WP3 identified services integration plan

- What does integration mean?

- What are we integrating?

- How do we integrate things?

What does integration mean?

- When we say "integration" , we really mean "interoperability"

- Assumptions:

  - There will be many services

  - They will be hosted by several domains

  - They will speak different "languages"
  The user's experience of these boundaries should be minimised

What does integration mean?

- Some design considerations:
  - Users should authenticate "at home"
    - An identity provider that speaks a specific protocol
  - Services authenticate and authorise users at the point of usage
    - Services should not *have* to be re-tooled to speak arbitrary protocols
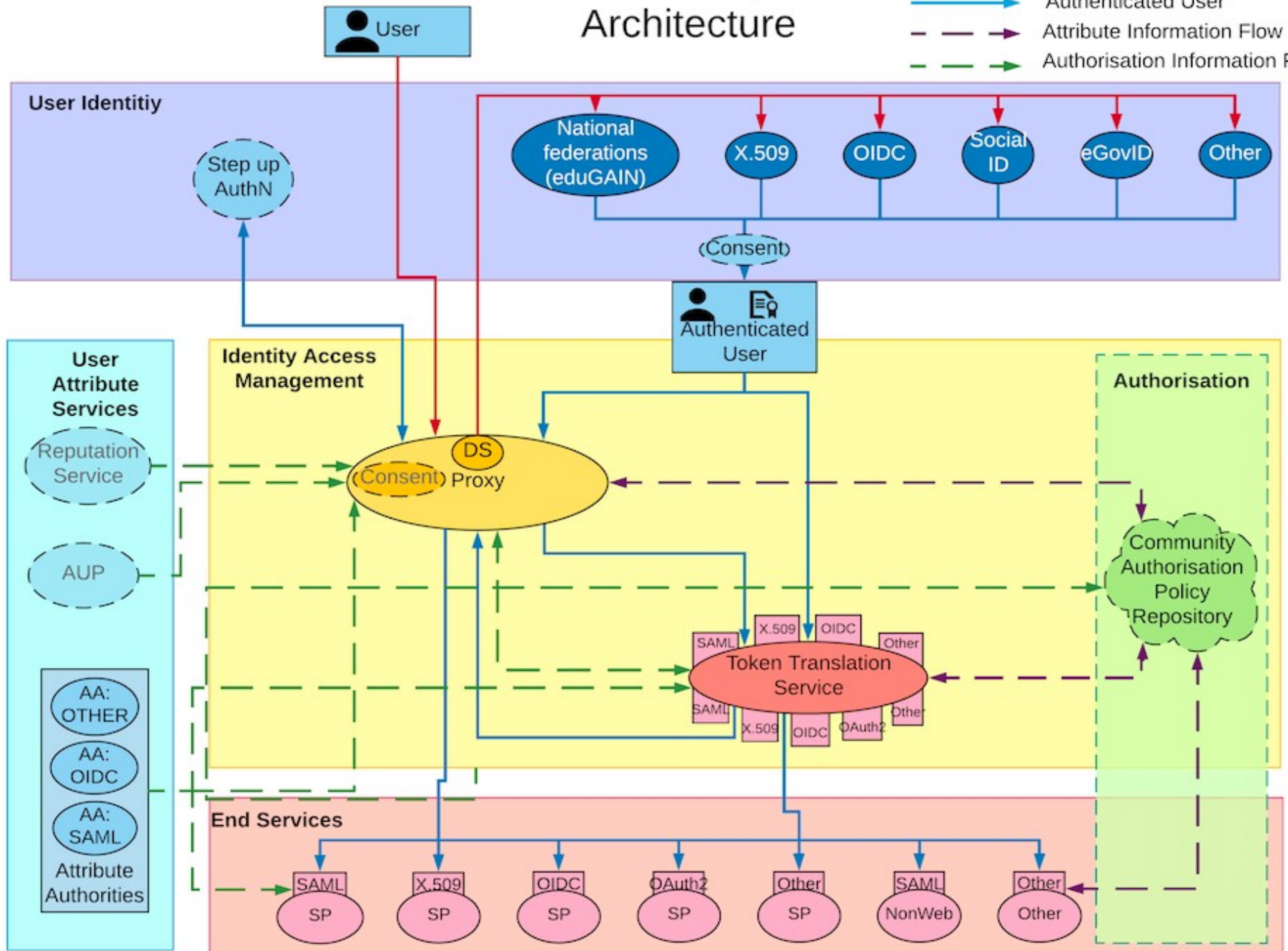   - Services should not have to deal with an arbitrary number of endpoints.

