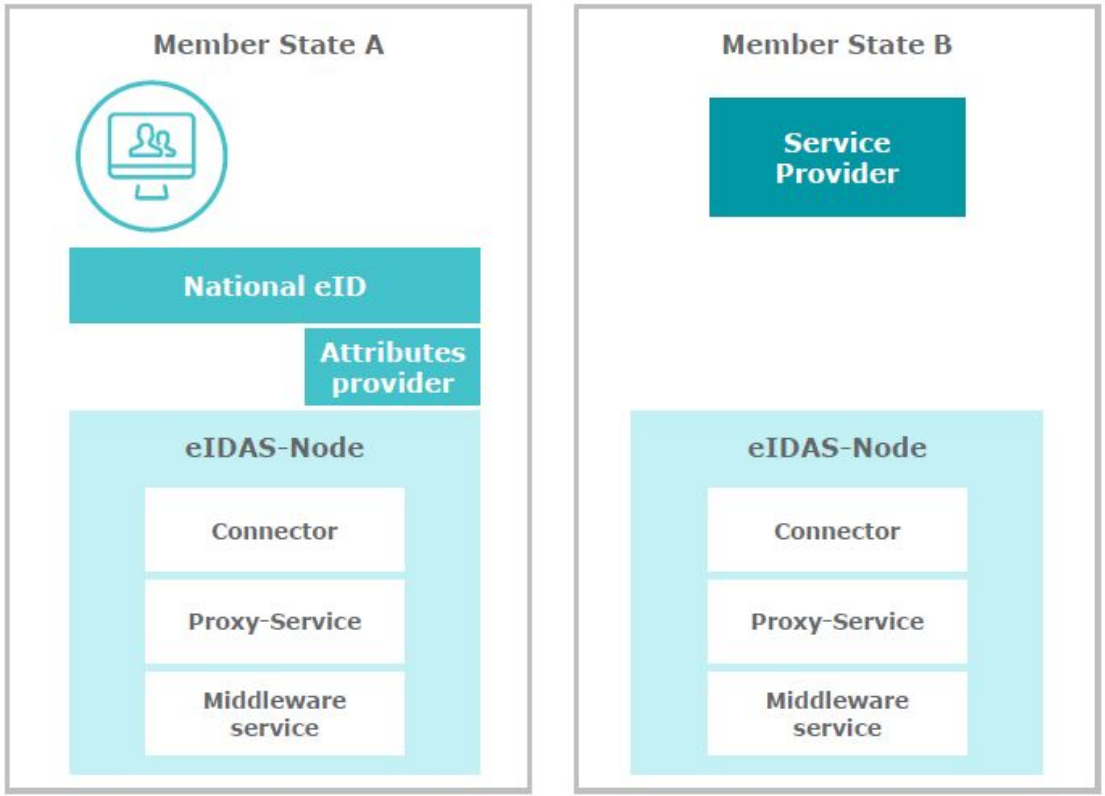
TOOP eID module

Detailed Design Approach

Petros KAVASSALIS (UAegean)
Nikos TRIANTAFYLLOY (UAegean)
Harris PAPADAKIS (UAegean)

Some eIDAS terminology



The eIDAS Network consists of a series of eIDAS-Nodes implemented at the Member State level.

- eIDAS Node:
  - Request a cross-border authentication
  - Provide a cross border authentication
- Service Provider (SP):
  - Offers services to eIDAS authenticated users (i.e. relying party)
  - Connected to national eIDAS node
- National eID:
  - National Identity Provider of the user's Member State

Architecture

Functionality:

The eID module acts as a proxy service between the Data Consumer (DC) Interface and the eIDAS node:

1. Handles **communication** with eIDAS network and **retrieval** of eIDAS attributes (for piloting we are assuming that the eIDAS minimum data set will be used but this can vary per business case), together with **pre-built UI**
2. **Propagates** the received eIDAS attributes back to the DC interface
3. DC includes received attributes to TOOP request message

Assumptions:

1. MS eIDAS node is available (for piloting non notified eIDAS scheme will be acceptable)
2. eIDAS node Connector Implementation based on SAML (eIDAS node v1.4 implementation)

Alternatives

In case the assumptions do not hold, the eID module can not be used as is, however eID integration in TOOP is generic enough to support:

1. Legacy (MS specific) means of eID.
    a. These can be used for piloting, however progression to eIDAS is desired
2. MS eIDAS connectors specific implementations can be used

In either case sample eID module software will need to be customised to specific MS needs or DC will need to integrate eID using a custom solution.

eID inclusion in TOOP requests is required in all cases

Provided Implementation

An implementation for the eID module can be found here as a Docker Image:

- https://hub.docker.com/r/endimion13/toop-eid/

And its source code can be viewed here:

- https://github.com/endimion/toopEid

Some minor configuration is required. Examples can be found on the description of the DockerImage

Provided Implementation

- At the end of the user authentication, the user is redirected back to the SP with redirect param “eidasAttributes" which contains the stringified version of the received attributes e.g.:

http://193.10.8.213:9084/ui?eidasAttributes=%7B%22eidasAttributes%22:[%7B%22friendlyName%22:%22familyName%22,%22loa%22:%22http://eidas.europa.eu/LoA/low%22,%22value%22:%22Garcia%22%7D,%7B%22friendlyName%22:%22firstName%22,%22loa%22:%22http://eidas.europa.eu/LoA/low%22,%22value%22:%22javier%22%7D,%7B%22friendlyName%22:%22dateOfBirth%22,%22loa%22:%22http://eidas.europa.eu/LoA/low%22,%22value%22:%221965-01-01%22%7D,%7B%22friendlyName%22:%22personIdentifier%22,%22loa%22:%22http://eidas.europa.eu/LoA/low%22,%22value%22:%22GF/GF/12345%22%7D],%22timestamp%22:null,%22audienceRestriction%22:null,%22issuer%22:null,%22country%22:null%7D#!loginSuccess

- Please note that there are available versions of the eID module that offer more secure ways of transfering the identification attributes back to the SP (e.g. using JWT)
- A Mock Version of this eID module will also be available (i.e. a version not requiring the integration with an actual eIDAS node) for easier testing of the system.