DIGITALISERINGSSTYRELSEN

NemLog-in

Integration with NemLog-in3

Contents

Change log

Date	Version	Change description	Initials
18-01-2021	0.1	First draft	Nets (TMNYM)
21-01-2021	0.2	Nets review and changes.	Nets (MDBEC/TMNYM)
22-01-2021	0.3	Added Encryption method section	Nets (TMNYM)
07-02-2021	0.4	Introduced changes cf. DIGST feedback	Nets (TMNYM)
19-03-2021	0.5	Changed section regarding requesting NSIS assurance level. Removed addressed comments.	Nets (TMNYM)
12-04-2021	0.6	Added details to section 9.	Nets (TMNYM)
04-05-2021	1.0	Added Local IdP to drawing and text in section 4.1. Added description of session handling and timeouts. Added specifics for private service providers regarding sessions. Handled comments deleted. OIOSAML in one word.	Nets (TMNYM)
05-05-2021	1.0a	Reformatting	Nets (SPEDE)
26-05-2021	1.1	Minor typo corrections and accept of new paragraph 9.6 Embedded authentication	Nets (SPEDE)
31-05-2021	1.2	Added new paragraph 7.2 New assurance level value.	Nets (TMNYM)
05-08-2021	1.3	Elaborated description of privileges migration process (section 6.7). Added description of OIO BPP scope value deviation from profile (section 9.5).	Nets (TMNYM)
30-08-2022	1.4	Updated description of NemID behaviour for NSIS LoA and Assurance Level based on revalidation of the user (section 9.2). Added new section 9.2.1 detailing the returned value of the AuthnContextClassRef in AuthnStatement of a SAML Assertion.	Nets (MDBEC)
24-10-2022	1.5	Updated reference links. Added page numbering.	Nets (SPEDE)
08-11-2022	1.6	Added sections 10.6. and 10.7 regarding attribute availability.	Nets (TMNYM)
08-05-2023	1.7	Added section 9.6 and Appendix A to document mobile app-switch behaviour of NemLog-in. Update section 9.5 to include example of how to start external browser in-app on Android and iOS.	Nets (MDBEC)
06-07-2023	1.7.1	Updated section 9.7 app-switch targeting App Links instead of Dynamic Links on Android.	Nets (MDBEC)
12-12-2023	1.8	Add new section 9.8 outlining how to specify local IdP in the AuthnRequest.	Nets (MDBEC)
08-01-2024	1.9	Add new section 9.8 ProviderName and updated reference to OIOSAML3.0.3	Nets (SSOMM)
22-03-2024	2.0	Removed all references to NemID OCES2	NETS (SSOMM)
24-04-2024	2.1	Added the OneTimeUse condition	NETS (SSOMM)

Date	Version	Change description	Initials
01-05-2024	2.2	Adjusted section 9.9 regarding allowed special characters in ProviderName	NETS (SSOMM)
01-10-2024	2.3	Adjusted section 9.9 – removed \ as supported character	NETS (SSOMM)

Update the footer w. date and version as well.

1 Introduction

This document describes how it-systems should be integrated with the NemLog-in3 OIO SAML interfaces.

NemLog-in3 supports two distinct SAML integration interfaces, supporting the two current OIO SAML profiles, OIO SAML 2.1.0 and OIO SAML 3.0.3. The document is an integration guide and does not provide all SAML profile details. These are described in the OIO SAML profile documents [OIOSAML3], [OIOSAML2.1.0].

1.1 Prerequisites

The reader is expected to be familiar with the most recent version of the OIO SAML 3 profile [OIOSAML3].

1.2 Intended audience

This document is a technical implementation guide aimed at architects and developers.

1.3 Terminologi

Term	Description
Identity Provider	An Identity Provider (IdP) is a trusted entity that authenticates users and generates authentication assertions or other assertions that vouch for a user's (subject's) identity.
Service Provider	A Service Provider (SP) is an entity that relies on assertions from an Identity Provider (IdP) to authenticate or authorize subjects' actions on its resources.
Assertion	Data structure produced by an Identity Provider (SAML authority) or similar regarding an act of authentication. The assertion provides information on the authentication performed by a User, attribute information about the User, and/or authorization permissions applying to the User with respect to a specified resource.
Metadata	Service Providers and Identity Providers gather the information needed to execute the SAML protocol in so-called metadata XML files. NemLog-in metadata contains: • EntityID – a unique identifier for the party (SP/IdP) in the federation • Cryptographic keys in the form of X.509 certificates - used for signing and encryption • Protocol endpoints

2 Reference implementations

Correct implementation of a SAML integration from scratch is a difficult task that require expertise and a substantial development and testing effort.

For this reason we strongly recommend that your integration with NemLog-in makes use of available SAML software, preferably one of the available OIO SAML reference implementations:

- OIO SAML for Java – [OIOSAML-Java]
- OIO SAML for C#/.NET – [OIOSAML-NET]

These software packages provide sample code and demonstration applications for both authentication and attribute query integrations.

Even if you plan to use another SAML implementation the reference implementations will be very useful to allow developers to familiarize themselves with the SAML protocol and OIO SAML specifics.

3 Available features

NemLog-in3 functionality is available to both public and private service providers. There are, however, some features that are only available to public services providers.

These are summarised in the table below.

Feature	Available to public service providers	Available to private service providers	Remarks
OIO SAML 2.1.0 interface	Yes	No	Backwards compatibility for public service providers is ensured by allowing access to OIO SAML 2.1.0 endpoints.
OIO SAML 3.0.3 interface	Yes	Yes	
Attribute Service	Yes	Yes	The NemLog-in Attribute Service should be used to enhance end-user privacy by allowing the SP to only request a minimal attribute set during authentication and then subsequently request additional attributes.
Single Sign-on	Yes	No	Private service providers are not allowed to participate in NemLog-in SSO. Note, that private service providers must nevertheless implement the Single log-out profile.
Single Logout (SLO)	Yes (required)	Yes (required)	Even though private service providers are not allowed participation in NemLog-in SSO they must implement support for SLO.
FBRS access rights	Yes	No	Private service providers are not allowed to specify the Privileges attribute in metadata. See [OIOSAML3], section 6.2.3.
CPR attributes	Yes	No	Private service providers are not allowed to specify the CPR attribute in metadata. This also prevents access to CPR attributes by Attribute Query.

4 Architectural overview

This section provides an overview of the architecture and shows the most common usage scenarios by example.

4.1 Integration overview

NemLog-in acts as an identity broker. Service providers use NemLog-in for authentication. OIOSAML assertions issued by NemLog-in as a result of the authentication process conveys information about the authenticated identity to the service provider. Identity information is loosely coupled to the identification means (MitID or local[1]) used in the authentication process. This allows the service provider to deal with the application centric aspects of authentication (who is authenticated, what was the authentication strength) and not to deal with different types of identification means, different authentication protocols etc.

