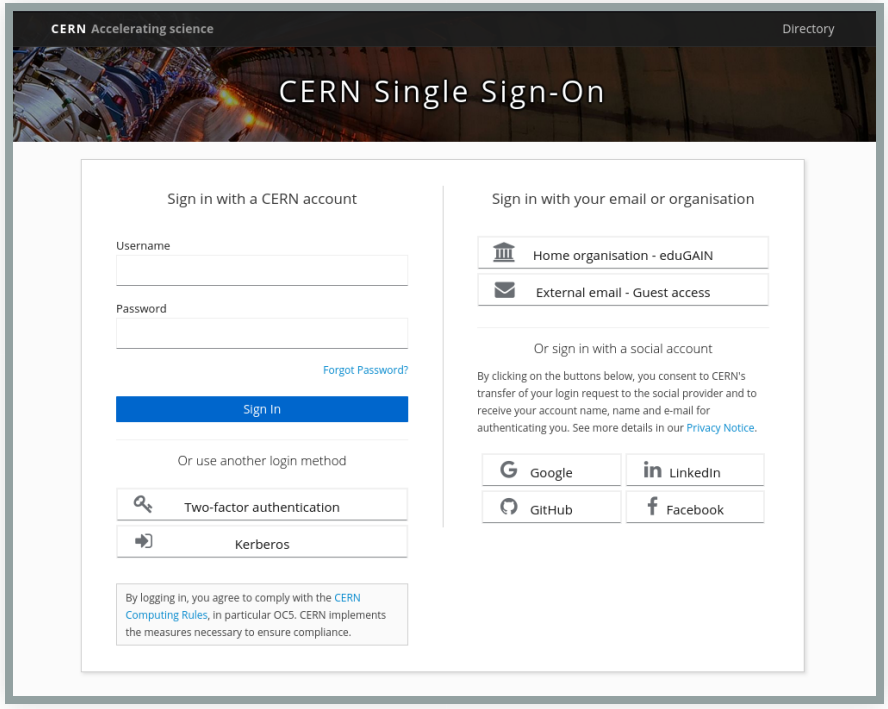
- 35,000 individual users
- 25,000 logins per day
- **More than 9,000 applications and websites**

CERN AAI Architecture



Single Sign-On

- Based on the open source product Keycloak
- Uses OpenID Connect (OIDC) and SAML standards
- Replaces the old SSO based on ADFS
- Customised to fit our use case



Overall Impression

- Positives
  - Upstream project of Red Hat Single Sign-On
  - Keycloak community is very active
  - Project applied to CNCF for incubation
  - Good compliance with OIDC and SAML standards
    - More focus on OIDC
  - More features are getting added
    - The initial releases needed more customization
  - Scalability and performance have improved

- Negatives
  - Occasional bugs causing some integrations to break
    - Recent bug in SAML response affecting the Microsoft federation
    - It was resolved very quickly
  - Uncertainty about future database support
    - MySQL and Oracle will be dropped

