

SSH certificates

!=

X-509 certificates

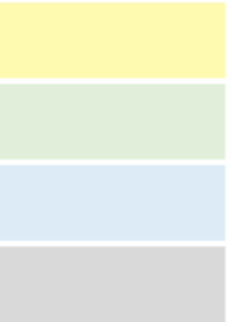
Scaling SSH access in EuroHPC by leveraging federated identity

Scale secure SSH access to resources by leveraging federated identity and SSH certificates in the EuroHPC Federation Platform

Mads Freek Petersen freek@wayf.dk



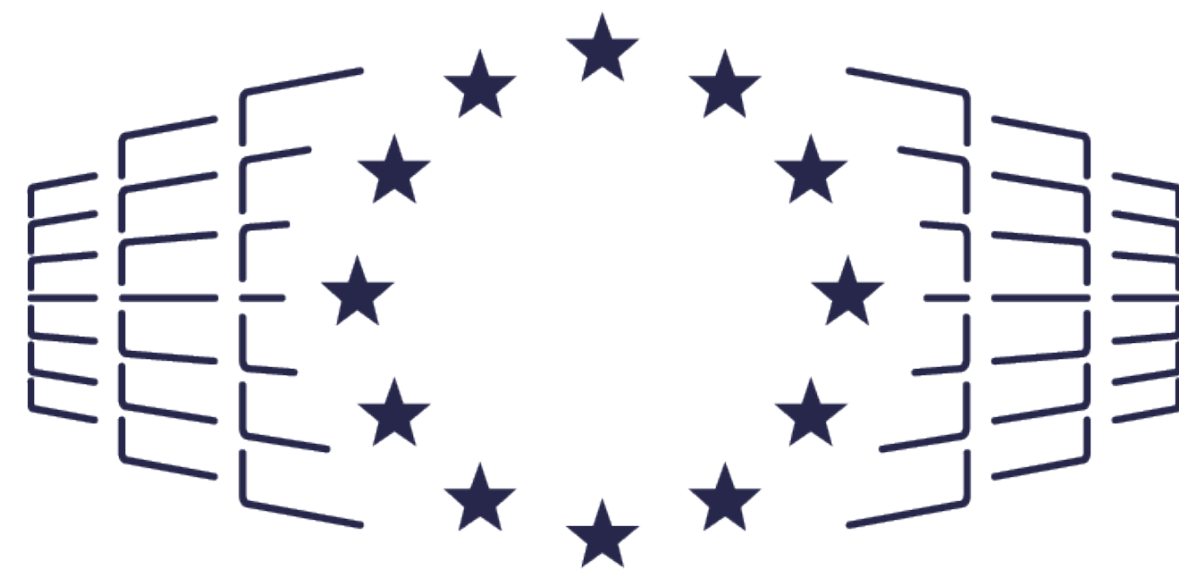
The EuroHPC Federation Platform (EFP)



"A one-stop-shop for accessing and utilizing EuroHPC systems and services"

A platform and federation framework which provides a **streamlined, single point of onboarding** and via which user can **access all systems and services** available to them in the EuroHPC ecosystem.

Direct access | Interactive interfaces | Programmatic APIs | Unified software stack

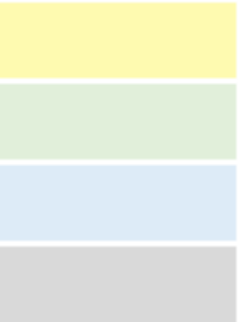


EuroHPC
Joint Undertaking

Federation Platform



EFP Main Features



Federated identity and Single-Sign-On (SSO)

Users utilize the same login and identity (e.g. granted via their home institution or national identity provider) to authenticate to all services and systems. Process for onboarding Industry.



Resource allocation, management and monitoring across systems

Users can view and manage their allocations and project members across all systems using a single interface.



Interactive web-based usage

with e.g., remote desktop, command line, AI training monitoring and Jupiter notebooks. Ability to launch batch jobs and browse files on the supercomputers.



Federated software catalogue

providing a pre-installed pseudo-uniform software stack on all systems. Advanced discoverability features for available software.



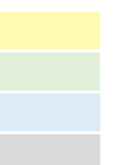
Advanced workflows and data transfer

Workflow execution and data transfers across systems, along with smart scheduling capabilities and high-level graphical interfaces for creating and managing workflows.



Direct access utilizing SSH certificates

Short lived certificates which are obtained via a login flow with Multi-Factor Authentication (MFA) for increased security.



EuroHPC Federation Platform (EFP)



Federation Platform

MyEFP

EFP
AAI



EFP
Allocations



EFP
Interactive



EFP
Workflows



EFP
Reporting



EFP
Helpdesk



EFP
Software Listing



EFP Core

Hosting Entities

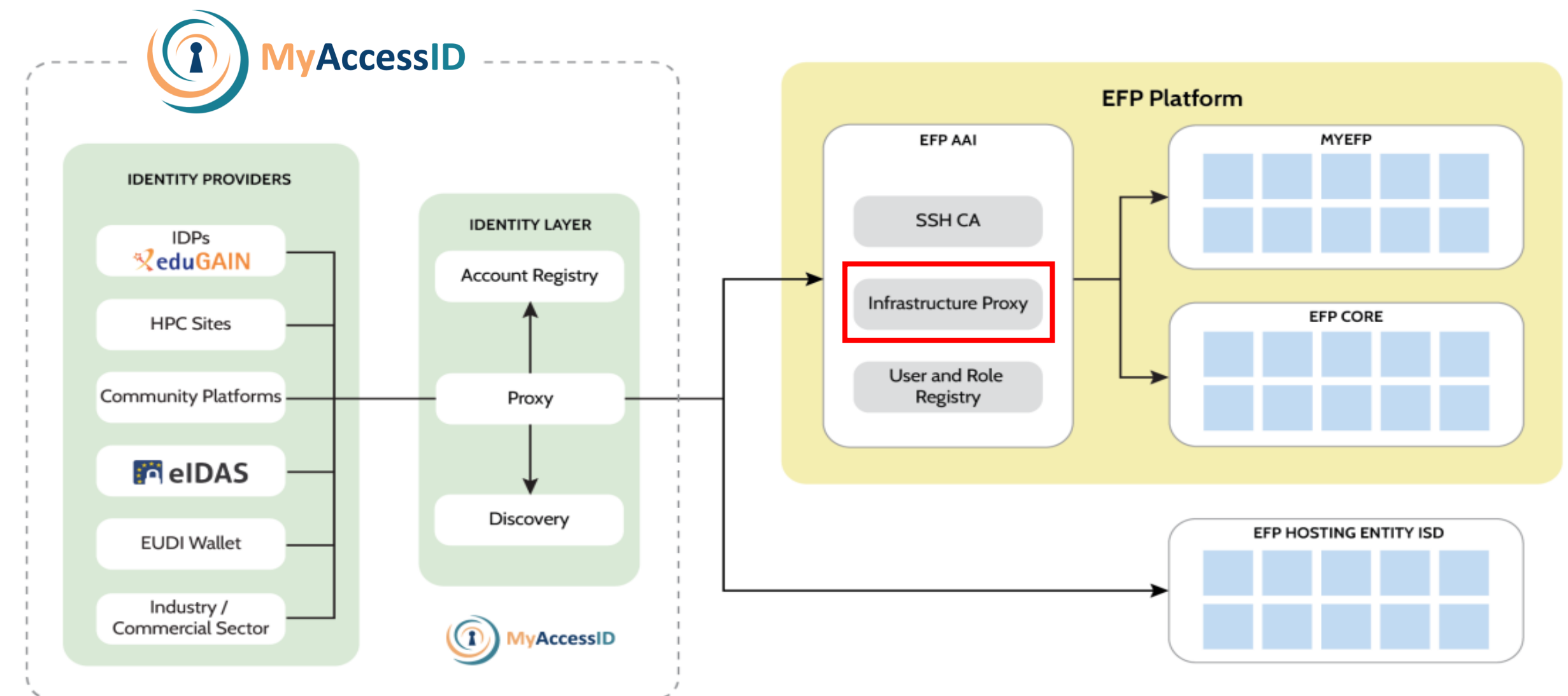
HPC

AI

Quantum

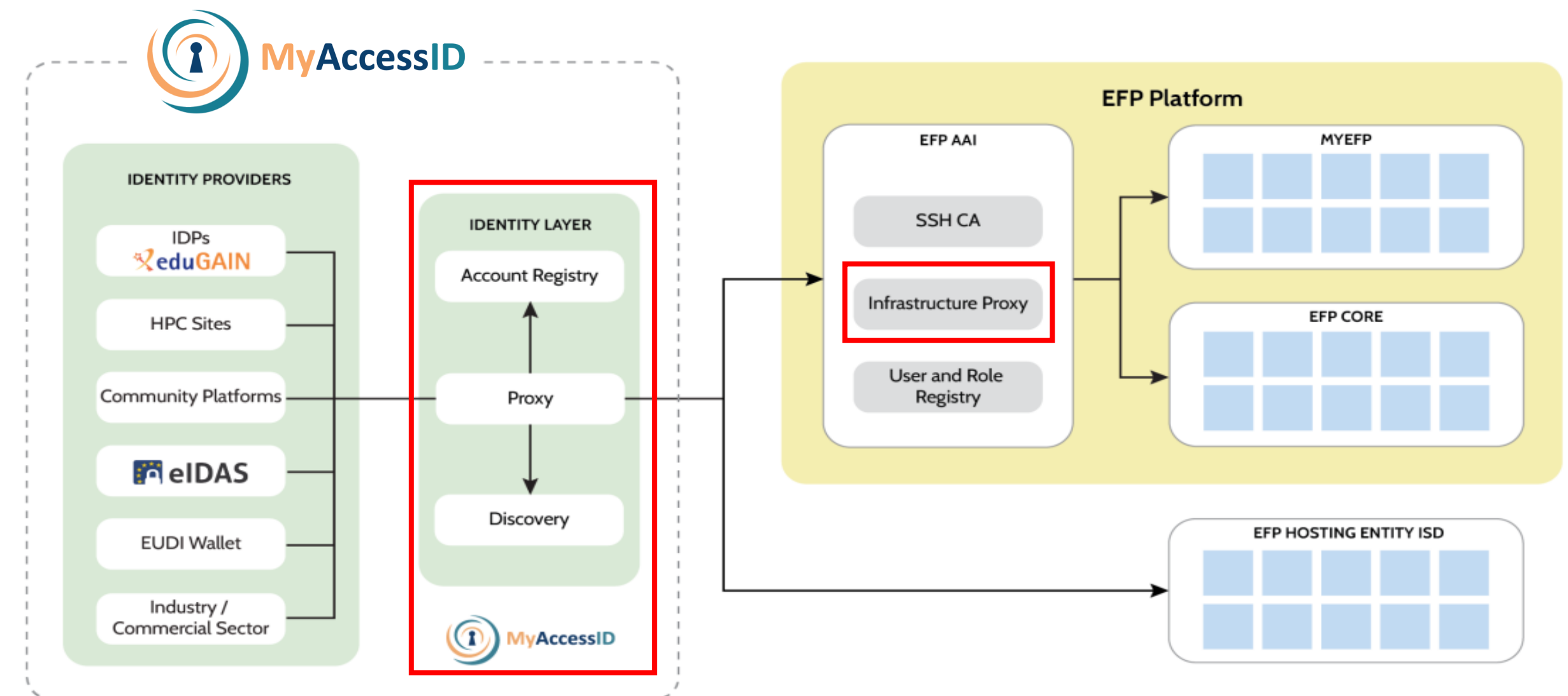
The key components of the EFP AAI

1. EFP Federation Infrastructure Proxy (FIP)
 - Based on GÉANT eduTEAMs, the EFP FIP connects every EFP service



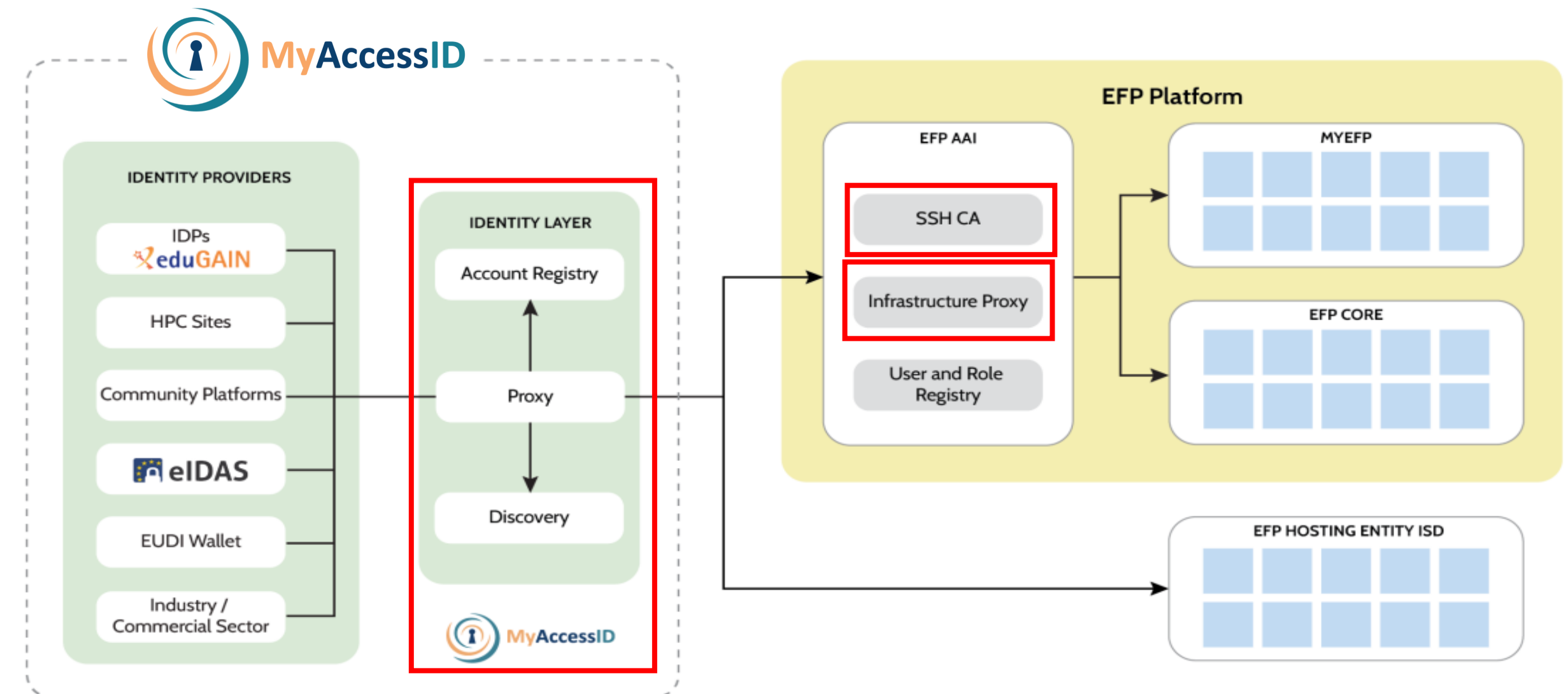
The key components of the EFP AAI

1. EFP Federation Infrastructure Proxy (FIP)
 - Based on GÉANT eduTEAMs, the EFP FIP connects every EFP service
2. MyAccessID
 - Provides the EFP FIP with end user identities