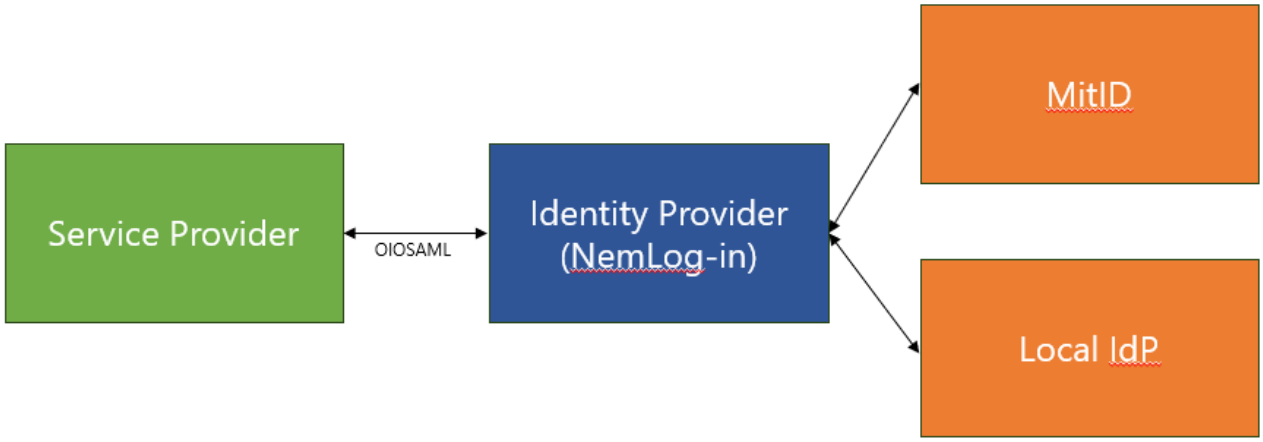

FIGURE 1: INTEGRATION OVERVIEW

[1] Local identification means will be available by the use of Local IdPs employed by individual user organisations.

4.2 Access with authentication

The first scenario shows the interaction where a user accesses a Service Provider directly (via her browser) to get a service with no prior session established.

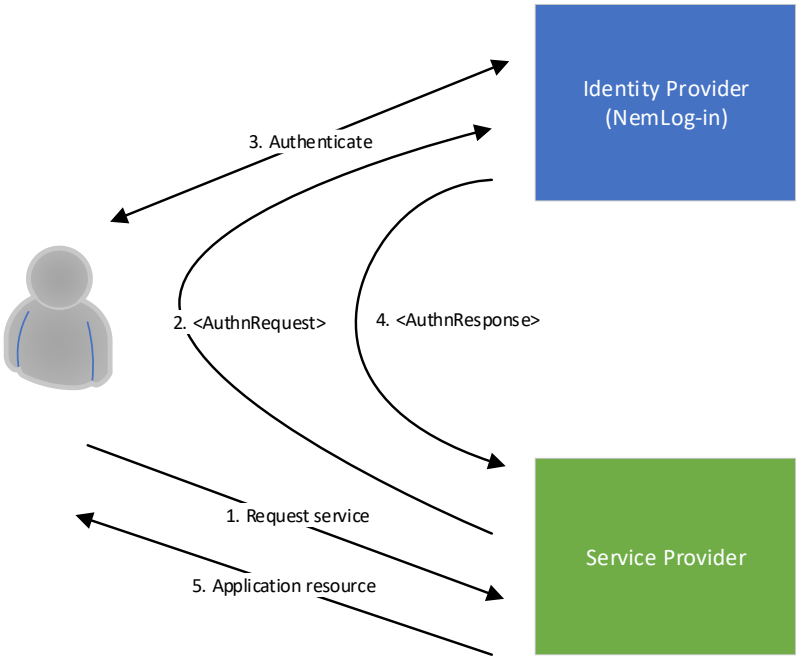

FIGURE 2: ACCESS WITH AUTHENTICATION

The steps are:

1) The user requests (via her browser) a web application resource from the Service Provider.
2) The Service Provider determines that the resource is protected and that the user has no current session. The Service Provider creates and signs an authentication request and redirects the user to the Identity Provider with the request as a parameter.
3) The Identity Provider receives the authentication request, learns that the user has no current (IdP) session, and therefore initiates authentication of the user. The user authenticates with valid credentials (MitID).
4) After successful authentication, the Identity Provider establishes a session and redirects the user back to the Service Provider with a response containing a signed SAML assertion. The Service Provider validates the assertion, creates a user session and performs an authorization check on the resource originally requested by the user.
5) If the authorization check succeeds, the requested application resource is returned to the user.

4.3 Access with single sign-on

The second scenario shows the interaction where a user accesses a Service Provider directly (via her browser) to get a service when an Identity Provider session has previously been established.

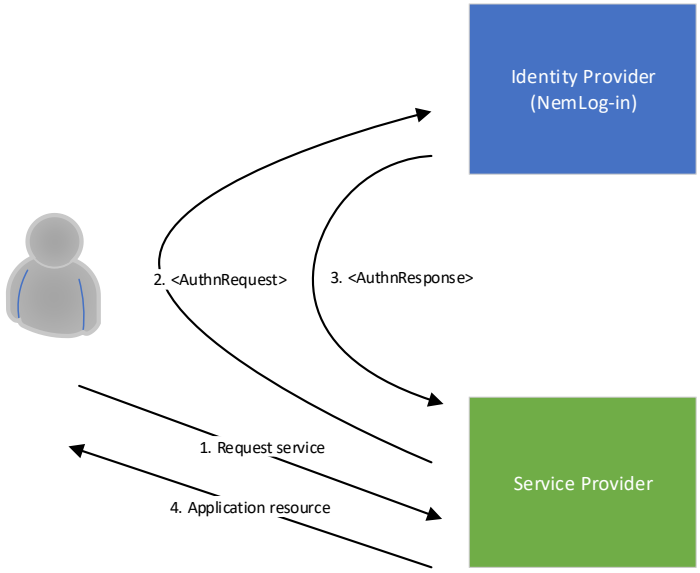

FIGURE 3: ACCESS WITH SINGLE SIGN-ON

The steps are:

1) The user requests (via her browser) a web application resource from the Service Provider.
2) The Service Provider determines that the resource is protected and that the user has no current session. The Service Provider creates and signs an authentication request and redirects the user to the Identity Provider with the request as a parameter.
3) The Identity Provider receives the authentication request, learns that the user has an active session, and therefore initiates single-sign on. The Identity Provider redirects the user back to the Service Provider with a response containing a SAML assertion.
4) The Service Provider validates the assertion, creates a user session, and performs an authorization check on the resource originally requested by the user. If the authorization check succeeds, the requested application resource is returned to the user.

