Trust & Identity Incubator
(De)Provisioning Users

Uros Stevanovic

Sprint Demo, 2020-03-19

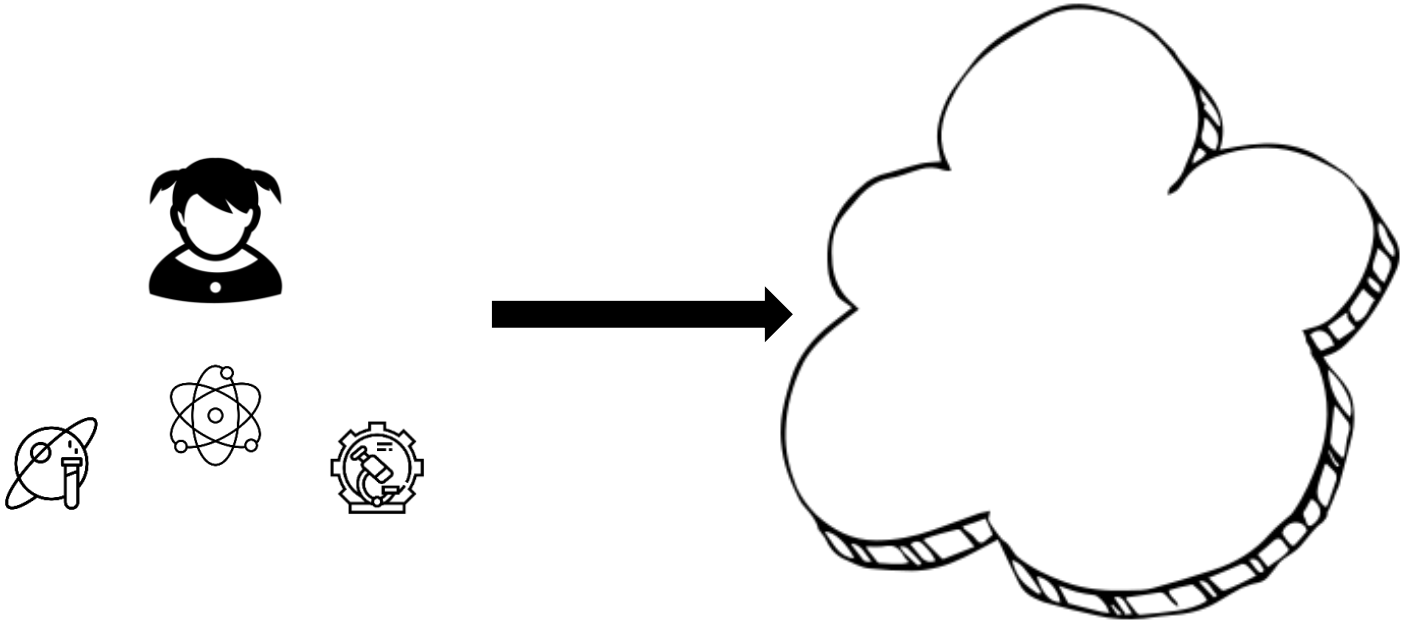
Public

www.geant.org

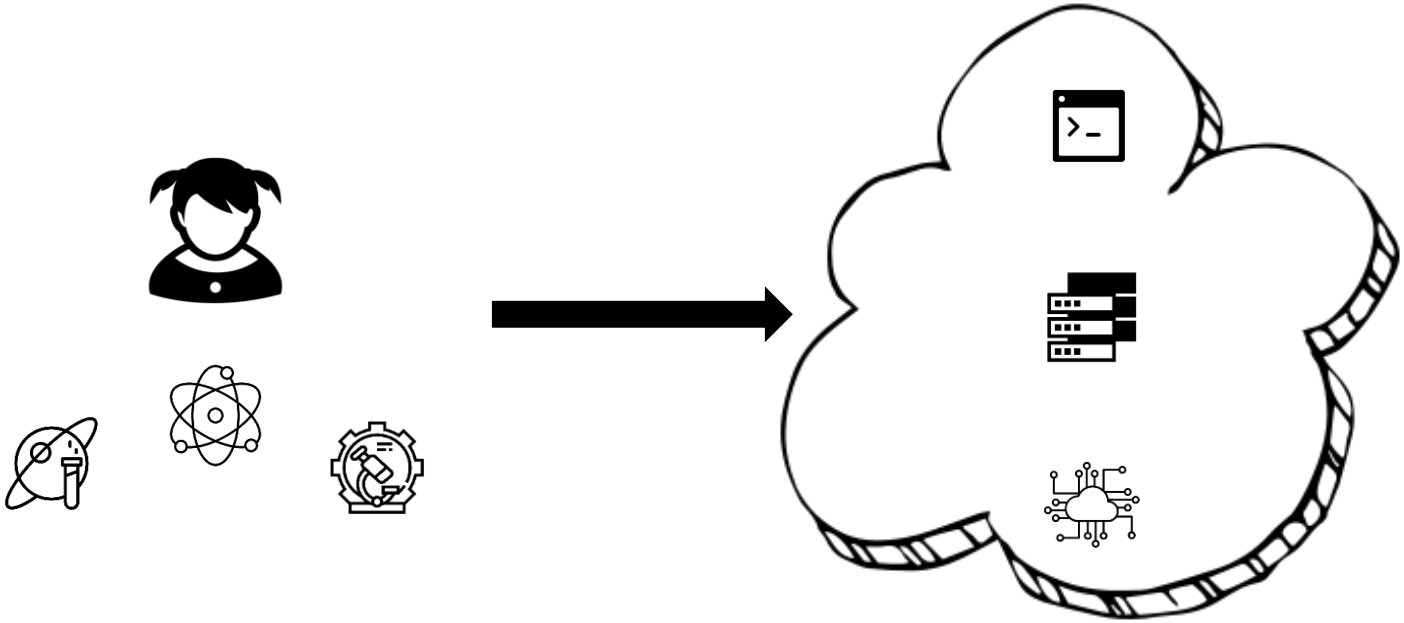
Instant user provisioning and deprovisioning

- (De)Provision users on remote systems
- Up-to-date user information
- Flexible approach
  - We’re “responsible” for the user information
  - Service is “responsible” for the user information
- Credential management
  - SSH

