Digitaliseringsstyrelsen

RETRIEVING CONSTRAINT VALUES THROUGH CALL-BACK SERVICE

Version: 1.2

Auther: Agency for Digital Government, NemLog-in

Published: April 2026

NemLog-in

Contents

Changelog

Date	Version	Change description	Initials
28-04-2026	1.2	Changed document design and added links to the order form and certificates	DIGST
04-03-2025	1.1	Updates to infrastructure instructions	NIRYHO
20-02-2024	1.0	Changed as per comments	SSOMM
08-02-2024	0.1	First draft	SSSOMM

1. Introduction

This document describes how service providers should implement call-back services for retrieving constraint values used in delegations.

1.1. Prerequisites

The reader is expected to be familiar with privileges and constraints as defined in [OIOBPP].

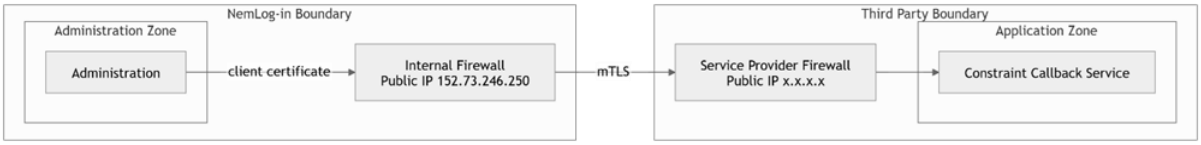
1.2. Intended audience

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

1.3. Terminology

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.
BPP	OIO Basic Privilege Profile is a way to describe privileges in a SAML assertion.

2. Infrastructure



This diagram illustrates the NemLog-in administration integration with external services for managing privileges and constraints. The Administration client in NemLog-in communicates securely with internal and external systems using mTLS client certificates.

Inside the NemLog-in boundary (IP: 152.73.246.250), the Administration client passes through an Internal Firewall to manage configurations and privileges.

The mTLS connection then crosses the Service Provider Firewall located within the Third Party Boundary (IP: x.x.x.x) to reach the Constraint Callback Service.

The Constraint Callback Service, contained in its own Application Zone, handles requests to retrieve or validate external constraint data.

2.1. Outgoing calls from NemLog-in

The Internal Firewall must allow outgoing connections to the Service Provider public IP. NemLog-in obtains this IP via an order form:

https://blanket.virk.dk/blanketafvikler/orbeon/fr/nem_v/85_2e113de92b0e192acbbcd870d99ced0c15c39a80/new?fr-language=da

2.2. Service Provider firewall and constraint validation

The Service Provider Firewall must allow incoming connections from the NemLog-in public IP (152.73.246.250). Additionally, the Constraint Callback Service must validate the client certificate presented by NemLog-in. A reference to the public part of the certificate can be found at: https://example.com/public-cert (The link can be updated when the certificate is available)

3. Call-back service implementation

To enable NemLog-in to retrieve constraint values for use in delegations from Service Providers IT-systems, the Service Providers can present an endpoint that NemLog-in can call to retrieve values. This endpoint will be called by delegation (Digital Fuldmagt og MitID Erhverv) upon a delegation being completed.

The service consists of a REST endpoint with one method: /getvalueset

The request query parameters will consist of the following:

Term	Definition	Description
constraintEntityId	String *Constraint EntityID (URN)* Required	The EntityId of the constraint as it is defined in the service providers IT-system. Example: urn:test:constraint
systemEntityId	String *System EntityID (URL)* Required	The EntityId of the system as it is defined in the service providers IT-system. Example: https://ping
privilegeName	String *Privilege Name (URN)* Required	The EntityId of the overlaying privilege as it is defined in the service providers IT-system. Example: urn:test:ping
scopeType	String *Scope type (CVR, SE, PU or CPR)* Required	Defines the scope type of the constraint. Example: CVR, SE, PU or CPR
scopeValue	String *Scope value* Required	Example: 1111111111 CPR-numbers must be without the dash

Two parameters are added to the header, they are both optional to implement for the Service Provider.

