Implementing a Nuts Use Case

This book describes how to implement a Nuts use case, from configuration, developing software to implement the use case's business rules, to enabling the use case for a subject.

- [Nuts Node Configuration](#)
- [Use Case Activation / Discoverability](#)
- [Requesting Access](#)

Nuts Node Configuration

A use case writer supplements you with a number of artifacts, that need to be configured:

- Service Discovery definitions
- Policy definitions

This page details how to configure each artifact.

Warning: do not alter these artifacts after receiving them, as they supply the Nuts node with the trust anchors as defined by the use case. Changes might lead to unsafe configuration.

Service Discovery Definitions

These JSON files instruct your Nuts node where to publish use case participation, and where to find other participants. They need to be provided in the Nuts node's discovery definitions directory (`discovery.definitions.directory`). The file name may be changed; it does not influence the definition. It's recommended to mount the files/directory in read-only mode if Docker is used.

Registration: after configuring the definitions, the Discovery Service for the applicable subjects through the Nuts node's VDR (v2) API, to make them discoverable.

Endpoints can also be discovered this way; definitions can specify additional parameters (in the form of a `DiscoveryRegistrationCredential`), for instance API endpoints, or OAuth2 Authorization Server URLs. **Note**: existing use cases use DID document services instead.

Policy Definitions

These JSON files instruct your Nuts node which credentials a participant needs to present (to your Nuts node) to acquire an access token with a certain scope. They need to be provided in the Nuts node's policy directory (`policy.directory`). The file name may be changed; it does not influence the definition. It's recommended to mount the files/directory in read-only mode if Docker is used.

Use Case Activation / Discoverability

After configuring the Nuts node with the required artifacts, the use case can be activated for a subject. This means activating the related Discovery Service for the subject, making them discoverable.

Pre-requisites:

- You have created a subject for the organization, for which you want to activate the use case.
- You have configured the Nuts node for the use case (Discovery Service definition and Policy definition).
- You know which endpoints are required to be registered during activation.

Acquiring credentials

The definition specifies which Verifiable Credentials the subject needs to register on the Discovery Service. These are often a combination of, but not limited to, the following credentials: `NutsOrganizationCredential`, `NutsUraCredential`, and/or `DiscoveryRegistrationCredential`.

These need to be issued by a trusted party, which depends on the use case. In development or test environments, parties are typically allowed to issue the credentials to themselves. In acceptance or production environments, they're normally issued by a trusted third party.

Self-issuance

In case, the use case allows self-issuance of credentials (e.g., because there's no trusted third party), you can issue the credential yourself.

To issue or load a credential, you need to choose one of the subject's DIDs. Use the following API call to list the DIDs of a subject (replace `<subjectID>` with the ID of the subject):

```
GET http://<internal Nuts API>/internal/vdr/v2/subject/<subjectID>
```

Then, you can issue the credential:

```
POST http://<internal Nuts API>/internal/vcr/v2/issuer/vc
Content-Type: application/json
```

```
{
  "issuer": "did:web:example.com",
  "type": "NutsOrganizationCredential",
  "expirationDate": "2020-12-09T16:09:53+00:00",
  "credentialSubject": {
    "id": "did:web:example.com",
    "organization": {
      "name": "Because we care B.V.",
      "city": "Arnhem"
    }
  }
}
```

Make sure you provide the proper values for the credential fields:

- Replace `issuer` and `id` with a DID of the subject
- Replace `type` and `credentialSubject` with the applicable values for the credential
- Replace the `expirationDate` with an appropriate expiration date (in RFC3339 format)

Note: alternatively, you can issue credentials through the [Nuts Admin](#) application.

After issuing, you need to load the credential into the subject's wallet. See the next section, "Trusted third party", for how to do this.

Trusted third party

A third party issues a Verifiable Credential to one of the subject's DIDs using an out-of-band mechanism.

The issuer can share the Verifiable Credential in any applicable way to the receiving party, including chat or e-mail. They can then be loaded into the subject's wallet using the following API call:

```
POST http://<internal Nuts API>/internal/vcr/v2/holder/<did>/vc
Content-Type: application/json


<Verifiable Credential>
```

- Replace `<did>` with the DID to which the credential was issued.
- Replace `<VerifiableCredential>` with the VC as it was received from the issuer.

Discovery Service Activation

To make a subject discoverable, activate the particular Discovery Service. The following example activates a service, supplying an additional FHIR endpoint to be used for the use case.

```
POST http://<internal Nuts API>/internal/discovery/v1/<serviceID>/<subjectID>
Content-Type: application/json


{
  "registrationParameters": {
    "fhir": "https://example.com/fhir"
  }
}
```

- Replace `<serviceID>` with the ID of the Discovery Service, which can be found in the definition in the `id` field.
- Replace `<subjectID>` with the ID of the subject

Note that if the subject doesn't have the required credentials in its wallet, the call might succeed, but actual registration won't happen until the required credentials are present. In this case, the call returns status 202 (`Created`), with a message detailing what failed. This is also the case if the Discovery Service is unreachable.

Requesting Access

To access APIs secured through Nuts, callers need an access token issued by the OAuth2 Authorization Server of the API owner. This page describes how to acquire an access token.

Requesting Service Access Token

This section describes which value(s) need to be specified in the service access token request.

- In the request URL:
  - `subjectID`: the ID of the local requester, which was provided by the Nuts node when the subject and its DIDs was created.
- In the request body:
  - `authorization_server`: the OAuth2 issuer URL of the party that grants access, found in the service discovery search result.
  - `scope`: specifies what resources the access token will give access to. This is specified by the use case.
  - `credentials` (optional): one or more credentials to provide to the authorization server that are not in the requester's wallet. This is typically used to provide an `NutsEmployeeCredential` to the authorization server. See the section below for how to provide this.
  - `token_type` (optional): by default, tokens are of type [DPoP](#) that mitigate token theft. Alternatively, the `Bearer` token type can be specified, but you'll be more vulnerable to MITM attacks.

Providing additional credentials

The service access token request allows you to supply credentials to the request, that are not in the subject's wallet but required for authentication. For instance, an `EmployeeCredential` that contains information about the current logged-in user for logging purposes. These credential don't need to be signed: in that case they will be "self-attested" (e.g., the `EmployeeCredential`); the Verifiable Presentation's signature will provide authenticity.

Example

The example below shows an example access token request with an `EmployeeCredential`.

```
POST http://<nuts private API>/internal/v2/auth/<subjectID>/request-service-access-token
Content-Type: application/json

{
```

```
  "authorization_server": "https://example.com/oauth2/hospital_x",
  "scope": "eOverdracht-sender",
  "credentials": [
    {
      "@context": [
        "https://www.w3.org/2018/credentials/v1",
        "https://nuts.nl/credentials/v1"
      ],
      "type": ["VerifiableCredential", "NutsEmployeeCredential"],
      "credentialSubject": {
        "name": "John Doe",
        "roleName": "Nurse",
        "identifier": "123456"
      }
    }
  ]
}
```