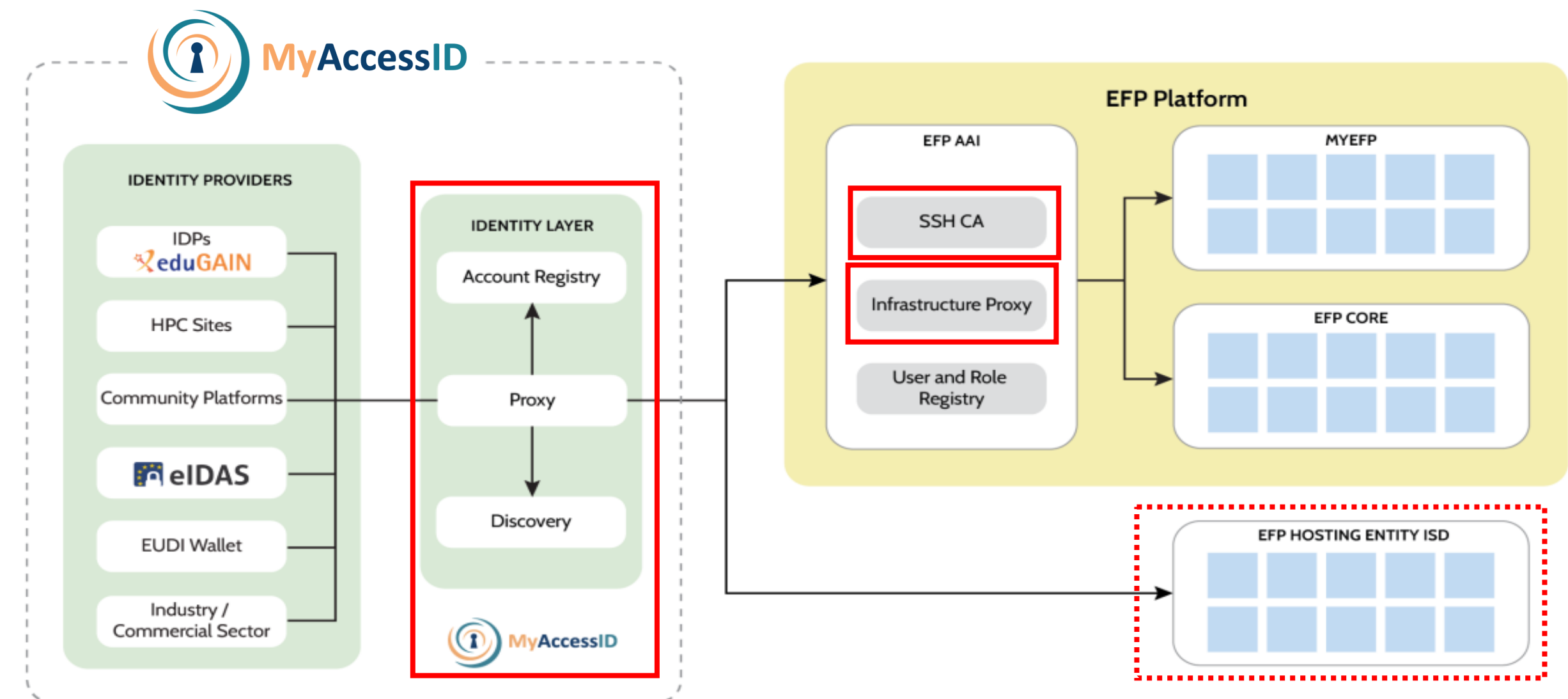
The key components of the EFP AAI

1. EFP Federation Infrastructure Proxy (FIP)
 - Based on GÉANT eduTEAMs, the EFP FIP connects every EFP service
2. MyAccessID
 - Provides the EFP FIP with end user identities
3. SSH Certificate Authority (CA)
 - Generates short-lived certificates for accessing EuroHPC Hosting Entities' systems



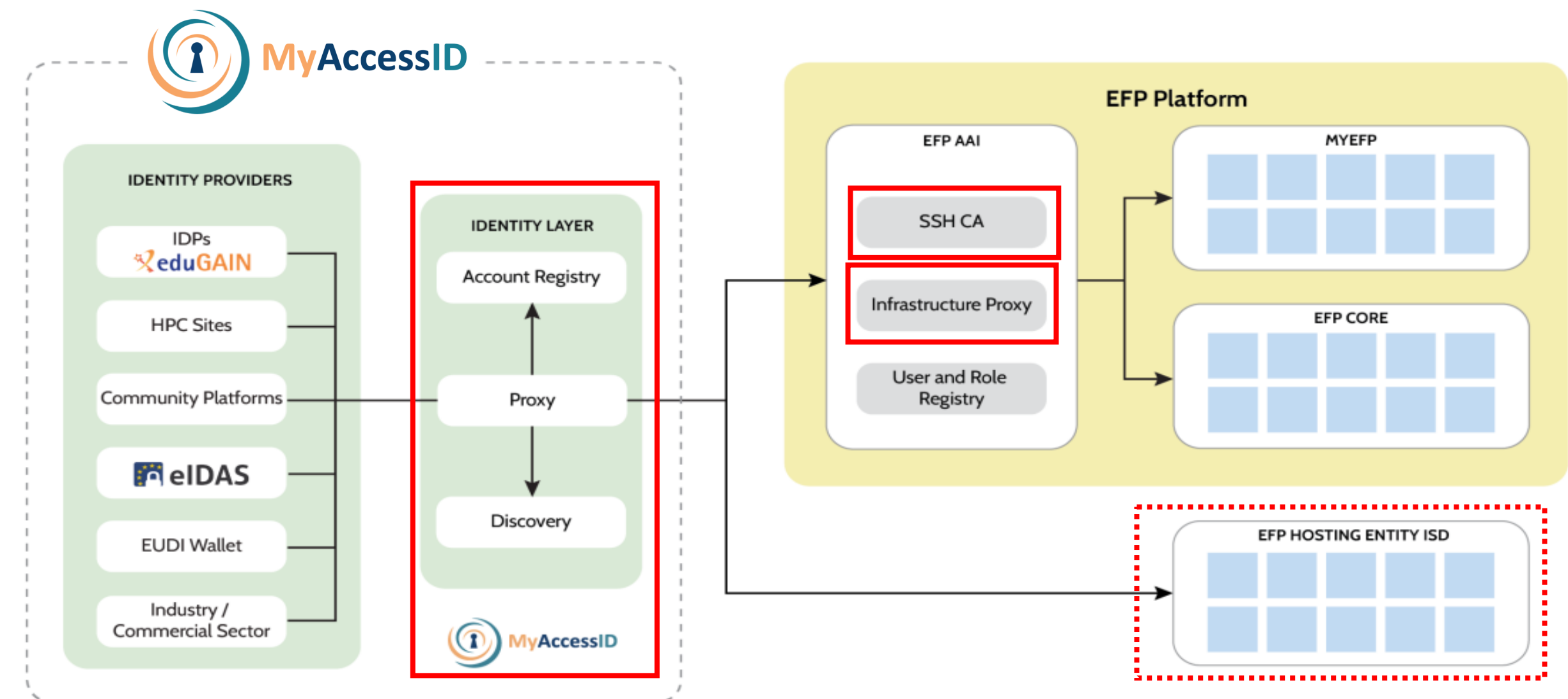
The key components of the EFP AAI

1. EFP Federation Infrastructure Proxy (FIP)
 - Based on GÉANT eduTEAMs, the EFP FIP connects every EFP service
2. MyAccessID
 - Provides the EFP FIP with end user identities
3. SSH Certificate Authority (CA)
 - Generates short-lived certificates for accessing EuroHPC Hosting Entities' systems
4. Hosting Entities' AAI
 - *Not part of the EFP AAI* but optionally Hosting Entities can also connect their own identity providers and clients to MyAccessID
 - This can add value in certain circumstances for Hosting Entities



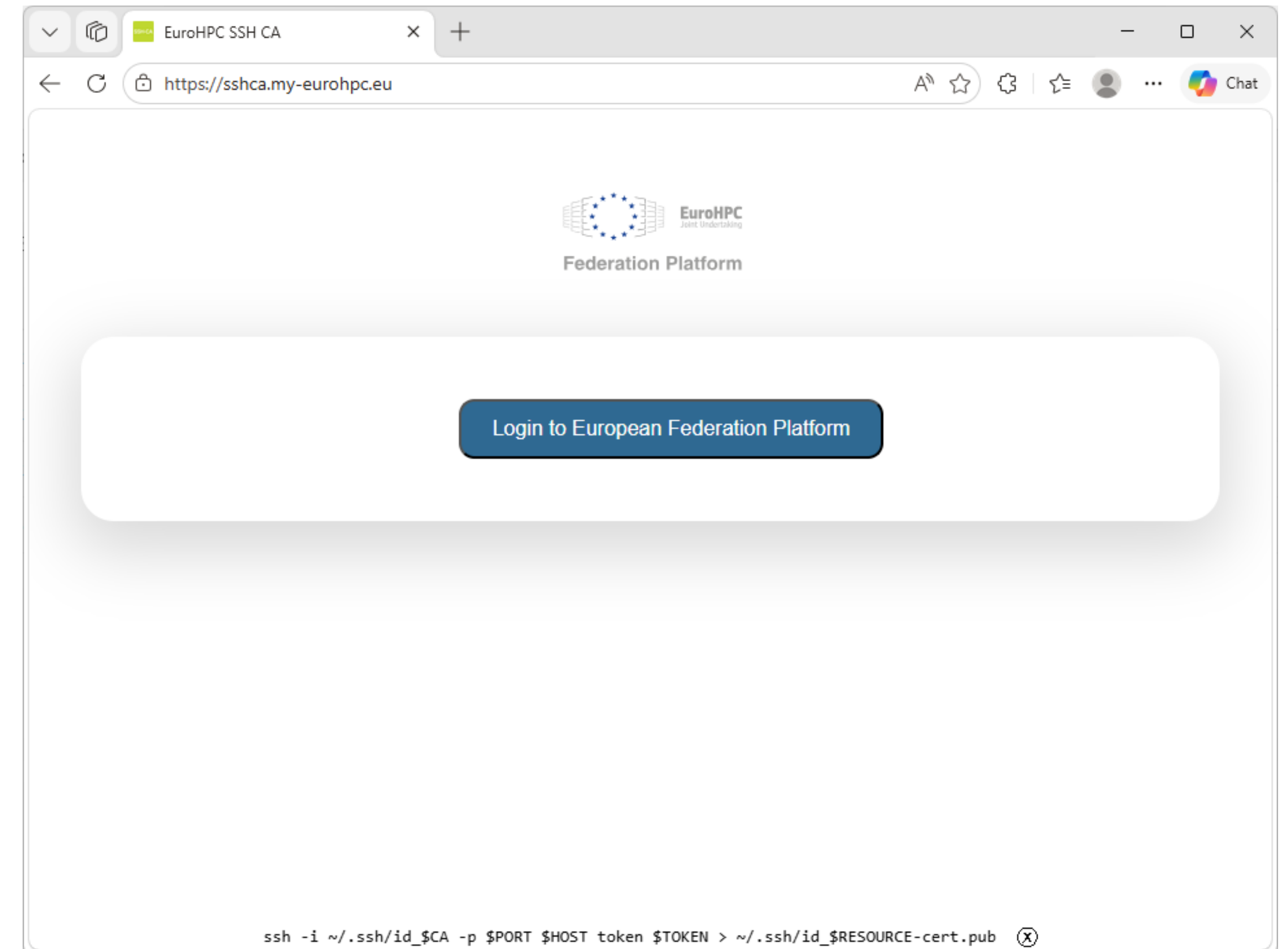
The key components of the EFP AAI

1. EFP Federation Infrastructure Proxy (FIP)
 - Based on GÉANT eduTEAMs, the EFP FIP connects every EFP service
2. MyAccessID
 - Provides the EFP FIP with end user identities
3. SSH Certificate Authority (CA)
 - Generates short-lived certificates for accessing EuroHPC Hosting Entities' systems
4. Hosting Entities' AAI
 - *Not part of the EFP AAI* but optionally Hosting Entities can also connect their own identity providers and clients to MyAccessID
 - This can add value in certain circumstances for Hosting Entities



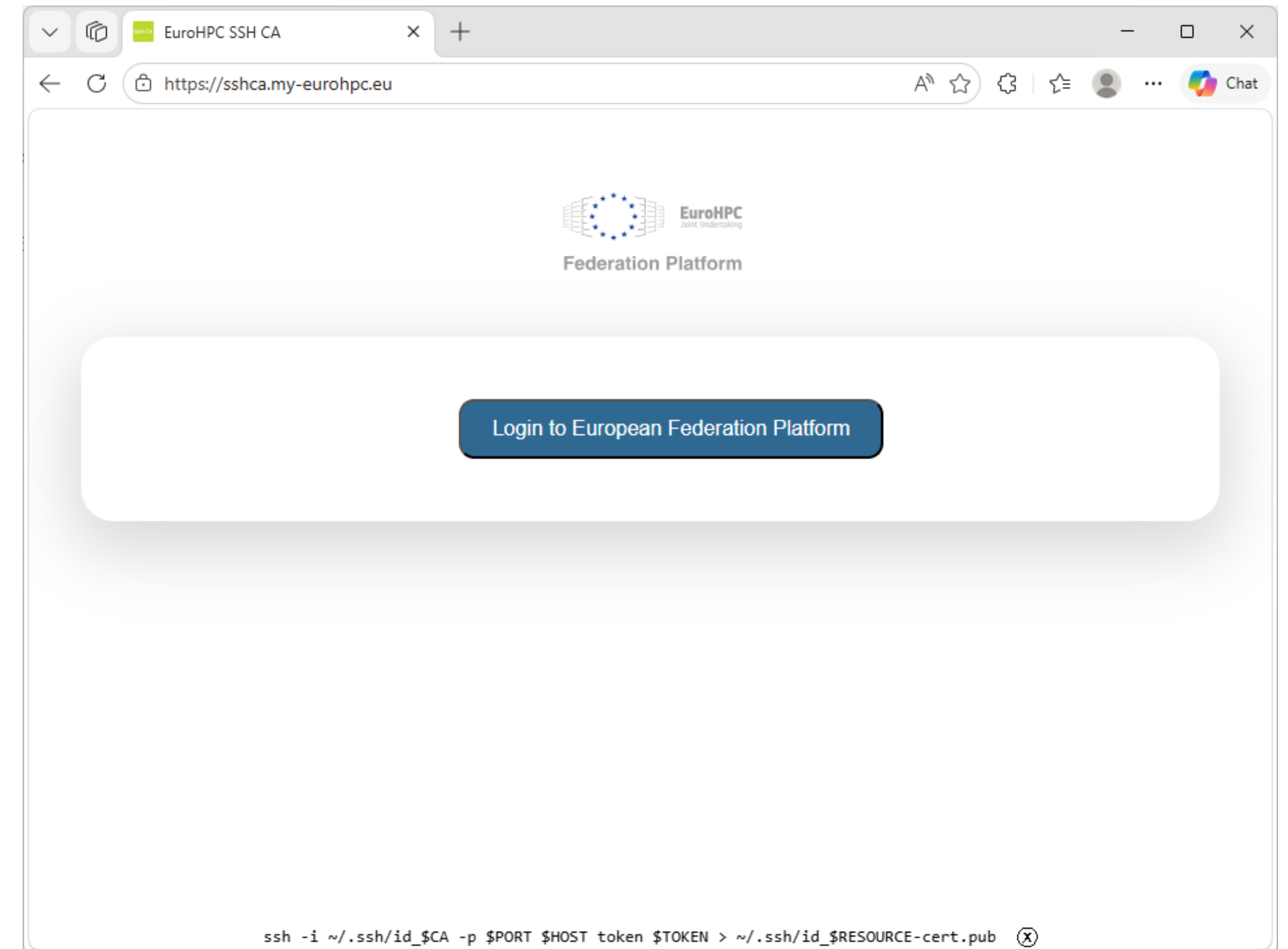
SSH Certificate Authority

- Using their MyAccessID account, the SSH CA enables users to access Hosting Entity's systems
- The process is secure and simple for end users and Hosting Entities
- Users are only offered certificates for resources that they are entitled to access



