Trust & Identity Incubator
(De)Provisioning users activity

Uros Stevanovic

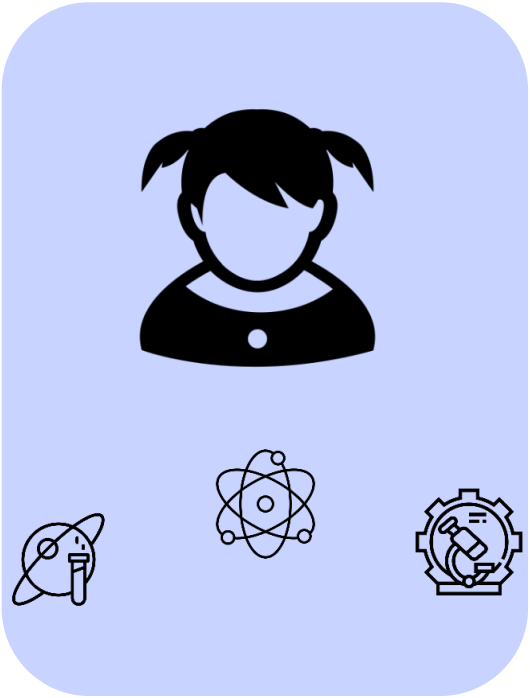
Sprint Demo, 2020-07-02

Public

www.geant.org

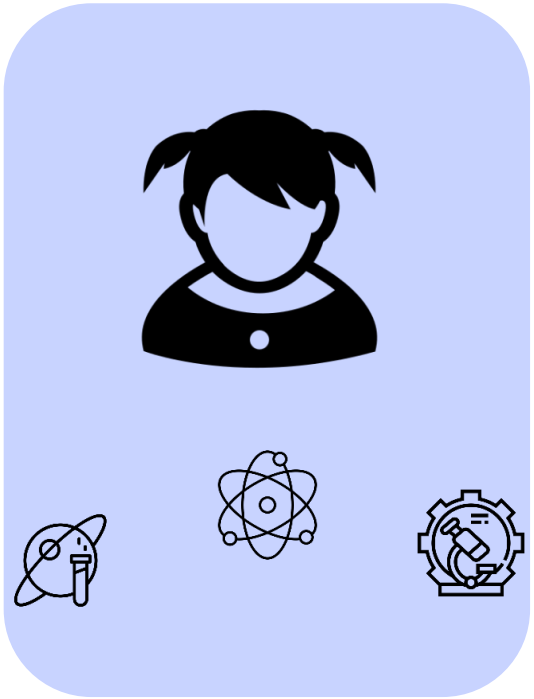
Remote access

Remote access

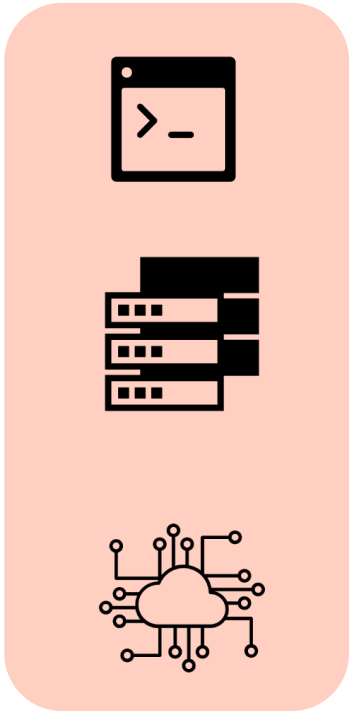


"*user*"

