EOSC AAI

Christos Kanellopoulos (GEANT)

GÉANT

GÉANT is the collaboration of European National Research and Education Networks (NRENs). Together we deliver an information ecosystem of infrastructure and services to advance research, education, and innovation on a global scale.

Network

Europe's essential terabit network for research and education **interconnects Europe's NRENs and links them to over 100 countries** in every region of the world.

Community

The **GÉANT Community Programme** facilitates **collaboration and knowledge sharing** between NRENs, user organisations, R&E institutions, and the commercial sector.

Services

GÉANT develops and operates a range of **connectivity**, **cloud**, and **identity services** that ensure a safe and secure environment **for researchers, educators, and students**.

eduroam eduGAIN

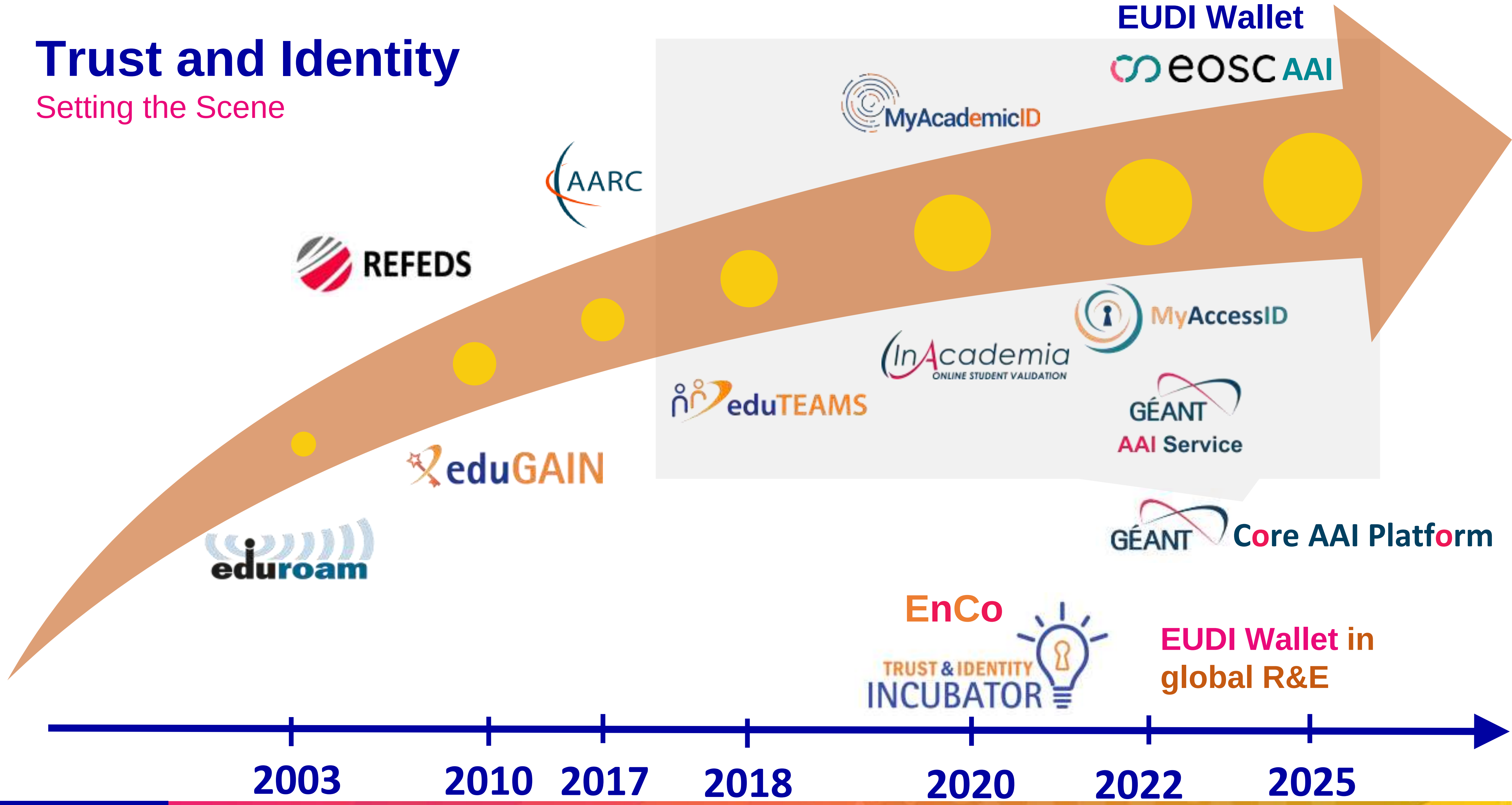
MyAccessID eduVPN eduSign eduTEAMS

Projects

GÉANT's **long-standing project relationship with the European Union (EU)** has benefited research, education, and innovation around the world **for over 25 years**.