4.4 Access via portal and use of attribute query

The third scenario shows the interaction where a user accesses a Service Provider via a portal and an Identity Provider session has previously been established

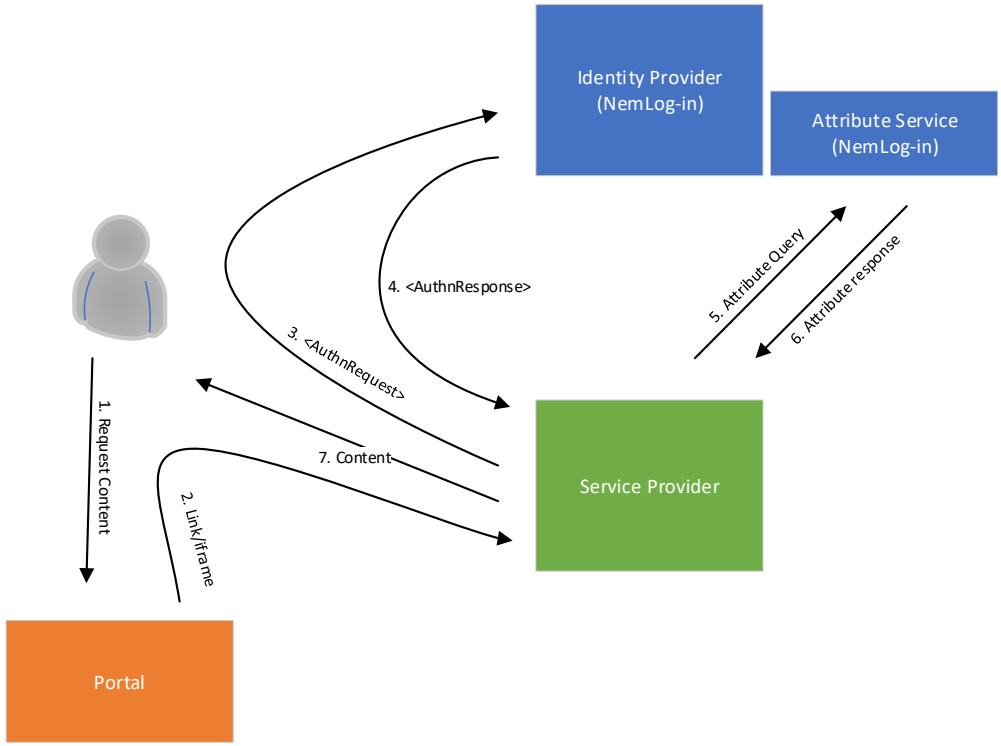

FIGURE 4: ACCESS VIA PORTAL AND USE OF ATTRIBUTE QUERY

The steps are:

1) The user accesses the Portal which aggregates content and services from different Service Providers.
2) Via the portal, the user requests an application resource from a Service Provider. In browser-based integration scenarios, the portal will either link to the Service Provider or frame its content (e.g. using an iframe). Web service integration is thus not considered.
3) The Service Provider determines that the resource is protected and that the user has no current session. The Service Provider therefore creates and signs an authentication request and redirects the user to the Identity Provider by posting this request.
4) The Identity Provider receives the authentication request, learns that the user has an active session, and therefore initiates single sign-on. The Identity Provider redirects the user back to the Service Provider with a response containing a SAML assertion. The Service Provider validates the assertion, creates a user session, and performs an authorization check on the resource originally requested by the user.
5) The Service Provider determines that it needs additional attributes about the user in order to either make an authorization decision or deliver its service, so it sends an attribute query to the Attribute Service co-located with the Identity Provider.
6) The Attribute Service authenticates and authorizes the query and returns an attribute assertion. The assertion is validated by the Service Provider and the attributes are extracted for use e.g. in an access decision.
7) The application resource originally requested by the user is returned (if access decision allows it).

4.5 Single Log-Out

A natural supplement to Single Sign-On is Single Logout whereby a user can terminate her current sessions with all Service Providers and Identity Providers.

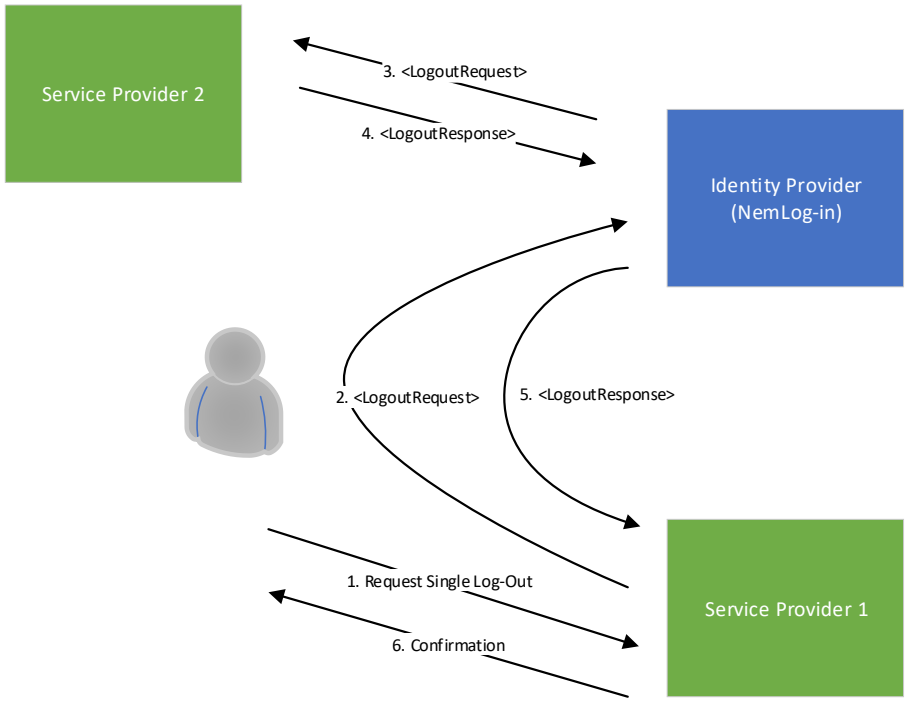

FIGURE 5: SINGLE LOG-OUT

The steps are:

1) The user contacts Service Provider 1 (e.g. by clicking 'Log out') to request Single Log out.
    1) Service Provider 1 contacts the user's Identity Provider to request Single Log out.
    2) The Identity Provider determines which additional Service Providers the user has active sessions with (Service Provider 2) and sends them a request for logout.
    3) Service Provider 2 terminates his user session and responds to the Identity Provider.
    4) The Identity Provider terminates his user session and responds to the Service Provider.
    5) The Service Provider responds with a confirmation to the user that all current sessions have been terminated.

5 NemLog-in interfaces

NemLog-in provides a set of interfaces available to service providers. These are summarized in the table below.

Interface/API	Description
OIOSAML 3.0.3interface	The interface used by service providers for authentication.
Attribute Service	This optional service implements the SAML 2.0 Attribute Query protocol. The service allows service providers to implement privacy friendly solutions that request a minimal attribute set during authentication and obtain additional attributes later.
Security Token Service (STS)	If your application needs access to public web services that utilize the OIO-IDWS standard for authentication and authorization, you must use the STS for token exchange. We refer to [OIOIDWS] for details.
Lookup services	A few service providers will need to use these new services to obtain attributes. To obtain access to these endpoints the service provider must register a web service client (WSC) in NemLog-in and apply for access. We refer to [NLSS] for details.
UUID-match services	With the introduction of privacy friendly identifiers in signing certificates and SAML subjects it may be required for a service provider to use one or more of these services to verify signer identitites. All service providers connected to NemLog-in will have access to these services using the signing certificates specified in metadata as a client certificate for authentication. We refer to [NLSS] for details.
Qualified signing service	NemLog-in's signing solution supports creation of qualified person- and employee signatures and seal generation. Service providers select if they wish to access the signing solution during connection of their system to NemLog-in. We refer to [NLSPS] for details.
Validation service	The validation service is a web interface and API for validating qualified signatures. We refer to [NLSPS] for details.