Remote access

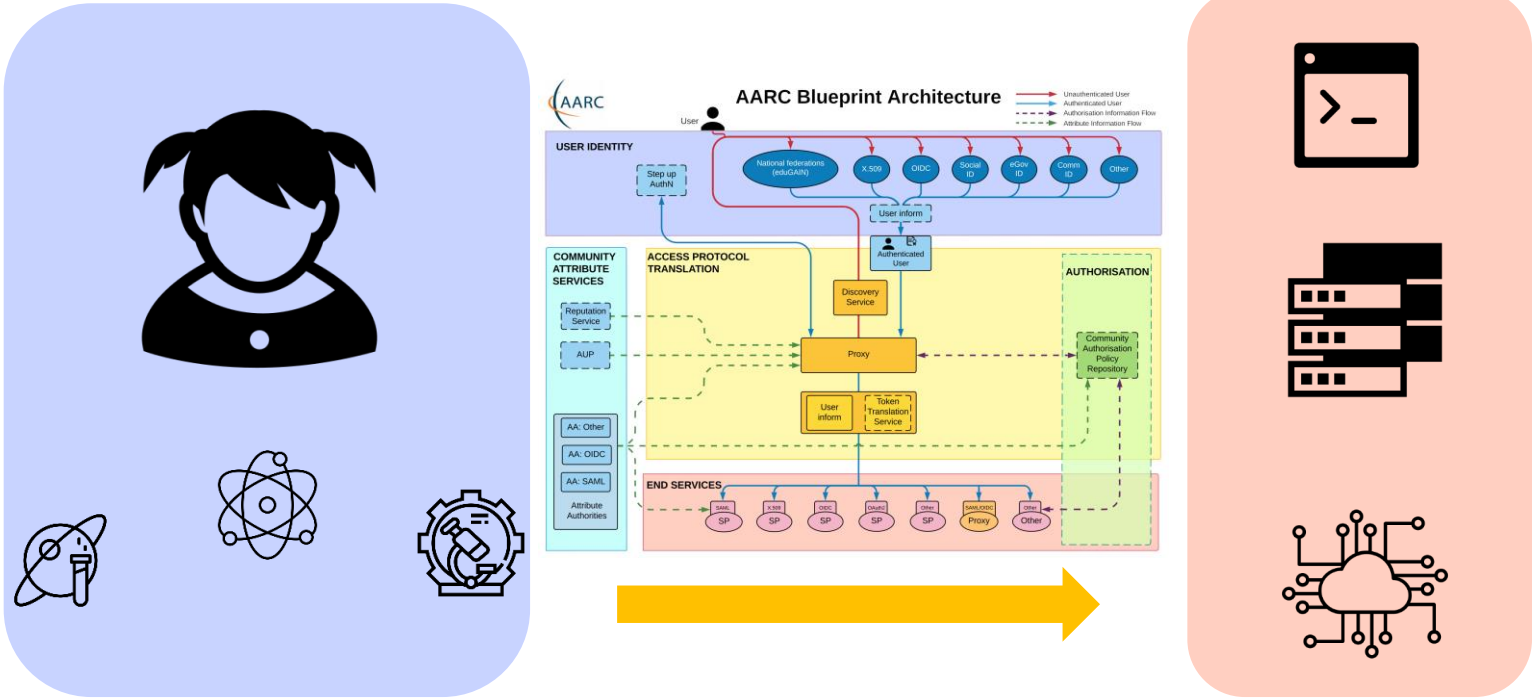


"*user*"



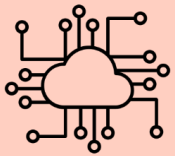
"*resources*"

Remote access



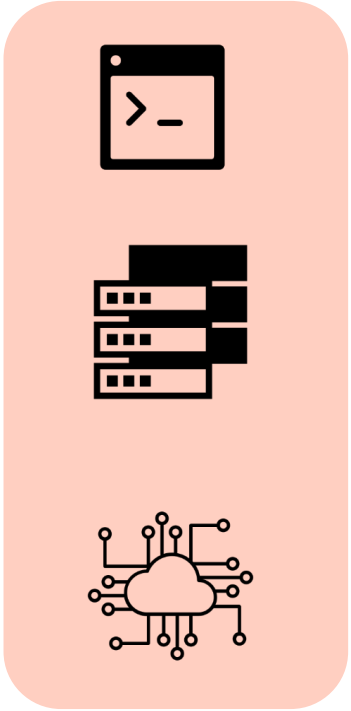
"*user*" "*AAI*" "*resources*"

