Karlsruhe Institute of Technology

Federated User Credential Deployment Portal (FEUDAL)

Lukas Burgey | August 29, 2018

STEINBUCH CENTRE FOR COMPUTING

Context

Helmholtz Data Federation (HDF)

Sites: KIT, FZJ, DKFZ, AKI, GSI, and DESY

Federated Identities using SP-IdP-Proxy (AARC BPA)

- User Authentication by the IdPs at the sites
- Extends user information from IdPs

Deployment

User deployment

- Account provisioning
- Deployment of user credentials
  - SSH public key
  - password
  - etc.

Requirements (1/2)

- Web Portal
- Deployment
  - Federated user authentication
  - Credentials: SSH public keys
  - Fault tolerant
  - Response time: Close to network latency
- Services
  - Services can be hosted at multiple sites
  - Sites can host multiple services

Requirements (2/2)

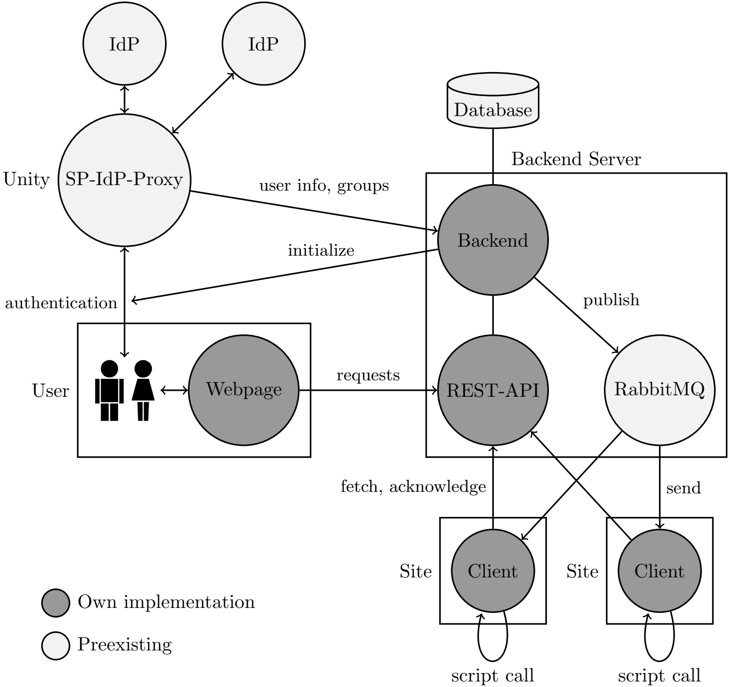
- At the sites:
  - Interface with all possible User Management Systems
  - Customisable by the local Administrator
  - Attractive to host services
  - No incoming connections
- Secure

FEUDAL Workflow

1. User: SSH public key upload
2. User: VO / service selection
3. Portal: Account provisioning at the services
4. Portal: Key deployment to the account
5. Portal: Display login information to the user
6. User: Can access the services with the public key

Architecture (1/2)

- Distributed:
  - FEUDAL clients
    - Every site hosts one or more clients
    - The clients execute the deployments
- Central elements:
  - Web portal
    - User interface
  - FEUDAL backend + database
    - Sends messages to the clients
    - Stores user information and credentials


IdP
IdP
Database
Backend Server
Unity
SP-IdP-Proxy
user info, groups
Backend
initialize
authentication
publish
User
Webpage
requests
REST-API
RabbitMQ
fetch, acknowledge
send
Site
Client
Site
Client
script call
script call
Own implementation
Preexisting

Architecture (1/2)

Technology

- SP-IdP-Proxy: OpenID Connect
- Backend: Django/Python
  - Inbuilt administration frontend
  - Simplifies usage of Database
  - Django REST Framework
- Clients: Go
  - Static linking
- Webpage: Angular/Typescript
  - Asynchronous requests

Messaging (1/4)

Messages (JSON):