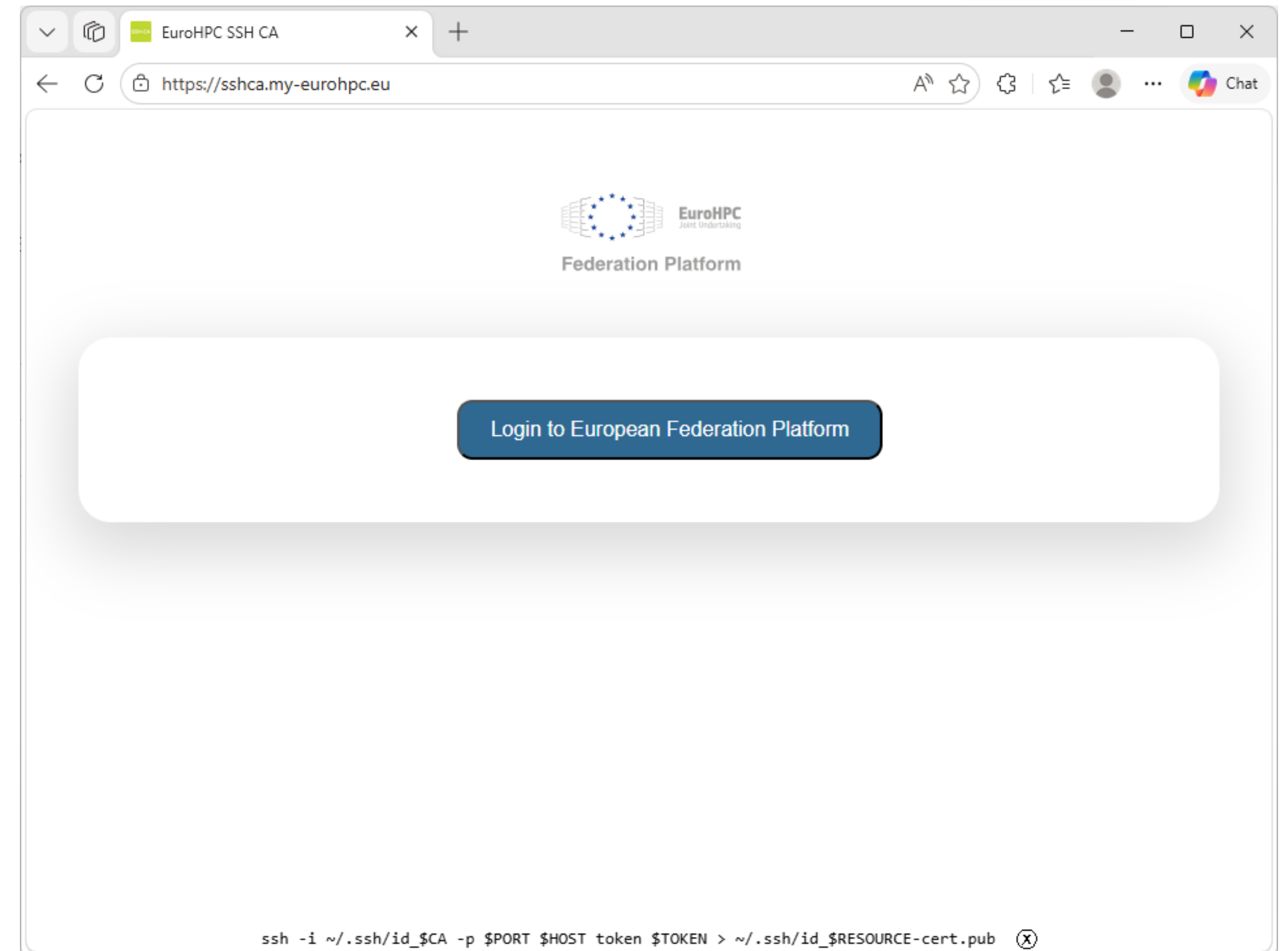
SSH Certificate Authority

- Using their MyAccessID account, the SSH CA enables users to access Hosting Entity's systems
- The process is secure and simple for end users and Hosting Entities
- Users are only offered certificates for resources that they are entitled to access
- Certificates are
 - provided to authorised users



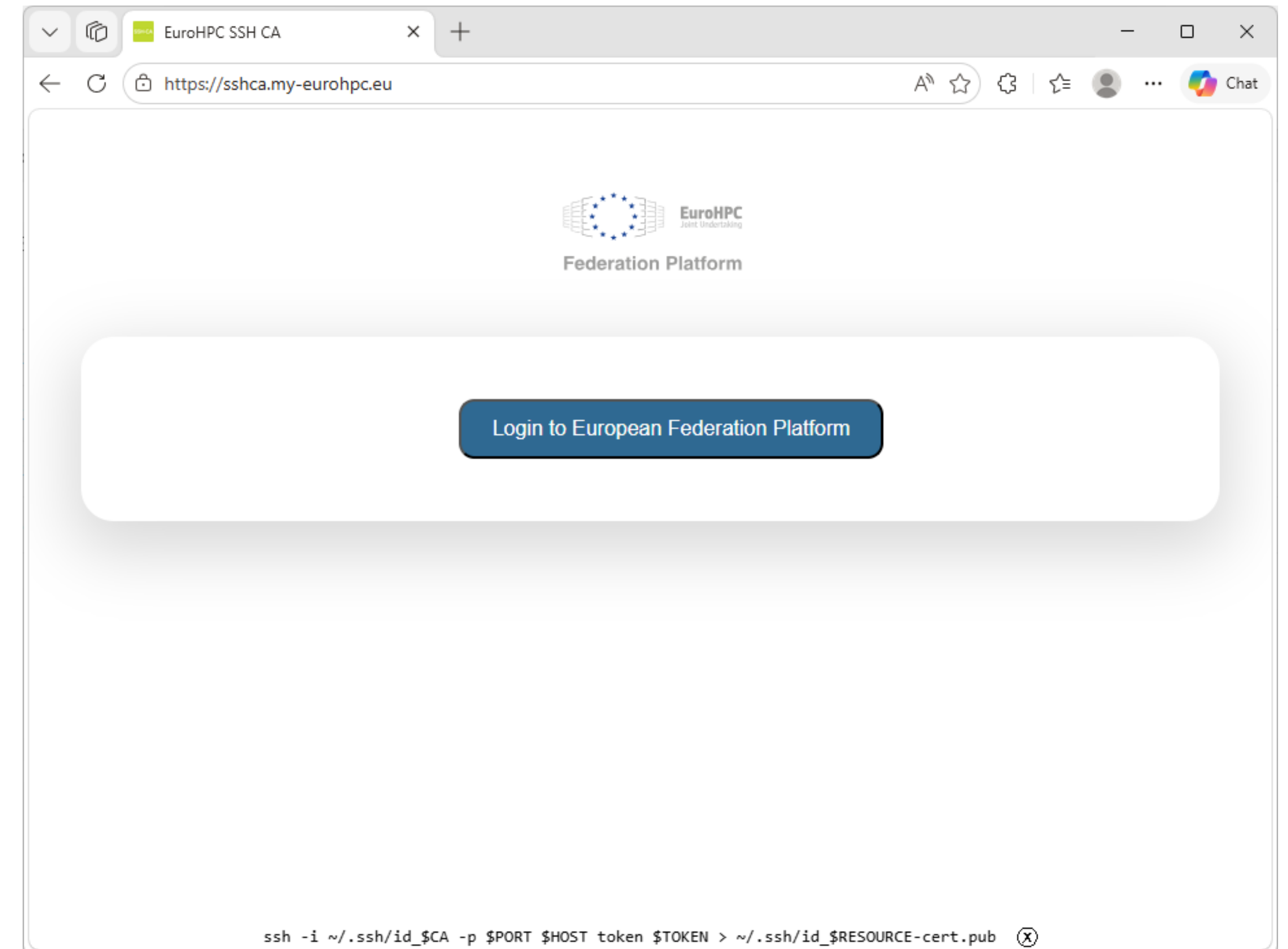
SSH Certificate Authority

- Using their MyAccessID account, the SSH CA enables users to access Hosting Entity's systems
- The process is secure and simple for end users and Hosting Entities
- Users are only offered certificates for resources that they are entitled to access
- Certificates are
 - provided to authorised users
 - scoped to each hosting entity



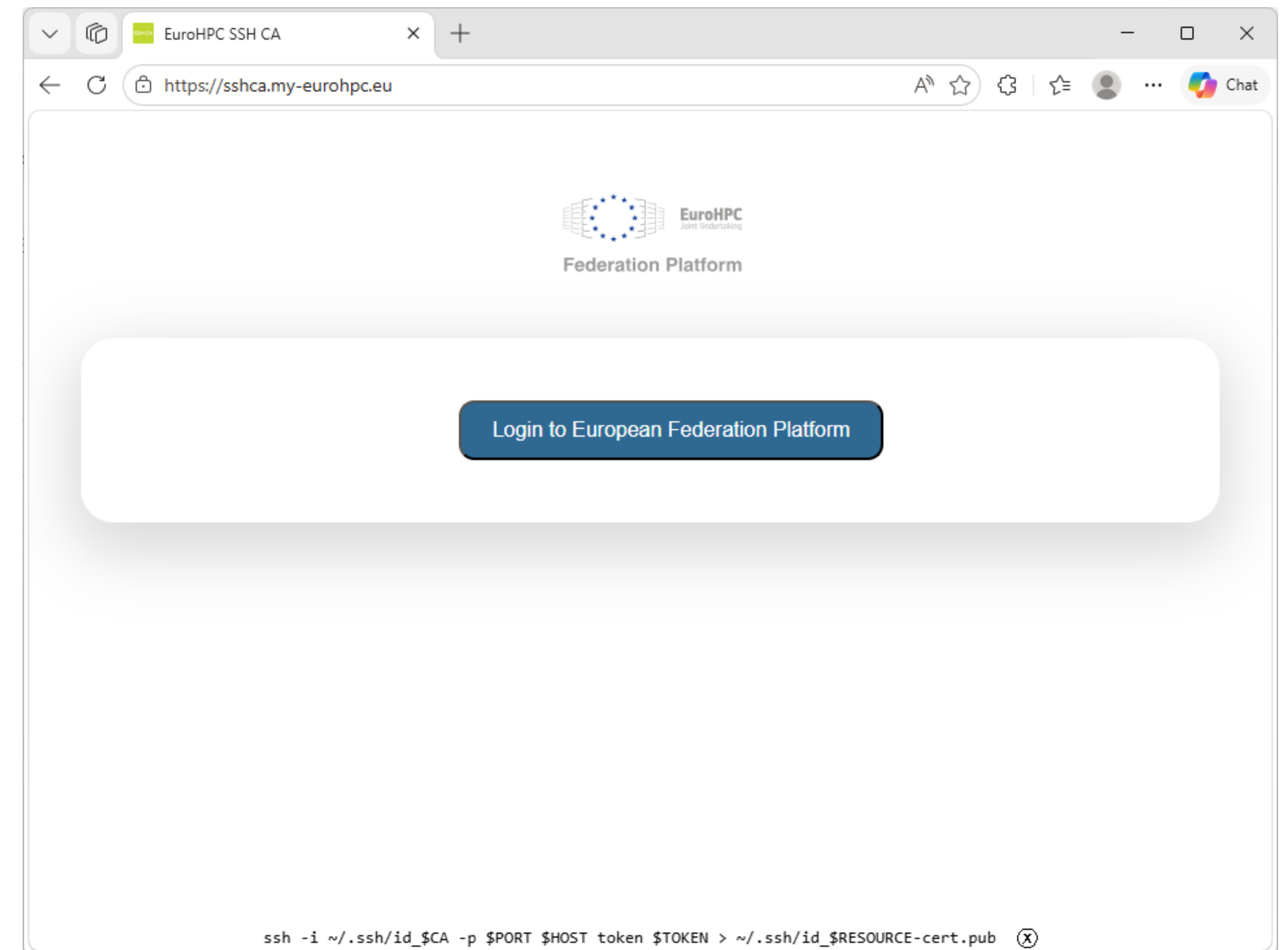
SSH Certificate Authority

- Using their MyAccessID account, the SSH CA enables users to access Hosting Entity's systems
- The process is secure and simple for end users and Hosting Entities
- Users are only offered certificates for resources that they are entitled to access
- Certificates are
 - provided to authorised users
 - scoped to each hosting entity
 - limited to a lifetime of one hour



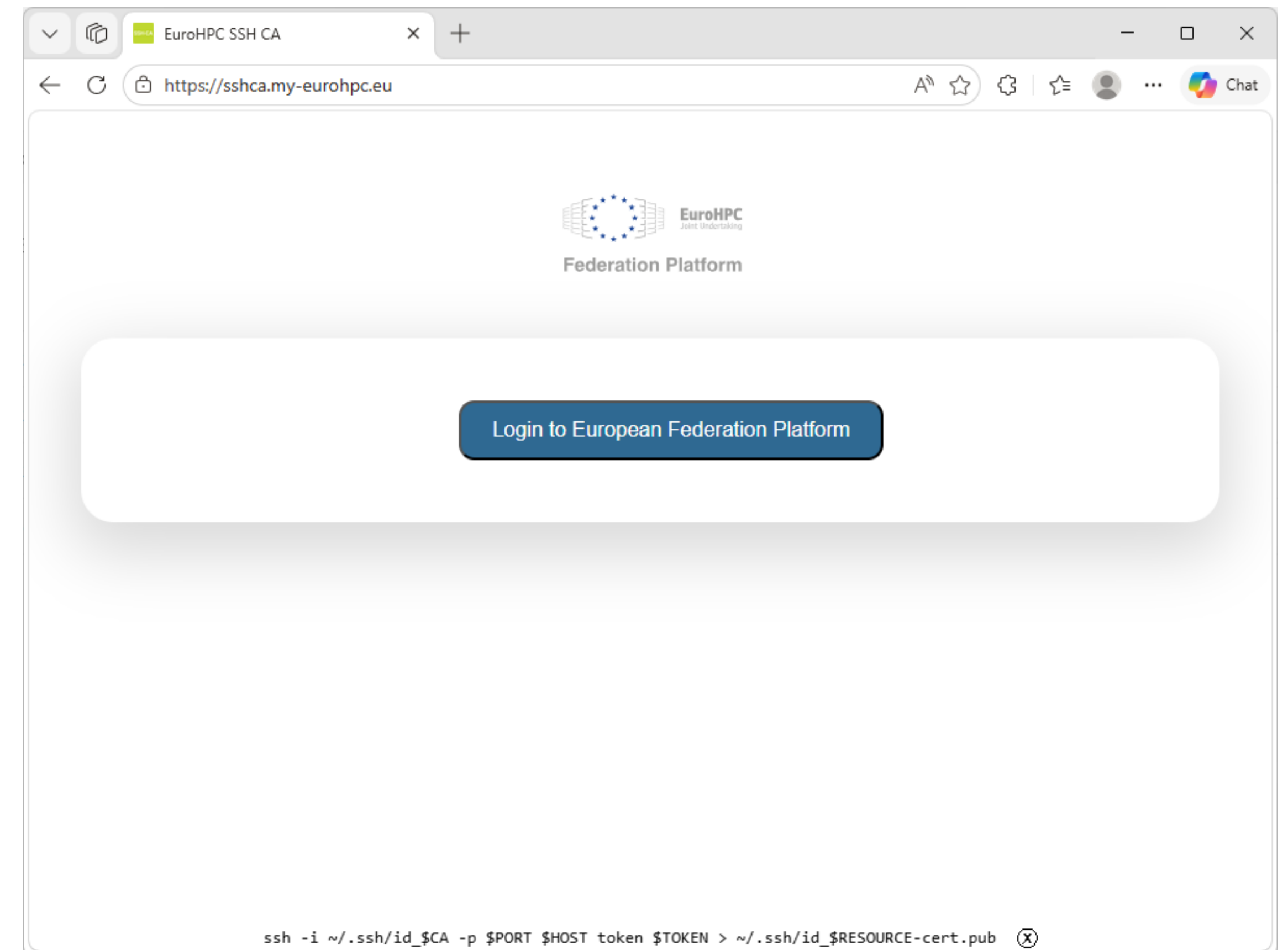
SSH Certificate Authority

- Using their MyAccessID account, the SSH CA enables users to access Hosting Entity's systems
- The process is secure and simple for end users and Hosting Entities
- Users are only offered certificates for resources that they are entitled to access
- Certificates are
 - provided to authorised users
 - scoped to each hosting entity
 - limited to a lifetime of one hour
 - and only issued after an MFA challenge



SSH Certificate Authority

- Using their MyAccessID account, the SSH CA enables users to access Hosting Entity's systems
- The process is secure and simple for end users and Hosting Entities
- Users are only offered certificates for resources that they are entitled to access
- Certificates are
 - provided to authorised users
 - scoped to each hosting entity
 - limited to a lifetime of one hour
 - and only issued after an MFA challenge
- The SSH CA is also used by other internal EFP components programmatically (workflows, etc)



So What are SSH Certificates?

And why is the DeiC's way of issuing them different?

Not new!

- djm@cvs.openbsd.org 2010/02/26 20:29:54

```
[PROTOCOL PROTOCOL.agent PROTOCOL.certkeys addrmatch.c auth-options.c]
[auth-options.h auth.h auth2-pubkey.c authfd.c dns.c dns.h hostfile.c]
[hostfile.h kex.h kexdhs.c kexgexs.c key.c key.h match.h monitor.c]
[myproposal.h servconf.c servconf.h ssh-add.c ssh-agent.c ssh-dss.c]
[ssh-keygen.1 ssh-keygen.c ssh-rsa.c ssh.1 ssh.c ssh2.h sshconnect.c]
[sshconnect2.c sshd.8 sshd.c sshd_config.5]
```

Add support for certificate key types for users and hosts.

