DID management

When you upgrade from V5 to V6, your DID documents will be migrated to a SQL database. A new VDR V2 API has also been added. It's no longer possible to set a controller on a DID document. All DID Documents are *self-controlled*. It's also no longer possible to fine tune key usage. You can add or remove a key from a DID document but can't control if it's used for changing the DID document or signing data. The only exception is encryption keys which will be added in a later version.

In V6 you register **subjects** not **DIDs**. Existing **did:nuts** identifiers will be migrated to subjects (with the same ID). Operations done on subjects through the VDR V2 API will affect all supported DID documents (currently web and nuts).

With V6, we also promote a different way of registering services: on the **discovery service**. Previously, you registered services in the DID document and might even use references between DID documents to control a set of endpoints. There's a international tendency to only register public keys in the DID document. Also with the introduction of **did:web**, DID documents are not signed. This difference is reflected by the VDR V2 API which won't add services to **did:web** documents, only to **did:nuts** documents.

Given all of the above, you might have a challenge if your adding a V6 style use case to your existing set of V5 use cases:

- Which DID methods do you enable?
- Should existing **did:nuts** subjects get a **did:web**?
- When and how to switch to the new VDR V2 API?
- When to switch from DID document services to service registration on the discovery service?

The V6 release notes state that you should not use the VDR V1 API together with the VDR V2 API. This is a general statement to prevent runtime problems. The V1 API only transforms **did:nuts** documents and not **did:web** documents. So if you're using **did:web** in use cases, you should no longer use the VDR V1 API.

The problem can be broken down in two parts:

- Create/Delete/KeyCreation operations on DID documents (subjects in V6)
- Service registration

Note: each time you restart the node, all changes to **did:nuts** will be migrated to **did:web** if they are not in sync.

Create/Delete/KeyCreation migration

The Create/Delete/KeyCreation operations are done with the following API calls:

- POST /internal/vdr/v2/subject
- DELETE /internal/vdr/v2/subject/{id}
- POST /internal/vdr/v2/subject/{id}/verificationmethod

After updating your node to V6 you may choose the moment you change these operations to the new VDR V2 API. After you've made the transition, restart the node once to make sure any **did:web** document is in sync with the **did:nuts** document for the subject. Any service registration in a **did:nuts** document will not be copied to the **did:web** document.

You might also want to change the designation of the identifier you store in your own DB, it used to be a DID, now it's a subject. After the transition you're also able to choose your own identifier.

Service registration

For service registration it's recommended to create a new separate management page for the discovery API. That way the old management page can use the VDR V1 service operations and the new management page can use the discovery API. Both can coexist, even for the same subject. It's then still recommended to switch the old management page to the new VDR V2 API.

Being able to register service on the discovery service does not solve the problem how a use case migrates from V5 style to V6 style services.

Recommendation

Given the little amount of work in migrating the VDR API and usage of **did:web**

- update your node to V6 with **did:web** disabled (at least for staging/production)
- switch all VDR V1 API calls to the VDR V2 API in a big bang
- enable **did:web**
- add a new management page for service discovery to support new use cases
- Never use DID Document service registration on use cases which support **did:web**

Revision #5
Created 5 November 2024 07:32:47 by Wout Slakhorst
Updated 5 November 2024 09:14:25 by Wout Slakhorst