User Management

Juan J. Doval
DEIMOS SPACE S.L.U.

NextGEOSS, September 25th 2017
Agenda

- Introduction
- User Management
- Federation Objectives
Introduction

NextGEOSS

• High-Level Architecture
DataHub harvest and register data, providing links to original sources.

Discovery Enablers empower search on DataHub from users.

Access Enablers allow community hubs to create data buckets for access.

Enhanced distributed gateway from research and operational infrastructures.

Processing Enablers allow community hubs to deploy distributed ICT technologies.

Publishing Appliances deliver to the community hubs processed results.

Community Portals register selected products and services to GEOSS.
User Management

- Context - User Stories
- Objectives
- Main Functionality
- State-of-the-art protocols
- Logical Architecture
- Implementation Status
- KPI Analytics
As a **GEOSS user**, I want to be able to **register** myself in the GEOSS community so that the user information is provided to a centralized authentication server to support single sign-on (SSO) with GEOSS providers.

As a **GEOSS user**, I want to be able to **authenticate and authorize** me in the GEOSS community with single sign-on (SSO) so that I can access to resources (data and some services)

A GEOSS user can be a data provider or a final user.
Objectives

- Current state-of-the-art technologies
- Support SSO: for minimizing the impact on data users to access and usage: register and login once in the GEOSS community
- Support federation
**Main functionality**

- Allows **registration** of users into the GEOSS community providing user information (user name, family name, email, telephone number, gender, ...)

- Allows **authentication and authorization** mechanisms based on GEOSS user credentials

- Provides **SSO** capability that enables a registered GEOSS user to log in once, and access multiple GEOSS applications without being required to authenticate for each application separately.

- Allows dynamic **client registration** of GEOSS services (i.e. harvesting, discovery, access and processing data) to be able to use the authentication and authorization mechanisms

- Allows integration of **social network login** (Google, Twitter, Facebook, LinkedIn).

- Allows integration of other SSO systems to provide a **federation** (e.g. ESA-https://eo-sso-idp.eo.esa.int, NASA-https://urs.earthdata.nasa.gov/).

- Is compatible with **different protocols**: OIDC, SAML2, Oauth2, ....
State-of-the-art protocols (I)

**Authentication viewpoint**

OpenID Connect turns SSO into a standard OAuth-protected identity API

**SAML 2.0, OpenID 2.0**
- Initiating user's login session
- Not responsible for collecting user consent
- High-security identity tokens (SAML only)
- Distributed and aggregated claims
- Dynamic introduction (OpenID only)
- Session timeout

- ✔ Initiating user's login session
- ✔ Collecting user's consent to share attributes
- ✔ No identity tokens per se
- ✔ No claims per se; protects arbitrary APIs
- ✔ Dynamic introduction
- ✔ Session timeout (in the works)

- ✗ Not responsible for session initiation
- ✗ Collecting user's consent to share attributes
- ✗ No identity tokens per se
- ✗ No claims per se; protects arbitrary APIs
- ✗ Dynamic introduction
- ✗ Session timeout (in the works)

**OAuth 2.0**
- Not responsible for collecting user consent
- No identity tokens per se
- No claims per se; protects arbitrary APIs
- Client onboarding is static
- No sessions per se
- Dynamic introduction
- Session timeout (in the works)

**OpenID Connect**
- Initiating user's login session
- Collecting user's consent to share attributes
- High-security identity tokens (using JSON Web Tokens)
- Distributed and aggregated claims
- Dynamic introduction
- Session timeout (in the works)

**References normatively as an option**
- (sharing some features as a result)

**Authentication/Authorization viewpoint**

- ✔ References normatively as an option
- (sharing some features as a result)

- OpenID Connect
  - You achieve federated single sign-on and login-time attribute exchange
  - You control access to claims about you

- OAuth 2.0
  - Not responsible for collecting user consent
  - No identity tokens per se
  - No claims per se; protects arbitrary APIs
  - Client onboarding is static
  - No sessions per se
  - Dynamic introduction
  - Session timeout (in the works)

- SAML 2.0
  - Initiating user's login session
  - Collecting user's consent to share attributes
  - High-security identity tokens (using JSON Web Tokens)
  - Distributed and aggregated claims
  - Dynamic introduction
  - Session timeout (in the works)

- UMA
  - You control access to apps operated by anyone
  - You can grant access to a variety of protected resources
  - You can grant access by setting policies and terms ahead of time
  - The authorization function is standard and centralizable
  - Authorization is based on claims

- UMA
  - You can grant access to apps operated by you
  - Authorization is based on authenticated identity
  - Apps can get access after you go offline

- UMA
  - You achieve federated single sign-on and login-time attribute exchange
  - You control access to claims about you

- UMA
  - You control access to web APIs
  - You grant access by consenting to terms at run time
  - You can grant access to apps operated by you

- UMA
  - The authorization function is local to protected resources
  - Apps can use a variety of access token types

- UMA
  - You can grant access to apps operated by anyone
  - You can grant access by setting policies and terms ahead of time
  - The authorization function is standard and centralizable
  - Authorization is based on claims
State-of-the-art protocols (II)

The RS exposes whatever value-add API it wants, protected by an AS.

The RPT is the main "access token" and (by default – it’s profitable) is associated with time-limited, scoped permissions.
Logical Architecture

Based on claims/scopes
NextGEOSS SSO allows tracking User Management usage.

- Number of authentications
- Authentication delay
- Registered users and clients
- Filters by IDP, client...
- User Accesses to Resources!
Federation Objectives

• Use Cases

• Proposed Approach
Use Cases

▪ As a **user**, I want to be able to **authenticate** myself in GEOSS using my credentials from NASA/ESA SSO service for supporting single sign-on (SSO).

▪ As a **user** with an active session started in NASA/ESA SSO service, I want to be able to **automatically access** GEOSS when selecting NASA/ESA login method.

▪ As a **user**, I want to be able to **authenticate** myself in NASA/ESA using my credentials from GEOSS SSO service for supporting single sign-on (SSO).
Proposed Approach (I)
NASA/ESA user profile information will be used for dynamic registration in our UM system (LDAP) and for internal usage in NextGEOSS.

Required user attributes:

- Username
- First Name
- Last Name
- E-mail
Proposed Approach (III)

Required information from ESA/NASA IDP:

- Client ID
- Client secret
- Authorization endpoint
- Token endpoint

Required matching parameter:

- Callback URL: https://nextgeoss-sso.elecnor-deimos.com/auth/nasa/callback
Thanks!

- Questions?