OpenSSH certificate key types are not X.509 certificates, but a much simpler format that encodes a public key, identity information and some validity constraints and signs it with a CA key. CA keys are regular SSH keys. This certificate style avoids the attack surface of X.509 certificates and is very easy to deploy.

...

OpenSSH certificate key types are not X.509 certificates, but a much simpler format that encodes a public key, identity information and some validity constraints and signs it with a CA key. CA keys are regular SSH keys. This certificate style avoids the attack surface of X.509 certificates and is very easy to deploy.

Not Invented Here!

- SmallStep - smallstep.com (“If you’re not using SSH certificates you’re doing SSH wrong”)
- Teleport - goteleport.com
- HashiCorp Vault - www.vaultproject.io

SSH Certificates 101

- A SSH certificate is a structure which contains a public key and some additional information signed by a SSH CA encoded according to rfc4251
- 2 types - user and host
- Only 1 level - i.e. only “root” keys that signs certificates
- Additional information
 - Principals - user names or host domain names
 - Validity period
 - Critical options
 - Extensions

SSH Certificate Authority

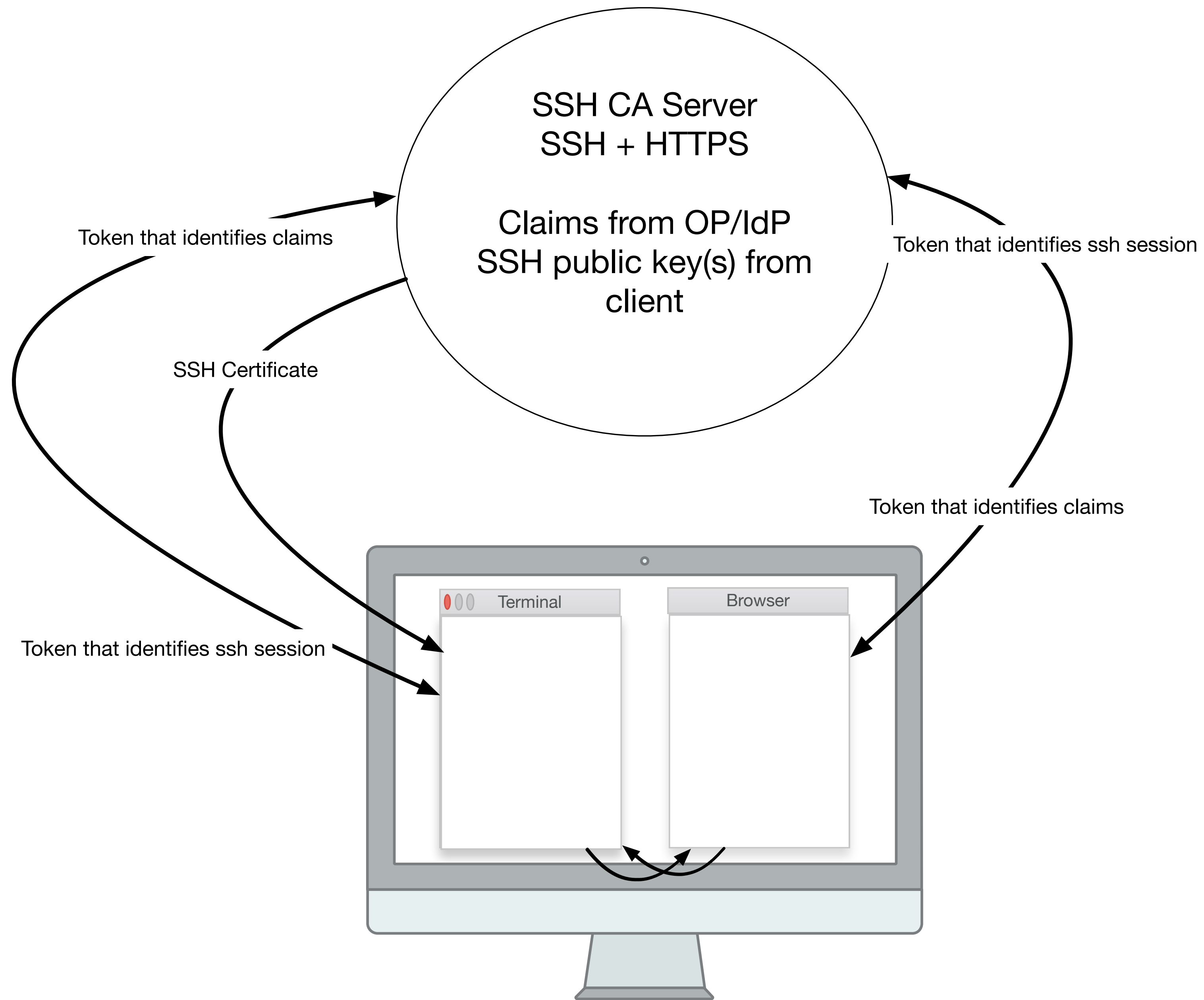
- A SSH Certificate Authority issue certificates based on
 - Knowledge of the user (principal / Key ID)
 - Policy (valid from - to)
 - Policy (extensions, critical options)
- The Deic SSH CA server is a go based http- and sshserver.

Why is the DeiC's way of issuing them different?

- Using SSH to “upload” public key

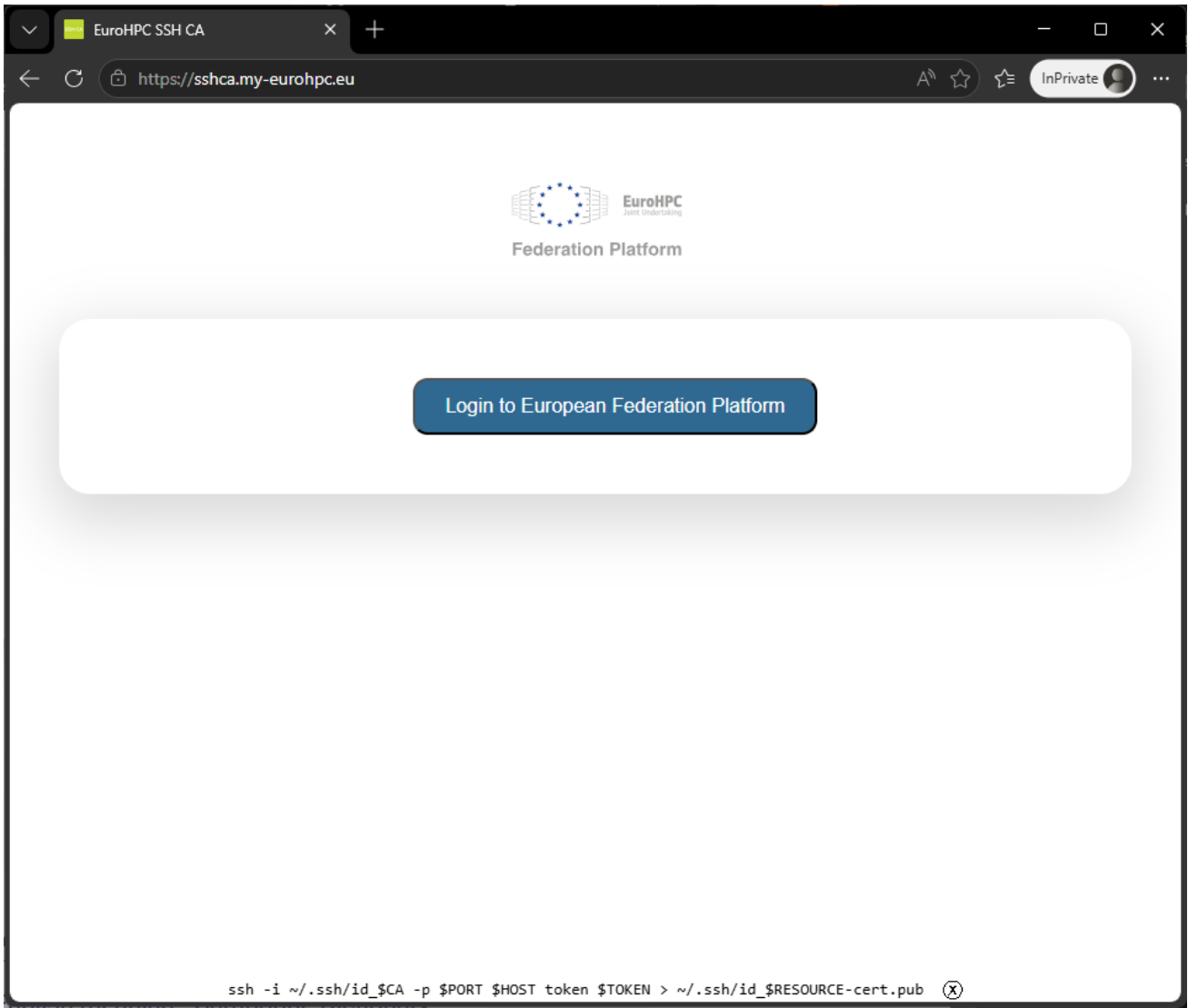
(but we support other methods - e.g. the API MyAccess uses)

- No extra code on client or server

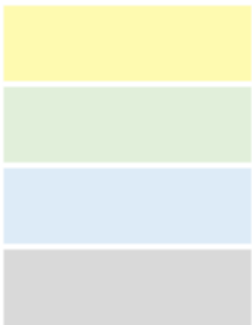


EFP SSH Certificate Authority

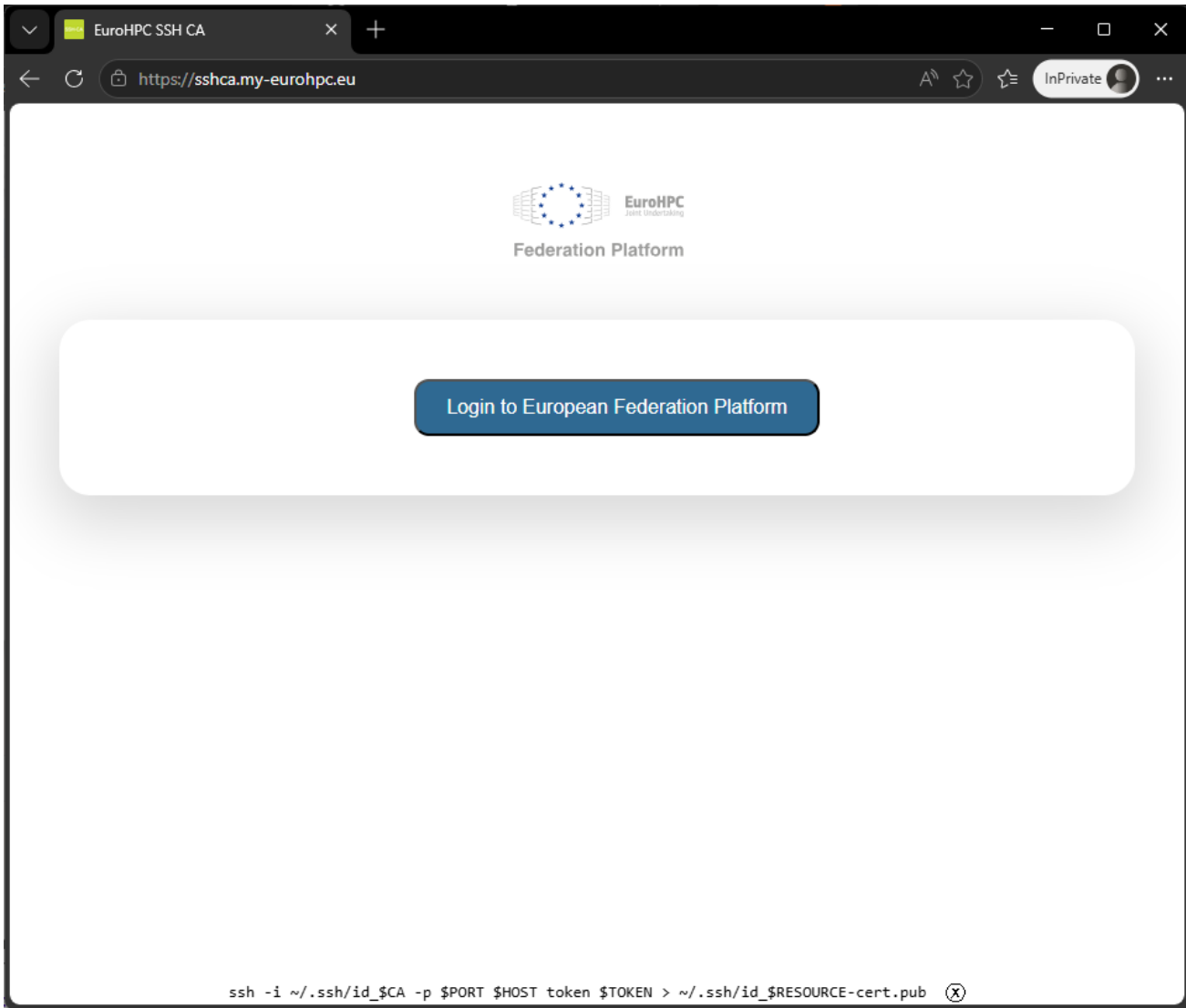
User browses to web SSH CA



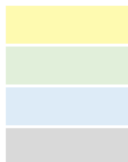
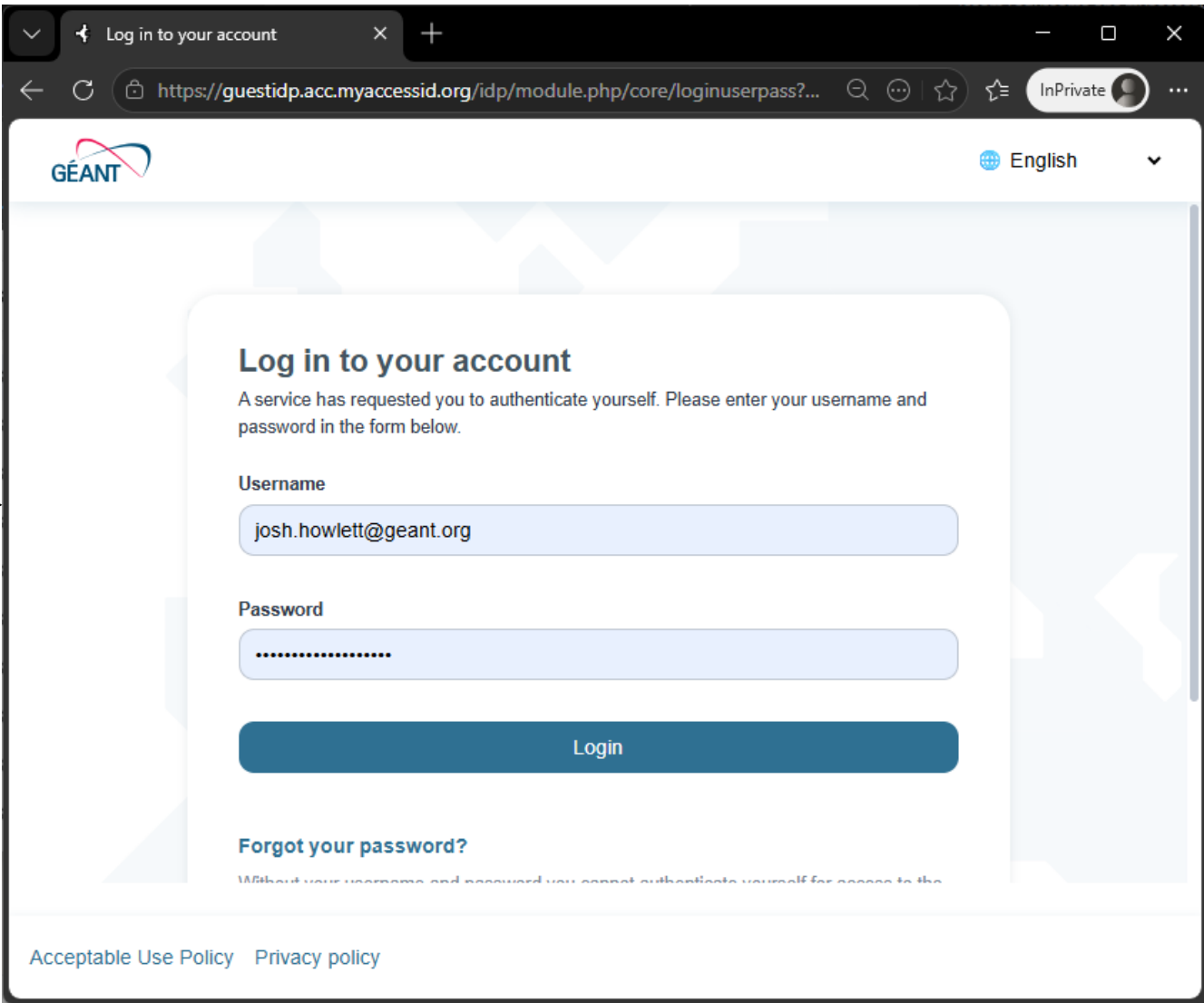
EFP SSH Certificate Authority