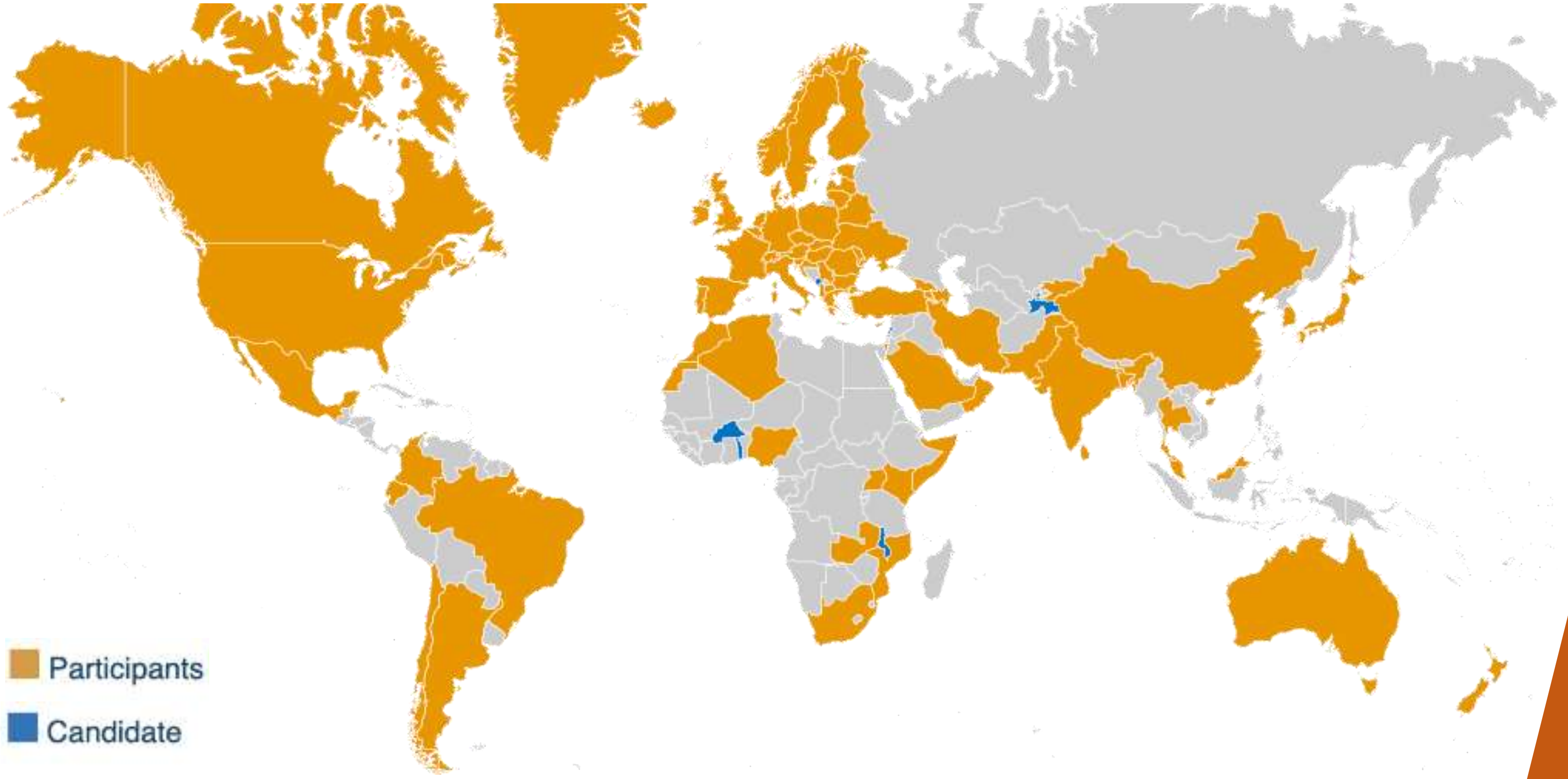
Trust and Identity

Setting the Scene



eduGAIN

Service uptake



78
Identity Federations

6000+
Identity Providers

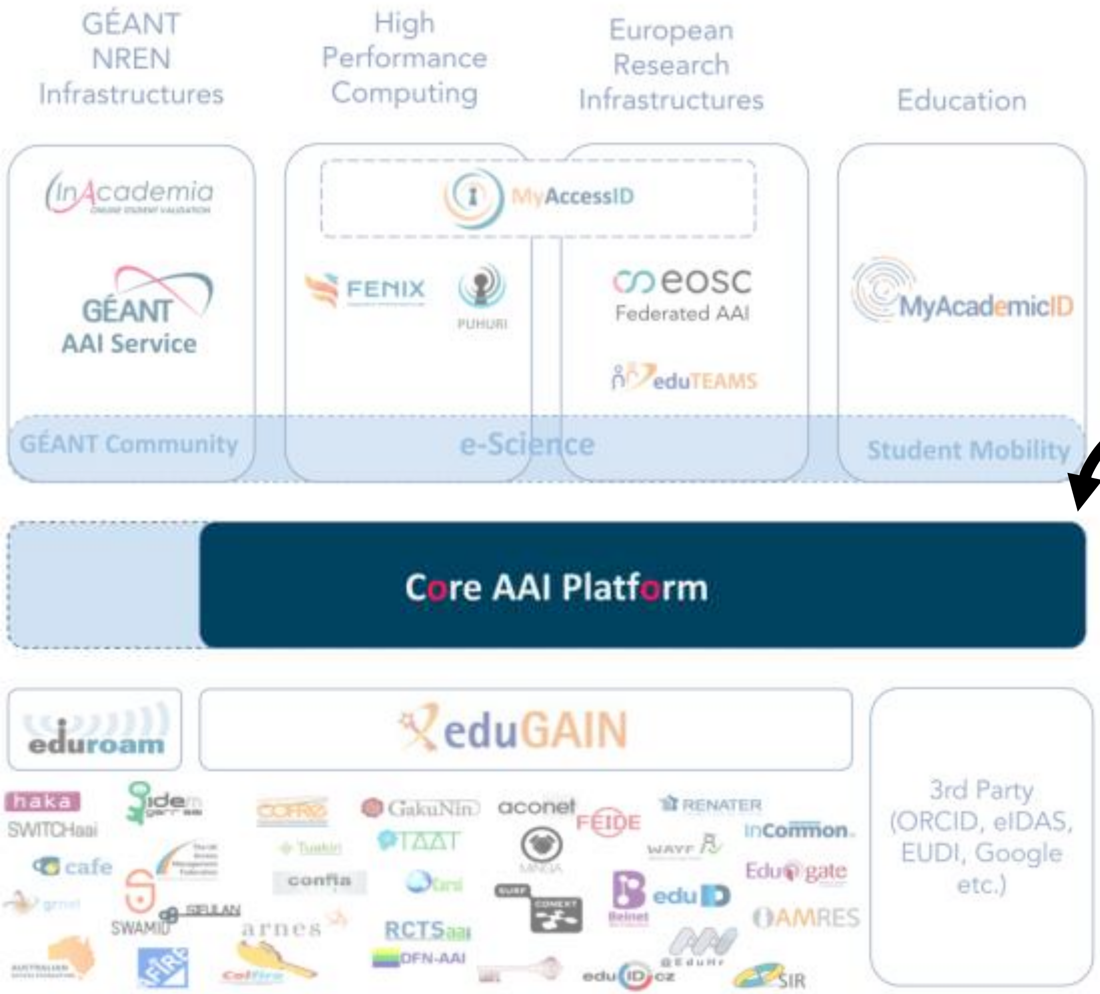
3700+
Service Providers

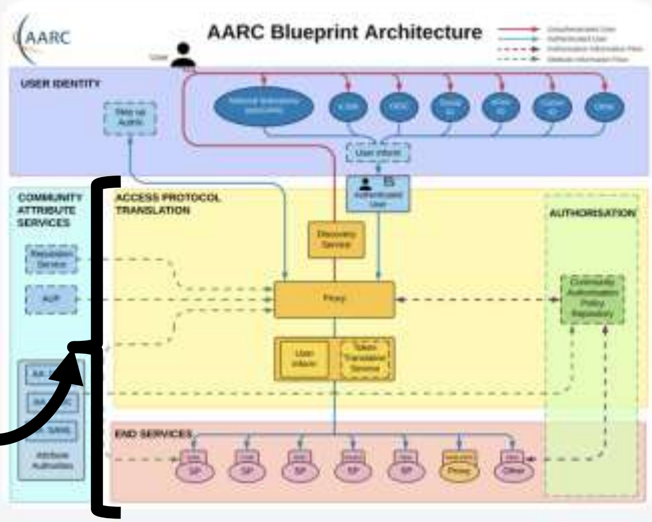
Core AAI Platform

eduroam

eduGAIN

3rd Party (ORCID, eIDAS, EUDI, Google etc.)





Community Attribute Services:

- How should attributes from multiple sources be aggregated? AARC-G003
- How should I express the home institute of a user? AARC-G025
- How should I express the identifier of a user AARC-G026
- What are the best practices for running my Attribute Authorities securely? AARC-G071
- Which Acceptable Use Policy should I use to facilitate interoperability? AARC-I044
- How should I infer the affiliation of a user? AARC-G057

User Identity:

- How should I integrate Social Media Identity Providers? AARC-G008
- How should users link accounts, and how does that affect Assurance? AARC-G009
- How should services indicate that they would like users to authenticate with multifactor authentication, and how should my proxy forward that information? AARC-G029

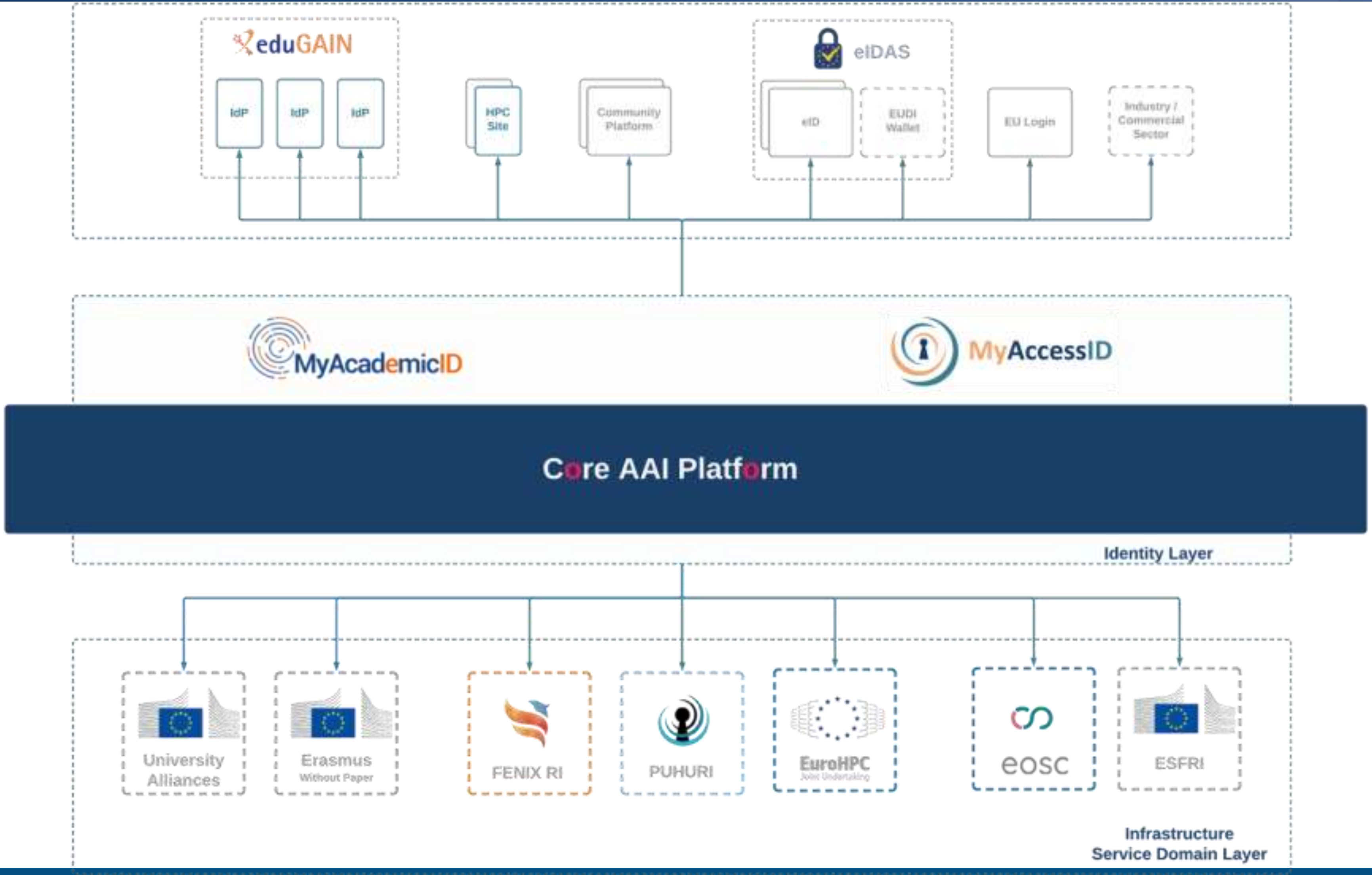
Assurance:

- How should assurance information of external identities be calculated? AARC-G031
- What can I say about assurance of identities from social media accounts? AARC-G041
- How is assurance impacted by account linking? AARC-G009
- How should assurance information be shared with other infrastructures? AARC-G021
- Which Assurance Profiles should I use, there are so many! AARC-I050

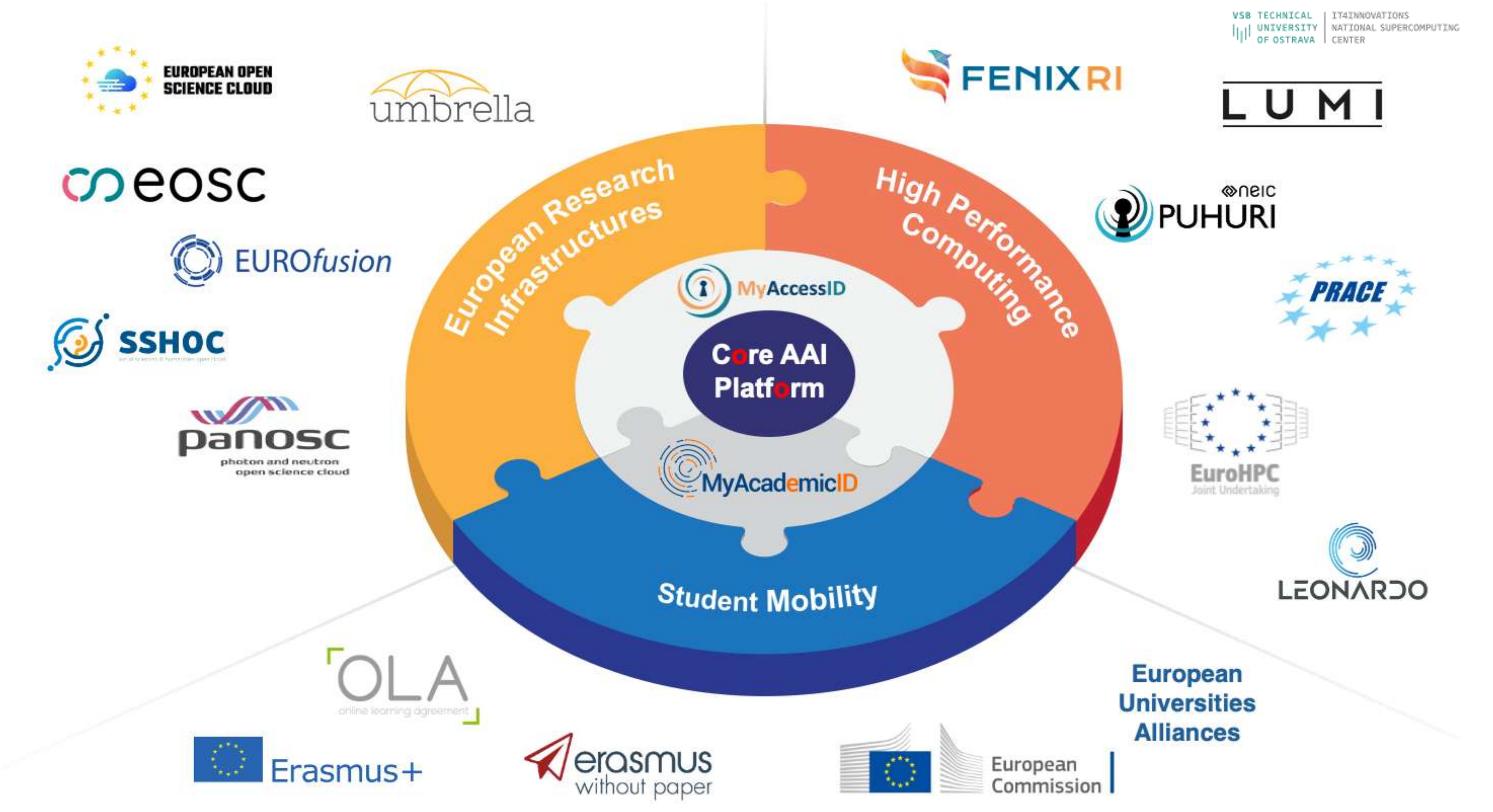
Access Protocol Translation:

- Which best practices should I follow for my Token Translation Services? AARC-G004