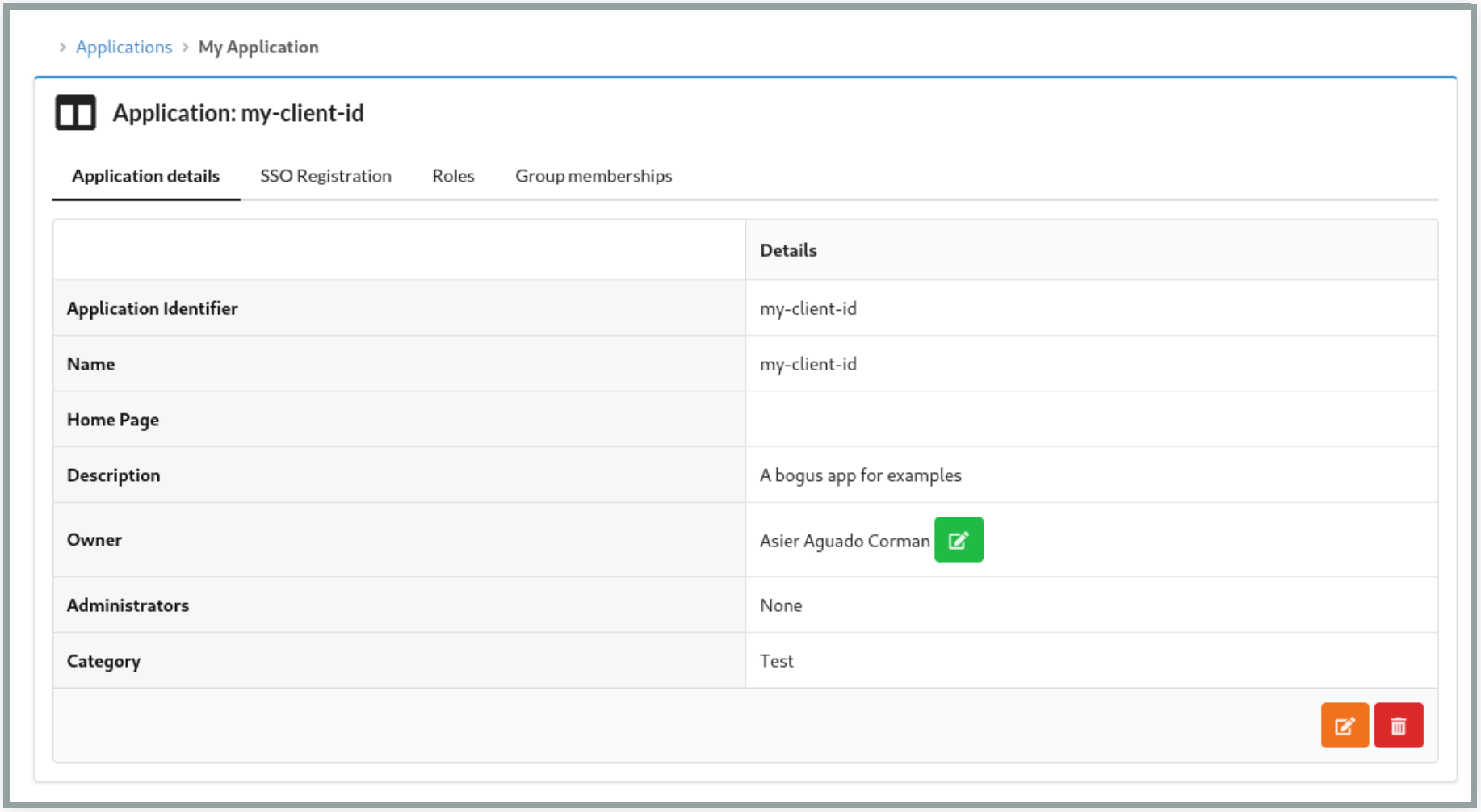
SSO Customisations

- Keycloak extensions: Service Provider Interfaces (SPIs)
  - Mappers to fetch user properties externally
  - Groups and Roles from outside Keycloak
  - Custom Level of Assurance implementation (before LoA was introduced in Keycloak)
  - Endpoint to get API tokens for any audience: uses Client Credentials Grant with an `audience` field
  - Endpoint to validate OTP codes: we use it together with a PAM module for 2FA over SSH
  - Compromised password detection: we compare input passwords with a database of leaked passwords. This allowed us to stop annual password changes.

- Keycloak Configuration
    - Optional 2FA in a second realm: we will move to step up 2FA now that is has been introduced
    - Account management is disabled: we use our own external database, API and portals
- Wrappers
    - A Keycloak REST Adapter developed over the Keycloak Admin API, e.g. to reset 2FA. This was made available on Github for other communities.

