Thursday 21 March 2019

Attending

- Markus Sabadello
- Jonathan Holt
- Drummond Reed
- Joe Andrieu
- Michael Herman
- Stephen Curren
- Dmitri Zagidulin
- Dan Burnett
- Nader Helmy
- Amy Guy
- Adrian Gropper
- Chris Boscolo
- Yancy Ribbens

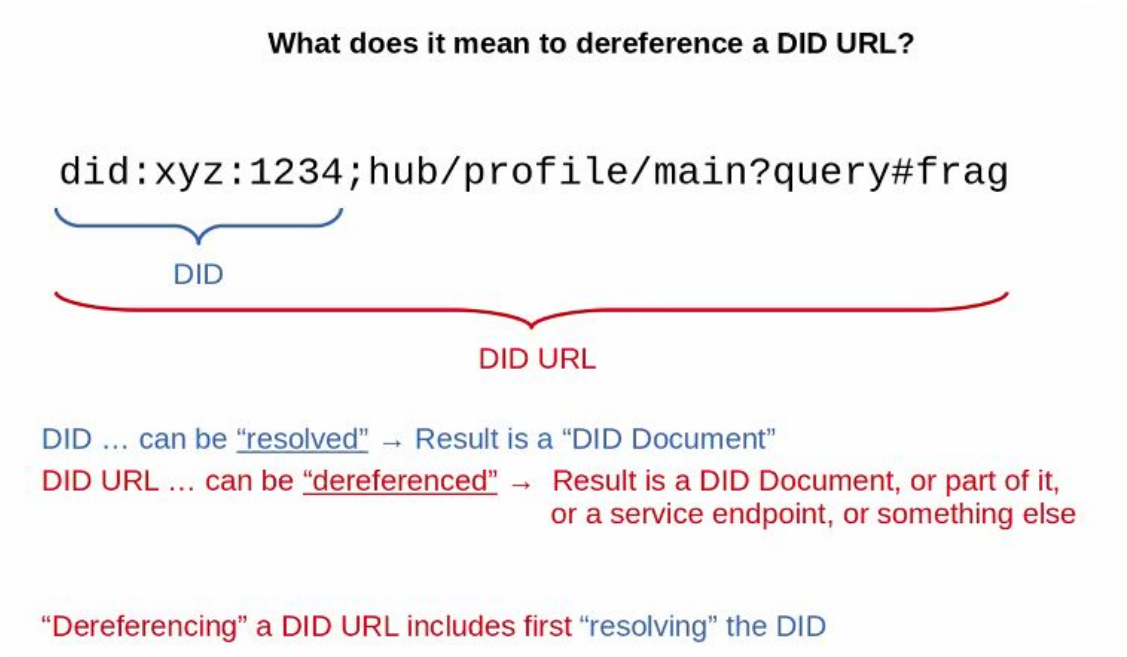
Agenda

1. Community governance: issue resolution and closure policies
2. What does it mean to dereference a DID URL?
3. Presentation of the above README: did-url-spec - Decentralized Identifier URL (did-url) Specification as well as the spreadsheet: https://github.com/mwherman2000/did-url-spec/tree/master/src (latest numbered version)

Meeting Notes

Topic #1: Understanding DID URLs and “Resolution” and “Dereferencing”

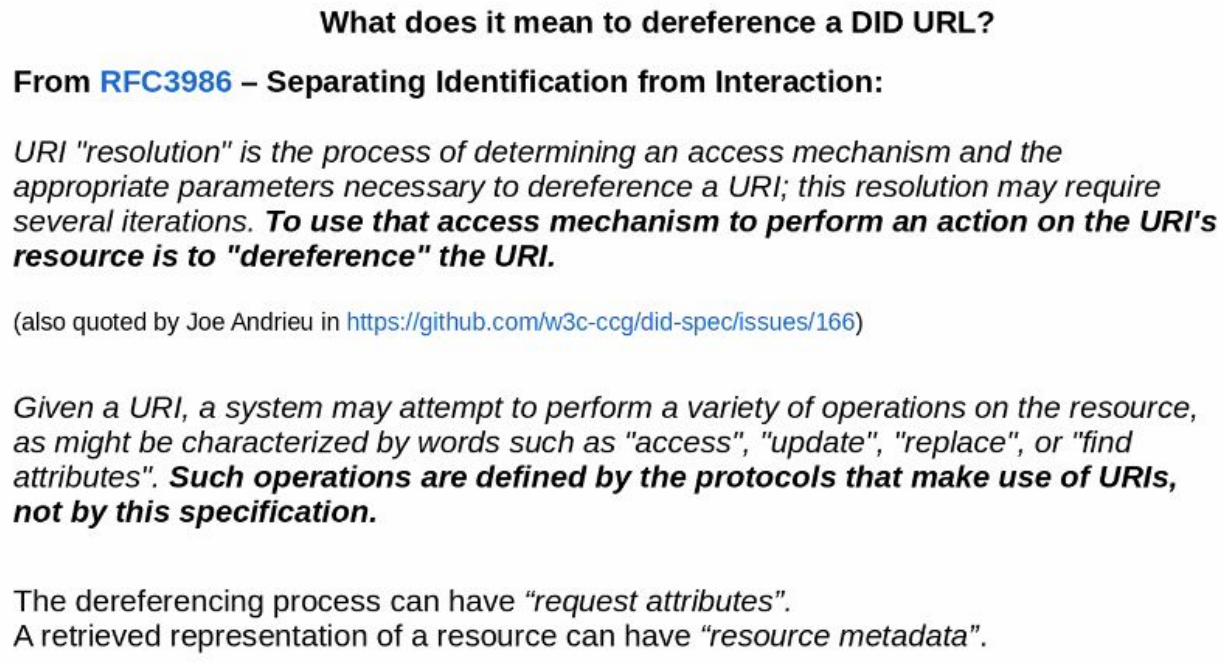
Markus explained the overall topic using this slide:



Michael pointed out that the "defererencing" category also splits into two:

1. A DID URL that dereferences to a subcomponent of a DID document.
2. A DID URL that undergoes a transformation to address a resource external to the DID document via a service endpoint.