AAI Platform as a Service: Core AAI

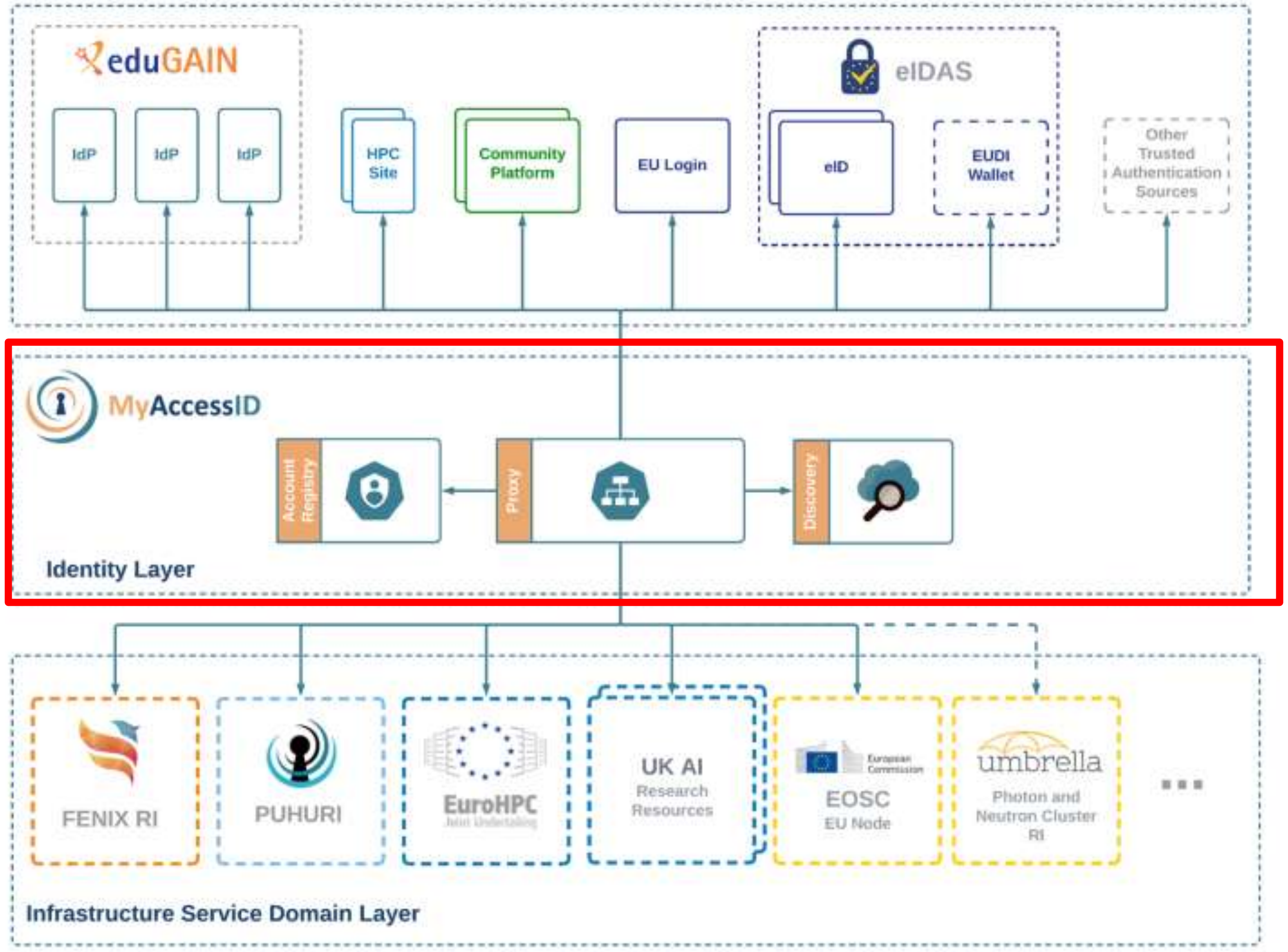


- Consistent Attributes
- Multi-protocol support *(SAML, OIDC, OAuth2)*
- Additional Authentication Sources (eIDs, EU Login, Community Platforms, ORCID, Social Logins, IdPs of Last Resort)
- Strong Identity and Authentication assurances (where needed)
- Trusted Platform

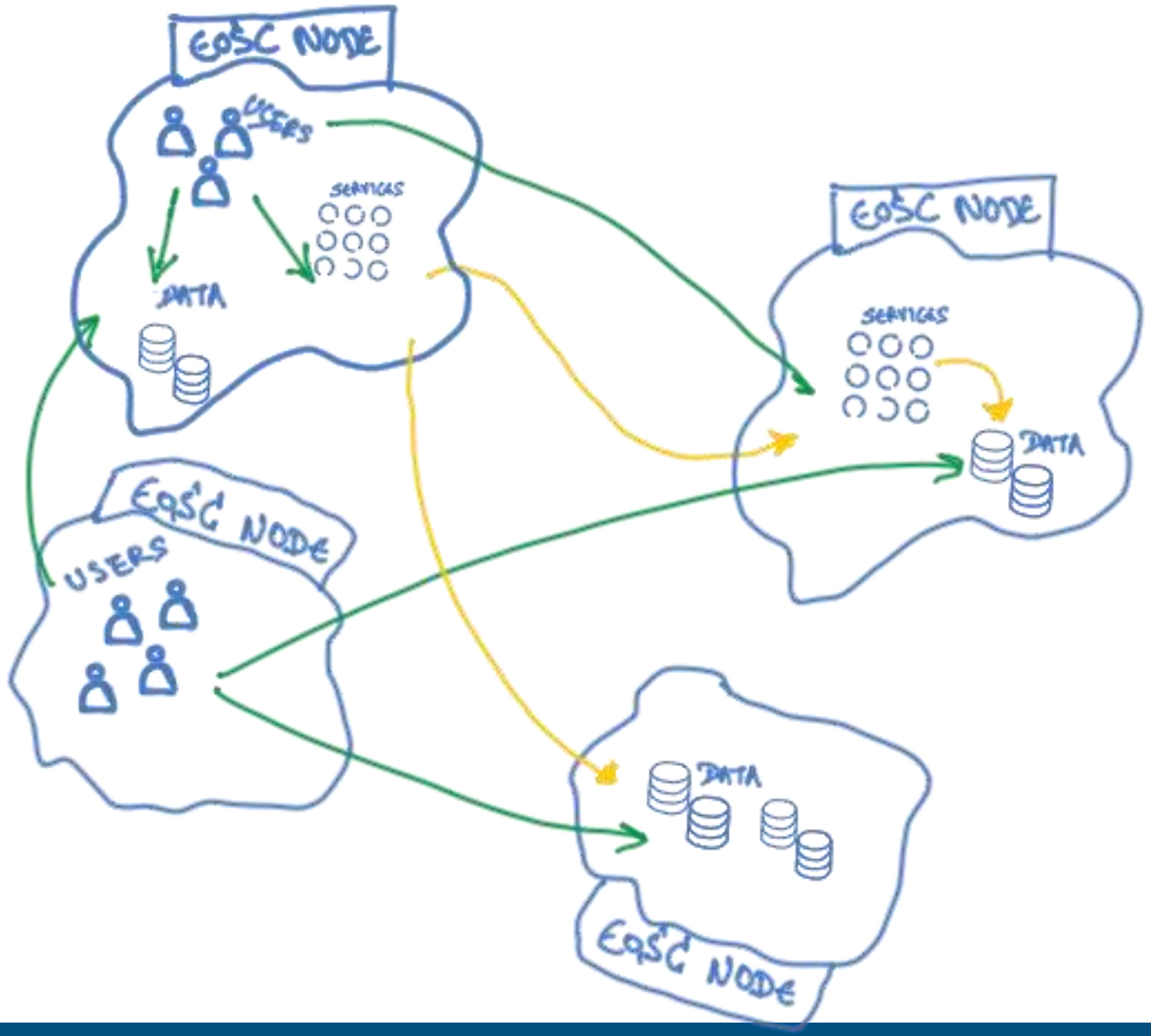

European Research Infrastructures
High Performance Computing
Student Mobility
MyAccessID
Core AAI Platform
MyAcademicID
EUROPEAN OPEN SCIENCE CLOUD
umbrella
eosc
EUROfusion
SSHOC
panosc
photon and neutron open science cloud
FENIX RI
VSB TECHNICAL UNIVERSITY OF OSTRAVA
IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER
LUMI
neic
PUHURI
PRACE
EuroHPC
Joint Undertaking
LEONARDO
OLA
online learning agreement
Erasmus+
erasmus without paper
European Commission
European Universities Alliances

MyAccessID: A common Identity Layer for Science

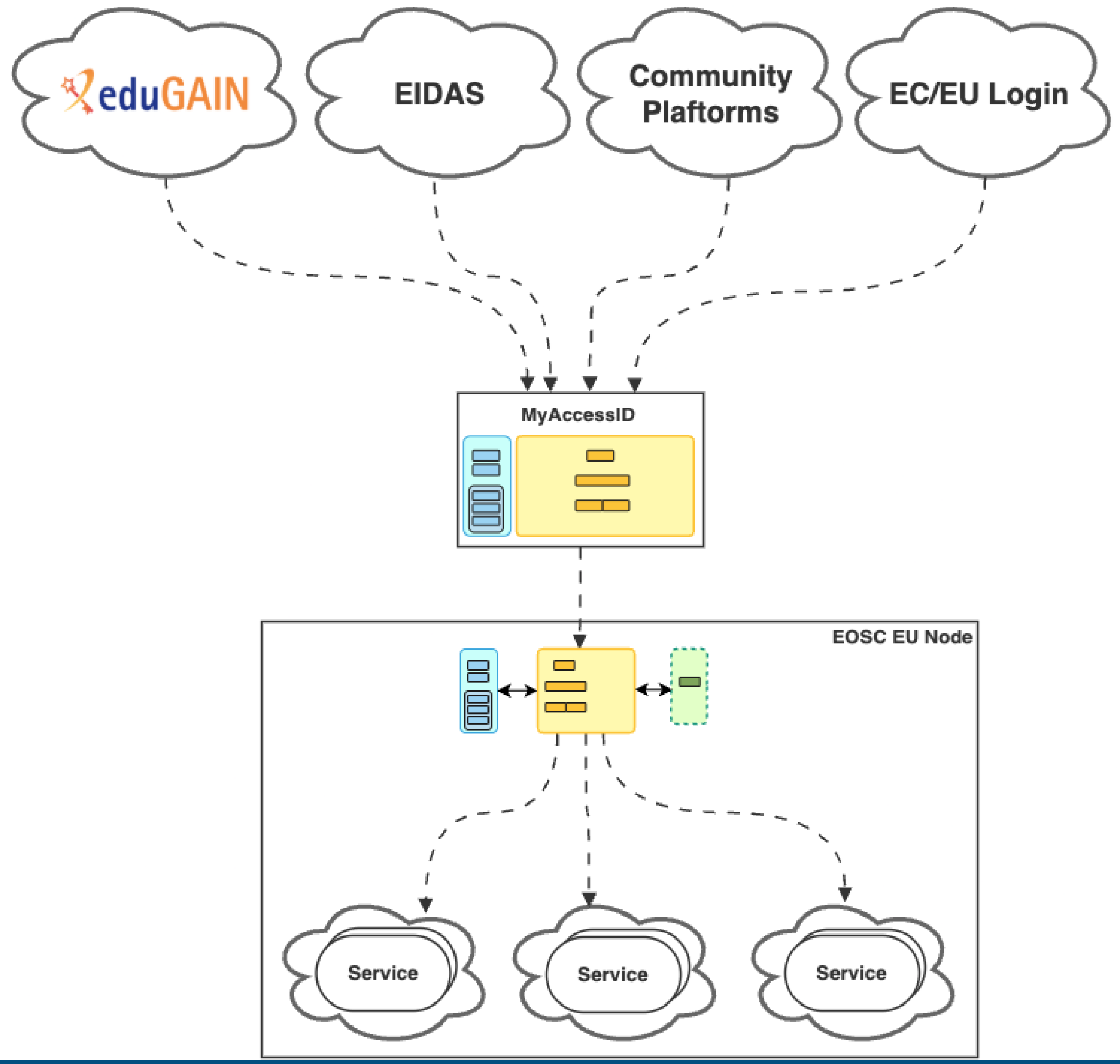
- HPC Datacenters are in the process of transforming to **Infrastructure Service Providers** with **a diverse Service Portfolio**
- These services become available in different administrative and policy domains, which we call **Infrastructure Service Domains**
- **A common Authentication and Authorization Infrastructure** enables uniform accessibility to scientists and engineers at European scale