User browses to web SSH CA

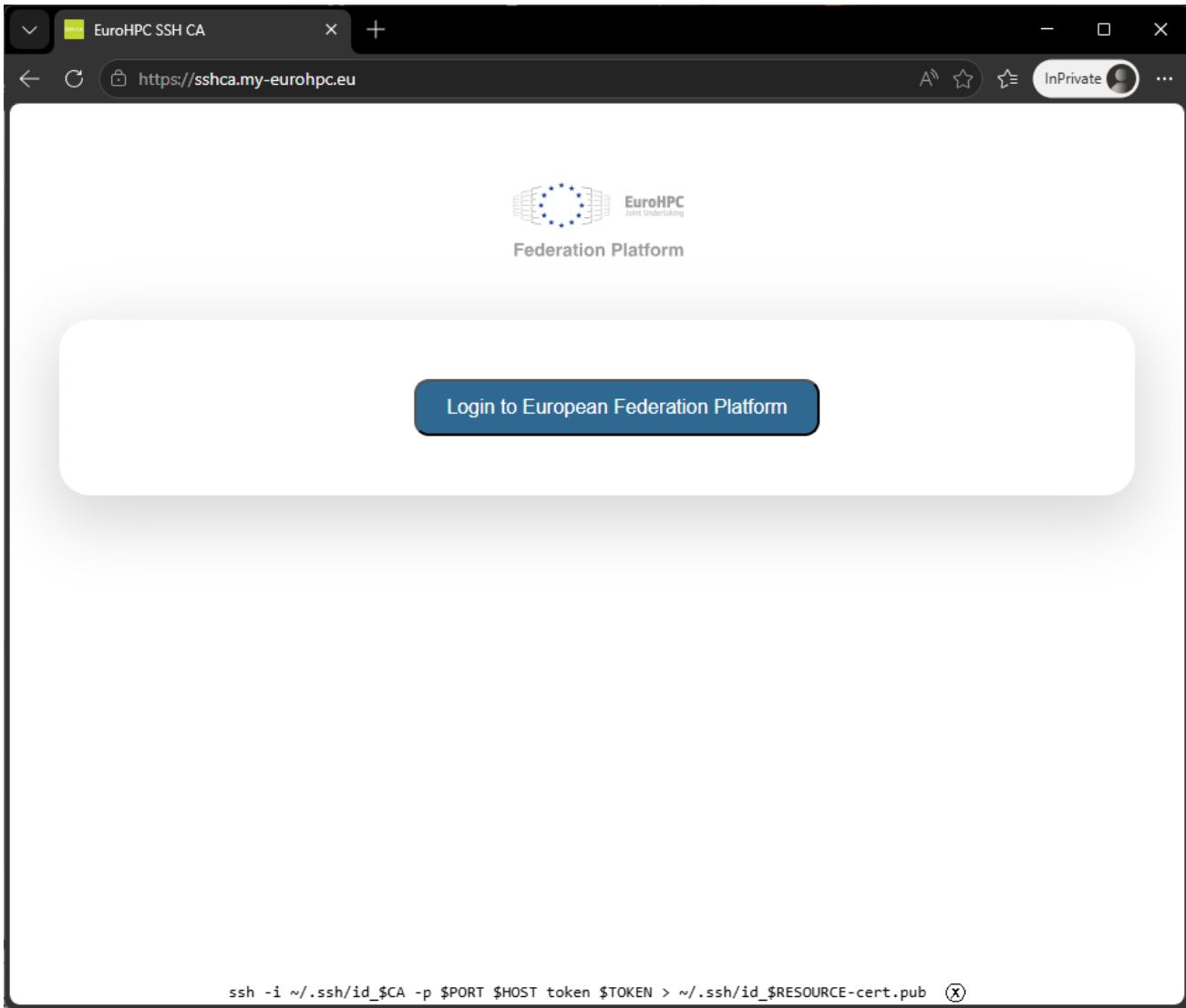


User authenticates against their identity provider

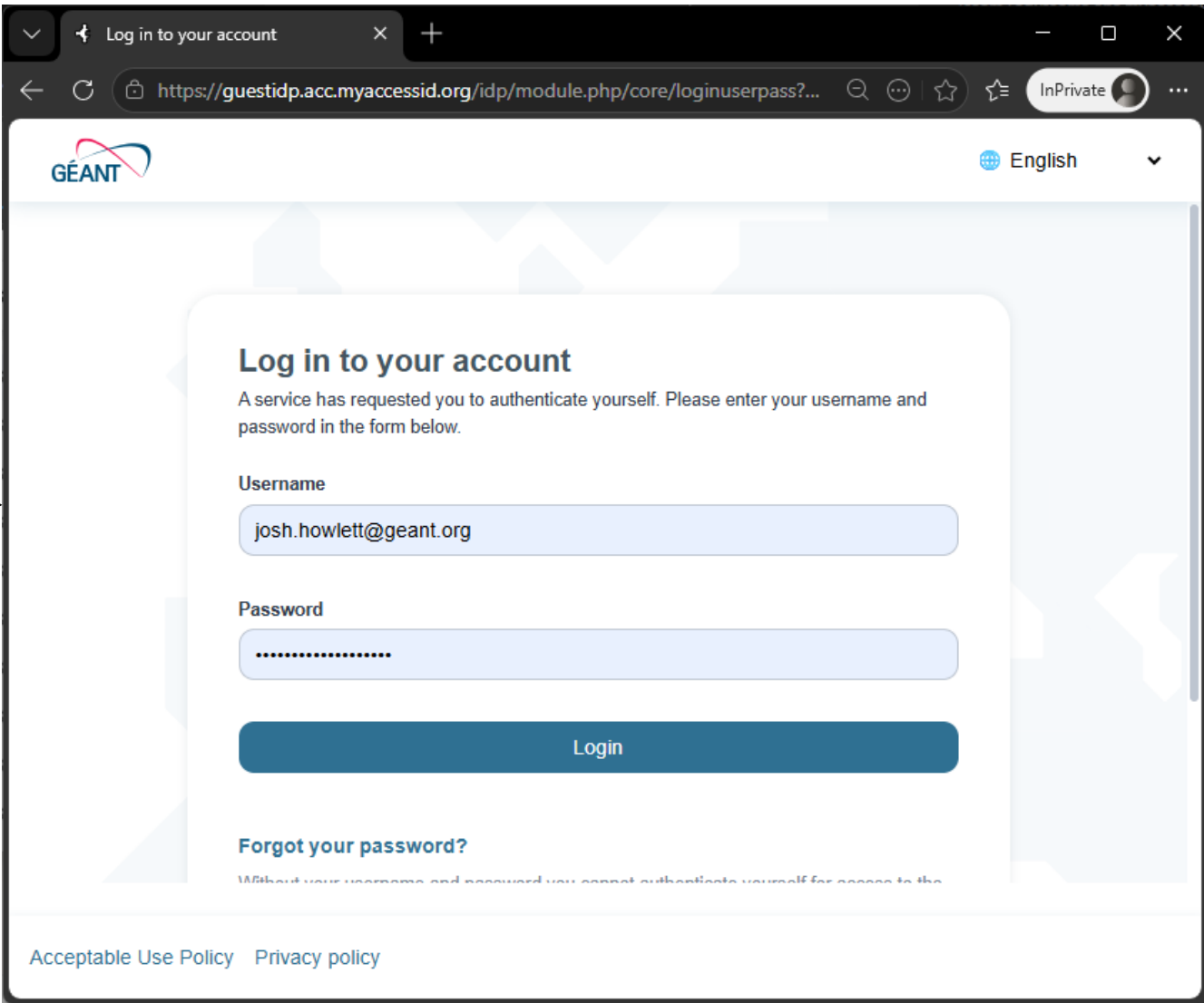


EFP SSH Certificate Authority

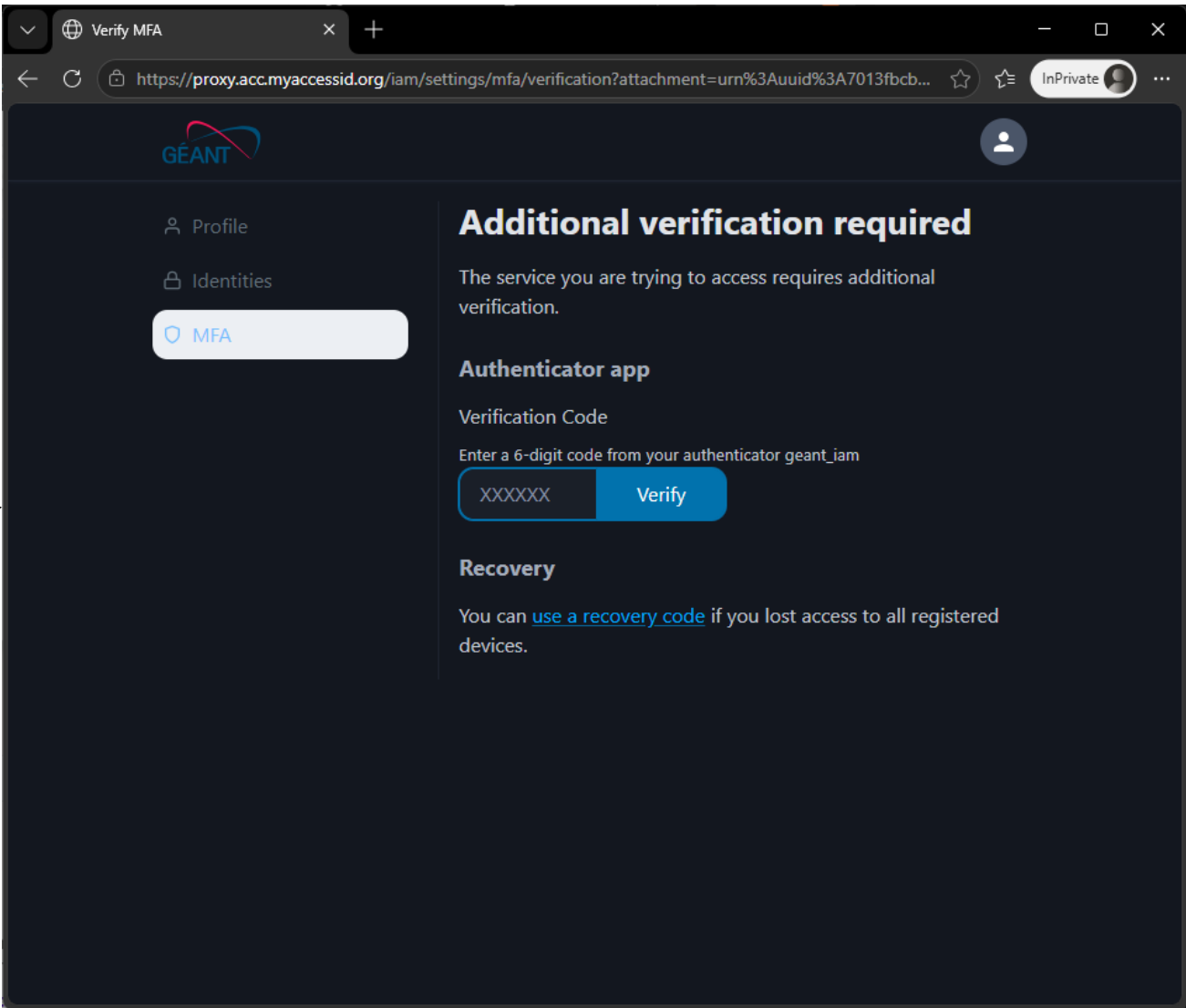
User browses to web SSH CA



User authenticates against their identity provider

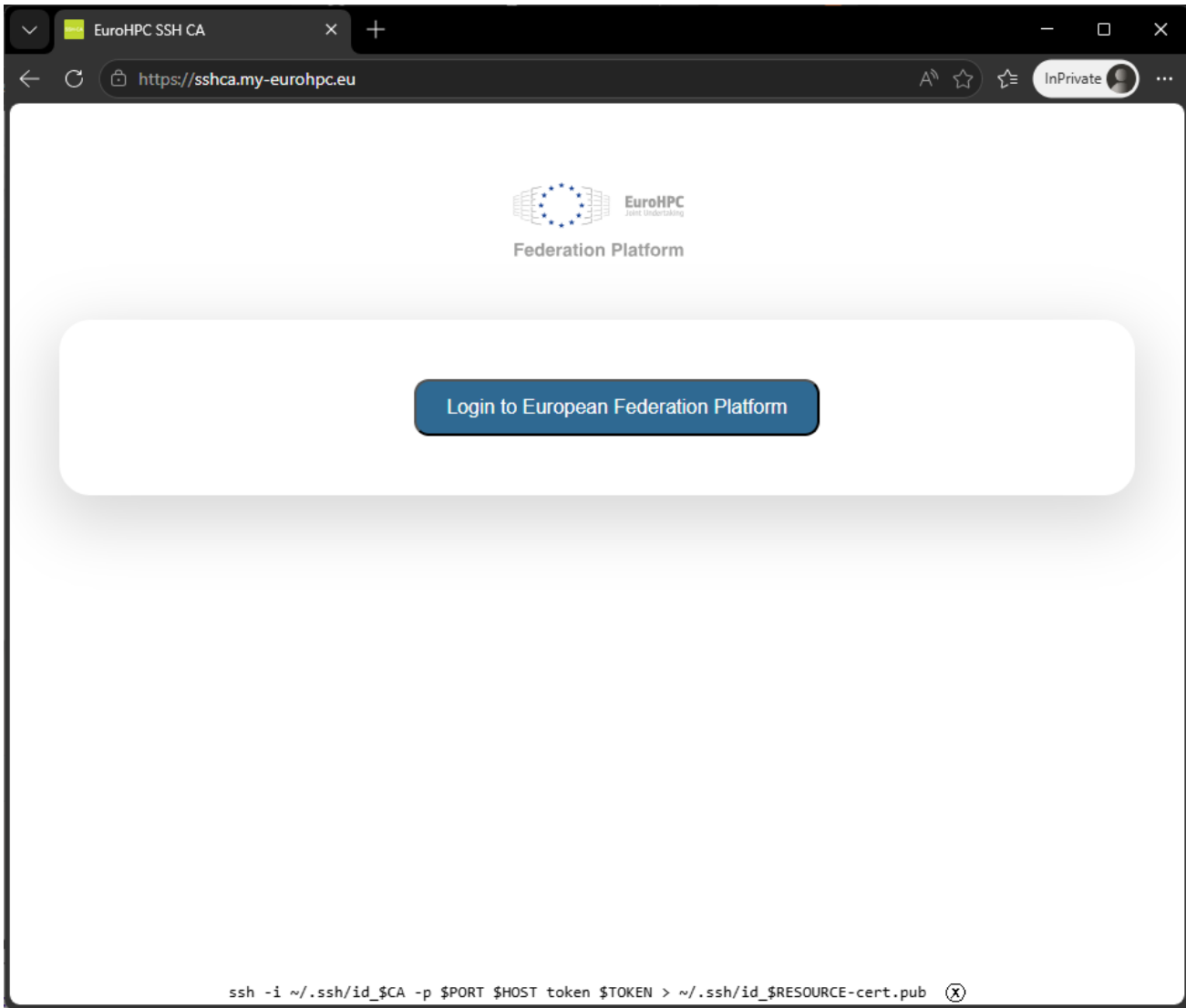