- Backend → Client:
  - identifier
  - action $\in$ {“deploy”, “remove”}
  - service
  - SSH public key
  - user info (from OpenID Connect)
  - group memberships (from Unity)
- Backend ← Client: Acknowledgement
  - identifier
  - login information

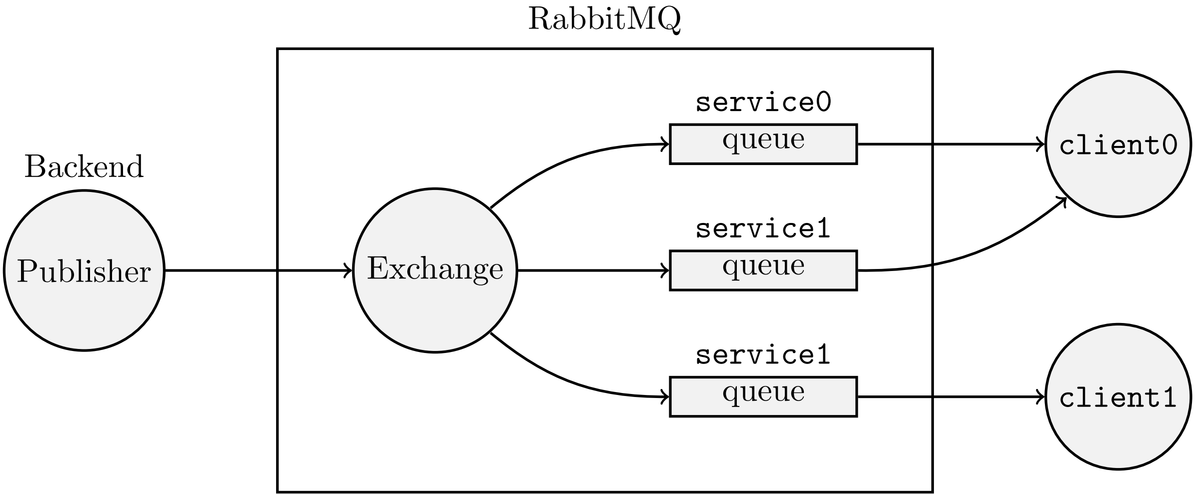
Messaging (2/4)

Publish Subscribe

- Quick transmission (close to network latency)
- Only outgoing connections at the clients
- Dedicated message broker: RabbitMQ
    - Delegated authentication of clients
    - Inbuilt message routing

Messaging (3/4)

Message routing



Messaging (4/4)

Clients manually fetch messages

- On startup
  - Missed deployments
- Per interval (e.g. 30 minutes)
  - Result: Unacknowledged deployments are retried

Security Considerations (1/2)

- Confidentiality & Integrity
  - TLS for all transmissions
- Authentication
  - User: OpenID Connect
  - FEUDAL Client: password

Security Considerations (2/2)

- Authorisation
  - User
    - groups from Unity
    - service ↔ groups
  - FEUDAL Client: configuration
- Trust
  - Service provider need to trust
    - SP-IdP-Proxy
    - Backend

  → Future work: Confirm data from the backend

Demo Time

Demo Time

Key Features

- Asynchronous deployment: if a site is down for a while, all deployments are retransmitted, once the site is back up.
- Future deployments: New machines can receive all users in the supported VO.
- Realtime deployments: http sockets to push information.
- Integration to local usermanagement left to the site-admin