6 NemLog-in changes for OIOSAML 3

This section provides a summary of important changes introduced in NemLog-in in the OIOSAML 3 interface.

6.1 Identity Provider Discovery

Identity Provider Discovery is not implemented in the NemLog-in OIOSAML 3 interface. The OIOSAML 2 interface which is supported by NemLog-in supports IdP Discovery as described in [OIOSAML2.1.0] but will be phased together with the support for the OIOSAML 2 interface.

6.2 Attribute profiles

OIOSAML 3 introduces two different attribute profiles, the *natural person profile* for representing private persons and the *professional person profile* representing persons associated to an organisation (often employees). The service provider may request a specific profile in the authentication request.

6.3 Session timeout

The sessions established in both IdPs (OIOSAML2 and 3) are given two expiry times, a *soft* and a *hard* expiry time. A session that is renewed before the soft expiry by another (or the same) service provider requesting a (not-forced) authentication, will be given a new expiry time of same duration as the initial session, provided that the total time elapsed since the session was first established does not extend beyond the hard expiry time.

This, in effect, will ensure that the user must provide new credentials when the hard expiry is reached.

NemLog-in currently employ the following soft and hard session lifetimes:

Session LoA	Soft (initial) session lifetime	Hard (maximum) session lifetime
Low	1 hour	8 hours
Substantial/3	1 hour	8 hours
High	1 hour	8 hours

As the table shows session lifetimes are the same regardless of LoA. This may, however, change with future releases in which case the service providers will be notified in advance.

6.4 Support for multiple sessions

With the introduction of the person and professional profiles in OIOSAML 3 and support for private service providers, it has become necessary for NemLog-in to support multiple simultaneous sessions for a single user/browser.

Consider, for example, the following scenario:

1) User logs into service provider A using person profile. Session A is established.
2) User visits private service provider B that requests authentication and the professional profile (employee log in).
3) NemLog-in performs a new authentication (private service providers are not allowed to participate in SSO) and establishes a new session B (employee).

Since the user has not requested logout, it would not be reasonable to terminate session A during step 3). After step 3) the user thus has two sessions with NemLog-in: Session A with service provider A (as private) and session B with service provider B (as an employee).

Now the user selects 'Log out' at service provider A. This will cause NemLog-in to, in turn, terminate both sessions and in that process send a logout request (according to binding specified in service provider B metadata) to service provider B.

6.5 All service providers must support single logout

As the example in section 6.4 above illustrates, private service providers must support SAML logout according to OIOSAML 3 even though they are not allowed to participate in single sign-on.

When a user logs out of any service provider, NemLog-in will ensure, that all sessions established by the users browser with NemLog-in are terminated, and all service providers participating in these sessions are notified according to the SAML protocol.

This design ensures that the user is not required to be aware of the nature of the individual service providers: When the user choses to log out, all sessions established with NemLog-in are terminated.

6.6 Migration of user accounts

Service providers currently using OIOSAML 2 would usually register user accounts to PID (for private users). These service providers must implement a migration process to associate such user accounts to the new OIOSAML 3 UUID-based subject name identifiers (e.g.

https://data.gov.dk/model/core/eid/professional/uuid/123e4567-e89b-12d3-a456-426655440000).

Since the new subject name ids cannot be obtained up front, the migration process must be implemented in the service authentication code. The process is sketched below.

1) Request the RID attribute in metadata when the service is registered.
2) Request the *persistent* name id format to make sure, that the same subject identifier is received whenever the user logs in.
3) Change the data model to allow user accounts to be associated to a subject name id.
4) When the OIOSAML 3 assertion is received from the authentication process, check if an account is registered to the persistent subject name id.

6.7 A similar process must be used to migrate private user accounts associated to PID.Migration of privileges

The NemLog-in FBRS model for normal privileges is sketched below.

A Service may have 0-many privileges associated. A privilege is owned by exactly one Service. Privileges from different services are collected into Roles. The figure does not comprise the special model for sharing privileges between services.

The Service Provider decides which privileges are used and understood by her Service, the NemLog-in Administration is responsible for managing Roles.

When services are migrated to OIOSAML3 they must be connected to NemLog-in as new services. This means that it is necessary to add equivalent privileges to the new service in NemLog-in Administration and subsequently add these privileges to all roles containing their OIOSAML2 counterparts to ensure that end users receive the same privileges for the service, regardless of whether the OIOSAML2 or OIOSAML3 integration handles the authentication.

The situation is illustrated by the figure below.

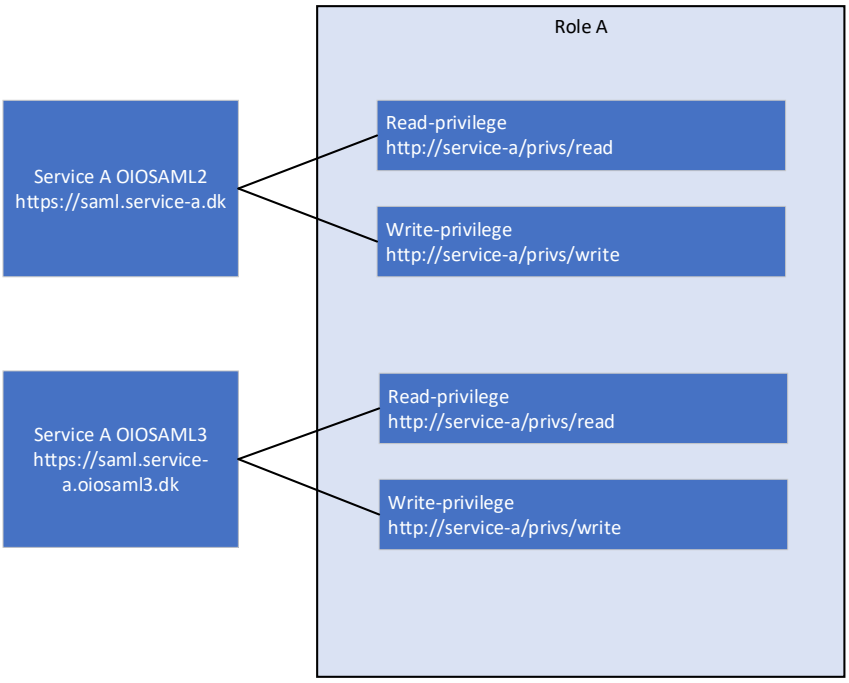

FIGURE 6: SERVICE USES PRIVILEGES

A service provider owns “Service A”. The SP prepares migration to OIOSAML3 by creating a new service in NemLog-in Administration – denoted “Service A OIOSAML3” in the figure. This service is assigned a new EntityID “https://saml.service-a.oiosaml3.dk” since EntityIDs must be unique in NemLog-in.