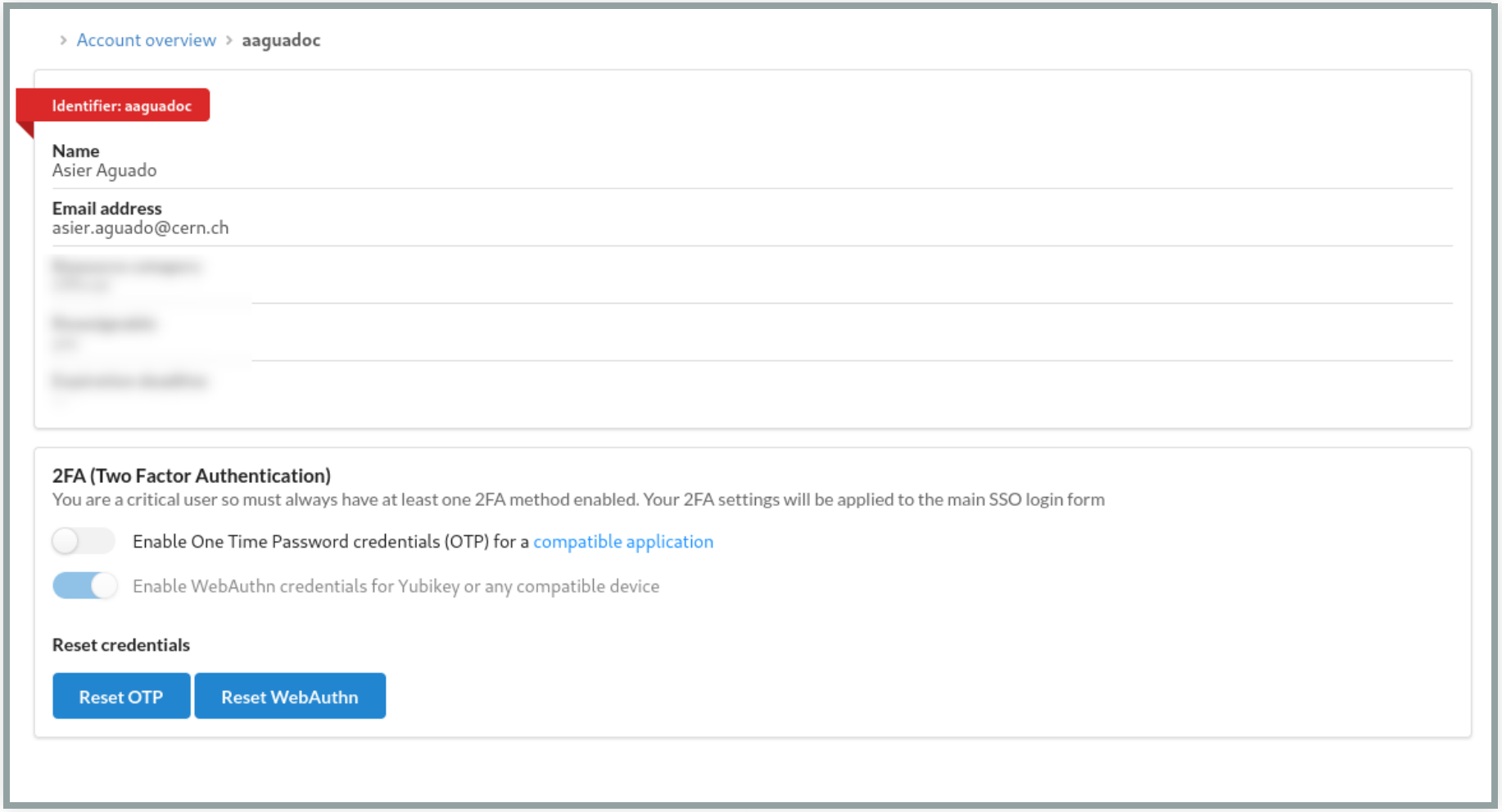
Application Portal

Users can register their applications in this portal.



2FA settings in Users Portal

Account console is disabled in Keycloak.



HTTP Endpoint to verify 2FA codes

Adds new possibilities for two-factor authentication.

```
$ ssh aaguadoc@aiadm.cern.ch
(aaguadoc@aiadm.cern.ch) Your 2nd factor (aaguadoc): 123456
```

Machine to machine authentication

A service authenticates to another service where it **doesn't necessarily have control or credentials**.

- Old approach: Cookie based authentication, using a command line tool logs into the SSO using Kerberos and SPNEGO and saves cookies into a file.
    - We developed the same tool for the new SSO.
- Modern approach: OAuth2 and JWT.
    - OAuth2 Client Credentials Grant.
    - Custom *API Access* endpoint to replace audiences.
        - Role based access control

Before:

```
$ kinit -t myservice.keytab myservice@CERN.CH
$ auth-get-sso-cookie -u https://myapi.cern.ch -o cookies.txt
$ curl -L -b cookies.txt https://myapi.cern.ch/foobar
```

Now:

```
$ curl --location --request POST "https://auth.cern.ch/auth/realms/cern/api-acc
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode "client_id=my-client-id" \
--data-urlencode "client_secret=bdebbdcc-c33c-11ed-b0a8-3822e22949e4" \
--data-urlencode "audience=myapi" | jq -r .access_token > token.txt
$ token=$(<token.txt)
$ curl -X PUT "https://myapi.cern.ch/api/foobar" -H  "authorization: Bearer $tc
```

Command line access

Accessing protected web resources from a CLI through `wget`, `curl` or similar.