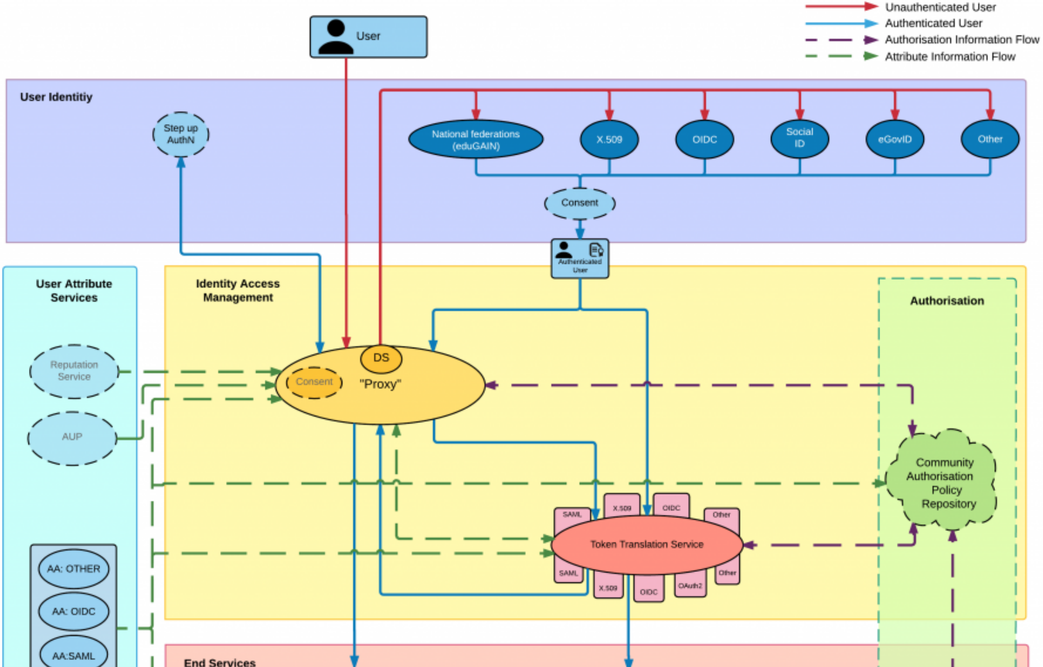
Questions?

Questions?

Backup Slides

Backup Slides

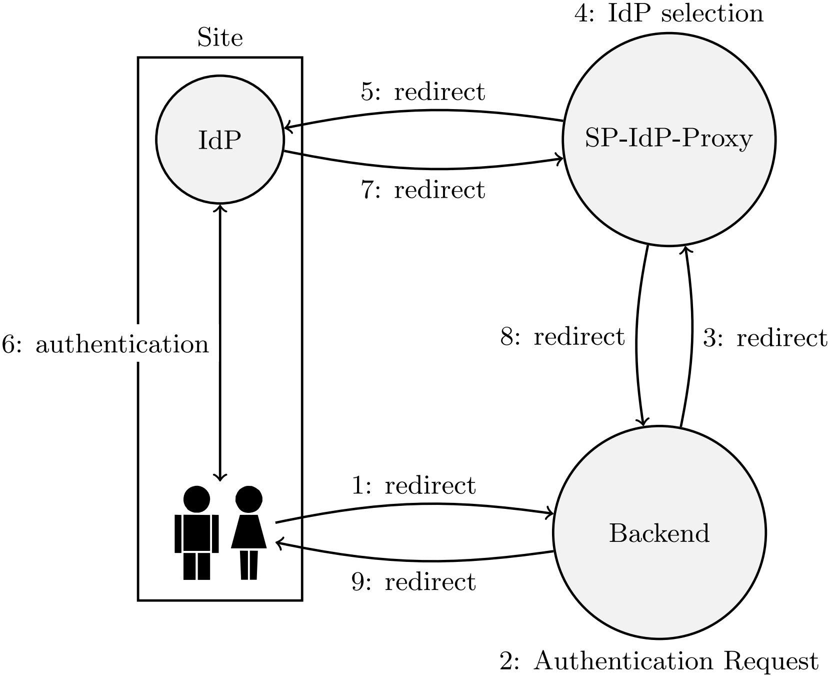
AARC Blueprint Architecture



WaTTS

Token Translation Service (AARC BPA)

- Uses plugins to translate tokens
- Plugins can be used to do deployment
  - Not optimal


4: IdP selection
Site
IdP
5: redirect
SP-IdP-Proxy
7: redirect
6: authentication
8: redirect
3: redirect
1: redirect
Backend
9: redirect
2: Authentication Request