Federated access for SSH with OpenID Connect

Diana Gudu, Marcus Hardt (KIT)
Damian Kaliszan, Paweł Wolniewicz (PSNC)

October 2021

Motivation

- Federated access to shell-based services

Motivation

Federated access to shell-based services

- Most prominent example: **SSH**
  - HPC
  - Cloud (IaaS)

Context

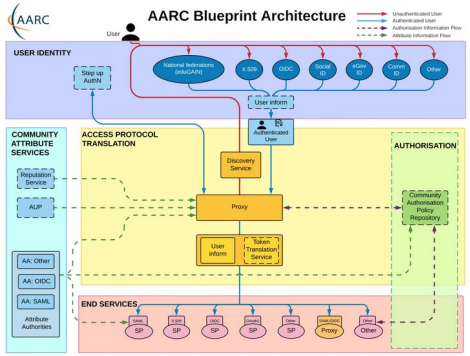
- European and national initiatives

Context

- European and national initiatives
  - Federated identities via eduGAIN

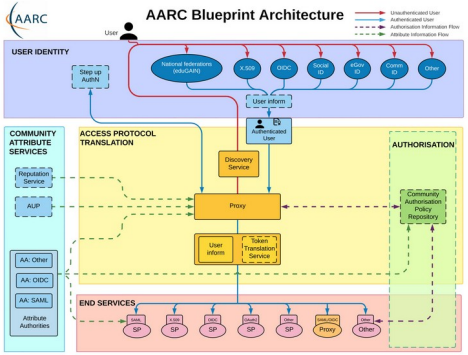
Context

- European and national initiatives
  - Federated identities via eduGAIN
  - Proxy-based AAI cf. **AARC Blueprint Architecture**
    - Unity
    - EGI Check-in



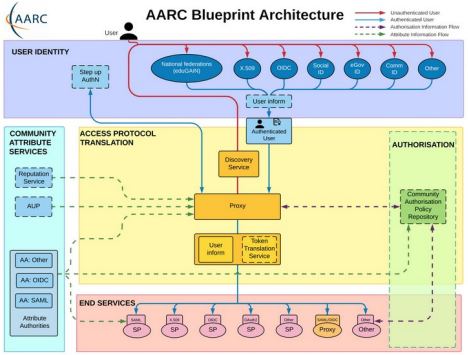
Context

- European and national initiatives
  - Federated identities via eduGAIN
  - Proxy-based AAI cf. **AARC Blueprint Architecture**
    - Unity
    - EGI Check-in
  - VO-based **authorisation**
    - AARC-G002 entitlement schema
    - Groups and roles inside VOs



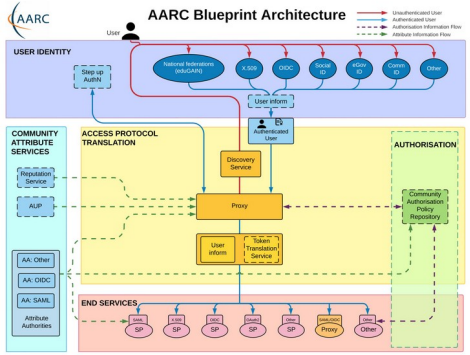
Context

- European and national initiatives
  - Federated identities via eduGAIN
  - Proxy-based AAI cf. **AARC Blueprint Architecture**
    - Unity
    - EGI Check-in
  - VO-based **authorisation**
    - AARC-G002 entitlement schema
    - Groups and roles inside VOs
  - REFEDS Assurance Framework



Context

- European and national initiatives
  - Federated identities via eduGAIN
  - Proxy-based AAI cf. **AARC Blueprint Architecture**
    - Unity
    - EGI Check-in
  - VO-based **authorisation**
    - AARC-G002 entitlement schema
    - Groups and roles inside VOs
  - REFEDS Assurance Framework
  - OpenID Connect (**OIDC**) & SAML



Context

- HPC

Context

- HPC
  - Access via SSH
    - password, SSH keys, …

Context

- HPC
  - Access via SSH
    - password, SSH keys, …
  - Accounts managed locally
    - Local UNIX users, LDAP, …

Context

- HPC
  - Access via SSH
    - password, SSH keys, …
  - Accounts managed locally
    - Local UNIX users, LDAP, …
  - Authorisation
    - Project-based
    - Requires approval process

Challenges

Integration of HPC service in federated context
requires local identities

Challenges

Integration of HPC service in federated context **requires local identities**

- mapping federated to local **identities**
- mapping federated to local **authorisation models**
- automated **provisioning & deprovisioning** processes for local identities

Goal

Enable SSH access using federated identity (OIDC)