User performs mandatory multifactor authentication

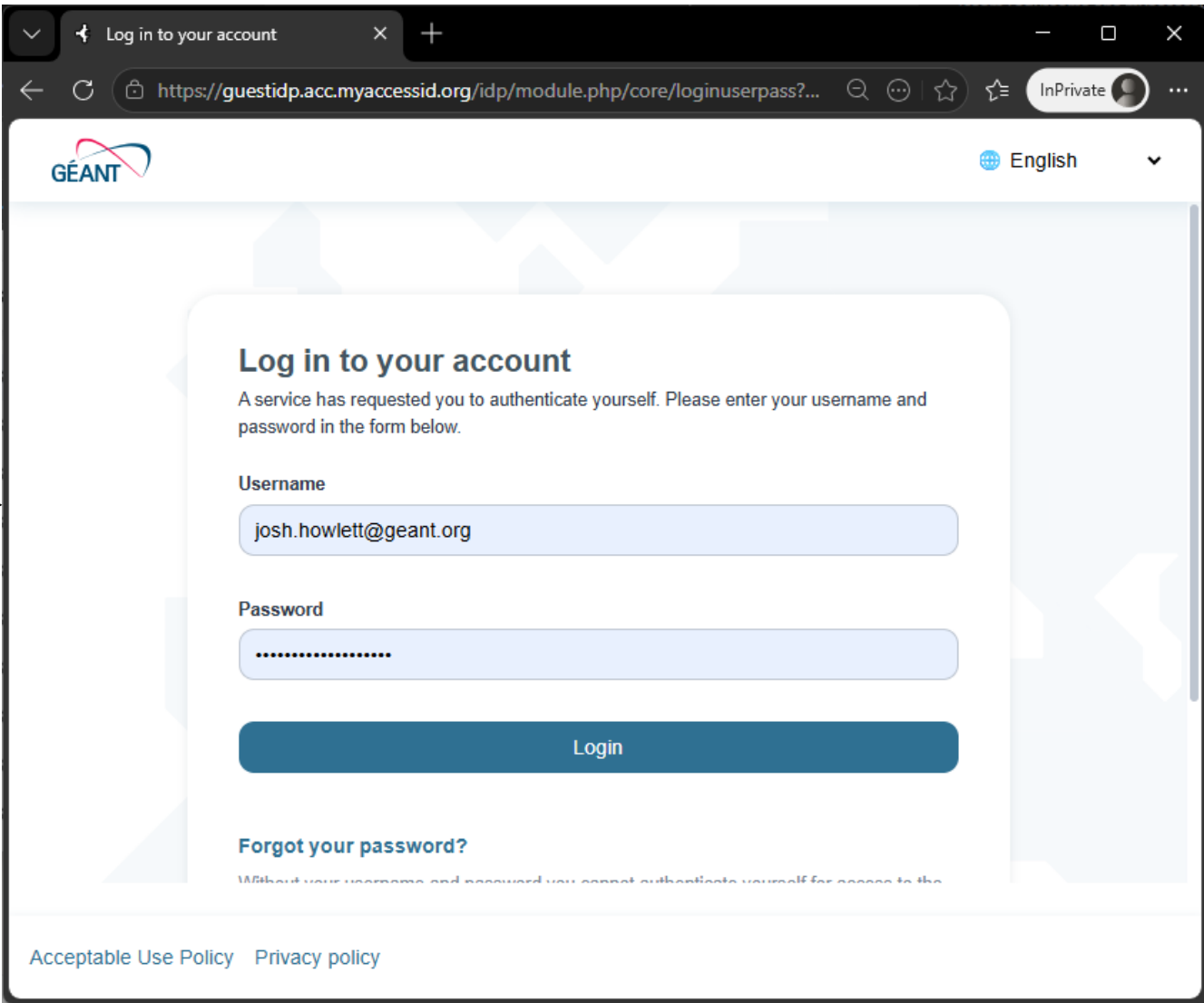


EFP SSH Certificate Authority

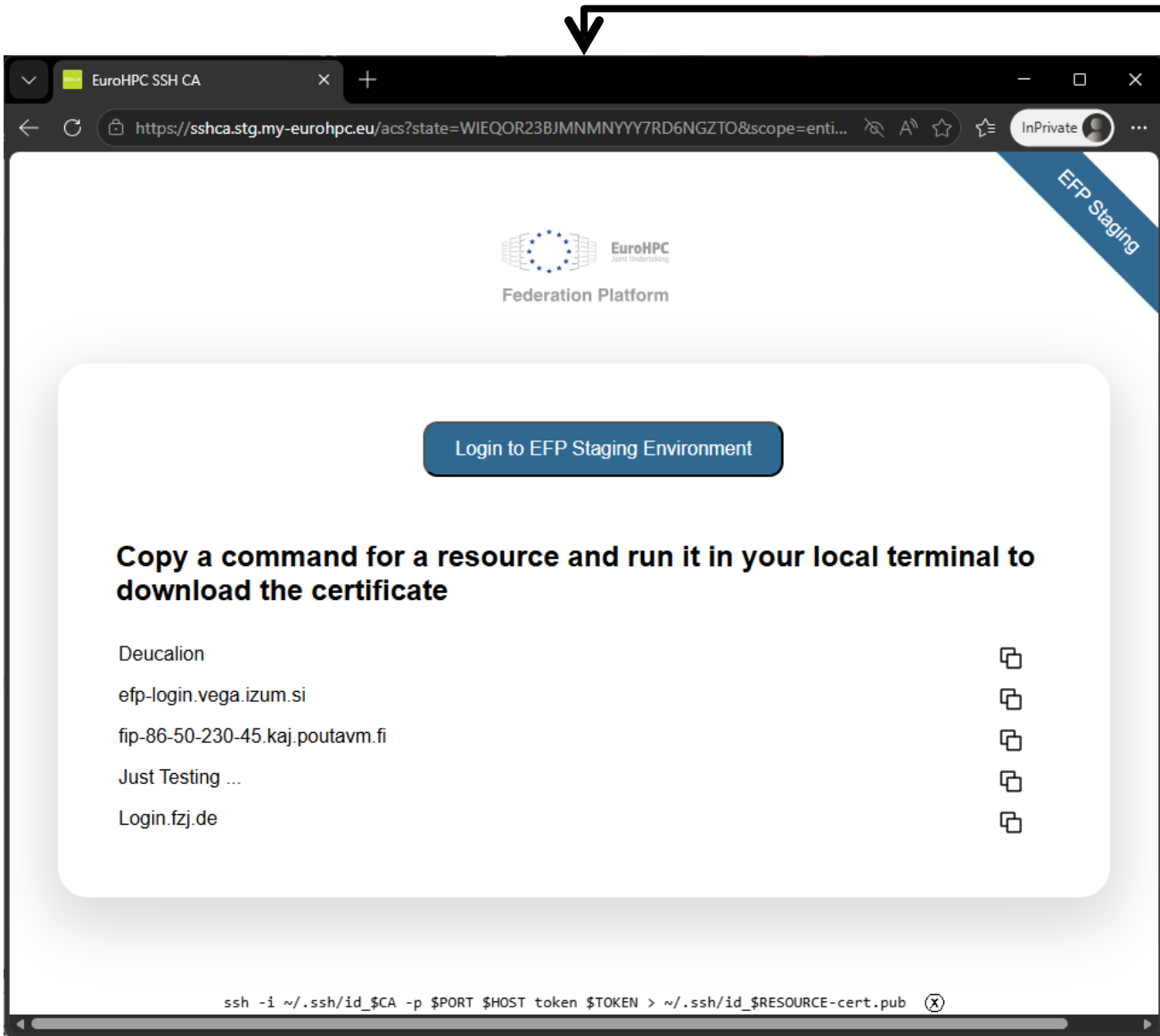
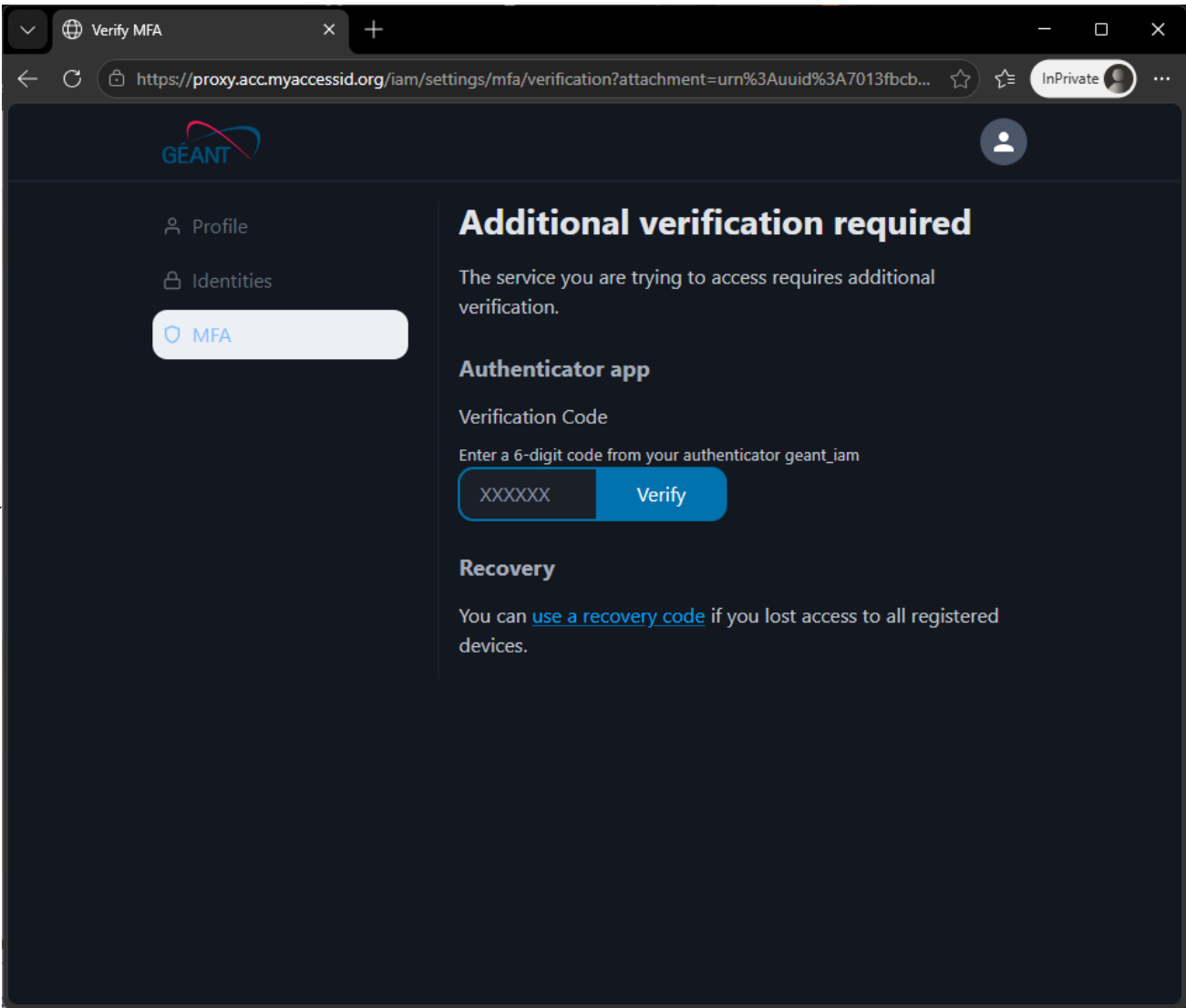
User browses to web SSH CA



User authenticates against their identity provider



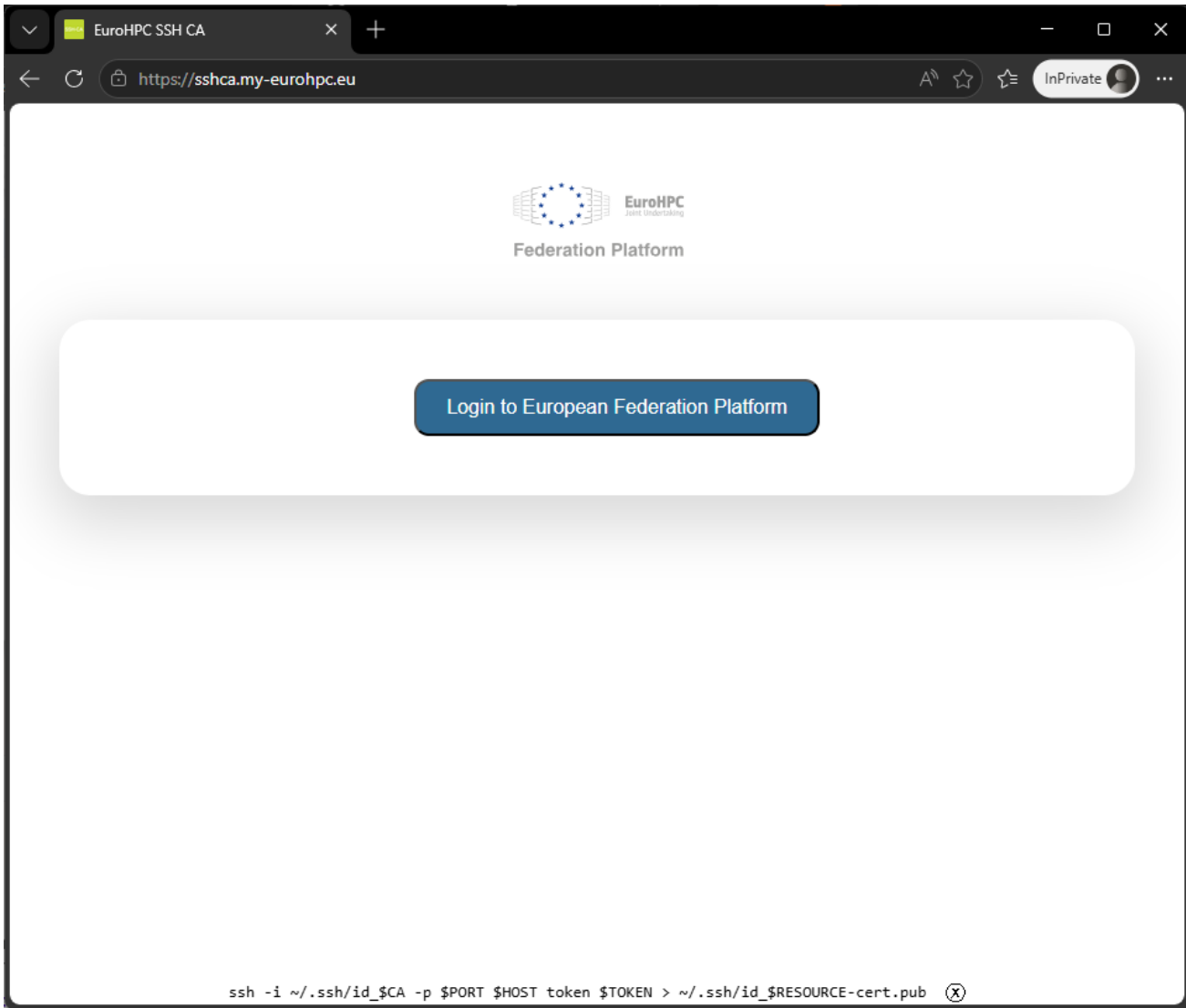
User performs mandatory multifactor authentication