The service makes use of two privileges; a “read” and a “write” privilege. These privileges are associated to Role A. When configuring “Service A OIOSAML3” the service provider creates two new privileges associated to “Service A OIOSAML3” and assigns the *same* URIs as used for the existing OIOSAML2 privileges. Note, that the current version of NemLog-in Administration requires that privilege URIs are globally unique. This constraint is lifted in the upcoming version allowing URI reuse.

Note, that at this point the new privileges are not yet added to Role A, this is handled by the NemLog-in administrators at Digitaliseringsstyrelsen.

This example illustrates the situation for *normal* privileges but the situation is equivalent for delegation privileges (“fuldmagtsprivilegier”).

6.7.1 Migration process for services using privileges

When public service providers migrate their services to OIOSAML 3 they must register the OIOSAML 3 endpoint by adding a new service in NemLog-in Administration. Public service providers that use FBRS privileges (“normal” and/or “delegation” privileges) must carefully follow the process described below when migrating their OIOSAML 2 services to OIOSAML 3. The process is referred to as *privilege migration.*

1) Implement OIOSAML support in the service. *Do not* change the privilege names or semantics of the privileges used in the application.
2) In NemLog-in Administration, add a new service of type Web SSO (OIOSAML3). Use a new EntityID for the service and add metadata.
3) Define privileges for the service - use the same names as used in the OIOSAML2 service configuration - and provision the service to Integrationtest.
4) Test the integration, then fill in and upload the testreport. Apply for Production.
5) Contact DIGST Systemforvaltning and request that existing roles using privileges for the OIOSAML 2 service are extended with privileges for the new OIOSAML 3 endpoint *without version change.*
6) DIGST Systemforvaltning confirms that privileges have been migrated.
7) Enable the OIOSAML 3 integration in your own production environment. (Optionally only for pilot users).
8) (Optional): Conduct a pilot test to verify that roles are indeed migrated correctly.

Note, that it is an important prerequisite, that neither privilege names nor privilege semantics are changed in the process.

7 NemLog-in changes in OIOSAML 2.1.0 interface

This section summarizes important changes from OIOSAML 2.0.9 to OIOSAML 2.1.0. Note that the OIOSAML 2 interface is only available to public service providers.

7.1 Reduced attribute set

Public service providers connected to NemLog-in are connected according to the OIOSAML 2.0.9 specification. The OIOSAML 2.0.9 provides a set of attributes that are closely related to the NemID certificates. With the introduction of MitID where no end-user certificates are involved in authentication it is necessary to remove the support for these attributes. The new OIOSAML 2.1.0 [OIOSAML2.1.0] profile describes the new reduced attribute set.

The following attributes are no longer supported in OIOSAML 2.1.0:

- UserCertificate (urn:oid:1.3.6.1.4.1.1466.115.121.1.8)
- Certificate issuer (urn:oid:2.5.29.29)
- (Certificate) serial number (urn:oid:2.5.4.5)
- IsYouthCert (dk:gov:saml:attribute:IsYouthCert)
- UniqueAccountKey (dk:gov:saml:attribute:UniqueAccountKey)
- Postal address (urn:oid:2.5.4.16)
- Title (urn:oid:2.5.4.12)
- Organization unit (urn:oid:2.5.4.11)
- UserAdministratorIndicator (dk:gov:saml:attribute:UserAdministratorIndicator)

Public service providers are still allowed to upload metadata with the unsupported attributes, but NemLog-in Administration will give a warning.

All systems that are already connected to NemLog-in must be able to support OIOSAML 2.1.0 when NemLog-in3 goes live.

7.2 New assurance level value

With the advent of MitID, public service providers that integrate with NemLog-ins OIOSAML 2.1.0 interface may experience SAML assertions passing an Assurance Level value of “2” instead of the usual value “3”. An example of such a situation is described below.

Consider two service providers:

A. Public service provider (participating in SSO) using OIOSAML 3
B. Public service provider (participating in SSO) using OIOSAML 2

And the following flow:

1) User visits a resource at service provider A.
2) Service provider requires NSIS LoA Low to grant access to that resource and redirects the user to NemLog-in requesting that assurance level in the <AuthnRequest>.
3) User has no active session with NemLog-in and is thus requested to authenticate. User selects MitID and authenticates at assurance level Low
4) User is redirected to service provider A with SAML assertion stating LoA Low.
5) User uses the same browser to access resource at service provider B
6) Service provider requires authentication to allow access to that resource and redirects the user to NemLog-in and does not request ForceAuthn = true in the <AuthnRequest>.

7) NemLog-in discovers that the user has an active session and since authentication is not forced immediately issues a SAML assertion for service provider B. No numerical Assurance Level is (well)defined for the session, thus requiring NemLog-in to translate an NSIS LoA to a numerical (non-NSIS) assurance level. Since LoA is Low for the current session, NemLog-in responds with an assertion claiming Assurance Level = "2".

In general, NSIS assurance levels are translated according to the table below.

NSIS LoA	Corresponding OIOSAML 2 Assurance Level
Low	2
Substantial	3
High	3

Table 1: Translation of NSIS LoA

Note that both NSIS LoA Substantial and High are translated to "3": Service providers must use OIOSAML 3 to distinguish Substantial and High.

8 Metadata definitions

The SAML service provider metadata specifies information needed by NemLog-in to authenticate users for the service provider. We refer to [OIOSAML3] for details, and to the NemLog-in portal for examples.

This section emphasizes NemLog-in specific details regarding metadata.

8.1 NameID format

With NemID we were used to identifying users by the familiar global identifiers: PID for private identities and {CVR, RID} for employee identities. In OIOSAML 2 these identifiers enter the Subject NameID by the profiles use of the X509SubjectName NameID format.

To enhance privacy OIOSAML 3 introduces non-global identifiers instead. Service providers must choose between a *transient* and a *persistent* subject identifier. The transient identifiers are session specific: With each new session, a new transient identifier is generated. The persistent identifiers are service provider specific: With each new session, assertions issued for the same identity will contain the same persistent identifier. However, when the user logs in at a different service provider, another persistent identifier is used (if that other service provider also uses persistent identifiers).

The service provider must specify the desired NameIDFormat in the metadata XML file that is registered with NemLog-in.

8.2 Requested attributes

The service provider must also specify the set of attributes that he wishes to receive in the assertion issued during authentication in the metadata XML file.

Note, that specifying an attribute in metadata does not guarantee that the attribute will be present in all assertions. We refer to section 10 for details.

Private service providers are not allowed to receive CPR and Privilege attributes. If a private service provider attempts to register a set of metadata containing these attributes with NemLog-in Administration, the metadata will be rejected.

Private service providers that need access to CPR must instead request the PID attribute in metadata, obtain the user CPR e.g. by asking the user to enter it in the UI, and use the NemLog-in PID-CPR match service to verify [PIDCPR].

The NemLog-in Attribute Query Service allows all attributes in the attribute profiles to be requested with the following exceptions:

- Private service providers are not allowed to request CPR or Privilege attributes.
- No service providers are allowed to request Assurance Level, NSIS LoA, NSIS IAL, or NSIS AAL attributes.