- without recompiling OpenSSH
- with a clear authorisation concept

Our approach

Developed client & server-side tools that work with **standard** SSH software to enable:

- Transparent and smooth user experience
- Streamlined and extensible server-side processes for
  - AuthN, AuthZ & account management

Client-side

- **No need** for:
  - Prior registration
  - SSH keys or service password
  - Knowing your local username
  - Changing your current SSH workflows

Client-side

- **No need** for:
  - Prior registration
  - SSH keys or service password
  - Knowing your local username
  - Changing your current SSH workflows
- **Required**:
  - obtain an access token from the federated AAI → `oidc-agent`[1]
  - our SSH client wrapper → `mccli`

```
$ mccli ssh hostname
```

[1] https://github.com/indigo-dc/oidc-agent

Client-side

- **No need** for:
  - Prior registration
  - SSH keys or service password
  - Knowing your local username
  - Changing your current SSH workflows
- **Required**:
  - obtain an access token from the federated AAI → `oidc-agent`[1]
  - our SSH client wrapper → `mccli`

```
$ mccli ssh hostname
```

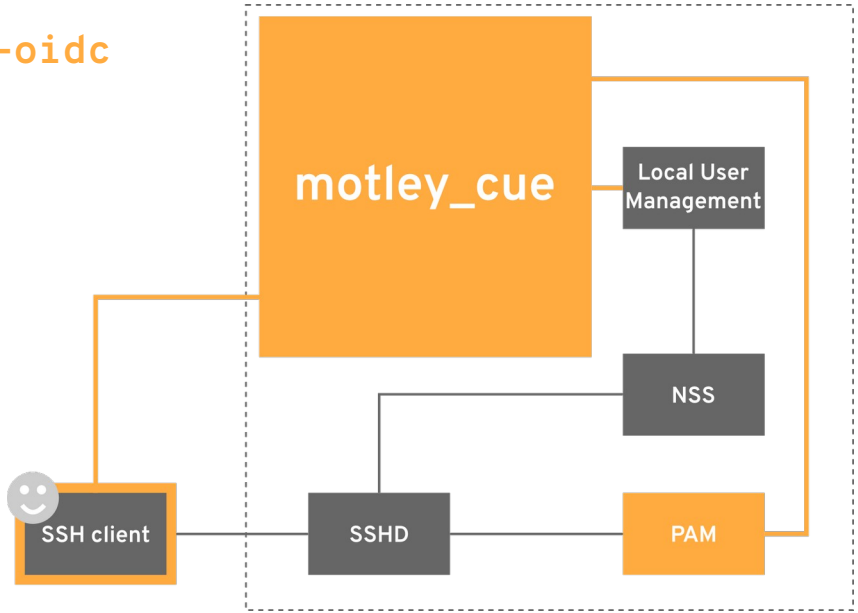
- Currently integrating additional SSH clients (incl. **PuTTY** on Windows[2])

[1] https://github.com/indigo-dc/oidc-agent

[2] See talk today by Dmytro Dehtyarov: OIDC Support for Windows using Putty

Server-side

- New service: `motley_cue`
  - highly configurable and extensible
- New PAM for SSH authentication: `pam-oidc`
  - accepts Access Tokens

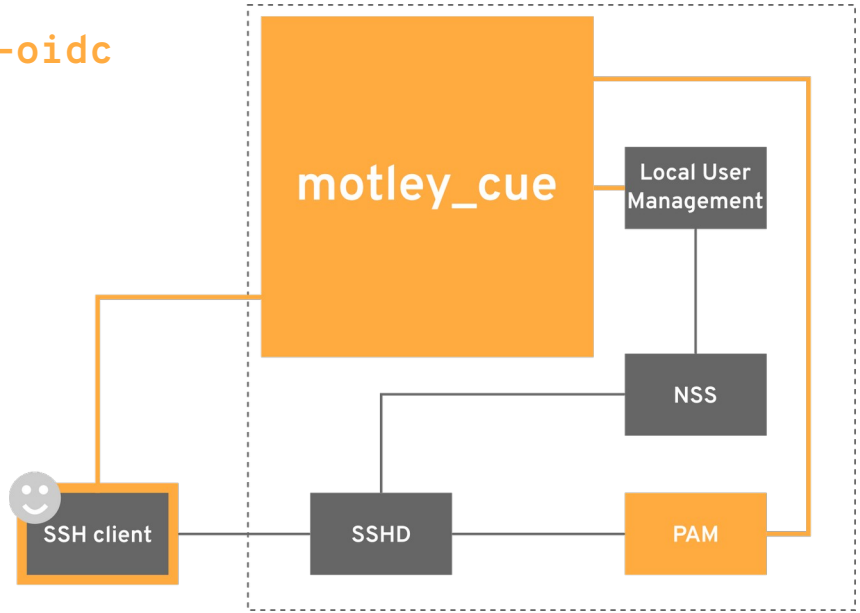


Server-side

- New service: `motley_cue`
  - highly configurable and extensible
- New PAM for SSH authentication: `pam-oidc`
  - accepts Access Tokens