Web vs non-web services



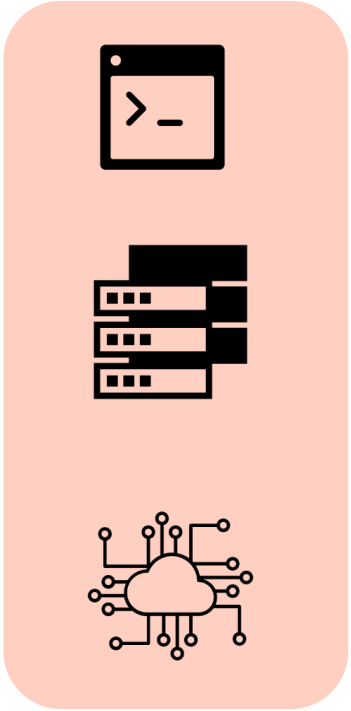
- Using web services typically involves “in-time” release of attributes
  - Up-to-date info (e.g. behind a proxy/IdP)
  - Flows are well defined
  - Numerous activities are solving this problem
- Non-web services require “specialized” flow
  - Additional credentials (e.g. SSH keys)
  - Once created, users “bypassing” IdPs (or proxies)

Users using *resources*



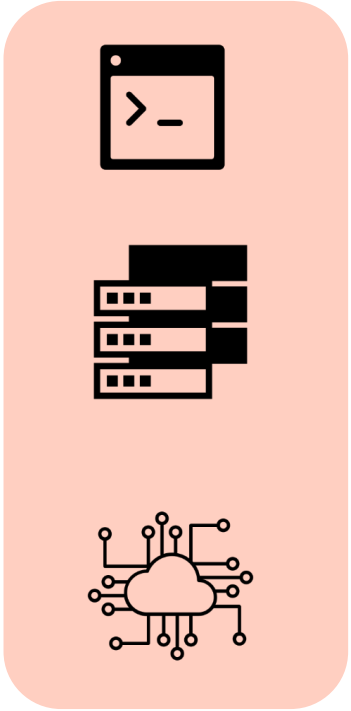
- Provision (and deprovision) users:
  - Create accounts
  - Provide necessary info to services
  - Provide access credentials (e.g. SSH keys)

Users using *resources*



- Provision (and deprovision) users:
  - Create accounts
  - Provide necessary info to services
  - Provide access credentials (e.g. SSH keys)
- Information up-to-date?

Users using *resources*



- Provision (and deprovision) users:
  - Create accounts
  - Provide necessary info to services
  - Provide access credentials (e.g. SSH keys)
- Information up-to-date?
- Deprovision users?