EOSC Federation



The EOSC EU Node and the Core AAI Platform



First deliverable of the new EOSC-A Task Forces

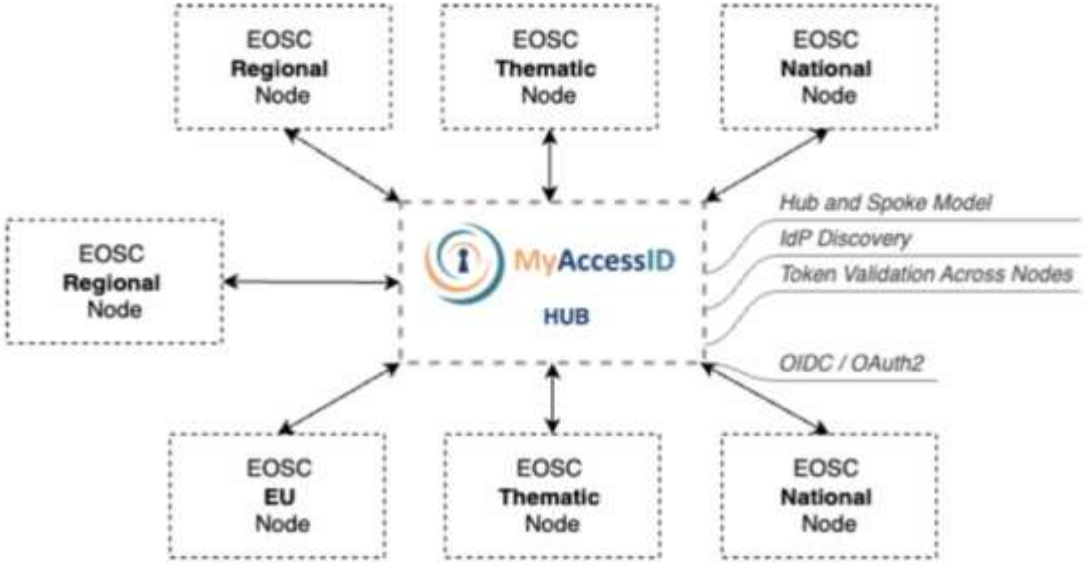
eosc | Technical and Semantic Interoperability Task Force

DOI

EOSC AAI Architecture 2025

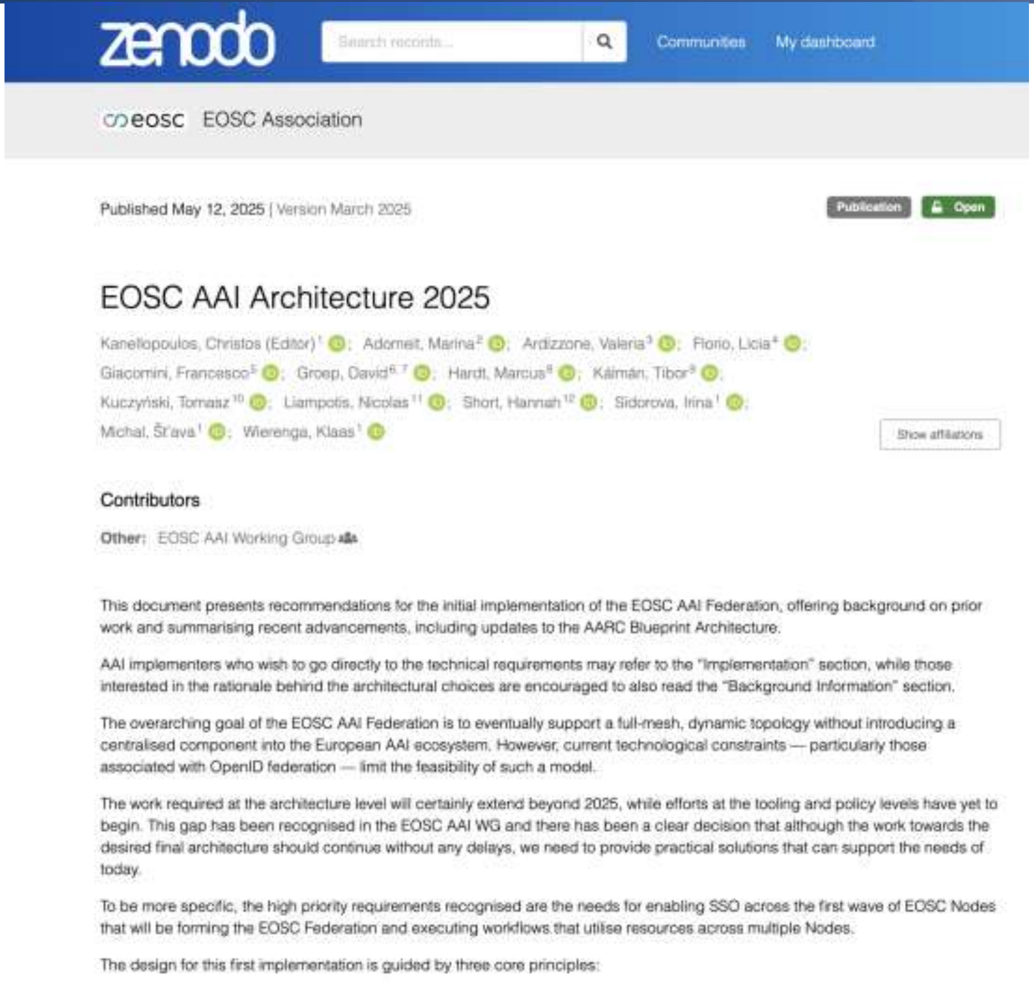
May 12, 2025

- The document presents **recommendations for the initial implementation of the EOSC AAI Federation,** offering background on prior work and summarising recent advancements, including updates to the AARC Blueprint Architecture.
- It is intended as a **practical guide for candidate EOSC Nodes,** outlining the steps necessary to connect with the EOSC AAI Federation. In the EOSC model, Nodes act as the primary integration points for services as it is described in the EOSC Federation Handbook, services are onboarded to individual Nodes rather than directly to the Federation.
- The overarching goal of the EOSC AAI Federation is to eventually support a **full-mesh, dynamic topology** without introducing a centralised component into the European AAI ecosystem.



EOSC AAI Federation "hub-and-spoke" model

EOSC AAI Architecture - March 2025 version



zenodo

Search records... Communities My dashboard

eosc EOSC Association

Published May 12, 2025 | Version March 2025

Publication Open

EOSC AAI Architecture 2025

Kanellopoulos, Christos (Editor)[1]; Adomeit, Marina[2]; Ardizzone, Valeria[3]; Florio, Licia[4]; Giacomini, Francesco[5]; Groep, David[6,7]; Hardt, Marcus[8]; Kálmán, Tibor[9]; Kuczyński, Tomasz[10]; Liampotis, Nicolas[11]; Short, Hannah[12]; Sidorova, Irina[1]; Michal, Šťava[1]; Wierenga, Klaas[1]

Show affiliations

Contributors

Other: EOSC AAI Working Group

This document presents recommendations for the initial implementation of the EOSC AAI Federation, offering background on prior work and summarising recent advancements, including updates to the AARC Blueprint Architecture.

AAI implementers who wish to go directly to the technical requirements may refer to the "Implementation" section, while those interested in the rationale behind the architectural choices are encouraged to also read the "Background Information" section.

The overarching goal of the EOSC AAI Federation is to eventually support a full-mesh, dynamic topology without introducing a centralised component into the European AAI ecosystem. However, current technological constraints — particularly those associated with OpenID federation — limit the feasibility of such a model.

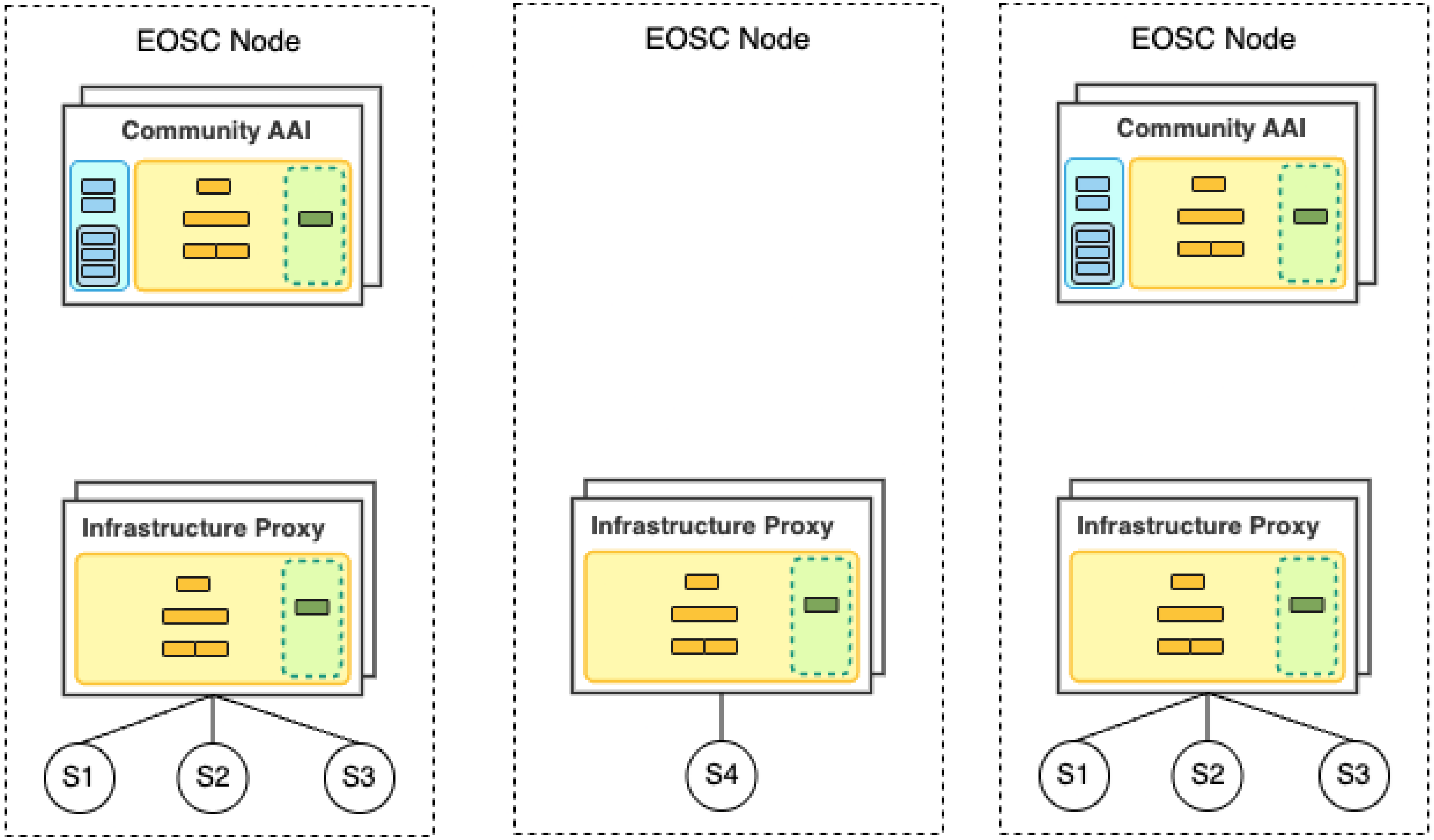
The work required at the architecture level will certainly extend beyond 2025, while efforts at the tooling and policy levels have yet to begin. This gap has been recognised in the EOSC AAI WG and there has been a clear decision that although the work towards the desired final architecture should continue without any delays, we need to provide practical solutions that can support the needs of today.