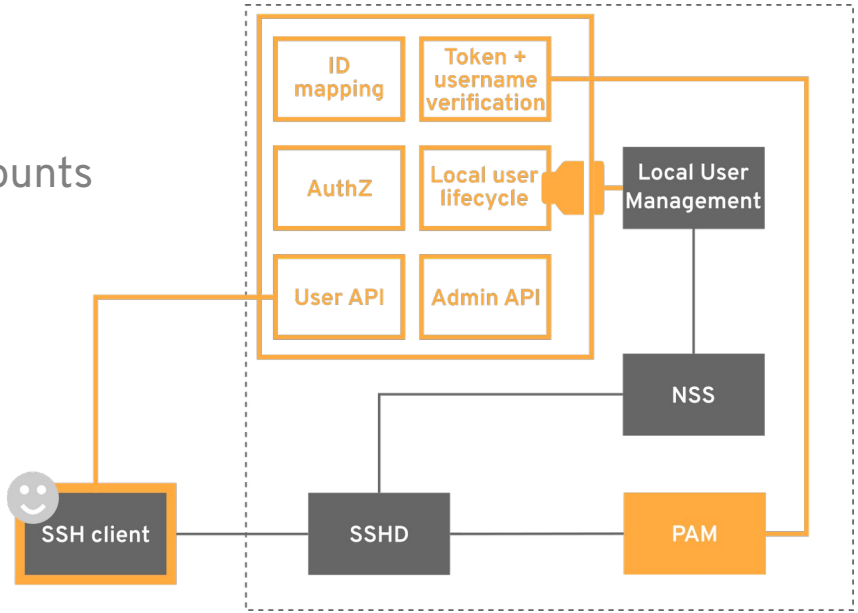
Unmodified SSH server

Integrate local site policies



Server-side

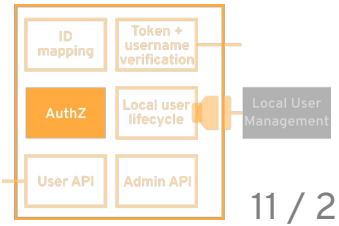
- New service: `motley_cue`
  - highly configurable and extensible
- What it does
  - handles the ID mapping
  - manages the lifecycle of local accounts
  - provides a slim REST interface
  - handles authorisation



motley_cue: authorisation

- support for multiple OPs
- based on VO membership
- individual users via `sub+iss`

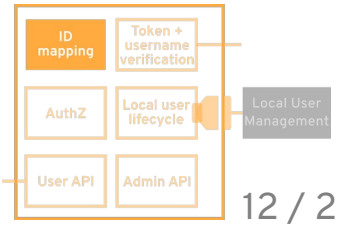
```
####################
[authorisation.egi]
####################
op_url = https://aai.egi.eu/oidc

## list of VOs whose users are authorised to use the service
authorised_vos = [
        "urn:mace:egi.eu:group:eosc-synergy.eu"
    ]

## the OIDC claim containing the VOs specified above
vo_claim = eduperson_entitlement

## Individual users are authorised by 'sub' claim.
## These users don't have to be members of the authorised VOs.
authorised_users = [
        "c2370093c19496aeb46103cce3ccdc7b18...@egi.eu"
    ]
```



motley_cue: mapping

- identity mapping

```
sub+iss → local account
```



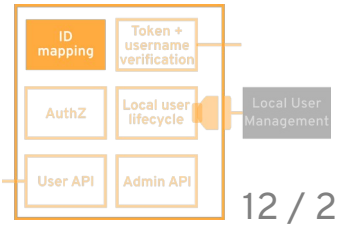
motley_cue: mapping

- identity mapping

  `sub+iss → local account`

  - stored at Local User Management

local UNIX → **gecos** field in `/etc/passwd`
LDAP → custom **attribute**



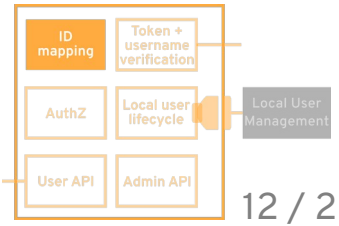
motley_cue: mapping

- identity mapping

  `sub+iss → local account`

  - stored at Local User Management
  - username generation strategies:
    - **friendly**: `preferred_username, first_last, …`
    - **pooled**: `egi001, egi002, …`

