

Caching in DID Resolution

Markus Sabadello, markus@danubetech.com

Introduction

In the DID Core specification, DID Resolution is defined as an abstract function with inputs and outputs. This is how DID Resolution is different from e.g. DNS Resolution, which is a single concrete protocol.

Additional aspects of DID Resolution are covered in the DID Resolution specification, and the “main” step (obtaining a DID document for a DID) is entirely method-specific, which is what makes DID architecture abstract, universal, and interoperable.

Various extensions to the DID Resolution process (e.g. resolution options, metadata) have already been listed in the DID Spec Registries and are being proposed in various places such as DIF’s list of extensions.

One topic in DID Resolution that has achieved little attention yet is caching. Should a DID resolver always retrieve a “fresh” version of the DID document by accessing the underlying verifiable data registry and executing the resolution steps defined by the DID method? Or can a DID resolver decide to cache the DID document (plus metadata) and return it without going through a “full” resolution process? This document mentions a few ideas as a basis for defining concrete DID Resolution extensions for a caching framework.

Main features

DID resolver configuration settings

The following are proposed configuration settings that control a DID resolver’s behavior:

- `cacheEnabled` = true/false (default is true)
- `cacheMaxTtl` = max. time-to-live in seconds, i.e. how long can a DID document for a DID be cached (default is 3600)

Whenever a DID is resolved, the DID document is first looked up from the cache. If it exists in the cache and its cache time is below the max. time-to-live time, then the cached DID document is returned and no actual resolution happens (no verifiable data registry is accessed). Otherwise, the DID is resolved as usual from the underlying verifiable data registry, and the resulting DID document is added to the cache.

DID resolution options

The following is a proposed new DID resolution option related to caching:

- `noCache` = true/false (default is false)

If this DID resolution option is passed to a DID resolver, then the DID document is always resolved as usual from the underlying verifiable data registry, and it is not looked up from the cache, or added to the cache.

DID document metadata

The following is a proposed list of DID document metadata properties related to caching:

- `cacheEnabled` = true/false (default is false)
- `cacheMaxTtl` = max. time-to-live in seconds, i.e. how long can a DID document for a DID be cached (default is 3600)

If these DID document metadata properties are returned after resolving a DID from its verifiable data registry, then they “override” the global resolver-wide configuration settings of the same name. This allows a DID controller to control caching behavior of “their” DID, i.e. a DID controller can disable caching of their DID, or set a custom time-to-live for their DID (similar to DNS).

Potential additional features

Excluding DID methods

Potential additional DID resolver configuration setting:

- `cacheExcludeMethods` = (default is “key”)

This means that the cache would not be used for certain DID methods such as `did:key`.

Disallowing DID document metadata

Potential additional DID resolver configuration setting:

- `cacheIgnoreDidDocumentMetadata` = true/false (default is false)

If this is set to true, then the cache-related DID document metadata properties are ignored, and only global resolver-wider configuration settings apply.

Active caching

Potential additional DID resolver configuration setting:

- `cacheActiveDids` = (default is empty list)
- `cacheActiveInterval` = the interval of active caching in seconds (default is 3600)

This contains a list of DIDs that are pro-actively resolved and cached on a regular basis, via some kind of background thread, and before actual client resolution requests come in.

Acknowledgements

Part of Danube Tech's work on DIDs, SSI, etc. has been supported by eSSIF-Lab.