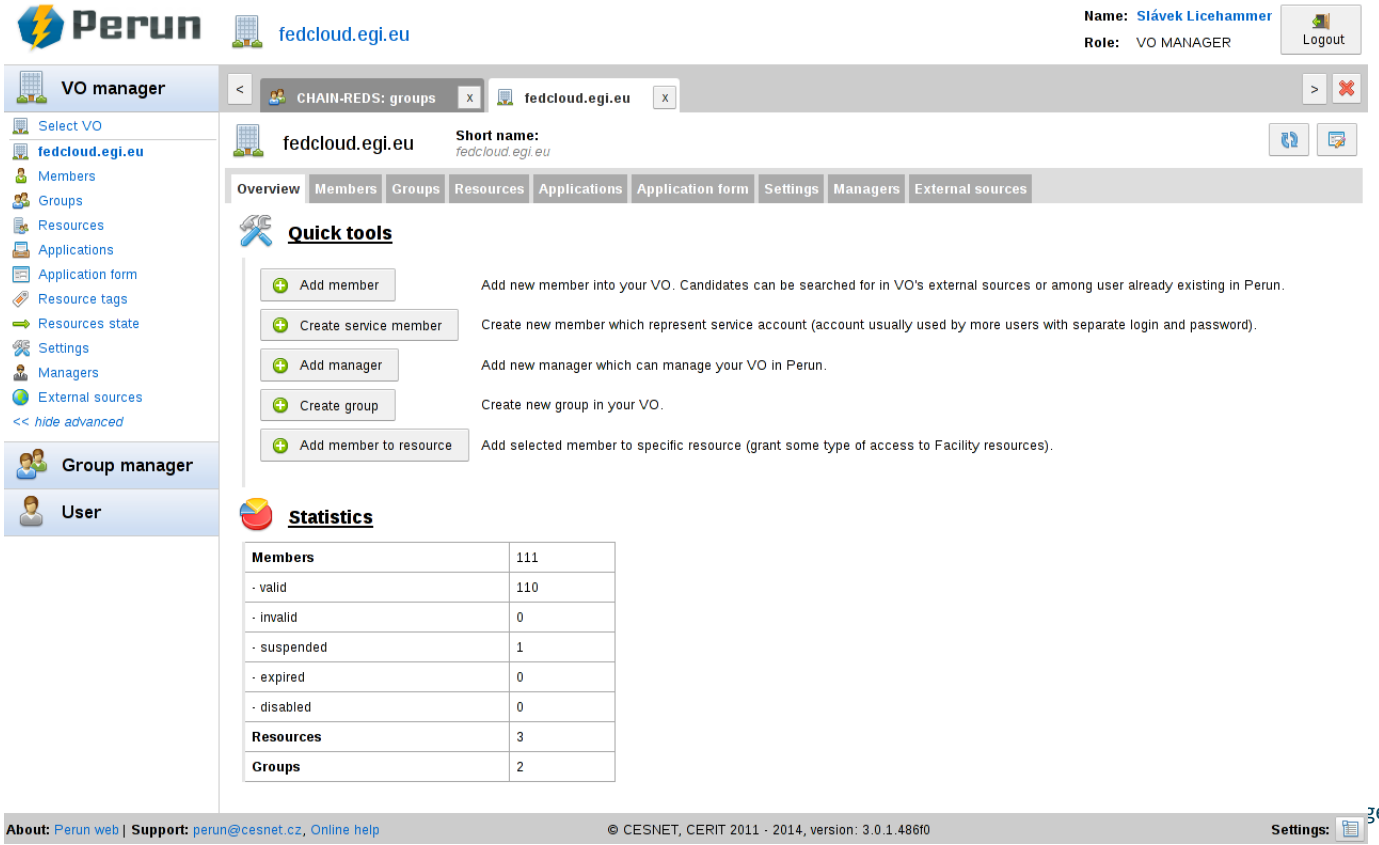
(De)Provision users activity

- Two (at least) solutions:
  - PERUN
  - FEUDAL
- Decentralized vs Centralized
- Asynchronous vs sequential
- General vs per-service data format