Term	Definition	Description
NemLogin-Calling-Identity-Language	String *Culture identifier*	The language culture the user is signed in with Example: 'da-DK'
CorrelationManager.CorrelationId	String *CorrelationID (UUID)*	The CorrelationID of the current operation, log for troubleshooting. Example: 858d8568-cc17-4620-81ea-a76dfb82830b

The response must consist of an array of ConstraintValueData consisting of the following properties.

Term	Definition	Description
ConstraintValueUuid	string (uuid)	The UUID returned must be unique to the specific constraint value. It will later be the value returned in the constraint attached to the users SAML assertion. Example: 2979bb52-d115-4d61-aeeb-d4037fea7332
ConstraintValueDescription	String	The description will be presented to the user creating the delegation. As such it should be meaningful for them in that context and make it easy for them to choose the correct constraint value. This value will only be presented to the user on entry and will not be persisted, only the UUID will. We recommend descriptions no longer than 120 characters to make facilitating the descriptions in the UI easier. Example: 24/12345 – Sag vedr. håndtering af xyz

An Open-API spec is provided with the guide (GetValueService.json). We recommend building a server stub from this specification to assure correct implementation of the service.

Also of note NemLog-in will call the service every time a user needs to fill out a constraint on a delegate, and will not provide any caching. If the call to the call-back service results in expensive operations in underlying systems with the service provider, it will be their responsibility to implement proper caching in their end.

3.1. Authentication

Calls to the service will originate from the NemLog-in backend via HTTPS. The endpoint must adhere to TLS protocol and encrypt the communication with a valid TLS certificate. Following one of the two TLS RFCs:

- RFC 5246: TLS Protocol Version 1.2
- RFC 8446: TLS Protocol Version 1.3

To ensure security in both ends NemLog-in backend will authenticate with an OCES3 client certificate issued to NemLog-in. By verifying this certificate the service provider can ensure that no unauthorized third party can gain access to the value service.

NemLogin will call with two different certificates depending on which environment: DevTest4 (PreProduction) or Production that is in question. The service providers must pin the authentication to the subject serial number of the certificates, this approach facilitate seamless certificate renewal processes without the need to issue entirely new certificates.

Download the certificates here:

https://cms.nemlog-in.dk/media/tbdgjawg/nemlog-in-fbrs-getconstraintvaluesetclient.zip

Please note that both certificates are in the zip-file.

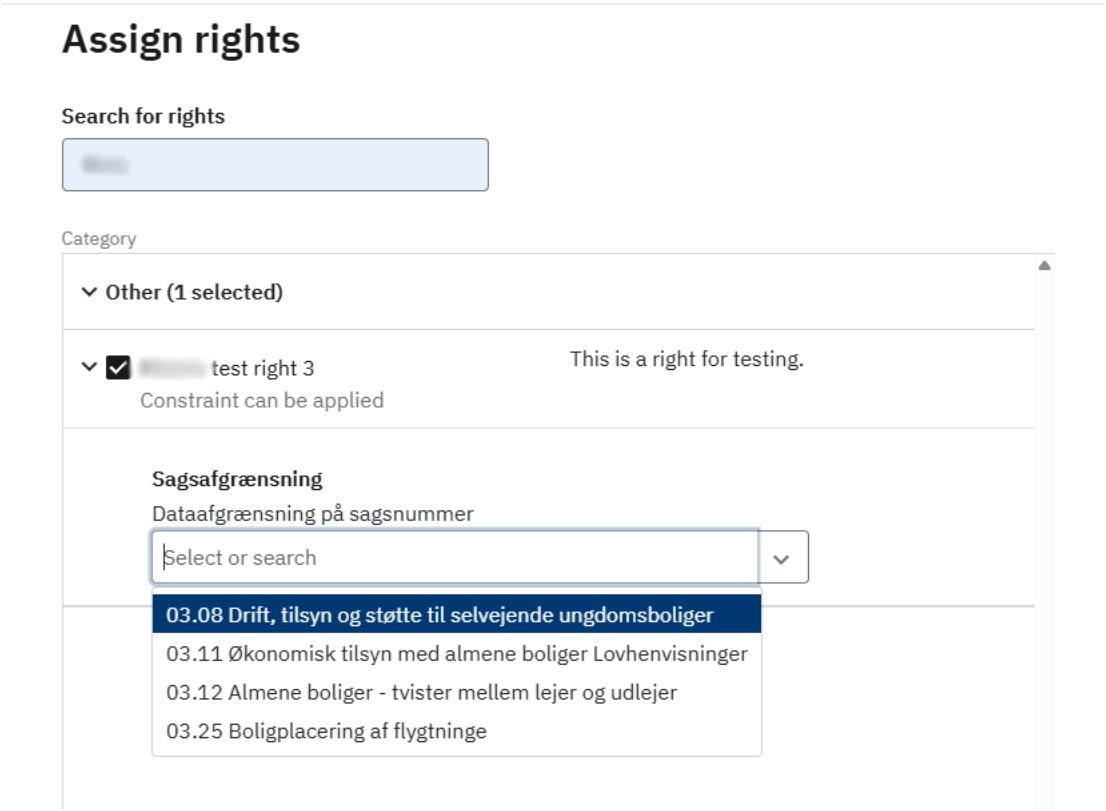
- NemLog-in FBRS.GetConstraintValueSetClient - Test - Exp.2028 (All environments except production)
- NemLog-in FBRS.GetConstraintValueSetClient - Exp.2028.cer (Production environment)