To be more specific, the high priority requirements recognised are the needs for enabling SSO across the first wave of EOSC Nodes that will be forming the EOSC Federation and executing workflows that utilise resources across multiple Nodes.

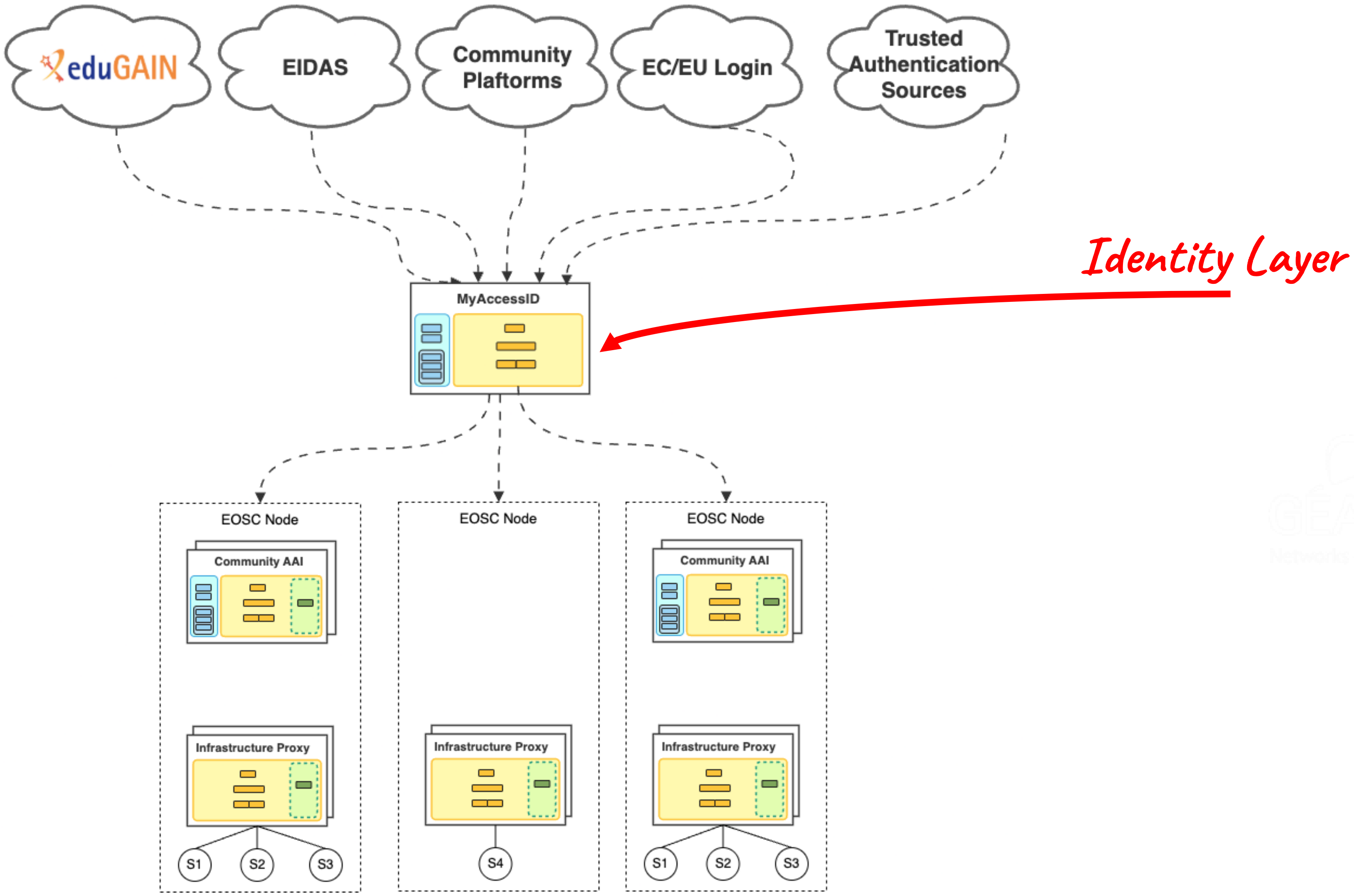
The design for this first implementation is guided by three core principles:

https://doi.org/10.5281/zenodo.15388270

EOSC AAI Nodes



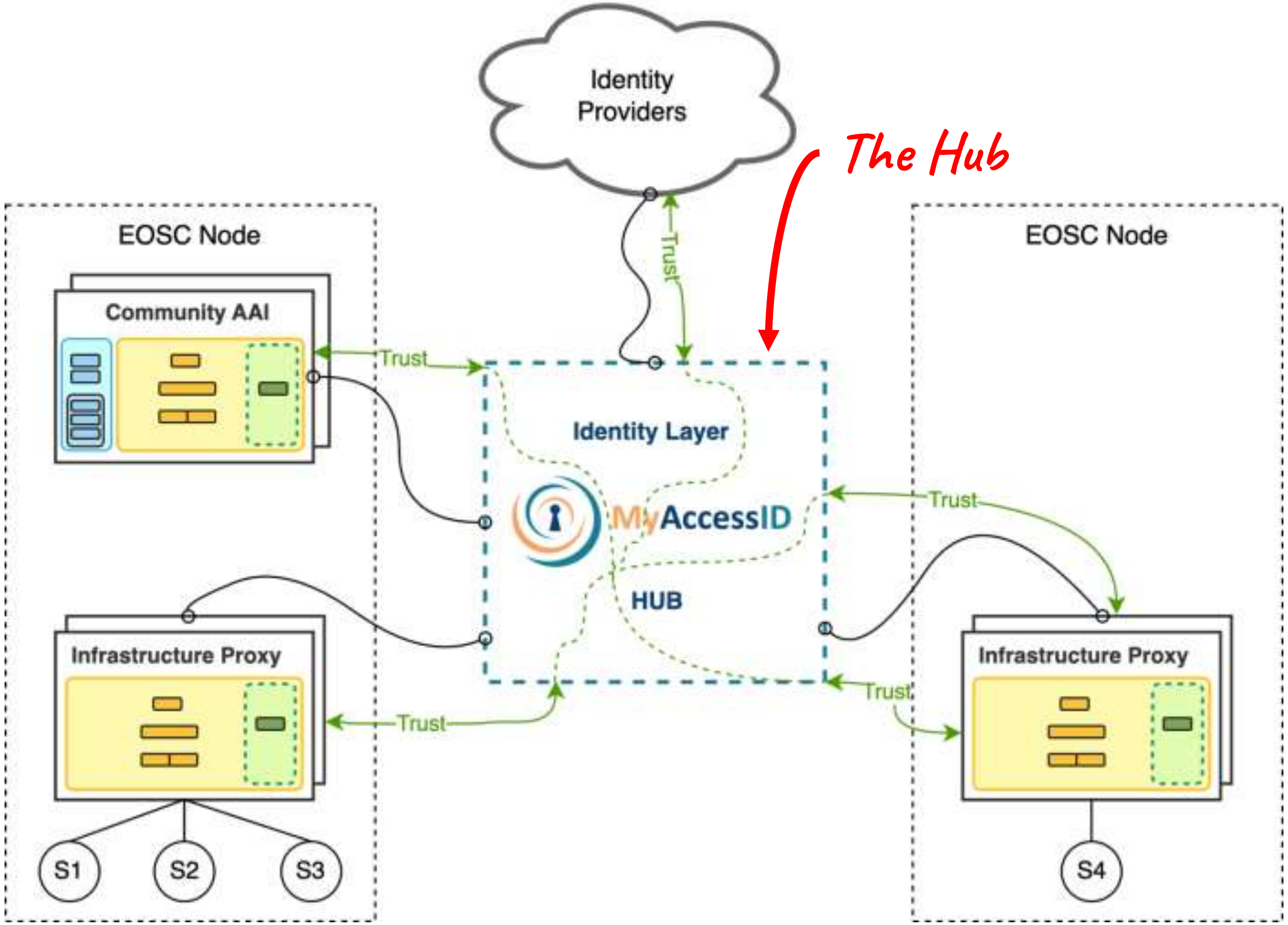
EOSC AAI Federation - The Identity Layer



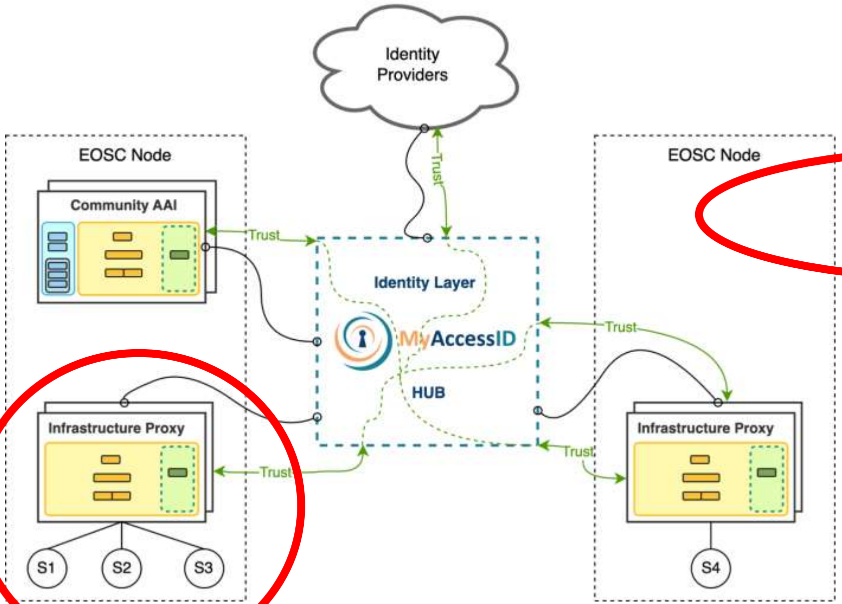

zenodo

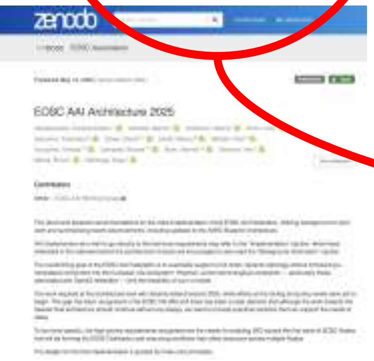
EOSC AAI Architecture 2025

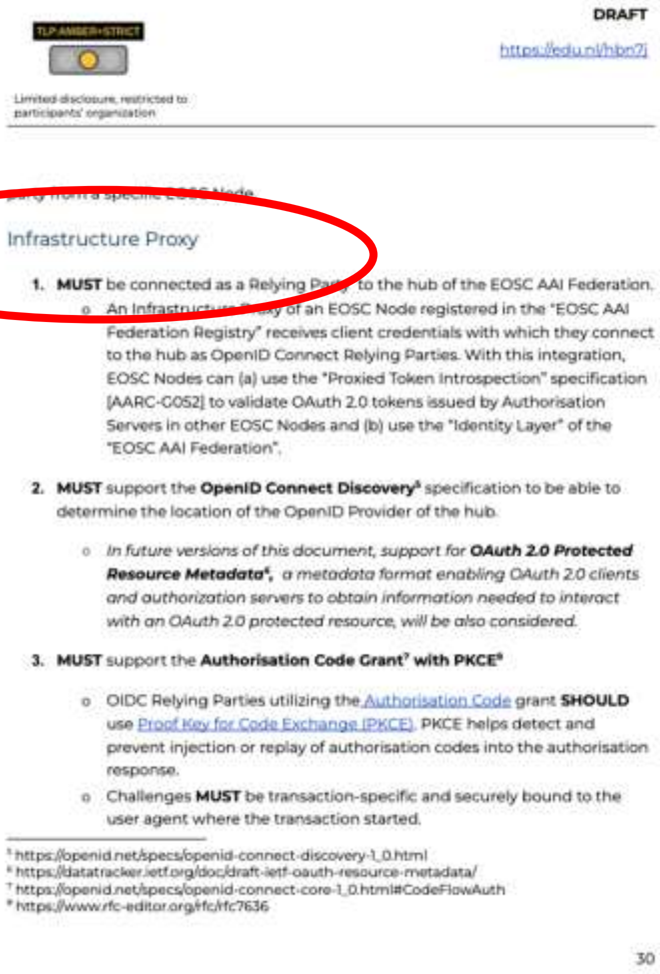
EOSC AAI Federation - Hub and Spoke model