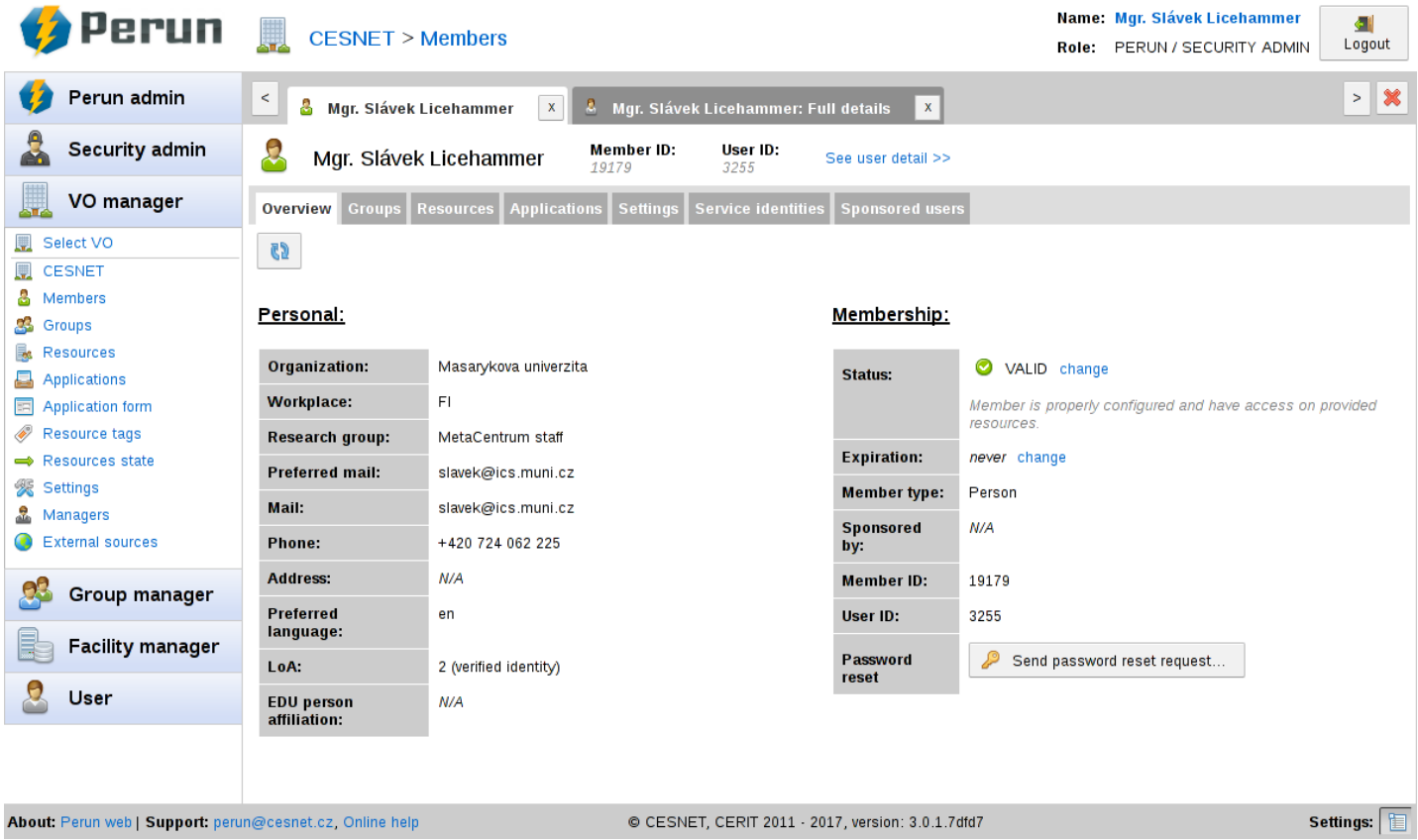
PERUN

- Membership Management Service
  - VO and group management
  - User management
  - Service management
  - Customizable
- Synchronization to/from external source (e.g. LDAP)
- REST API
- Developed and maintained by CESNET
- One use case: MMS for a proxy
- Typically “hidden” from a “regular” user

PERUN


Perun
fedcloud.egi.eu
Name: Slávek Licehammer
Role: VO MANAGER
Logout
VO manager
Select VO
fedcloud.egi.eu
Members
Groups
Resources
Applications
Application form
Resource tags
Resources state
Settings
Managers
External sources
<< hide advanced
Group manager
User
CHAIN-REDS: groups
fedcloud.egi.eu
Short name:
fedcloud.egi.eu
Overview
Members
Groups
Resources
Applications
Application form
Settings
Managers
External sources
Quick tools
Add member
Add new member into your VO. Candidates can be searched for in VO's external sources or among user already existing in Perun.
Create service member
Create new member which represent service account (account usually used by more users with separate login and password).
Add manager
Add new manager which can manage your VO in Perun.
Create group
Create new group in your VO.
Add member to resource
Add selected member to specific resource (grant some type of access to Facility resources).
Statistics
Members 111
- valid 110
- invalid 0
- suspended 1
- expired 0
- disabled 0
Resources 3
Groups 2
About: Perun web | Support: perun@cesnet.cz, Online help
© CESNET, CERIT 2011 - 2014, version: 3.0.1.486f0
Settings:

PERUN



nt.org

GÉANT

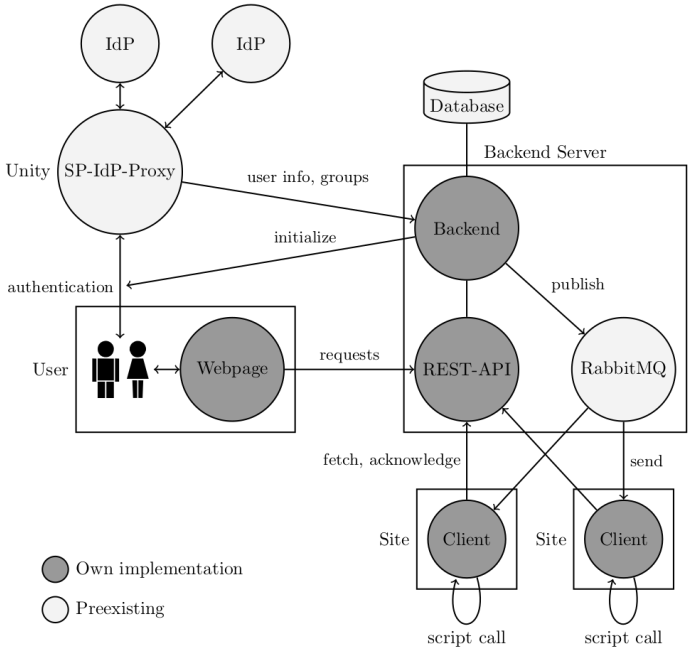
PERUN User Provisioning

- Flow is not “user-centric”, i.e. deployment is typically not decided by the user
- Centralized model
  - Master + slave model
  - More tightly integrated (akin to “business environment”)
  - Trust level required between sites and PERUN is high
  - Customized communication (format per service)
- Synchronous deployment
  - Service needs to be online
  - Typically SSH connection to services

FEUDAL

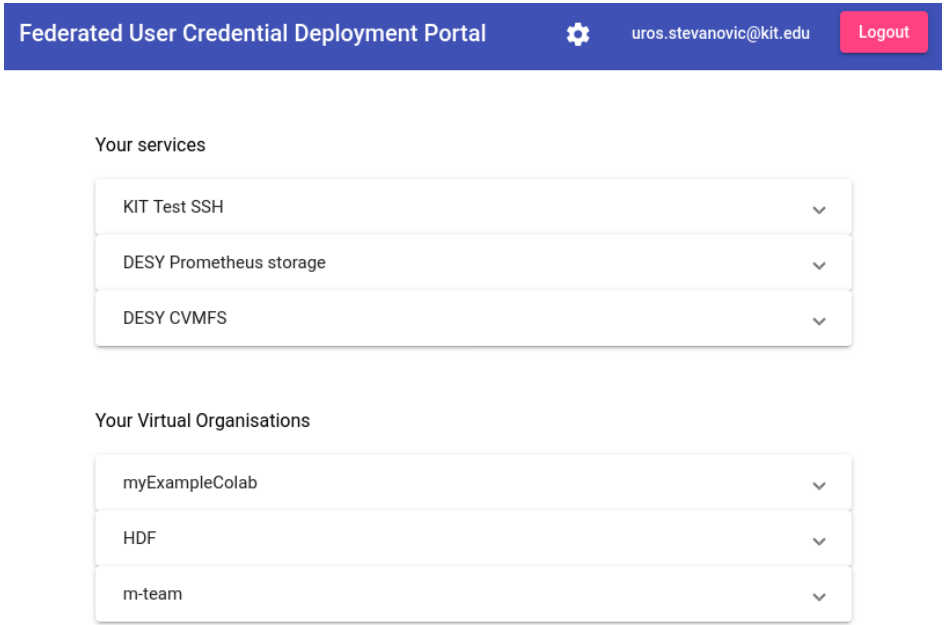
- Federated User Credential Deployment Portal
  - Web application (OIDC client)
  - Provision/deprovision users
  - Deploy credentials (e.g. SSH)
  - AuthZ discrimination
- Architecture
  - Web portal (UI)
  - Backend+database (user info and credentials)
  - Clients (deployed service side) + “adapters”
- Pub-Sub