Hello ~~Kitty~~ Proxy

AARC Blueprint Architecture

Legend:
- Unauthenticated User (red solid arrow)
- Authenticated User (blue solid arrow)
- Attribute Information Flow (purple dashed arrow)
- Authorisation Information Flow (green dashed arrow)

User

User Identitiy
- Step up AuthN
- National federations (eduGAIN)
- X.509
- OIDC
- Social ID
- eGovID
- Other
- Consent

Authenticated User

User Attribute Services
- Reputation Service
- AUP

Attribute Authorities
- AA: OTHER
- AA: OIDC
- AA: SAML

Identity Access Management
- DS
- (Consent) Proxy

Token Translation Service
- SAML
- X.509
- OIDC
- Other
- SAML
- X.509
- OIDC
- OAuth2
- Other

Authorisation
- Community Authorisation Policy Repository

End Services
- SAML — SP
- X.509 — SP
- OIDC — SP
- OAuth2 — SP
- Other — SP
- SAML — NonWeb
- Other — Other

What are we integrating?
Services identified by WP3

- Compute:

  - Workflow Manager (actually: work**load** manager)

  - Endpoints:

    - HPC

    - HTC

  - Workflows / pipelines (complex)

What are we integrating?
Services identified by WP3

- Data:

    - Data Process Manager

    - Data Catalogue

    - Transfer

    - Online data
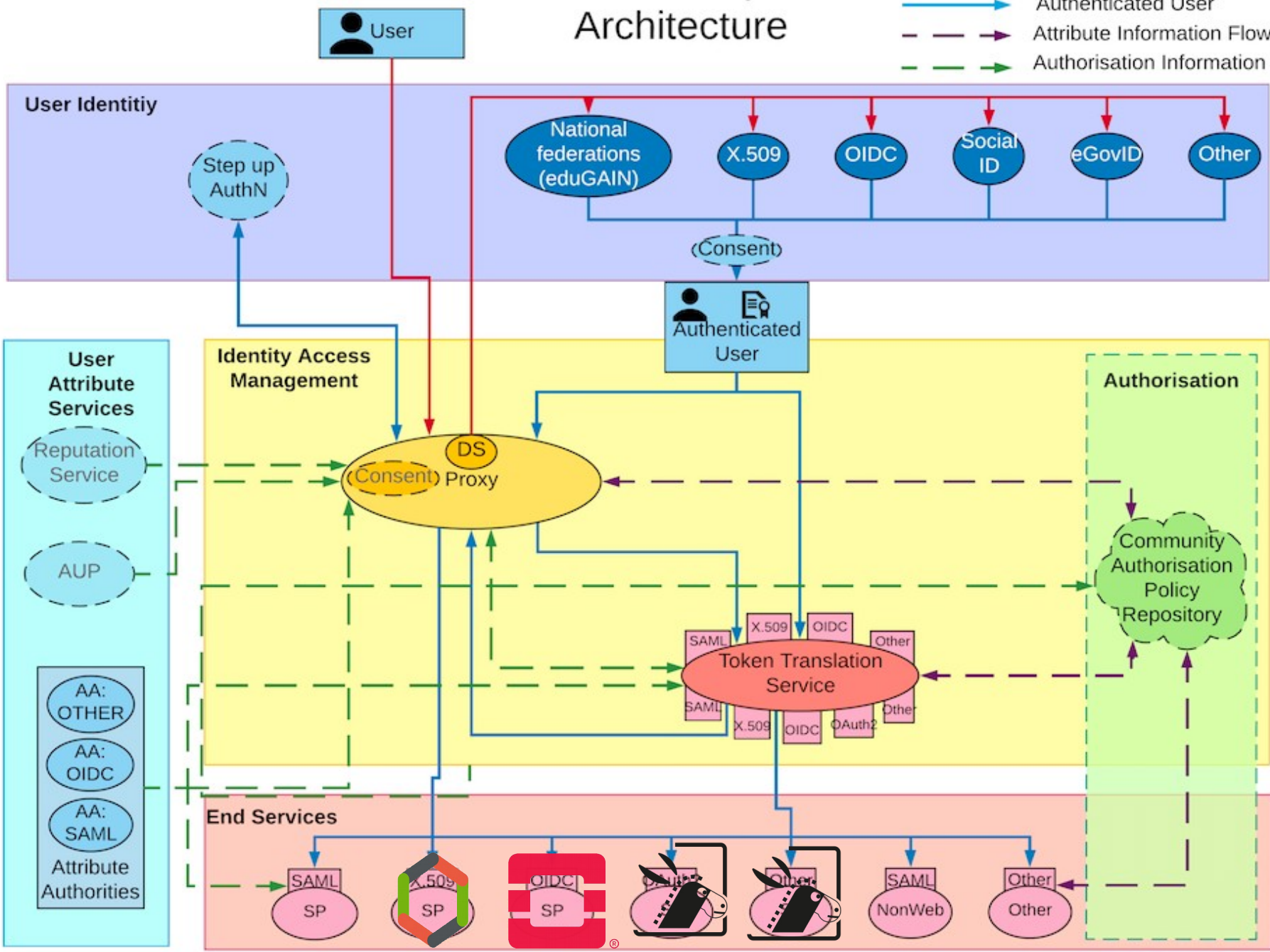
    - Federator

Not mentioned – all the things to make it go.

- Ops is hard, but doesn't have to be – **if you build a proper service management system**
  - Applications
    - Integration, QA, Delivery
  - Accounting
    - Compute, storage, bandwidth, *etc*
  - Monitoring
    - Alarms: SLA, SLO violations, infrastructure, etc.
  - Configuration and deployment

AARC Blueprint Architecture

Legend:
- Unauthenticated User
- Authenticated User
- Attribute Information Flow
- Authorisation Information Flow

User

User Identitiy
- Step up AuthN
- National federations (eduGAIN)
- X.509
- OIDC
- Social ID
- eGovID
- Other
- Consent

Authenticated User

User Attribute Services
- Reputation Service
- AUP
- AA: OTHER
- AA: OIDC