Infrastructure Proxy







TLP:AMBER+STRICT

Limited disclosure, restricted to participants' organization

DRAFT

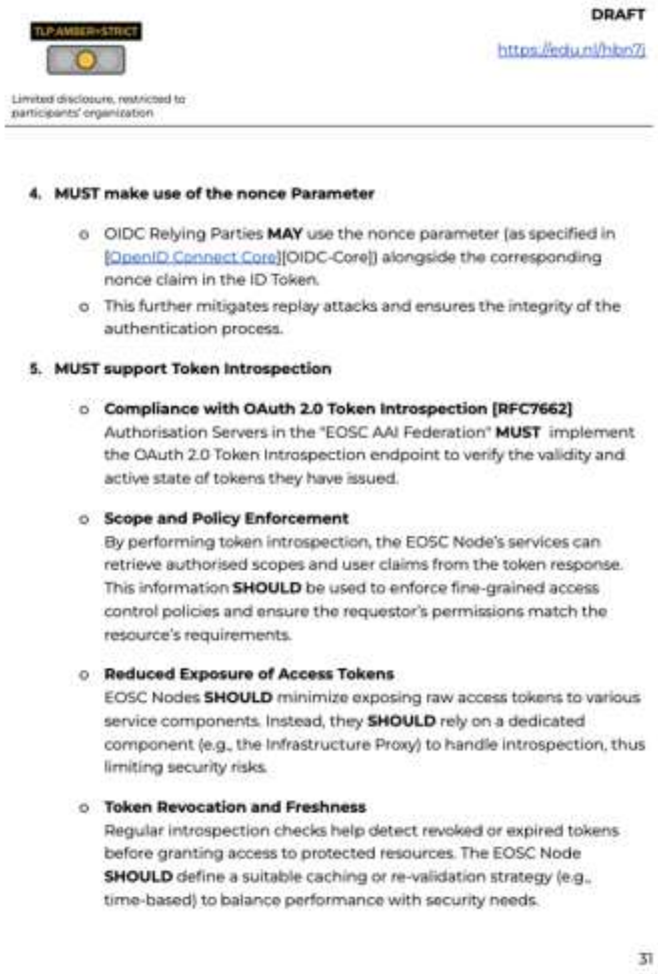
https://edu.nl/hbn7j

Infrastructure Proxy

1. **MUST** be connected as a Relying Party to the hub of the EOSC AAI Federation.
   - An Infrastructure Proxy of an EOSC Node registered in the "EOSC AAI Federation Registry" receives client credentials with which they connect to the hub as OpenID Connect Relying Parties. With this integration, EOSC Nodes can (a) use the "Proxied Token Introspection" specification [AARC-G052] to validate OAuth 2.0 tokens issued by Authorisation Servers in other EOSC Nodes and (b) use the "Identity Layer" of the "EOSC AAI Federation".
2. **MUST** support the **OpenID Connect Discovery**[5] specification to be able to determine the location of the OpenID Provider of the hub.
   - *In future versions of this document, support for **OAuth 2.0 Protected Resource Metadata**[6], a metadata format enabling OAuth 2.0 clients and authorization servers to obtain information needed to interact with an OAuth 2.0 protected resource, will be also considered.*
3. **MUST** support the **Authorisation Code Grant**[7] **with PKCE**[8]
   - OIDC Relying Parties utilizing the Authorisation Code grant **SHOULD** use Proof Key for Code Exchange (PKCE). PKCE helps detect and prevent injection or replay of authorisation codes into the authorisation response.
   - Challenges **MUST** be transaction-specific and securely bound to the user agent where the transaction started.

[5] https://openid.net/specs/openid-connect-discovery-1_0.html
[6] https://datatracker.ietf.org/doc/draft-ietf-oauth-resource-metadata/
[7] https://openid.net/specs/openid-connect-core-1_0.html#CodeFlowAuth
[8] https://www.rfc-editor.org/rfc/rfc7636

30



TLP:AMBER+STRICT

Limited disclosure, restricted to participants' organization

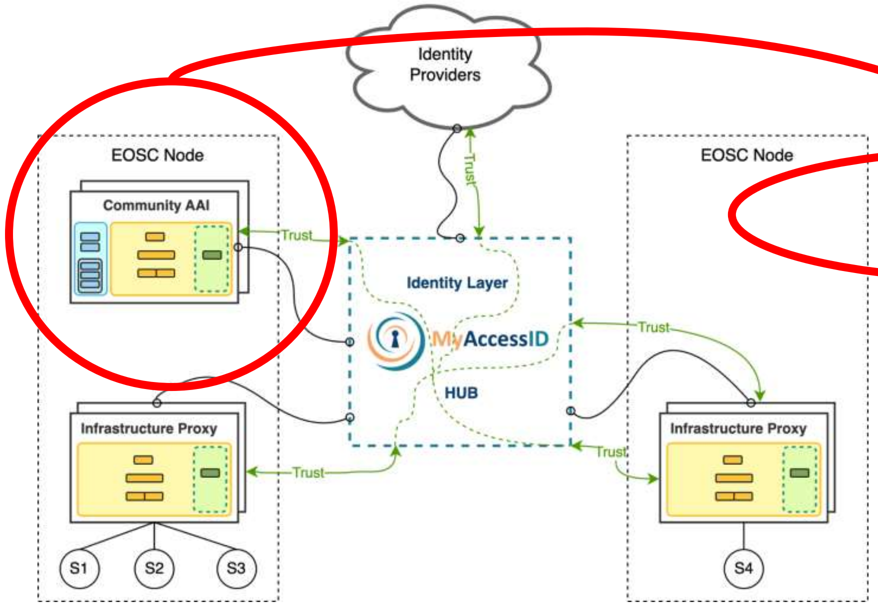
DRAFT

https://edu.nl/hbn7j

4. **MUST make use of the nonce Parameter**
   - OIDC Relying Parties **MAY** use the nonce parameter (as specified in [OpenID Connect Core][OIDC-Core]) alongside the corresponding nonce claim in the ID Token.
   - This further mitigates replay attacks and ensures the integrity of the authentication process.
5. **MUST support Token Introspection**
   - **Compliance with OAuth 2.0 Token Introspection [RFC7662]**
     Authorisation Servers in the "EOSC AAI Federation" **MUST** implement the OAuth 2.0 Token Introspection endpoint to verify the validity and active state of tokens they have issued.
   - **Scope and Policy Enforcement**
     By performing token introspection, the EOSC Node's services can retrieve authorised scopes and user claims from the token response. This information **SHOULD** be used to enforce fine-grained access control policies and ensure the requestor's permissions match the resource's requirements.
   - **Reduced Exposure of Access Tokens**
     EOSC Nodes **SHOULD** minimize exposing raw access tokens to various service components. Instead, they **SHOULD** rely on a dedicated component (e.g., the Infrastructure Proxy) to handle introspection, thus limiting security risks.
   - **Token Revocation and Freshness**
     Regular introspection checks help detect revoked or expired tokens before granting access to protected resources. The EOSC Node **SHOULD** define a suitable caching or re-validation strategy (e.g., time-based) to balance performance with security needs.

31

Community AAI



media accounts and passport vetted identities. The support of RAF and AARC-G021 allows typical LoA requirements for access to top notch HPC resources vs. wiki pages.

EOSC Nodes **MAY** have one or more Community AAIs whose users should be able to access services provided by other EOSC Nodes. In such cases the Community AAIs:

1. **MUST** connect their OpenID Connect Providers / OAuth 2.0 Authorisation Servers to the Hub of the EOSC AAI Federation.

   The Collaboration Platform (Community AAI) registered in the "EOSC AAI Federation Registry" may provide or receive client credentials with which the central hub of the EOSC AAI Federation is connected as an OpenID Connect Relying Party / OAuth 2.0 client. With this integration:

   1. OAuth 2.0 tokens issued by these Community AAIs can be used to access services provided by other EOSC Nodes, enabling cross-Node workflows.
   2. The central hub of the "EOSC AAI Federation" will be able to authenticate users at and/or link the collaboration identities of the users of these Community AAIs

2. **MUST** support the **OpenID Connect Discovery**[16] specification so that the central hub of the EOSC AAI Federation can dynamically determine the location and related trust metadata of the OpenID Provider of the Collaboration Platform (Community AAI).

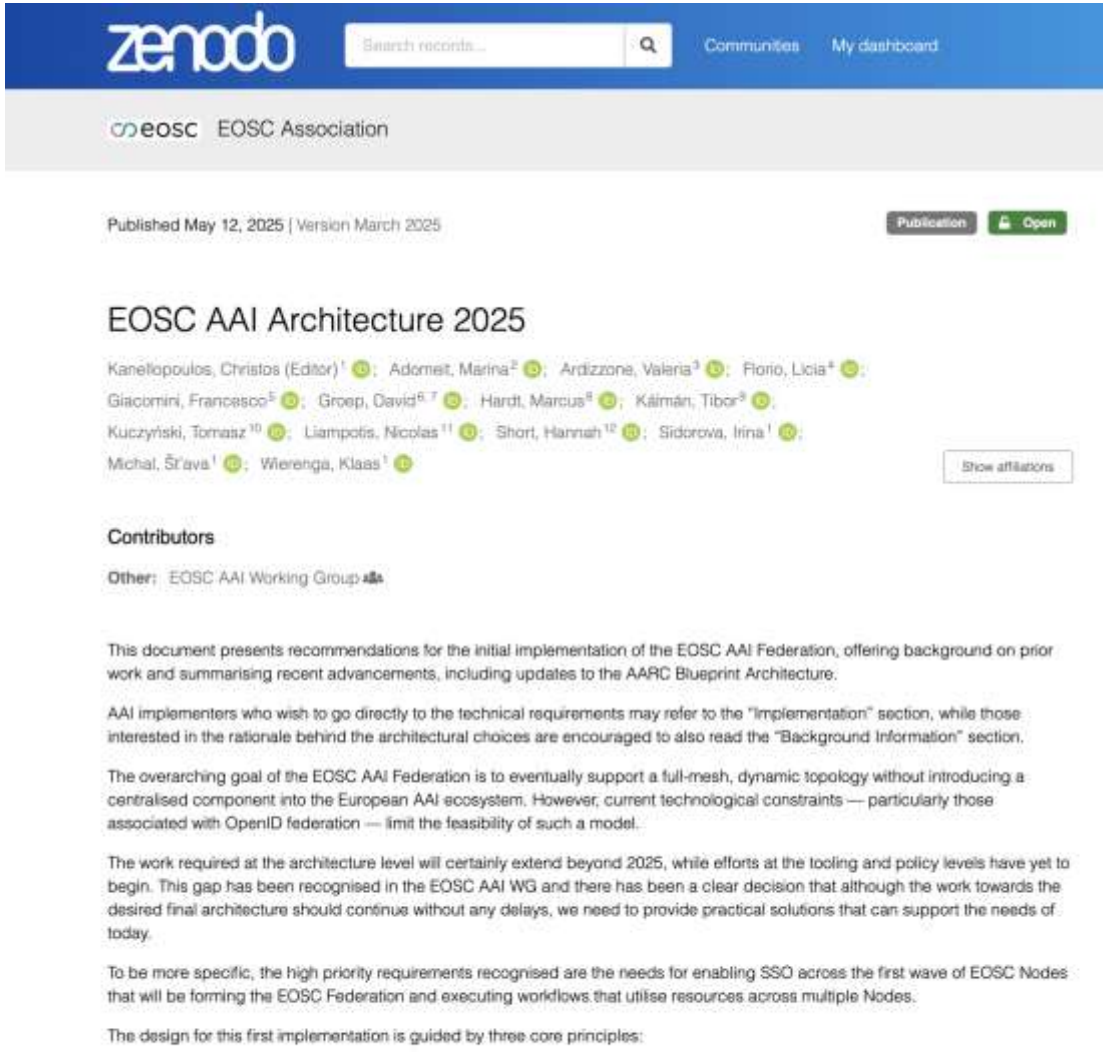
   *In future versions of this document, support for **OAuth 2.0 Protected Resource Metadata**[17], a metadata format enabling OAuth 2.0 clients and authorization servers to obtain information needed to interact with an OAuth 2.0 protected resource, will be also considered.*