FEUDAL



- Web portal (user interaction point)
- REST API
- Backend
  - Django
- RabbitMQ (Pub-Sub)
- Clients (Go, Python, etc)
- Scripts
  - Adapters
- JSON data format:
  - Status
  - User info
  - credentials

FEUDAL



FEUDAL Privacy Policy

FEUDAL

- OIDC client
- User-centric flow
  - Typically user is in control
  - Deployment per service, per VO
- Decentralized model
  - Server + client model
  - Clients runs at sites (admin control), trust level not necessarily very high
  - Client only receives the info (user_info, JSON)
  - Standardized communication
- Asynchronous communication
  - Pub-sub, outgoing connection at clients
  - Flexible messaging (resending upon failure, onboarding, etc)

Side-by-side comparison

FEUDAL

- OIDC client
- User-centric flow
  - Typically user is in control
  - Deployment per service, per VO
- Decentralized model
  - Server + client model
  - Clients runs at sites (admin control), trust level not necessarily very high
  - Client only receives the info (user_info, JSON)
  - Standardized communication
- Asynchronous communication
  - Pub-sub, outgoing connection at clients
  - Flexible messaging (resending upon failure, onboarding, etc)

PERUN

- MMS (Membership management service)
  - Flow is not very “user-centric”, i.e. deployment is typically not decided by the user
- Centralized model
  - Master + slave model
  - More tightly integrated (akin to “business environment”)
  - Trust level required between sites and PERUN is higher
  - Customized communication (format per service)
- Synchronous deployment
  - Service needs to be online
  - Typically SSH connection to services

Usage consideration

- FEUDAL and PERUN have complementing flows/use cases
- Tight integration, easy-to-understand deployment, easy VO deployment → PERUN
- Flexible model, user may decide, asynchronous decentralized communication → FEUDAL

Usage consideration

- FEUDAL and PERUN have complementing flows/use cases
- Tight integration, easy-to-understand deployment, easy VO deployment → PERUN
- Flexible model, user may decide, asynchronous decentralized communication → FEUDAL

How to proceed?

Use cases

- Not PERUN or FEUDAL, but both
- Centralized model needed/expected/reasonable → PERUN:
  - Cloud apps (e.g. GSuites)
  - Mail lists
  - LDAP (executed by PERUN)
  - Windows apps
- Decentralized model → FEUDAL:
  - Provision users for SSH access to VMs
  - LDAP (executed on site's side)
  - Mail lists (via LDAP)
  - Further plugins

Up-to-date info / Deprovision users

- PERUN + FEUDAL:
  - PERUN is an MMS (users' info is up-to-date)
  - FEUDAL is a "client" of PERUN (or other MMS)
- Centralized Model → PERUN directly executes action
- Decentralized Model → PERUN via FEUDAL updates info

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.

API

- JSON based API, userinfo
- 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" ]
   }

}
```

DEMO

- FEUDAL “look and feel” + SSH use case
- FEUDAL API (MMS + FEUDAL)
- FEUDAL update of user info
- FEUDAL LDAP use case

Achievements

- Provision users, via PERUN, FEUDAL, or PERUN+FEUDAL:
  - Cloud applications
  - LDAP (+ Mail lists)
  - Access to VMs (SSH)
  - Windows applications
- Centralized + Decentralized
- Up-to-date info
- Deprovision users

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).