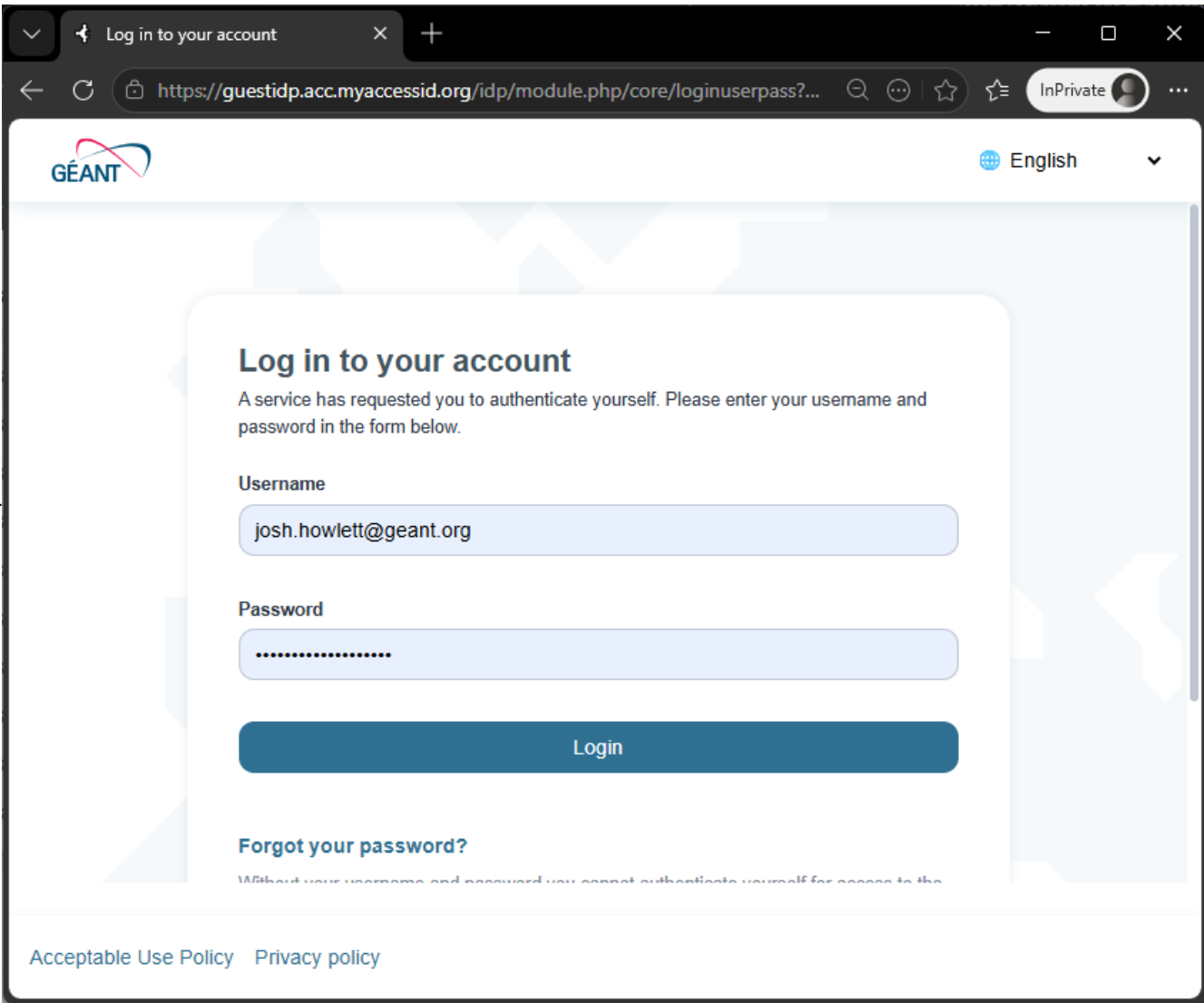
User downloads a certificate for the resource

EFP SSH Certificate Authority

User browses to web SSH CA

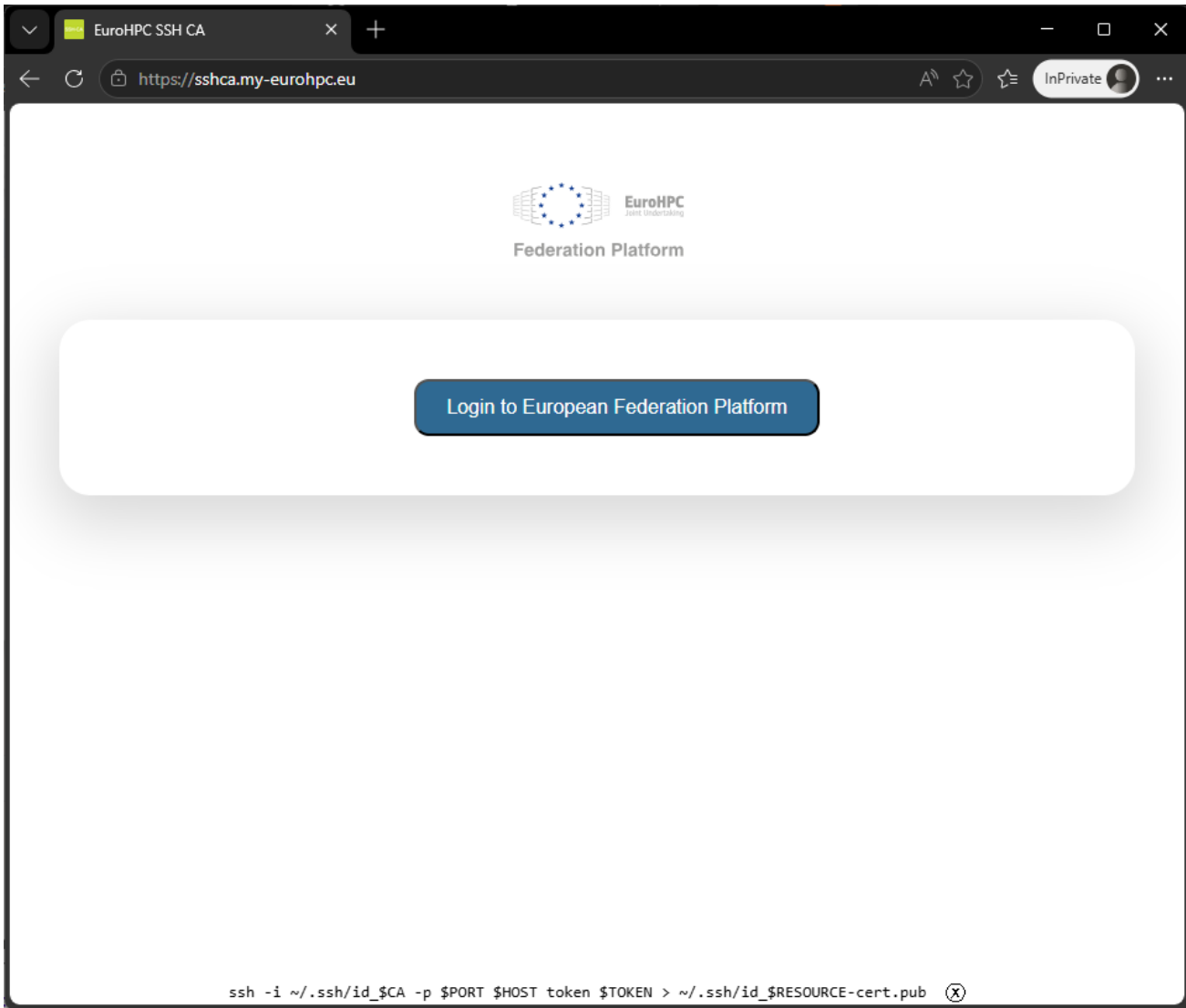


User authenticates against their identity provider

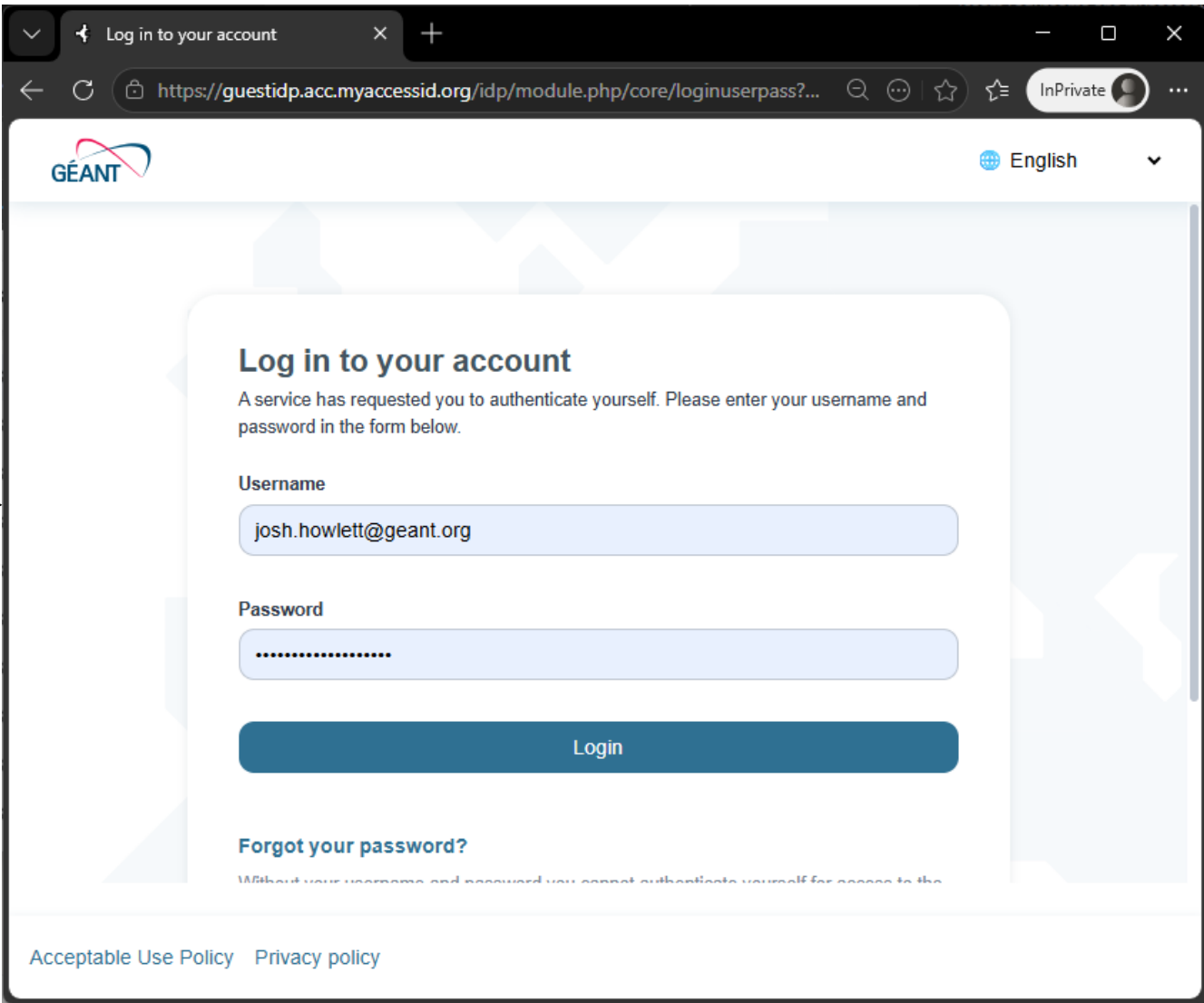


EFP SSH Certificate Authority

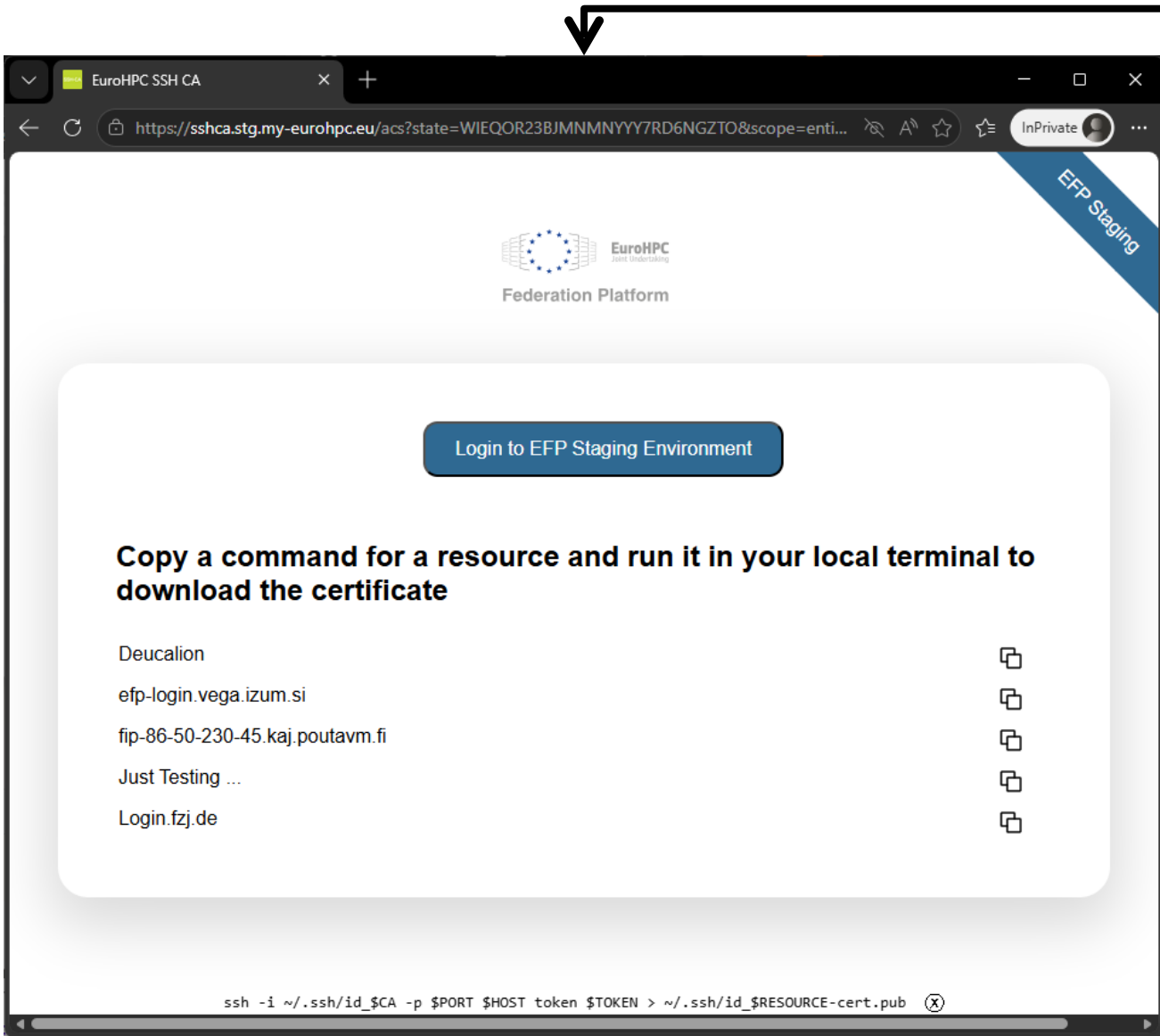
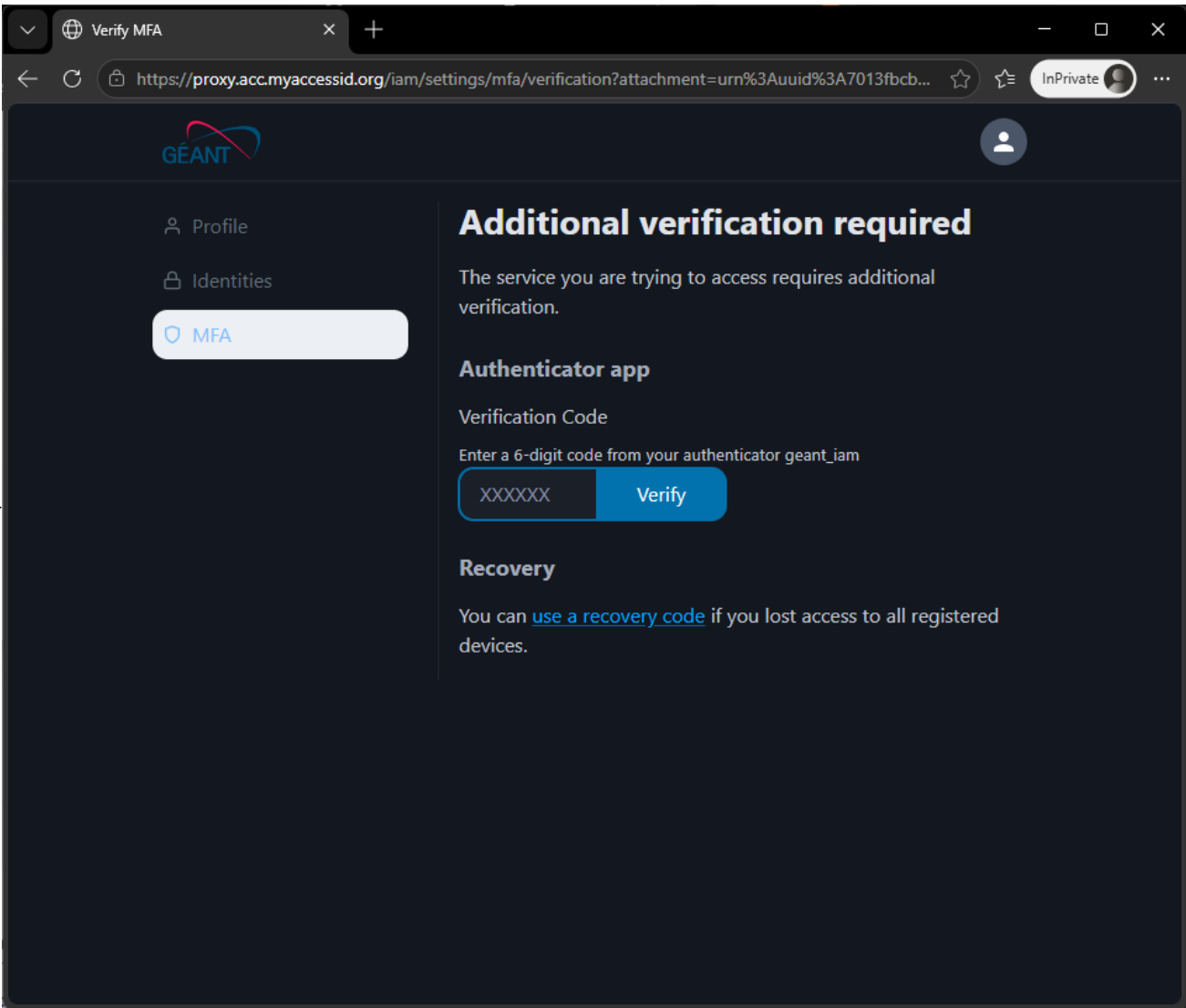
User browses to web SSH CA