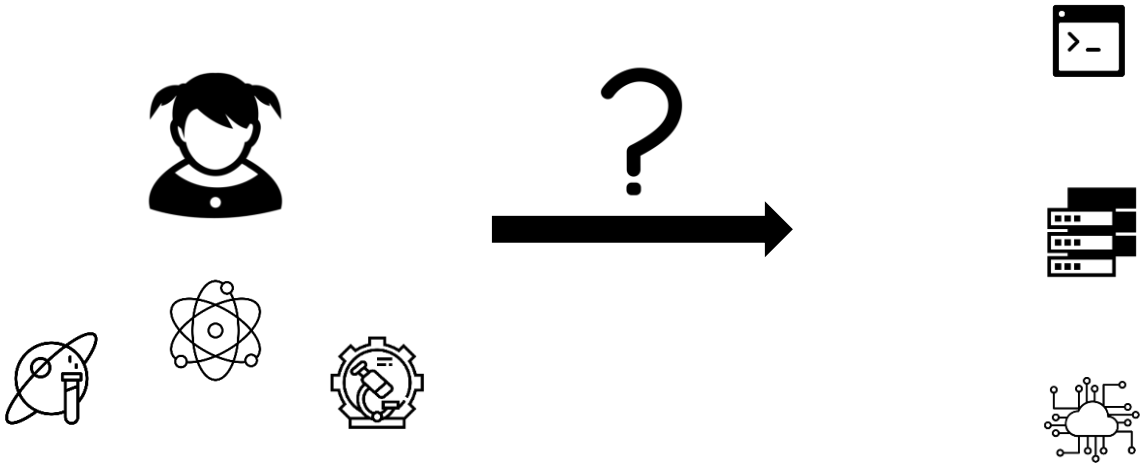
Overview



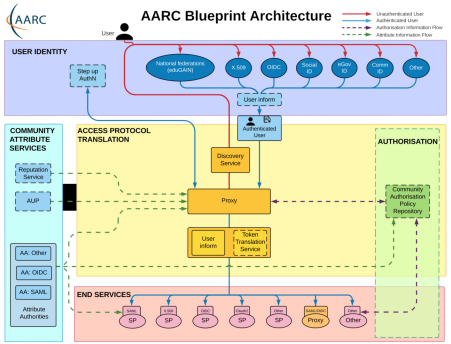
Overview



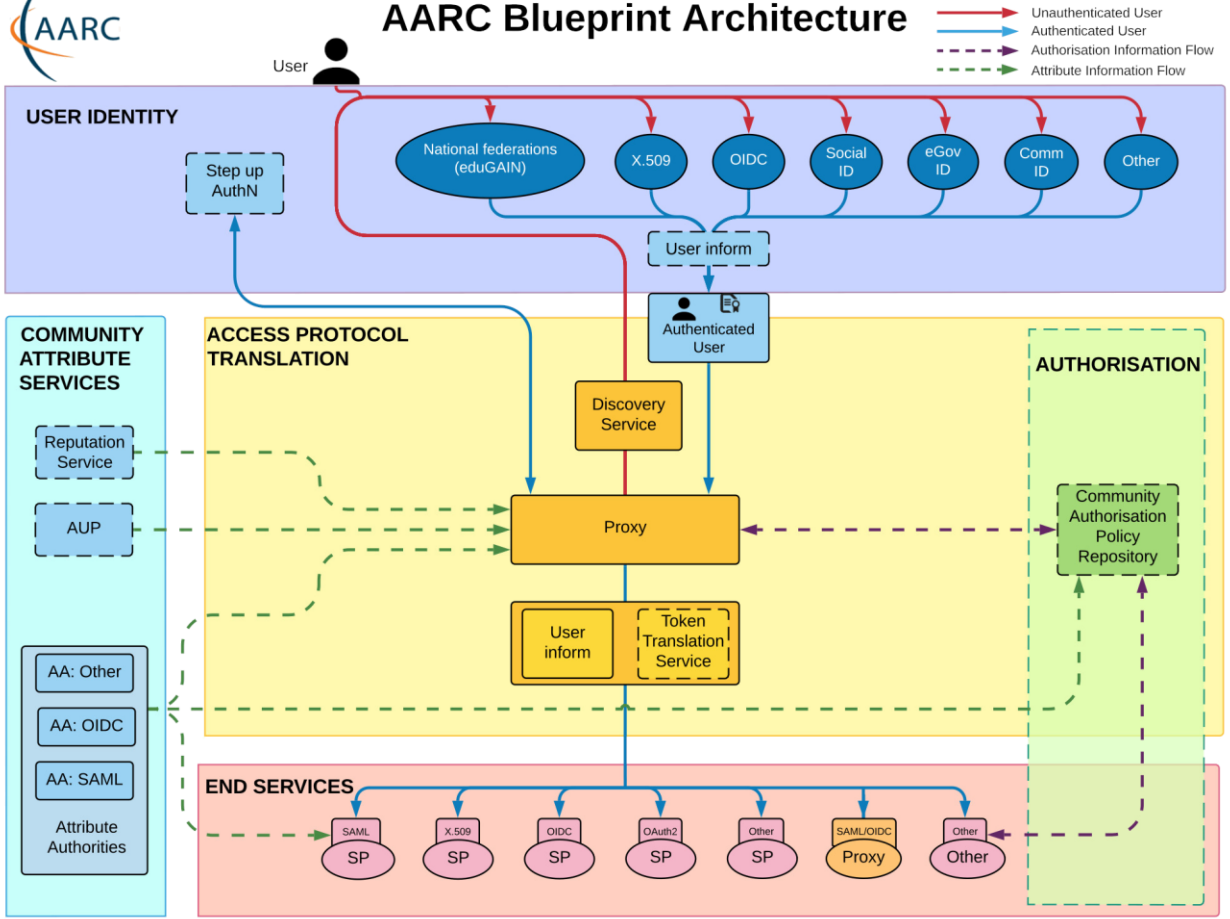
Overview

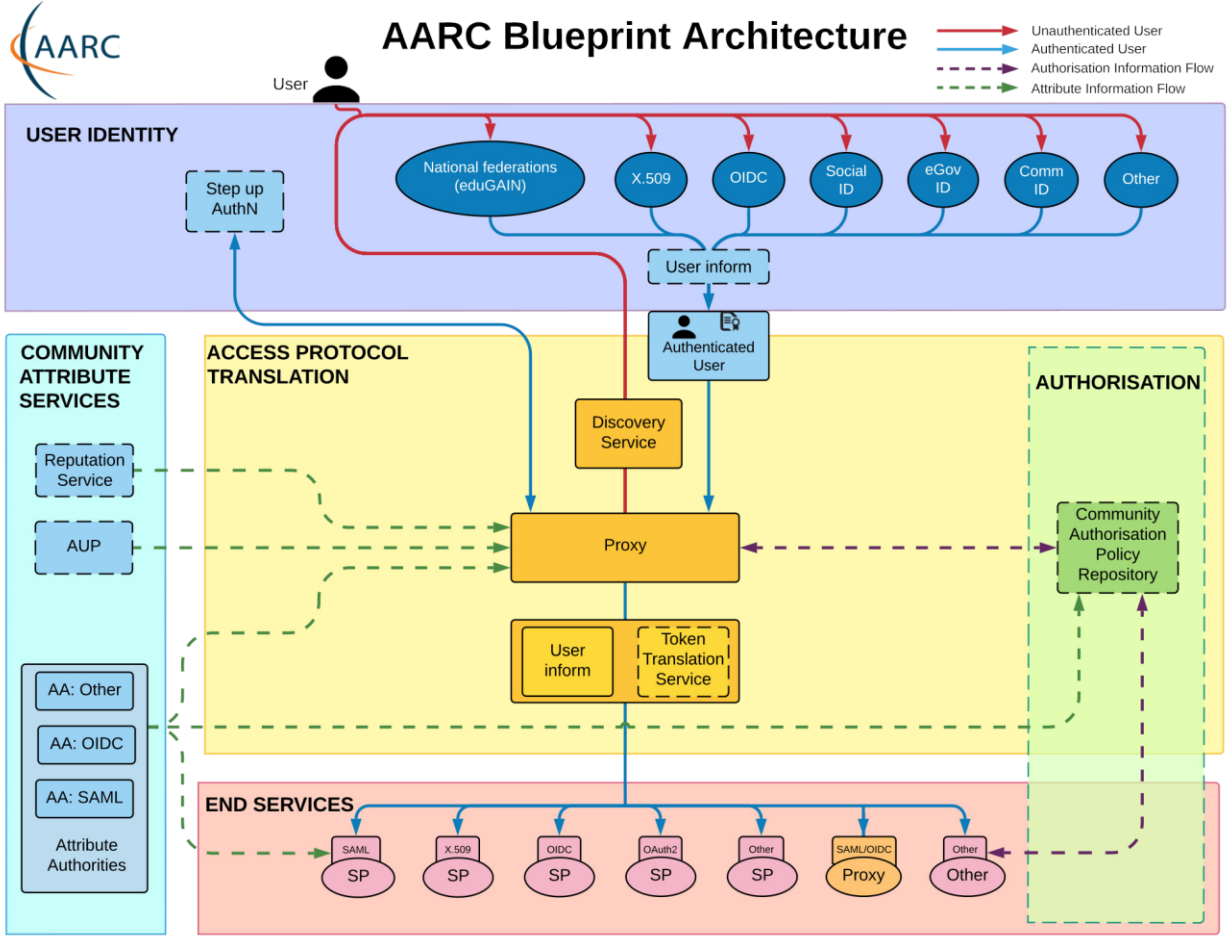


Overview

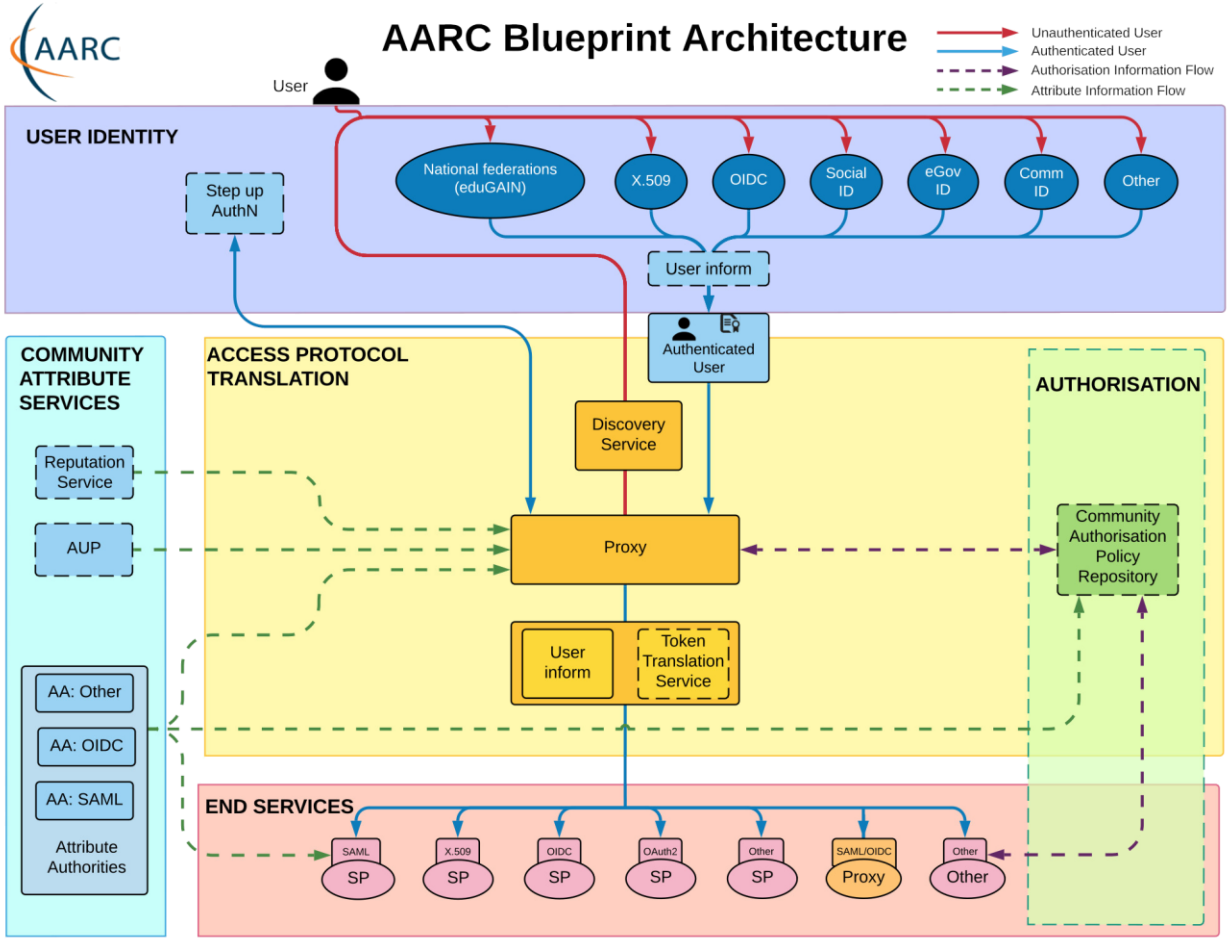

AARC
AARC Blueprint Architecture
User
Unauthenticated User
Authenticated User
Authorisation Information Flow
Attribute Information Flow
USER IDENTITY
Step up AuthN
National federations (eduGAIN)
X.509
OIDC
Social ID
ACeS ID
Comm ID
Other
User inform
Authenticated User
COMMUNITY ATTRIBUTE SERVICES
ACCESS PROTOCOL TRANSLATION
AUTHORISATION
Reputation Service
AUP
Discovery Service
Proxy
Community Authorisation Policy Repository
User inform