[16] https://openid.net/specs/openid-connect-discovery-1_0.html
[17] https://datatracker.ietf.org/doc/draft-ietf-oauth-resource-metadata/

3. **MUST** support the **Authorisation Code Grant**[18] **with PKCE**[19]
   - OIDC Providers utilizing the Authorisation Code grant **MUST** support Proof Key for Code Exchange (PKCE). PKCE helps detect and prevent injection or replay of authorisation codes into the authorisation response.
   - Challenges **MUST** be transaction-specific and securely bound to the user agent where the transaction started.
4. **MUST use the nonce Parameter**
   - OIDC Providers **MUST** support the use of the nonce parameter (as specified in [OpenID Connect Core][OIDC-Core]) alongside the corresponding nonce claim in the ID Token.
   - This further mitigates replay attacks and ensures the integrity of the authentication process.
5. **MUST support Token Introspection**
   - **Compliance with OAuth 2.0 Token Introspection [RFC7662]**
     The Community AAIs as OpenID Connect Provider / OAuth 2.0 Authorisation Servers issuing OAuth 2.0 access tokens that are used across services in the "EOSC AAI Federation" **MUST** implement the OAuth 2.0 Token Introspection endpoint for central hub of the EOSC AAI Federation to verify the validity and active state of tokens they have issued.
   - **Scope and Policy Enforcement**
     By supporting token introspection, the Infrastructure Proxies of other EOSC Nodes, via the central hub of the EOSC Federation, can retrieve authorised scopes and user claims from the introspection responses.

[18] https://openid.net/specs/openid-connect-core-1_0.html#CodeFlowAuth
[19] https://www.rfc-editor.org/rfc/rfc7636

EOSC AAI Architecture - Next steps


zenodo
Search records... | Communities | My dashboard

eosc EOSC Association

Published May 12, 2025 | Version March 2025

Publication | Open

EOSC AAI Architecture 2025

Kanellopoulos, Christos (Editor)[1]; Adomeit, Marina[2]; Ardizzone, Valeria[3]; Florio, Licia[4]; Giacomini, Francesco[5]; Groep, David[6,7]; Hardt, Marcus[8]; Kálmán, Tibor[9]; Kuczyński, Tomasz[10]; Liampotis, Nicolas[11]; Short, Hannah[12]; Sidorova, Irina[1]; Michal, Šťava[1]; Wierenga, Klaas[1]

Show affiliations

Contributors

Other: EOSC AAI Working Group

This document presents recommendations for the initial implementation of the EOSC AAI Federation, offering background on prior work and summarising recent advancements, including updates to the AARC Blueprint Architecture.

AAI implementers who wish to go directly to the technical requirements may refer to the "Implementation" section, while those interested in the rationale behind the architectural choices are encouraged to also read the "Background Information" section.

The overarching goal of the EOSC AAI Federation is to eventually support a full-mesh, dynamic topology without introducing a centralised component into the European AAI ecosystem. However, current technological constraints — particularly those associated with OpenID federation — limit the feasibility of such a model.

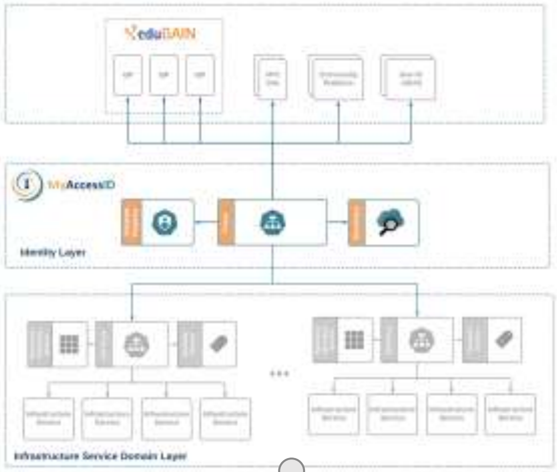
The work required at the architecture level will certainly extend beyond 2025, while efforts at the tooling and policy levels have yet to begin. This gap has been recognised in the EOSC AAI WG and there has been a clear decision that although the work towards the desired final architecture should continue without any delays, we need to provide practical solutions that can support the needs of today.

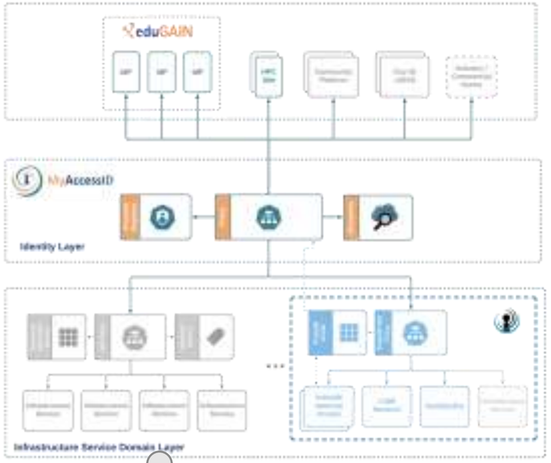
To be more specific, the high priority requirements recognised are the needs for enabling SSO across the first wave of EOSC Nodes that will be forming the EOSC Federation and executing workflows that utilise resources across multiple Nodes.

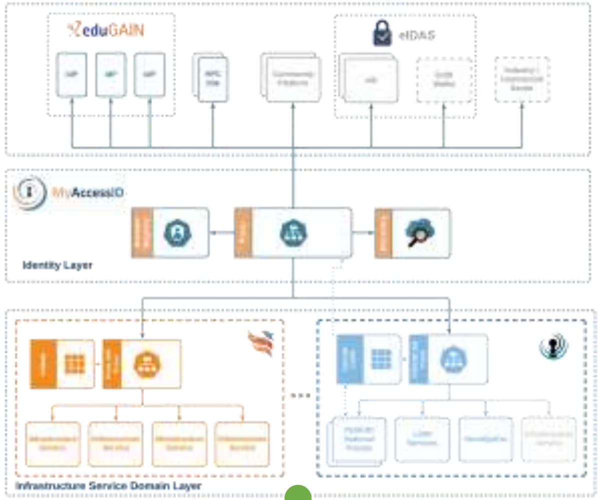
The design for this first implementation is guided by three core principles:

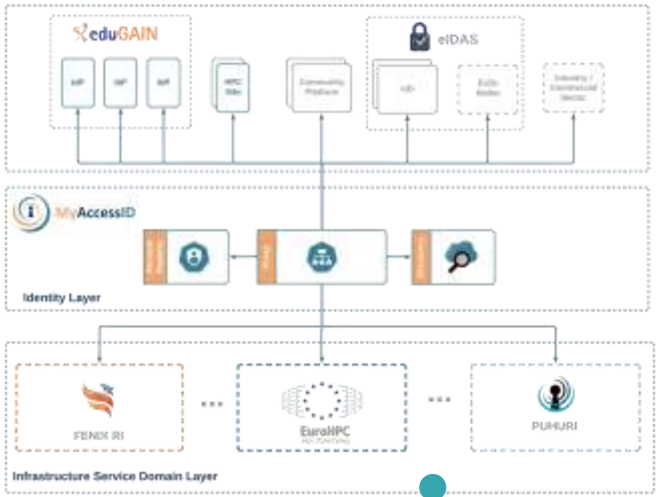
- *EOSC Federation Build-up Groups continues with the EOSC AAI WG*
- *Work with the candidate nodes on the feedback from the initial implementation*
- *Next update: end of 2025*
- *Continue contributions to the AARC BPA*