Markus went on to explain the following slide:



Jonny asked how much DID resolution is actually parallel to DNS resolution.

Michael said that he has a set of proposals for how a set of DID resolution operations can work, e.g., how to query the capabilities of a resolver, or how to get a list of DID documents.

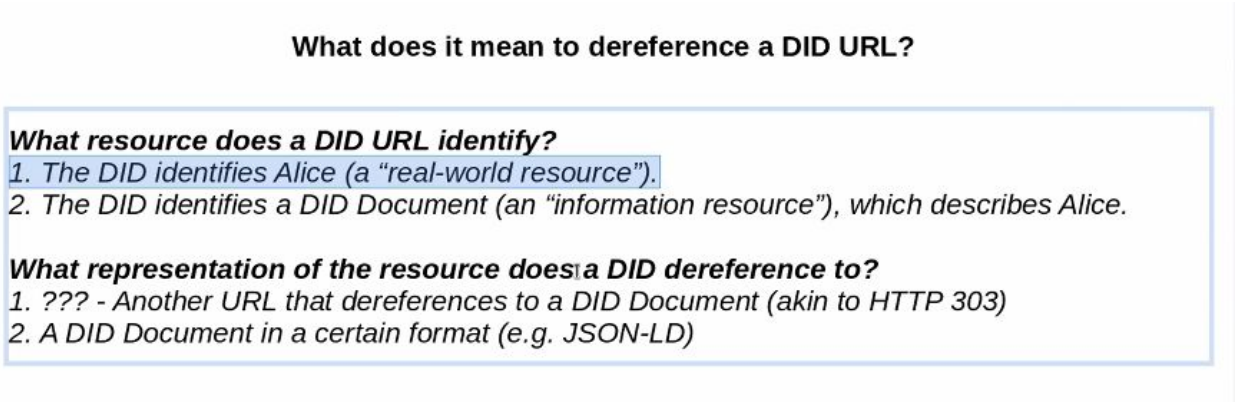
Michael whether returning a DID document is “access”?

Joe responded that access to a DID document is an open issue. Some proposals are to encrypt a DID document.

Drummond proposed that “DID resolution” be the set of steps that a DID method takes to return a DID document. He agreed with Joe that a particular DID method may in fact impose access control on obtaining a DID document. He then said that once a DID document is returned, you move into the “dereferencing” stage, and that can divide into the two types discussed above.

Dan: The DID document is not itself the resource, but a means of determining how to access the identified resource.

Markus next talked about this slide:



Drummond: Let’s avoid the HTTP Range 14 rathole! Let’s be clear that a DID never identifies a DID document! A DID identifies a DID Subject. That DID Subject MAY be a network-accessible resource. In that case a DID document describes how to access that resource. But the DID document may identify a non-network resource (like a person), in which case the DID document describes how to access service endpoints associated with the resource.

Jonny: IPLD is a type of resource that a DID may identify.

Joe: The resource when a DID is dereferenced is always on the network.

Markus spoke to this slide:

What does it mean to dereference a DID URL?

From WebID 1.0:

When using URIs, it is possible to identify both a thing (which may exist outside of the Web) and a Web document describing the thing. For example, the person Bob is described on his homepage. Alice may not like the look of the homepage, but may want to link to the person Bob. ***Therefore, two URIs are needed, one for Alice (a "real-world resource") and one for the homepage or a RDF document describing Alice (an "information resource").***

Joe and Drummond discussed the analogy with DNS and URL-identified resources. Drummond said that the DNS record that is retrieved in order to access the web page or RDF document. The DNS record is not the resource being identified by a URI. Therefore it is the equivalent of a DID document.

Markus next discussed this slide:

What does it mean to dereference a DID URL?

From HttpRange-14:

According to the HTTP specification, a response code of 303 indicates that "the response to the request can be found under a different URI ...". It provides the URI where we can look for that response.

The new URI is not a substitute reference for the originally requested resource.

→ A HTTP URI can dereference to a representation of the resource (HTTP 200).
→ A HTTP URI can dereference to another URI (HTTP 303).

Dan dove into this "HttpRange-14" issue by suggesting that a DID subject was not actually a person. "+1 that action on a DID for a physical resource could be to return a URL for a file ."

Drummond suggested that if a DID always identifies a DID subject, and that DID subject may be **either** a **real-world resource** (non-network-retreivable) or a **network resource**. If a DID identifies a real-world resource, then the DID document describes options for interacting either with that real-world resource via some type of network connection, or accessing other descriptions of that real-world resource. But the DID itself still identifies the real-world resource.

If the DID identifies a network-retrievable resource, then resolving the DID to the DID document will enable obtaining the actual URL that **also** identifies that network-retrievable resource. That means the DID and the URL would actually be “synonyms”, but at different layers of abstraction.

Drummond also said this addresses the “two URI” problem from Web ID. The DID that identifies Alice is URI #1, and the URL that identifies some resource describing Alice (**not** the DID document) is a second URI.

Joe mentioned that this means that a “naked DID” by itself should not be considered a URL. We should need to add a delimiter at the end of the naked DID to turn it into a URL for the DID document.

Drummond seconded that suggestion. It could be either the empty path, or the empty fragment.

Dan said that he had clarity that a DID document is about how to **describe how to interact with a resource**, but not describing the resource itself. That’s why a DID document is never the resource identified by the DID.