The optional attributes NSIS IAL and NSIS AAL must instead be obtained during authentication.

8.3 Encryption algorithms

NemLog-in supports the encryption algorithms allowed in the OIOSAML 3 specification, i..e. AES-CBC and AES-GCM for block encryption (with various key sizes) and RSA-OAEP-MGF1 or RSA-OAEP for key transport. See [OIOSAML3].

By default NemLog-in will use the strongest ciphers (AES-GCM-256 and RSA-OAEP) when encrypting the SAML assertion to service providers. If your system does not support these algorithms you may specify an <EncryptionMethod> element in the your metadata file to use alternative algorithms (AES-CBC and/or RSA-OAEP-MGF1). See example below.

```
<KeyDescriptor use=”encryption”>
  <ds:KeyInfo>…</ds:KeyInfo>
    <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc"/>
    <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p"/>
</KeyDescriptor>
```

TABLE 2: KEYDECRIPTOR ELEMENT WITH ENCRYPTIONMETHOD SPECIFICATION

9 Authentication requests

The overall requirements for requesting authentication are described in the OIO profiles, e.g. [OIOSAML3] section 4.1.

The OIOSAML 3 profile allows the SP to tailor the specific authentication flow to his needs. SP may request a particular user type (attribute profile), either private or professional, and SP may request authentication with a particular NSIS assurance level. Furthermore, the SP may request a specific local IdP perform the authentication. The options are described in the following.

9.1 Requesting an attribute profile

The service provider may specify a specific attribute, this is done by adding a <RequestedAuthnContext> element to the AuthnRequest as specified in [OIOSAML3], OIO-SP-07. NemLog-in will restrict authentications to identities of the requested type, but the service provider MUST nevertheless validate, that the Subject NameID element corresponds to the requested profile, as described in [OIOSAML3] requirement OIO-IDP-15.

If the user already has a session with NemLog-in and a new public service provider requests unforced authentication, a single signon occurs (see section 0 above) even though the previous authentication does not satisfy the profile requested. In this case it is the service providers responsibility to act accordingly and e.g. instruct the user to log out and log in again.

9.2 Requesting assurance level

Similar to requesting a specific profile, the service provider may request a specific NSIS assurance level (LoA) as described in [OIOSAML3], OIO-SP-06.

When a specific NSIS LoA is requested, NemLog-in will attempt to obtain an authentication satisfying that request.

If no NSIS LoA is requested it is equivalent to requesting NSIS Substantial.

To ensure a smooth transition to MitID, NemLog-in will allow authentications satisfying OIOSAML 2 Assurance Level “3” when NSIS Substantial is requested. I.e. if NSIS Substantial is requested NemLog-in will aim at providing an authentication satisfying either NSIS Substantial or High or Assurance Level “3”.

It is very important to emphasise that the service provider *must always* verify that the assurance level conveyed in the SAML assertion is sufficient to allow the user to access the requested resource, *regardless* of the assurance level requested!

If the service provider wishes to allow MitID users to authenticate, the appropriate choice is to request NSIS Substantial and accept SAML assertions with either Assurance Level “3”, NSIS LoA Substantial or NSIS LoA High.

9.3 Forced authentication

The service provider may force a user authentication by setting the forceAuthn attribute in the authentication request. See [OIOSAML3].

9.4 OneTimeUse condition

The service provider may specify a OneTimeUse condition in the AuthnRequest, it must be specified alongside the forceAuthn attribute. It must be added as a SAMLCondition and have the following format:

```
<saml:Conditions>
  <saml:OneTimeUse />
</saml:Conditions>
```

Specifying the OneTimeUse condition will result in the session being created and set to expire immidiately. The result being that any subsequent logins will trigger a new login flow. Thus the login will not be reusable.

9.5 Passive authentication

A public service provider may instruct NemLog-in to not take control over the user interface by requesting a so-called passive authentication, allowing a single sign-on authentication to occur if the user has a valid session. If no session exists such a request is rejected with the NoPassive error message.

Since private service providers are not allowed to participate in SSO they are not allowed to request passive authentication. All requests for passive authentication from private service providers are rejected with the NoPassive error message. We refer to [OIOSAML3] for details.

9.6 Embedded authentication

Note that the OIOSAML standard as well as DIGST policies don't allow authentication requests to be issued inside an HTML frame or via any mechanism that would require the use of third-party cookies by the IdP to establish or recover a session with the browser.

This will typically imply that requests will involve a full-frame redirect, in order for the top-level window origin be associated with the IdP. The only allowed exception is passive logins (see section 9.4 above).

In addition, when authentication is performed in a Native App on a mobile device, the use of web views is not allowed since web views do not provide sufficient isolation between App and web context, and typically do not allow the users to inspect the address bar before entering their credentials.

Instead, authentication should be performed in an external browser. On iOS this could be achieved using the SFSafariViewController class or SFAuthenticationSession, and on Android using the Android Custom Tab feature.

See RFC 8252 (https://datatracker.ietf.org/doc/html/rfc8252) for details and references to open-source samples.

9.6.1 Open external browser in-app

9.6.1.1 Android

Custom Tabs can be implemented by following the Android's implementation guide [CUSTAB]. Below is a code snippet for how to open a URL in a Custom Tab.

```
CustomTabsIntent.Builder builder = new CustomTabsIntent.Builder();
CustomTabsIntent customTabsIntent = builder.build();
customTabsIntent.launchUrl(MainActivity.this, Uri.parse("SP_URL"));
```

9.6.1.2 iOS

Opening the SFSafariViewController is as simple as instantiating it with a URL and presenting it.

```
Guard let url = URL(string: "https://sp.app.site/page.html) else {
    return
}
let safariVC = SFSafariViewController(url: url)
self.navigationController?.pushViewController(safariVC, animated: true)
```

9.7 Mobile app-switch

NemLog-in is designed to target support for mobile app-switch using either App Links on Android [AndAppLinks] or Universal Links on iOS [UniLinks]. To this end the service provider must provide NemLog-in with the platform and return URL, where the return URL is the URL of the service provider app. The return URL is used as the return address for app-switch back to the service provider app when authentication finishes in third party apps like MitID.

If no return URL is provided by the service provider, then the end-user will have to manually go back to the original app when using MitID. For local IdPs the behaviour will depend on the implementation of the local IdP, but NemLog-in will communicate the platform and return URL through the SAML AuthnRequest send to the local IdP.

9.7.1 Return URL communication

To enable automatic app-switch back to the service provider app the service provider must as part of the SAML AuthnRequest send the platform and return URL. To this end a SAML extension, AppSwitch, must be used as data transfer object for these two pieces of information. The AppSwitch XML element takes the form as shown in example below.

```
<nl:AppSwitch xmlns:nl="https://data.gov.dk/eid/saml/extensions">
    <nl:Platform>Android</nl:Platform>
    <nl:ReturnURL>dk.serviceprovider.test</nl:ReturnURL>
</nl:AppSwitch>
```

The value for the Platform-element can either contain the value Android or iOS. The value of the ReturnURL-element must be the Universal Link or Android App Link URL of the service provider app. Prior to forwarding SAML AuthnRequest to NemLog-in the service provider should perform XML schema validation. An XML schema for validation of the AppSwitch-element SAML extension can be found in Appendix A.