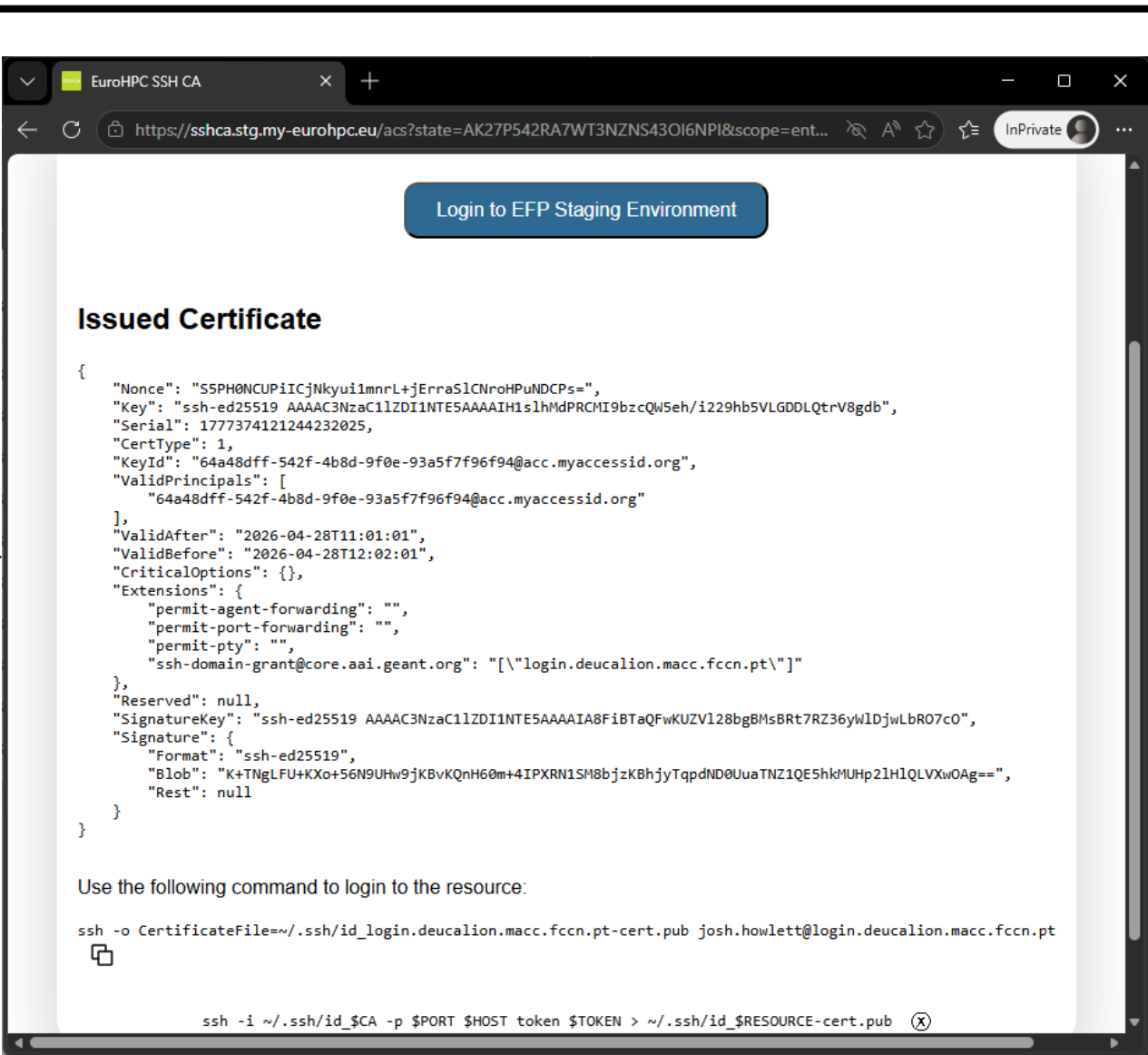
User authenticates against their identity provider



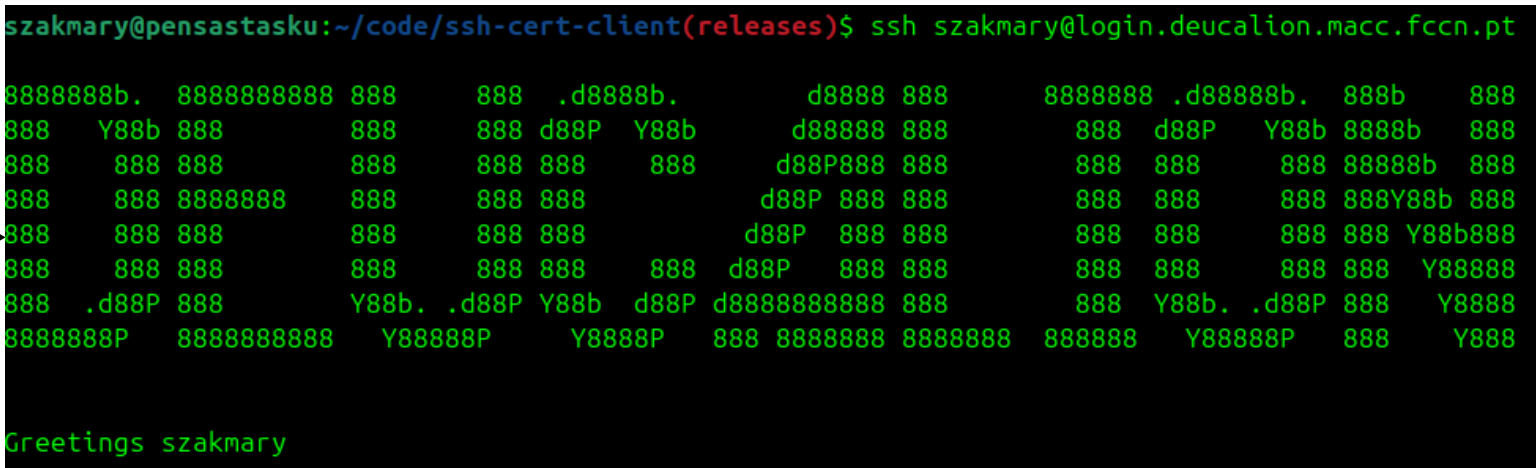
User performs mandatory multifactor authentication



User downloads a certificate for the resource



User copies the SSH command from the browser...



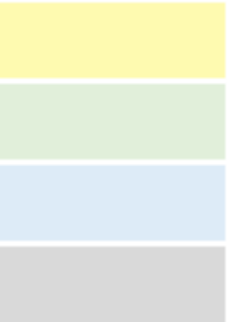
...and pastes the command into their terminal

SSH CA integration

- Hosting Entities must integrate their system with the SSH CA to accept EFP users
- The details of each Hosting Entity's integration will vary, but the general approach is the same

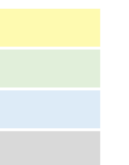
```
$ ssh-keygen -L -f ~/.ssh/id_efp.lumi.csc.fi-cert.pub
/home/josh/.ssh/id_efp.lumi.csc.fi-cert.pub:
    Type: ssh-ed25519-cert-v01@openssh.com user certificate
    Public key: ED25519-CERT SHA256:CihDEMsLg/zoIx/
e+d6jQFyHO8Mky5IEZIIz5gOgRiI
    Signing CA: ED25519
SHA256:zBERWNBCsqnQFo605mieee6sQHerFBhchTiIq8ymD+Y (using ssh-ed25519)
    Key ID: "64a48dff-542f-4b8d-9f0e-93a5f7f96f94@acc.myaccessid.org"
    Serial: 1774630421907173074
    Valid: from 2026-03-27T16:52:41 to 2026-03-27T17:53:41
    Principals:
        64a48dff-542f-4b8d-9f0e-93a5f7f96f94@acc.myaccessid.org
    Critical Options: (none)
    Extensions:
        permit-agent-forwarding
        permit-port-forwarding
        permit-pty
        ssh-domain-grant@core.aai.geant.org UNKNOWN OPTION:
000000135b226566702e6c756d692e6373632e6669225d (len 23)
```

SSH CA integration



- Hosting Entities must integrate their system with the SSH CA to accept EFP users
- The details of each Hosting Entity's integration will vary, but the general approach is the same
 1. Download the SSH CA public key and configure your SSH server(s) to trust it

```
$ ssh-keygen -L -f ~/.ssh/id_efp.lumi.csc.fi-cert.pub
/home/josh/.ssh/id_efp.lumi.csc.fi-cert.pub:
    Type: ssh-ed25519-cert-v01@openssh.com user certificate
    Public key: ED25519-CERT SHA256:CihDEMsLg/zoIx/
e+d6jQFyHO8Mky5IEZIIz5gOgRiI
    Signing CA: ED25519
SHA256:zBERWNBCsqnQFo605mieee6sQHerFBhchTiIq8ymD+Y (using ssh-ed25519)
    Key ID: "64a48dff-542f-4b8d-9f0e-93a5f7f96f94@acc.myaccessid.org"
    Serial: 1774630421907173074
    Valid: from 2026-03-27T16:52:41 to 2026-03-27T17:53:41
    Principals:
        64a48dff-542f-4b8d-9f0e-93a5f7f96f94@acc.myaccessid.org
    Critical Options: (none)
    Extensions:
        permit-agent-forwarding
        permit-port-forwarding
        permit-pty
        ssh-domain-grant@core.aai.geant.org UNKNOWN OPTION:
000000135b226566702e6c756d692e6373632e6669225d (len 23)
```



SSH CA integration

- Hosting Entities must integrate their system with the SSH CA to accept EFP users
- The details of each Hosting Entity's integration will vary, but the general approach is the same
 1. Download the SSH CA public key and configure your SSH server(s) to trust it
 2. Configure your SSH server to provision the user with a local account based on the certificate's principal; this takes a value of the user's MyAccessID identifier

```
$ ssh-keygen -L -f ~/.ssh/id_efp.lumi.csc.fi-cert.pub
/home/josh/.ssh/id_efp.lumi.csc.fi-cert.pub:
    Type: ssh-ed25519-cert-v01@openssh.com user certificate
    Public key: ED25519-CERT SHA256:CihDEMsLg/zoIx/
e+d6jQFyHO8Mky5IEZIIz5gOgRiI
    Signing CA: ED25519
SHA256:zBERWNBCsqnQFo605mieee6sQHerFBhchTiIq8ymD+Y (using ssh-ed25519)
    Key ID: "64a48dff-542f-4b8d-9f0e-93a5f7f96f94@acc.myaccessid.org"
    Serial: 1774630421907173074
    Valid: from 2026-03-27T16:52:41 to 2026-03-27T17:53:41
    Principals:
        64a48dff-542f-4b8d-9f0e-93a5f7f96f94@acc.myaccessid.org
    Critical Options: (none)
    Extensions:
        permit-agent-forwarding
        permit-port-forwarding
        permit-pty
        ssh-domain-grant@core.aai.geant.org UNKNOWN OPTION:
000000135b226566702e6c756d692e6373632e6669225d (len 23)
```

SSH CA integration

- Hosting Entities must integrate their system with the SSH CA to accept EFP users
- The details of each Hosting Entity's integration will vary, but the general approach is the same
 1. Download the SSH CA public key and configure your SSH server(s) to trust it
 2. Configure your SSH server to provision the user with a local account based on the certificate's principal; this takes a value of the user's MyAccessID identifier
 3. Configure your SSH server to require that the "domain grant" certificate extension matches your domain (as registered with EFP Allocations)

```
$ ssh-keygen -L -f ~/.ssh/id_efp.lumi.csc.fi-cert.pub
/home/josh/.ssh/id_efp.lumi.csc.fi-cert.pub:
    Type: ssh-ed25519-cert-v01@openssh.com user certificate
    Public key: ED25519-CERT SHA256:CihDEMsLg/zoIx/
e+d6jQFyHO8Mky5IEZIIz5gOgRiI
    Signing CA: ED25519
SHA256:zBERWNBCsqnQFo605mieee6sQHerFBhchTiIq8ymD+Y (using ssh-ed25519)
    Key ID: "64a48dff-542f-4b8d-9f0e-93a5f7f96f94@acc.myaccessid.org"
    Serial: 1774630421907173074
    Valid: from 2026-03-27T16:52:41 to 2026-03-27T17:53:41
    Principals:
        64a48dff-542f-4b8d-9f0e-93a5f7f96f94@acc.myaccessid.org
    Critical Options: (none)
    Extensions:
        permit-agent-forwarding
        permit-port-forwarding
        permit-pty
        ssh-domain-grant@core.aai.geant.org UNKNOWN OPTION:
000000135b226566702e6c756d692e6373632e6669225d (len 23)
```

SSH CA integration

- Hosting Entities must integrate their system with the SSH CA to accept EFP users
- The details of each Hosting Entity's integration will vary, but the general approach is the same
 1. Download the SSH CA public key and configure your SSH server(s) to trust it
 2. Configure your SSH server to provision the user with a local account based on the certificate's principal; this takes a value of the user's MyAccessID identifier
 3. Configure your SSH server to require that the "domain grant" certificate extension matches your domain (as registered with EFP Allocations)
- GÉANT provides tooling for parsing SSH certificates (by *openssh*) to facilitate steps 2 and 3

```
$ ssh-keygen -L -f ~/.ssh/id_efp.lumi.csc.fi-cert.pub
/home/josh/.ssh/id_efp.lumi.csc.fi-cert.pub:
    Type: ssh-ed25519-cert-v01@openssh.com user certificate
    Public key: ED25519-CERT SHA256:CihDEMsLg/zoIx/
e+d6jQFyHO8Mky5IEZIIz5gOgRiI
    Signing CA: ED25519
SHA256:zBERWNBCsqnQFo605mieee6sQHerFBhchTiIq8ymD+Y (using ssh-ed25519)
    Key ID: "64a48dff-542f-4b8d-9f0e-93a5f7f96f94@acc.myaccessid.org"
    Serial: 1774630421907173074
    Valid: from 2026-03-27T16:52:41 to 2026-03-27T17:53:41
    Principals:
        64a48dff-542f-4b8d-9f0e-93a5f7f96f94@acc.myaccessid.org
    Critical Options: (none)
    Extensions:
        permit-agent-forwarding
        permit-port-forwarding
        permit-pty
        ssh-domain-grant@core.aai.geant.org UNKNOWN OPTION:
000000135b226566702e6c756d692e6373632e6669225d (len 23)
```

SSH Client Configuration

This slide intentionally left blank

Scale

Access to large number of user via federated web-based login
Trust many users by trusting one or more SSH CAs

Secure

Certificates with time and identity limitations



Security Analysis Report:
SSH CA

Version: 1.2
09.12.2024

<https://hackmanit.de/en/penetrationtests/overview/>

<https://github.com/wayf-dk/sshca>

<https://www.wayf.dk/da/giv-ssh-adgang-fra-institutionslogin-med-ny-software-fra-deic>

<https://sshca.deic.dk/defaultCA>

```
ssh -p 2222 -o CertificateFile=~/.ssh/id_defaultCA-cert.pub demo@sshca.deic.dk demo
```