Trust & Identity Incubator
Provisioning activity

Uros Stevanovic

Sprint Demo, 2020-05-26

Public

www.geant.org

Requirements for non-web services

- Accessing service requires (at the very least):
  - User creation (and removal)
  - Credential deployment
- User “creation” → Account provisioning
  - Creating/assigning a user account on the Service side
  - E.g. name, group, home folder
- Deploying a credential for a user
  - SSH
  - Password
  - Tokens

Overview

- Provision users on federated downstream services (e.g. behind a proxy)
  - Create accounts
  - Provide necessary information to services
  - Deprovision users

- Two (at least) solutions:
  - FEUDAL
  - PERUN

Goals

- Provide a “robust” solution
- Test services, “real” use cases
- Deployment and usage scenarios
  - Provision local users
  - Provision users for local applications
  - Cloud applications
- Deprovisioning of the users

Achievements

- API is implemented
- JSON based, user_info based
- API:
  - Get all users (also per VO)
  - Update user info
  - Delete a user
  - Check if user exists

```
{
   "userinfo": {
      "iss": "https://proxy.acc.eduteams.org",
      "sub": “<sub>@eduteams.org",
      "name": "Uros Stevanovic",
      "given_name": "Uros",
      "family_name": "Stevanovic",
      "email": "uros.stevanovic@kit.edu",
      "ssh_key": “<some_key>",
      "eduperson_entitlement": [ “<group1>",
                                 “<group2>"
      ],
      "eduperson_targeted_id": [ “<some string>@eduteams.org" ],
      "eduperson_principal_name": [ "urost@acc.eduteams.org" ],
      "eduperson_scoped_affiliation": [ "member@acc.eduteams.org" ]
   }

}
```

API

PATH	METHOD	DESCRIPTION
at/	PUT	Update a user using an access token. The access token is used to retrieve an up-to-date userinfo.
userinfo/	PUT	Update a user using a plain userinfo.
users/ users/?vo=<vo>	GET	Retrieve the subjects of the registered users. Can be filtered by vo.
user/<sub>/	GET DELETE	Check if the user with sub <sub> is registered. Delete the user with sub <sub> from feudal.

Status and next steps

✓ Done:

- Connected FEUDAL to eduTEAMS Acceptance (both prod and dev)
- API implemented
- “Drop” use cases (GSuites will be worked on elsewhere)

▢ Next:

- PERUN updating user info via FEUDAL – will be done
- Provide “classic” LDAP use case for FEUDAL (PERUN already supports it) – work started
- Provide further use cases (create a list of users per VO)

⛔ Blocker:

- None

GÉANT
Networks • Services • People

Thank you

Any questions?

www.geant.org

The research leading to these results has received funding from
the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 856726 (GN4-3).