Token Translation Service
AA: Other
AA: OIDC
AA: SAML
Attribute Authorities
END SERVICES
SP
Proxy
DiRAP

AARC Blueprint Architecture





Web services ✓



Web services ✓

Non-web ?

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

Goals

- “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

Great! But how?

- Two (at least) solutions:
  - FEUDAL
  - PERUN
- FEUDAL
  - Decentralized model
- PERUN
  - Tightly integrated (centralized) model

Great! But how?

- Two (at least) solutions:
  - FEUDAL
  - PERUN
- FEUDAL
  - Decentralized model
- PERUN
  - Tightly integrated (centralized) model

DEMO FEUDAL

More details

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

Great again! So… why TII?

- Demonstrating the usefulness of the approach
- Potential additional use-cases
  - LDAP
  - SYMPA
  - Cloud apps
- But also…

Great again! So… why TII?

- Demonstrating the usefulness of the approach
- Potential additional use-cases
  - LDAP
  - SYMPA
  - Cloud apps
- But also…
  - Not FEUDAL or PERUN but FEUDAL + PERUN

Great again! So… why TII?

- Demonstrating the usefulness of the approach
- Potential additional use-cases
  - LDAP
  - SYMPA
  - Cloud apps
- But also…
  - Not FEUDAL or PERUN but FEUDAL + PERUN

FERUN ??

Functionality outline

- PERUN is an MMS → up-to-date user info
  - Deploy via FEUDAL
- Once user is in FEUDAL, PERUN can:
  - Update the user info (groups, credentials)
  - Remove the user from a group (and services)
  - Remove the user completely from FEUDAL
- Easy to use additional MMSs

Current status

- FEUDAL and PERUN are available (both dev and prod)
- FEUDAL is connected to eduTEAMS
- Agreed on the API
  - Update user information
  - Get all users per VO
  - Get all users
  - Remove user

Status and next steps

✓ Done:

- Connected FEUDAL to eduTEAMS
- API set (pretty much)

▢ Next:

- Implement API
- Connect PERUN and FEUDAL
- Provide “classic” LDAP use case for FEUDAL (PERUN already supports it)
- Investigate further use cases (SYMPA?)
- FEUDAL and Windows usage

⛔ 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).