A complete SAML AuthnRequest with the AppSwitch extension will take the form as shown in the example below.

```
<?xml version="1.0"?>
<samlp:AuthnRequest
    ID="id9eb5dd256c25461584a2796994feab1d"
    ...
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
  <saml:Issuer>https://sp3.dev-nemlog-in.dk</saml:Issuer>
  <samlp:Extensions>
    <nl:AppSwitch xmlns:nl="https://data.gov.dk/eid/saml/extensions">
      <nl:Platform>Android</nl:Platform>
      <nl:ReturnURL>dk.serviceprovider.test</nl:ReturnURL>
    </nl:AppSwitch>
  </samlp:Extensions>
  ...
</samlp:AuthnRequest>
```

9.7.2 Test MitID app presence on device

The service provider can choose to check if the MitID app is present on the device prior to sending the SAML AuthnRequest with the AppSwitch-extension. Below this check is described for each of the two supported platforms.

9.7.2.1 Android

In Android explicit intents can be used to switch to another app and we can reuse this approach to check if the MitID app is installed on the device.

```
public boolean deviceHasMitIDApp() {
    try {
        getPackageManager().getPackageInfo("dk.mitid.app.android", 0);
        return true;
    } catch (PackageManager.NameNotFoundException e) {
        return false;
    }
}
```

From Android 11 package visibility for apps changed such that intent to query for a specific app also must be stated in the manifest.

```
<manifest ...>
    <queries>
        <package android:name="dk.mitid.app.android" />
    </queries>

    <application .... />
</manifest>
```

9.7.2.2 iOS

On iOS it is possible to query for the URL scheme of the MitID app (“mitid-app”) to check if the MitID app is installed on the device.

```
func canOpenMitIDApp() -> Bool {
    guard let url = URL(string: "mitid-app://") else {
        return false
    }
    return UIApplication.shared.canOpenURL(url)
}
```

The app must declare the intent to query for this scheme by adding the URL scheme to the Info.plist of the app under the key LSApplicationQueriesSchemes; see Apple Developer documentation [iOScanOpenURL] for details.

9.8 Local IdP identification

A service provider can request a specific local IdP be used for the authentication either with some prior knowledge of EntityID of the local IdP or in the case of a broker the broker can retrieve a list of available local IdPs using the Identity Service API.

The identification of the local IdP is communicated to NemLog-in using the SAML IDPList-element as part the SAML AuthnRequest. Below is an example of the XML used as part of the SAML AuthnRequest.

```
<samlp:AuthnRequest ...>
  <saml:Issuer>https://saml.service-provider</saml:Issuer>
  ...
  <samlp:Scoping>
```

```
    <samlp:IDPList>
      <samlp:IDPEntry ProviderID="https://saml.idp.organization" />
    </samlp:IDPList>
    ...
  </samlp:Scoping>
</samlp:AuthnRequest>
```

Generally, if the SAML AuthnRequest identifies a local IdP no UI of NemLog-in will be shown. NemLog-in's UI will only be shown in the case of an error in the authentication flow between NemLog-in and the local IdP.

The SAML AuthnRequest from the service provider is validated and can result in three SAML Responses with status "Requester". The message of the Response will contain on the following three values:

- NLIDP-201 IDPList with multiple Local IdP entries specified.
- NLIDP-202 Identified Local IdP not found.
- NLIDP-203 Identified Local IdP does not support requested LoA.

The service provider must ensure that:

- the IDPList-element precisely contains one IDPEntry (NLIDP-201),
- the ProviderID-attribute of the IDPEntry-element contains SAML EntityID of a known to NemLog-in local IdP (NLIDP-202), and
- the FAL of the local IdP (the notified assurance level of the system) identified is the same level or higher than the requested LoA (NLIDP-203).

9.9 Providername

As of OIOSAML 3.0.3 ProviderName is now mandatory for Proxy IdP's, which is to be defined in the AuthnRequest. See [**OIO-SP-09**] in [OIOSAML3] for more details.

To support special characters the ProviderName must be defined as an UTF-8 string Base64 encoded in the attribute. It must consist of between 2 and 100 characters. The following character set is allowed:

- Letters and numbers including ÆØÅ
- Special characters: .,()-/
- Blank space

10 Authentication responses

This section describes topics that the service provider must consider when handling responses. Note, that the service provider must conform to formal requirements, processing rules, etc. specified in [OIOSAML3].

10.1 Attribute profile

With the introduction of the two different attribute profiles, it is important that the service provider validates the attribute profile by inspecting the Subject NameID - see [OIOSAML3], OIO-IDP-15 – to ensure that an assertion for a user with the expected profile is in fact received.

10.2 Assurance level

Not all identification means supported by NemLog-in comply with the NSIS standard. When these are used for authentication NemLog-in will issue an assertion without NSIS attributes. Instead, the assurance level is specified in terms of the OIOSAML 2 assurance level attribute:

```
<saml:Attribute
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
   Name="dk:gov:saml:attribute:AssuranceLevel">
   <saml:AttributeValue xsi:type="xs:string”>3</saml:AttributeValue>
</saml:Attribute>
```

We will in general distinguish the two assurance level attributes by referring to the OIOSAML 2 numeric attribute as Assurance Level and to the OIOSAML 3 attribute as NSIS LoA.

When the service provide receives the SAML assertion he is obliged to validate the assertion as specified in [OIOSAML3], and in that process, the assurance level attribute(s) must be validated.

We refer to the reference implementation documentation [OIOSAML-NET, OIOSAML-Java] for details regarding validation configuration.

10.2.1 Authentication Statement

In OIOSAML 3 the returned value of AuthnContextClassRef-element in the authentication statement (AuthnStatement-element) of the SAML Assertion will reflect whether an Assurance Level or a NSIS LoA was returned. The value returned is `urn:oasis:names:tc:SAML:2.0:ac:classes:X509` and `https://data.gov.dk/concept/core/nsis` respectively.

10.3 CPR attribute

Only public service providers are allowed to request the CPR attribute during authentication or by attribute query.

Private service providers must instead request the NemID PID attribute in metadata and use the NemID PID-CPR-match service. See [PIDCPR] for details.

10.4 Name attributes

Some NemLog-in identitites are anonymized (or pseudonymized). This is e.g. the case for citizens that are name and address protected.

For these identities the name attributes reflect this situation as shown in the table below.

Attribute	Value for anonymized identity	Value for non-anymized identity
Fullname	N/A	Full name for identity
Firstname	N/A	First name for identity
Lastname	N/A	Last name for identity
Alias	"Pseudonym"	N/A

TABLE 3: NAME ATTRIBUTES FOR ANONYMIZED IDENTITIES

We recommend requesting the Alias attribute and to use its value to distinguish anonymized identities.

10.5 OIO Basic Privilege Profile deviation

NemLog-in implements OIO Basic Privilege Profile v 1.1 [OIOBPP] but deviates from the profile definition of scope values ([OIOBPP] page 8).