3.2. Setup and Validation

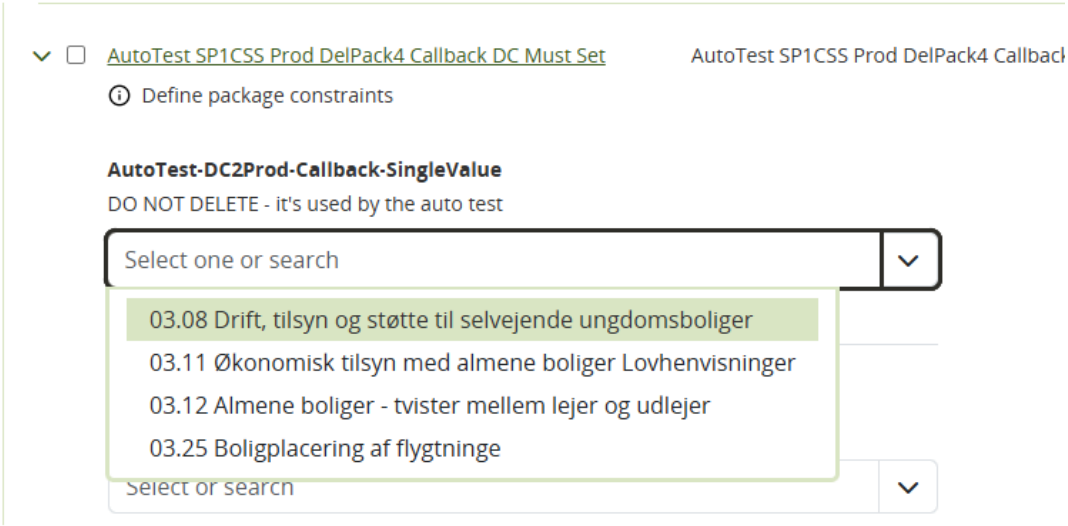
The constraint callback service is configured by the “forvalter” in the administration component. When ordering a new privileges, the service provider should define which constraint callback service delievers the constraints.

To facilitate more seamless development testing NemLog-in supplies a validation service for validating the service providers call-back services. It resides in the NemLog-in administrative portal where the constraint is set up itself, and thus the constraint can be set up and tested end-to-end.

- Test version provided by Nemlog-in -> Swagger ui link
- Note; that when ordering new privileges with constraints from a callback service, if you want to test without yet being ready with your own constraint service, then url you should order for is https://understoodconstraintcallbackservice.sp-devtest4-nemlog-in.dk//understoodconstraintcallbackdemo/getvalueset
- Privilege order form -> [HERE]
- On the "Assign Rights" page in MitID Erhverv Administration, you can select the constraint from the list.



Sample of the rights selection in MitID Erhverv when privilege points to a constraint callback service that is configured correctly.



Sample of the rights selection in Fuldmagt when privilege points to a constraint callback service that is configured correctly.

4. References

The references below will be updated continuously.

Term	Reference
[OIOBPP]	https://digst.dk/media/20999/oiosaml-basic-privilege-profile-1_2.pdf