Timeplan









FENIX Design

MyAccessID & PUHURI Design

FENIX Implementation

FENIX Pilot

MyAccessID & PUHURI Pilot

MyAccessID & PUHURI Production

FENIX Production

Support for Advanced Use Cases

MyAccessID FENIX Production

Federated SSH CA PoC

Identity Vetting PoC

Federated SSH CA Pilot

EOSC EU Node

MFA - Strong Authentication EUDI Wallet Pilot

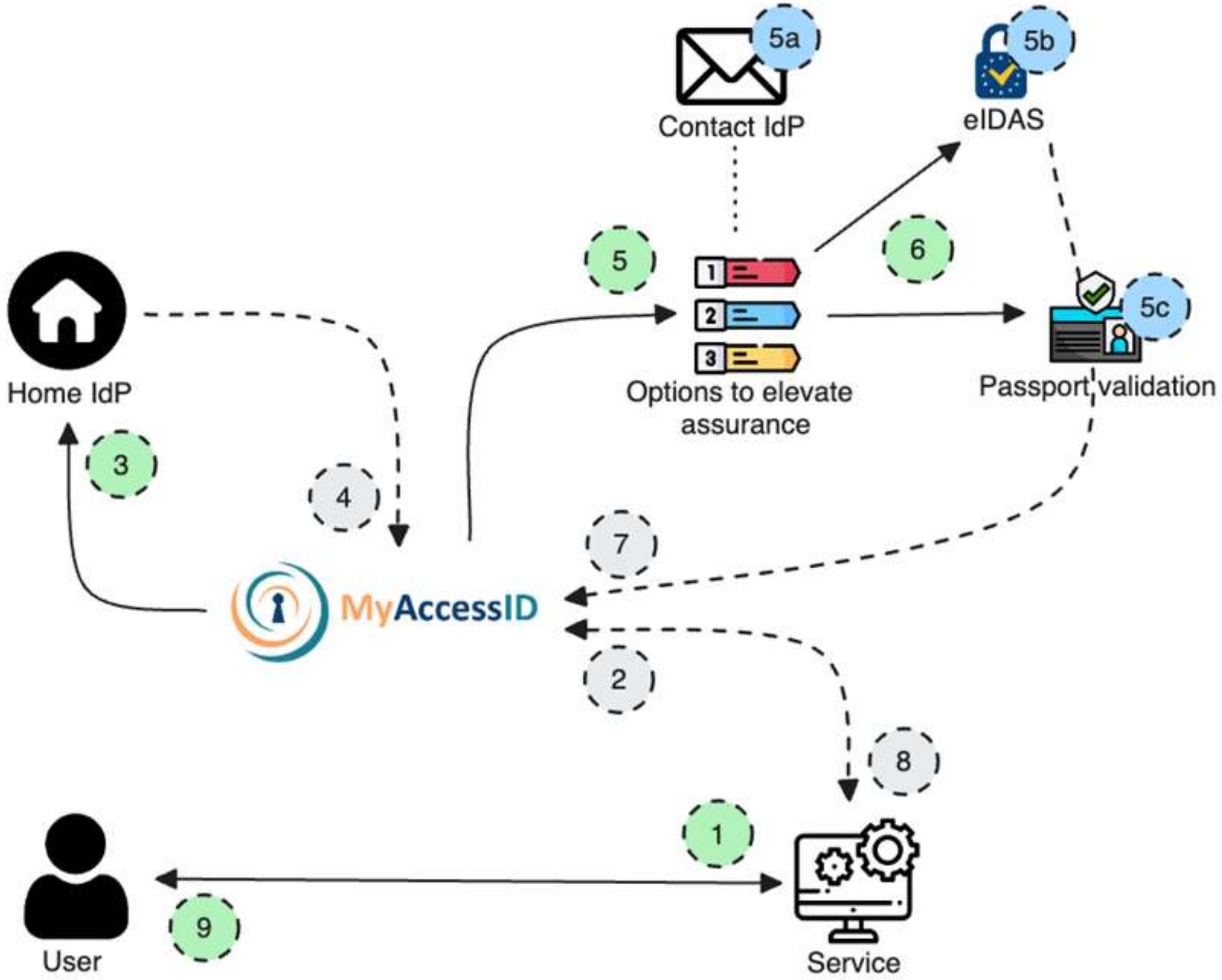
Identity Vetting Pilot

EOSC AAI Federation

EuroHPC Federated Platform

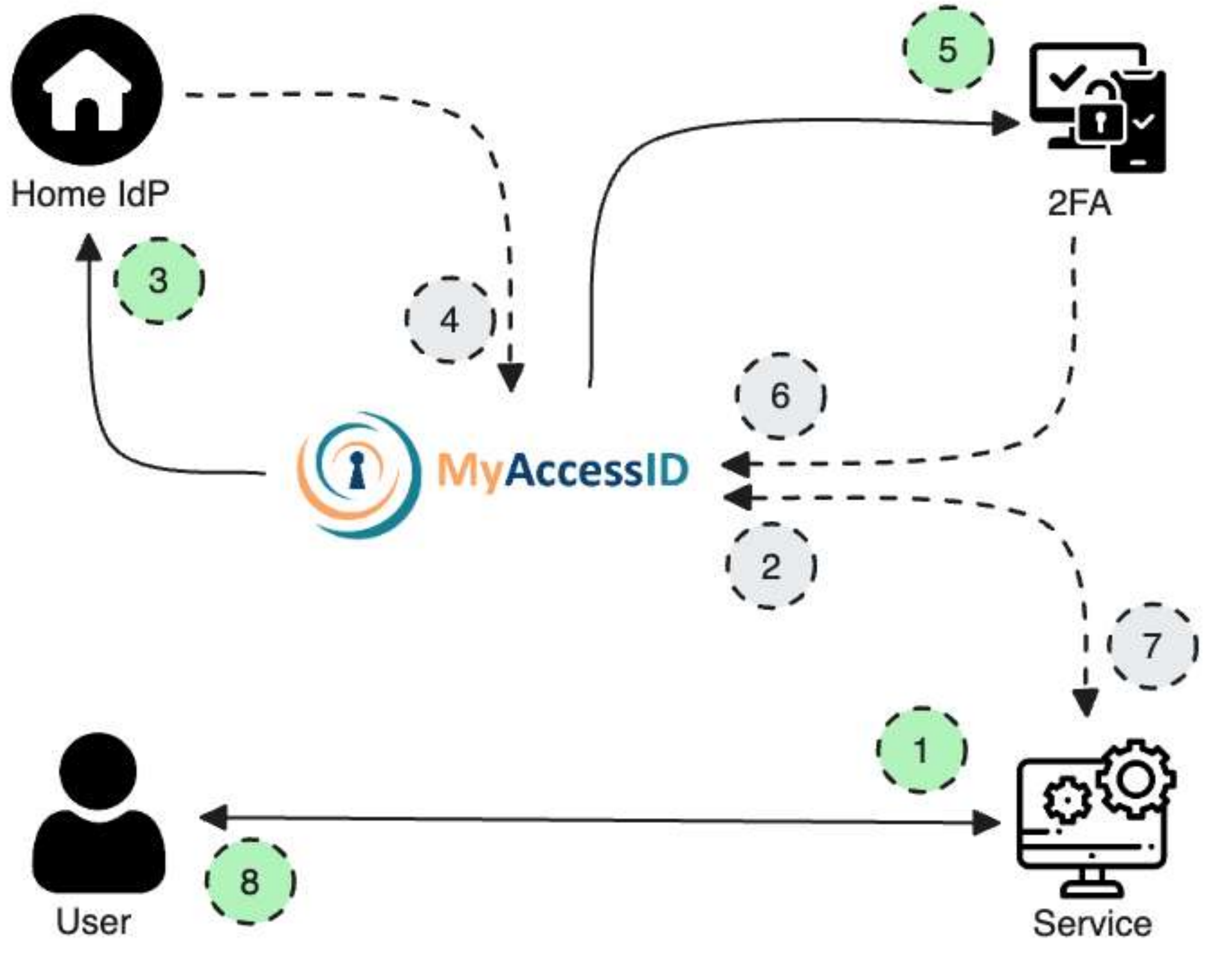
2018 | 2019 | 2020 | 2021 | 2022 | 2023 | 2024 | 2025 | 2026

Identity vetting



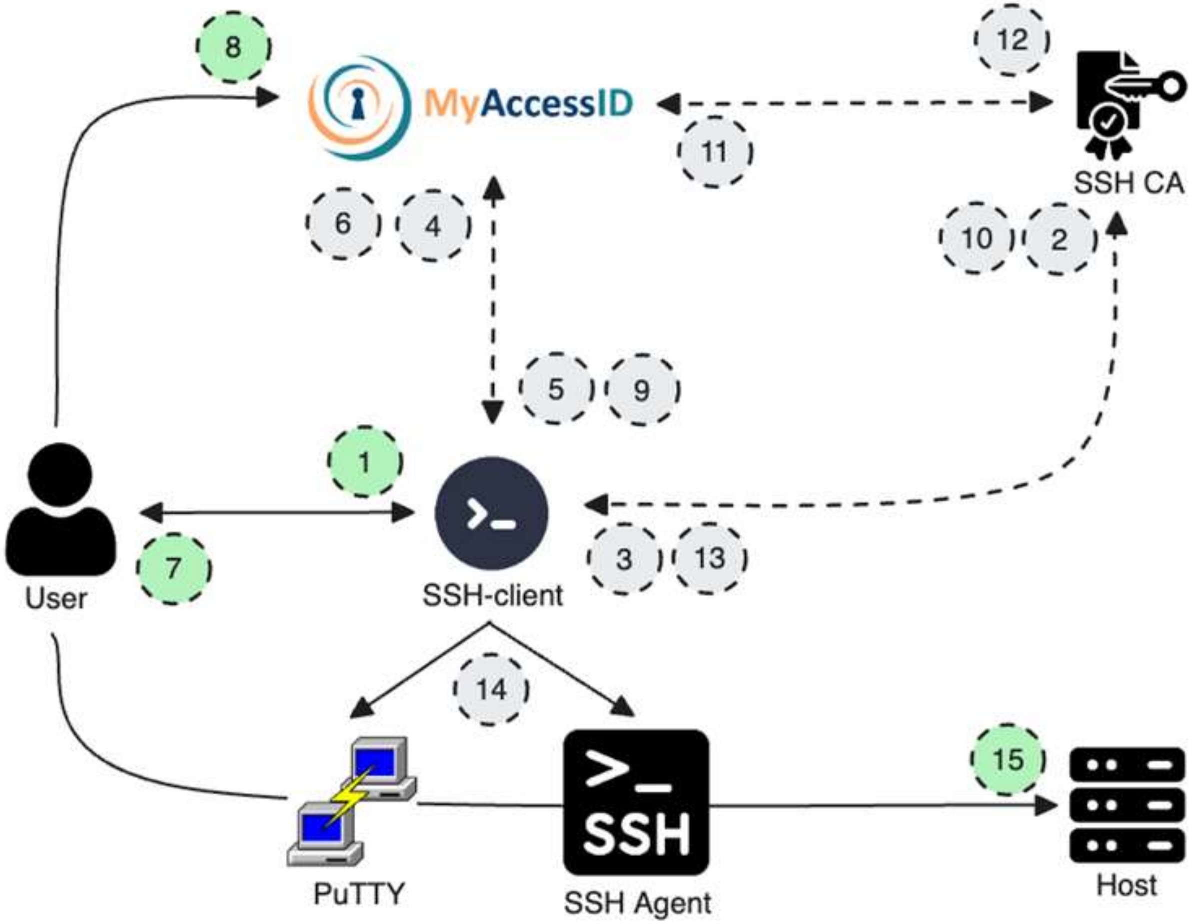
1. User accesses the target service
2. Service redirects the user to authenticate via MyAccessID expecting assured identity
3. MyAccessID lets the user to authenticate via user's home IdP
4. User authenticates to their home IdP with low assurance
5. MyAccessID redirects the user to options to elevate assurance of their identity
   a) contact the IdP to help with identity verificaiton
   b) use eIDAS
   c) use a governmental-issued certification
6. User chooses one mechanism and proceeds
7. MyAccessID verifies the assurance of the identity
8. MyAccessID provides assured identity information to the service
9. The target service lets user to access it

Multi-Factor Authentication



1. User accesses the target service
2. Service redirects the user to authenticate via MyAccessID requiring 2FA
3. MyAccessID lets the user to authenticate via user's home IdP
4. User authenticates to their home IdP without 2FA
5. MyAccessID checks if the 2FA has been provided so far and if not, it lets user to proceed with the 2FA through MyAccessID
6. MyAccessID verifies if 2FA was succesful
7. MyAccessID provides details about performed 2FA with the user's attributes to the service
8. The target service lets user to access it

Federated SSH Access



1. User uses the SSH-client
2. 3. SSH-client retrieves configuration details from the SSH CA
4. 5. SSH-client starts an OIDC device authorization flow
6. SSH-client starts polling MyAccessID waiting for user's authorization
7. SSH-client provides user with URL for authorization
8. User authenticates via MyAccessID and authorizes the operation
9. When authorized, MyAccessID polling returns access token to SSH-client
10. SSH-client provides SSH public key to SSH CA for signing it (send access token)
11. 12. SSH CA uses access token to get user's attributes
13. SSH CA provides a signed SSH certificate back to SSH-client
14. SSH-client sets up the environment (depends on the user's OS)
15. User accesses the host with the signed SSH certificate

GÉANT

Thank you!

May tanong?

Questions?

Вопросы?

¿Preguntas?

سوالات

კითხვები?

प्रश्न

Questões?

Vragen?

Ebibuuzo?

Maswali?

Frågor?

Kysymyksiä

有問題嗎？

Domande?

Ερωτήσεις?

Fragen?

christos.kanellopoulos@geant.org

www.geant.org