NemLog-in uses the following values for specifying scope for a privilege:

- urn:dk:gov:saml:CvrNumberIdentifier:<CVR number>
- urn:dk:gov:saml:ProductionUnitIdentifier:<P number>
- urn:dk:gov:saml:SeNumberIdentifier:<SE number>
- urn:dk:gov:saml:CprNumberIdentifier:<CPR number>

I.e. the first character after "urn:dk:gov:saml:" prefix is upper case, not lower case as specified in [OIOBPP].

10.6 Attributes for private identities

Attribute availability for service providers for private identities depend on two parameters:

- The applied user credential
- Whether the person is anonymized

The anonymized person attribute set will be applied if:

- The person is name- and address-protected in CPR

Pseudonymized identities can – for test purposes – be created in the Integrationtest MitID Simulator test-tool [MitIDSim] by creating an identity with the option 'Non-disclosure of name and address' selected.

Attribute friendly name	Person MitID	Anonymized person MitID

Specification version	X	X
STS Bootstrap Token	X	X
Privileges	X	X
LoA	X	X
IAL	X	X
AAL	X	X
Full name	X	
First name	X	
Last name	X	
Alias		X
Email		
CPR	X	X
Age	X	X
CPR UUID	X	X
Date of birth	X	X
PID	X	X
Assurance Level[2]		

Table 4: Attributes available for private identities

For attributes, that cannot be delivered consistently (missing X's in rows above), an explanation is given below.

10.6.1 Email

The email attribute is available for NemID authentications if and only if the email-address is contained in the POCES-certificate (subjectAlternativeName). The email address for an anonymized person is not returned even if present in the NemID certificate.

Email-addresses are not available when the user authenticates with MitID.

10.6.2 Alias

The alias attribute will be returned for anonymized identities.

[2] The numeric dk:gov:saml:attribute:AssuranceLevel attribute from OIO SAML 2.

10.7 Attributes for professional identities

Availability of attributes for professional identities is given in the table below.

Note, that MitID Erhverv introduces the novel concept of anonymous identities. For these, a limited set of attributes is available.

Attribute friendly name	MitID Privat til Erhverv	MitID Erhverv user	Anonymous MitID Erhverv user
Specification version	X	X	X
STS Bootstrap Token	X	X	X
Privileges	X	X	X
LoA	X	X	X
IAL	X	X	X
AAL	X	X	X
Full name	X	X	
First name	X*	X	X
Last name	X*	X	X
Alias			
Email	X***	X***	X
CPR	X*	X	X
Age	X*	X	X
CPR UUID	X*	X	X
Date of birth	X*	X	X
Persistent Identifier			X
RID number	X	X**	X
CVR number	X	X	X
Organization name	X	X	X
Production unit	X		X

Attribute friendly name	MitID Privat til Erhverv	MitID Erhverv user	Anonymous MitID Erhverv user
SE number	X		X
Authorized to Represent		X	
Assurance Level[3]	X		

Table 5: Attributes for professional identities

* These attributes are only available for MOCES with a registered CPR.

** The RID attribute contains the PID identifier.

*** Only if the e-mail address is registered in FBRS or available from the MOCES certificate.

For attributes, that cannot be delivered consistently (missing X's in rows above), an explanation is given below.

10.7.1 Assurance level attributes (LoA, IAL, AAL, OIO SAML 2 assurance level)

These are only supplied for identities that are NSIS compliant.

10.7.2 Anonymous MitID Erhverv user

For anonymous MitID Erhverv identities, NemLog-in will never convey potentially privacy infringing attributes. Note, that this includes Authorized to Represent.

The Alias attribute is only available for Pseudonym identitites with the fixed value "Pseudonym".

10.7.3 Persistent Identifier

Persistent Identifier is currently only available for MitID Erhverv users.

10.7.4 Authorized to Represent

As mentioned, Authorized to Represent is not available for anonymous MitID Erhverv users since it is privacy infringing.

[3] dk:gov:saml:attribute:AssuranceLevel from OIO SAML 2.

11 Single Log Out

According to [OIOSAML3] section 4.2 all service providers must support single log out. Note, that this is also the case for private service providers although they are not allowed to use the NemLog-in single sign-on feature.

NemLog-in will convey log-out requests to all service providers participating in SSO, regardless of whether the particular service provider uses OIOSAML 2 or 3.

We refer to the reference implementations and the OIO profiles for details.

12 References

The references below will be updated in the near future.

Term	Reference
[OIOSAML3]	https://digst.dk/it-loesninger/standarder/oiosaml-profiler/
[OIOSAML-Java]	https://github.com/digst/OIOIDWS.Java
[OIOSAML-NET]	https://github.com/digst/OIOSAML.Net
[OIOSAML2.1.0]	https://digst.dk/it-loesninger/standarder/oiosaml-profiler/
[OIOIDWS]	https://digst.dk/it-loesninger/standarder/oio-identity-based-web-services-12-oio-idws/
[OIOBPP]	https://digst.dk/it-loesninger/standarder/oiosaml-profiler/
[PIDCPR]	https://www.nets.eu/dk-da/l%C3%B8sninger/nemid/nemid-tjenesteudbyder/supplerende-tjenester/pid-rid-cpr-tjenester/Pages/muligheder-med-pid-cpr-tjenesten.aspx
[NLSS]	https://tu.nemlog-in.dk/oprettelse-og-administration-af-tjenester/signering-kvalificeret/ny-funktionalitet/
[NLSPS]	https://tu.nemlog-in.dk/oprettelse-og-administration-af-tjenester/signering-kvalificeret/tekniske-egenskaber/dokumentation/
[MitIDSim]	https://mitidsimulator.test-nemlog-in.dk
[AndAppLinks]	https://developer.android.com/training/app-links
[UniLinks]	https://developer.apple.com/ios/universal-links/
[CustomTab]	https://developers.google.com/web/android/custom-tabs
[iOScanOpenURL]	https://developer.apple.com/documentation/uikit/uiapplication/1622952-canopenurl

Appendix A – XML schema for public SAML extensions

```
<?xml version="1.0" encoding="UTF-8" ?>
<schema
        targetNamespace="https://data.gov.dk/eid/saml/extensions"
        xmlns:publicExtensions="https://data.gov.dk/eid/saml/extensions"
        xmlns="http://www.w3.org/2001/XMLSchema"
        elementFormDefault="qualified"
        blockDefault="substitution"
        version="2.0">
    <element name="Platform" type="publicExtensions:AppSwitchPlatformType" />
    <simpleType name="AppSwitchPlatformType">
        <restriction base="string">
            <enumeration value="Android" />
            <enumeration value="iOS" />
        </restriction>
    </simpleType>
    <element name="ReturnURL" type="anyURI" />
    <element name="AppSwitch" type="publicExtensions:AppSwitchType" />
    <complexType name="AppSwitchType">
        <sequence>
            <element ref="publicExtensions:Platform" />
            <element ref="publicExtensions:ReturnURL" />
        </sequence>
    </complexType>
</schema>
```