local UNIX → **gecos** field in `/etc/passwd`
LDAP → custom **attribute**



motley_cue: mapping

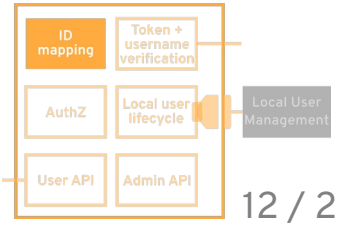
- identity mapping

  `sub+iss → local account`

  - stored at Local User Management
  - username generation strategies:
    - **friendly**: `preferred_username, first_last, …`
    - **pooled**: `egi001, egi002, …`

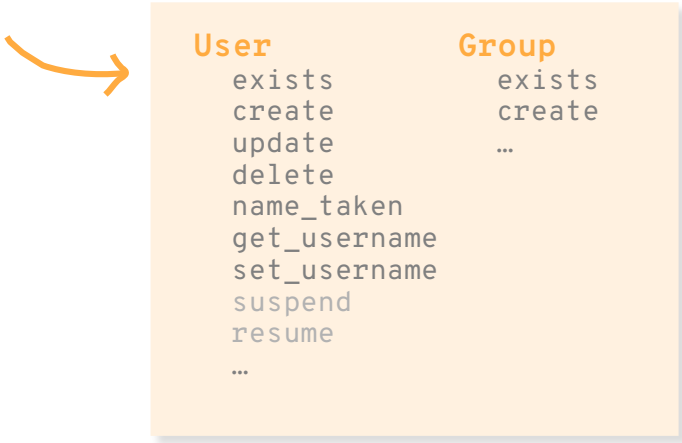
  local UNIX → **gecos** field in `/etc/passwd`
  LDAP → custom **attribute**

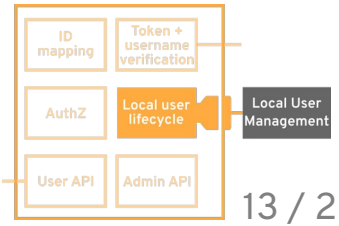
- VOs mapped to local groups



motley_cue: lifecycle of local accounts

- interface to site-local user management system
- generic & extensible
  - supported backends: local UNIX, bwIdM
  - additional backends can be developed



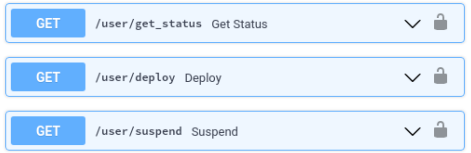


https://git.scc.kit.edu/feudal/feudalAdapterLdf

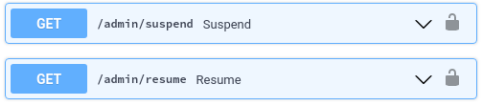
motley_cue: REST interface

- supplies the mapping information to user & service
- web-based
- authentication: OIDC Access Token as bearer