- AA: SAML
- Attribute Authorities

Identity Access Management
- DS
- Consent
- Proxy
- SAML / X.509 / OIDC / Other
- Token Translation Service
- SAML / X.509 / OIDC / OAuth2 / Other

Authorisation
- Community Authorisation Policy Repository

End Services
- SAML SP
- X.509 SP
- OIDC SP
- OAuth2
- Other
- SAML NonWeb
- Other

How do we integrate things?

- An AAI for AENEAS is needed

  - Use an existing one (aai.egi.eu)

  - Have a managed solution for the demos (fork aai.egi.eu)

  - Use something special for the demo

- Add service endpoints to the AAI

Describe scenarios that can fail in specific ways

- *As user with OIDC credentials, I should be able to move data, using Rucio, from endpoint A (xrootd) to endpoint B (xrootd)*

- Failure scenarios:

  - Rucio cannot speak OIDC

    - Use a token translation service?

    - User a different tool?

  - Rucio cannot speak to an xrootd endpoint

    - A different data federator is needed

Describe scenarios that can fail in specific ways

- *As a user, I should be able to access a DIRAC web interface with my SAML credentials*

- Failure scenarios:

  - DIRAC web interface cannot authenticate with SAML

    - Use a proxy to authenticate and speak another protocol to DIRAC web interface

  - DIRAC submission engine needs needs an X.509 certificate

    - Use a token translation service to issue a proxy certificate to the DIRAC submission engine

Wrap: Service Integration Plan

1) Have an AAI based on the AARC blueprint

2) Pick a service

    1) Understand it's access and authorisation model

    2) Have an instance of it

    3) Plug it into the AAI proxy

3) Define scenarios which have clear tests that can fail

4) Execute the tests

5) Evaluate the gaps – Where do we build the bridge across the gap ? Tool or AAI?

You will need an actual infrastructure

- Either build it yourself

- "simulate" the experience on EOSC with a pilot project.