- Old approach: Same command line tool as in the previous use case
- Modern approach: OAuth2 Device Authorization Grant.
  - Users log in using a web browser window outside the terminal.
  - Positive: Compatible with all modern 2FA protocols.
  - Negative: Resistance towards moving to this because it requires a round trip to a browser

Before:

```
$ kinit
Password for aaguadoc@CERN.CH:
$ auth-get-sso-cookie -u https://the-target-api.cern.ch -o cookies.txt
$ curl -L -b cookies.txt https://the-target-api.cern.ch/foobar
```

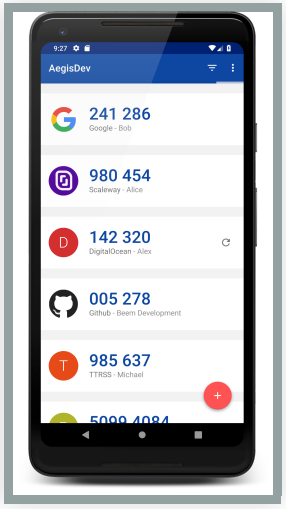
Now:

```
$ auth-get-user-token -c myapi -x -o token.txt
CERN SINGLE SIGN-ON

On your tablet, phone or computer, go to:
https://auth.cern.ch/auth/realms/cern/device
and enter the following code:
KFRX-JXIV

You may also open the following link directly and follow the instructions:
https://auth.cern.ch/auth/realms/cern/device?user_code=KFRX-JXIV

Waiting for login...

$ token=$(<token.txt)
$ curl -X PUT "https://myapi.cern.ch/api/foobar" -H  "authorization: Bearer $tc
```

Two-factor authentication

The new SSO supports

- Authenticator Applications (OTP)
- Security Keys and other hardware (WebAuthn)



New 2FA strategy

- CERN was previously using step-up 2FA, which was not supported by Keycloak.
- We decided to enforce 2FA to every login of a critical account:
  - Defined *critical users* instead of *critical applications*.
  - The change was well received by users who already used 2FA daily.
  - Poor reception by users who occasionally used *critical applications*, 2FA on every login means less convenience (especially for mobile).

- Possible changes after user feedback:
  - Step up 2FA defined by critical applications.
  - Non-critical users can opt-in for *always-on* 2FA.
  - Reassess critical user membership.

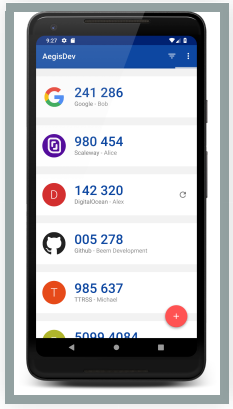
***Always-on* 2FA Rollout**

September 2022:
About 1600 users

March 2023:
2248 users

2FA Support Cases

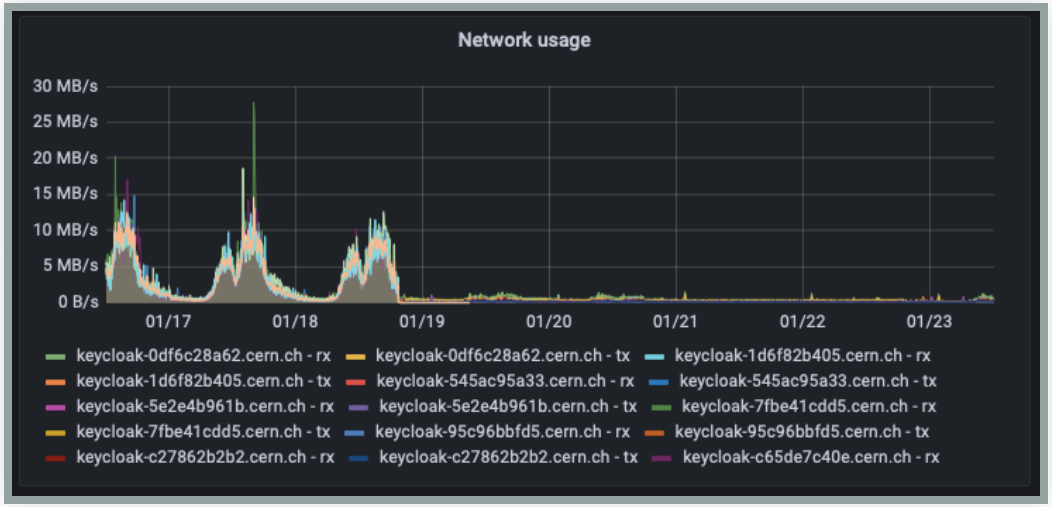
- TOTP cases where phone clock out of sync and codes invalid.
- Some clash between Yubikeys and native Windows/Mac WebAuthn fingerprint protocol.
- Plenty of lost tokens. Procedure established where the security team can verify an identity over Zoom before resetting the token.