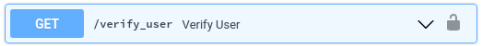
- user API

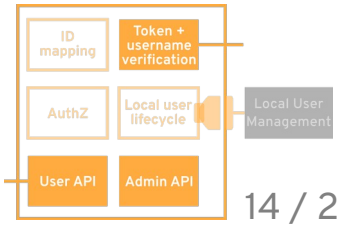


- admin API



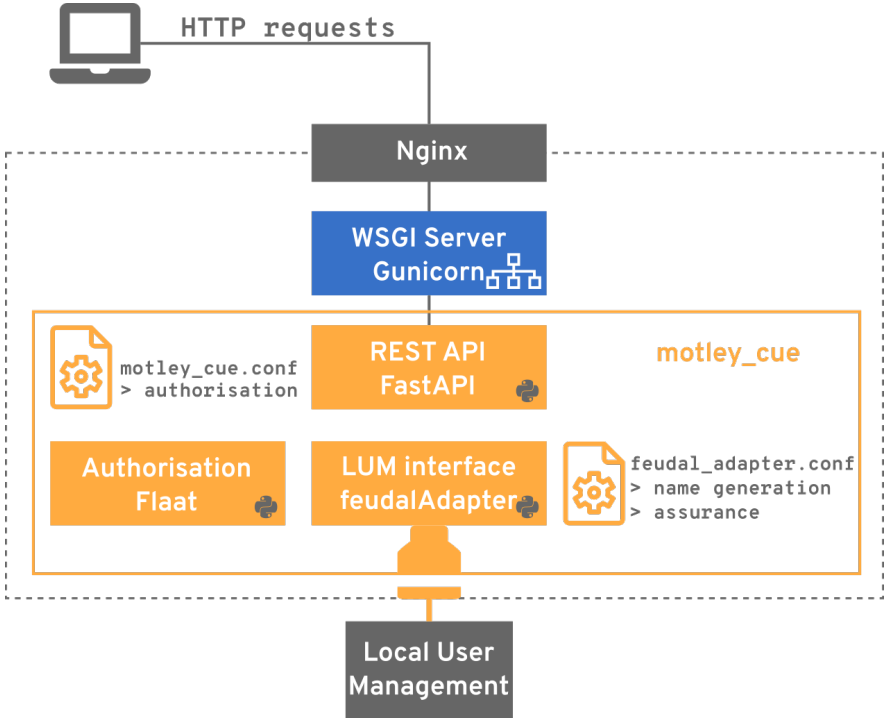
- service API





motley_cue: tech stack

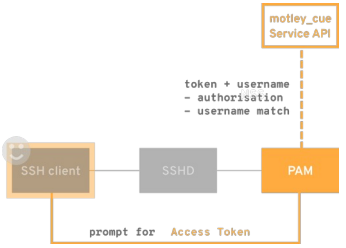
- Python-based
- FastAPI
  - REST APIs
  - fast
  - async support
- Gunicorn
  - WSGI server
  - scalability
- Nginx
  - reverse proxy
  - SSL handling



PAM-OIDC

- Based on OIDC access token authentication
  - user is prompted for an **Access Token** instead of Password
- Written in **C**
- Query **motley_cue** service API for:
  - token validation
  - authorisation
  - username match

```
$ curl -X 'GET' \
    $motley_cue_endpoint/verify_user&username=$username \
    -H “Authorization: Bearer $token”

{
  "state": "deployed",
  "verified": true
}
```



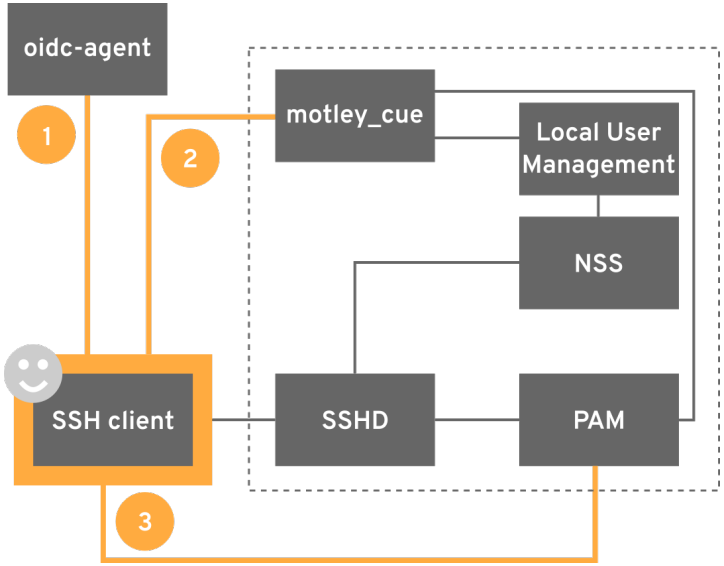
https://git.man.poznan.pl/stash/scm/pracelab/pam.git

User workflow

```
$ mccli ssh hostname
```

In the background:

1. Retrieve OIDC Access Token
2. Get local username and deploy local account if not deployed
3. Input Access Token when prompted



Demo

ssh-oidc-demo.data.kit.edu

Future work

- Support **other backends** for local user management
  - LDAP
- Support **approval** workflow for **account provisioning**
  - Local account in “pending” state awaiting approval
- Collaboration with initial HPC centres: PSNC, CESGA
- Local account **deprovisioning**
- **Windows**[1] support
- **mytoken**[2] integration
- Support **other services** besides SSH

[1] https://indico.egi.eu/event/5464/contributions/15659/
[2] https://indico.egi.eu/event/5464/contributions/15657/

Summary

- SSH access with federated identity
  - local account management fully transparent to user
  - on-the-fly account creation
- Full control for local admins
  - users can be filtered by VO, entitlements, assurance, …
  - can extend backends to integrate local processes

- Packages available for several distributions

Links

- Demo instance

https://ssh-oidc-demo.data.kit.edu/

- Documentation

https://github.com/EOSC-synergy/ssh-oidc

- Contact

m-contact@lists.kit.edu

Questions?


KIT
Karlsruher Institut für Technologie
PSNC
EOSC SYNERGY
HIFIS