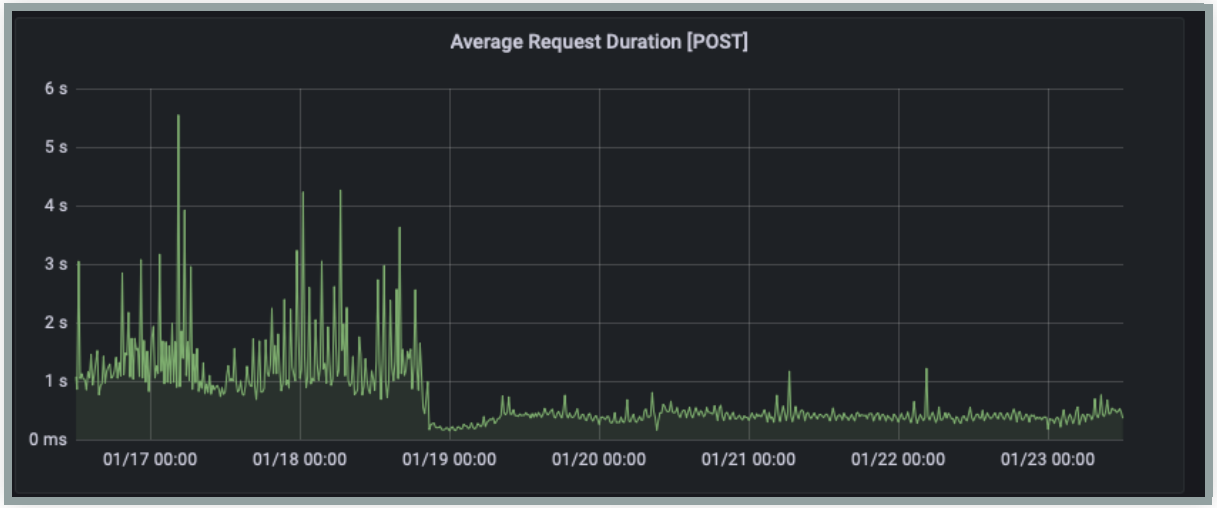


Other challenges

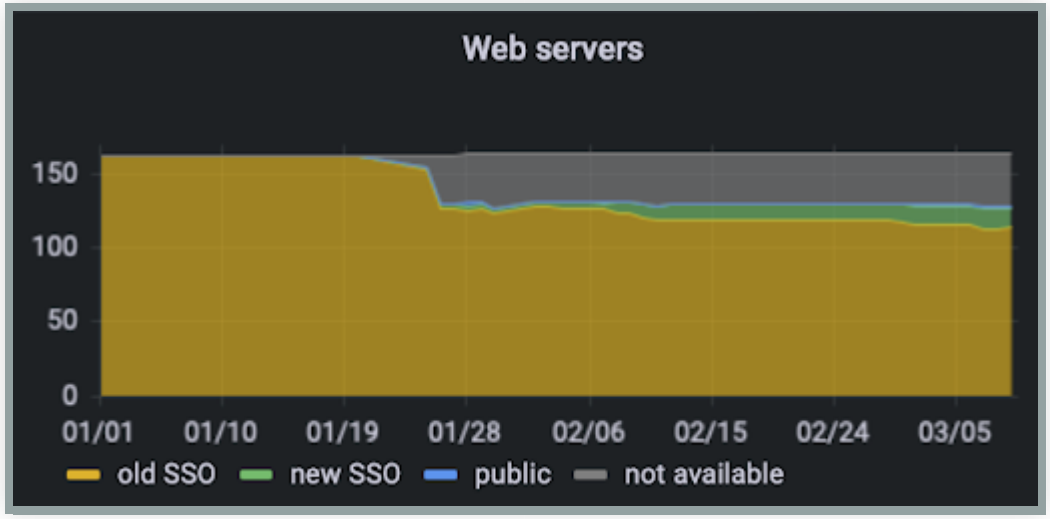
- Growing number user sessions
  - Mitigated with built-in user session limit
- High (and growing) number of clients (applications)
  - Number of clients seems usually lower in the Keycloak community
  - Performance issues in the past, it is getting better

Impact of Keycloak 19 upgrade





Old to new SSO migration status



Future roadmap

- Decomissioning old SSO by H2 2023
- Moving Keycloak to Kubernetes by H2 2023
- Remove the second realm for 2FA
- More details on our documentation page