DID Spec Community Final Draft Meeting Page

This page is for taking notes of weekly meetings held in March & April 2019 of members of the W3C Credentials Community Group who are collaborating to complete the Community Final Draft of the DID specification. Meeting notes are listed in reverse chronological order.

Note: This meeting is directly followed by the weekly DID Resolution Spec First Draft Meeting.

Call Information

Time: Every Thursday, 13:00-14:00 PT (20:00-21:00 UTC)

IRC channel: http://tinyurl.com/w3c-ccg

https://zoom.us/j/7077077007

Or iPhone one-tap:
US: +16465588656,,7077077007# or +16699006833,,7077077007#

Or Telephone:
Dial (for higher quality, dial a number based on your current location):
US: +1 646 558 8656 or +1 669 900 6833
United Kingdom: +44 (0) 20 3051 2874 or +44 (0) 20 3695 0088
Meeting ID: 707 707 7007
International numbers available: https://zoom.us/u/q6mghCSZ

Links (Generally Useful to the Group)

- DID Spec
- DID Matrix Parameters Syntax Proposal

Thursday 4 April 2019

Agenda

1. Agenda creation/review/prioritization
2. Overarching goals and scope of DID URLs
3. Design principles (suggested by Michael Herman)
    a. is readability an important principle?

   b. is compatibility with and leveraging of existing standards an important principle? …if so, which standards? etc. etc.
   c. is the #OpenToInnovation principle important/a pre-requisite?
   d. is RFC 3986 compatibility a required principle?
   e. "how big is the tent" from a **did-uri** grammar perspective? …for example, does the proposal support transform options at the **did-root** and **did-method** as well as **did** levels in the grammar?
   f. Other principles?

4. Syntax/grammar/ABNF discussion
   a. Manu's proposal (colon delimited)
   b. Michael Herman's proposal (based on OData syntax)
   c. DID Matrix Parameters Syntax Proposal
5. Other features to discuss
6. Call/agenda planning
   a. April 11
   b. April 16
   c. April 25
7. Plan for a Community Final Draft
   a. We need to move forward the outstanding PRs

Notes

TOPIC: Agenda review

DID Grammar thread was started this last weekend. There are 25-30 messages on the thread.

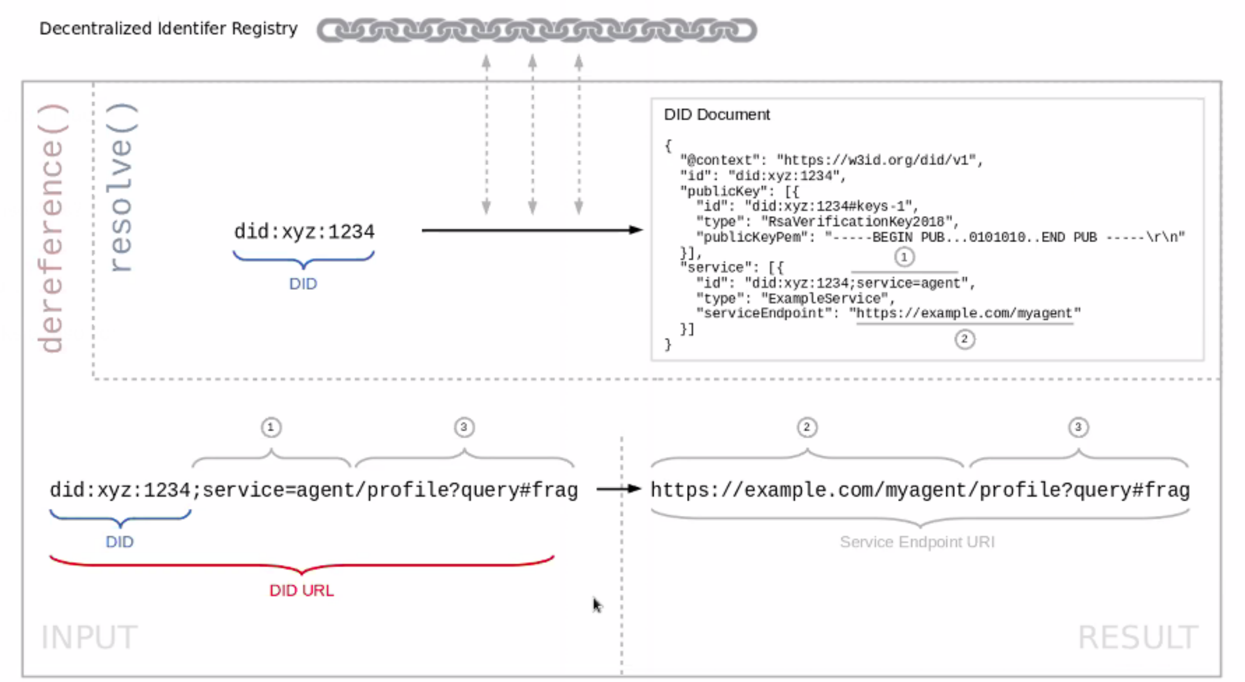
Michael Herman suggested some questions regarding design principles.

Jonny: What are the goals of the DID URL discussions? Do we need to arrive at consensus? Can we allow some issues to be determined in the DID method?

Drummond: Part of this could be handled in the DID resolution part of the second call.

Markus: I have prepared a few diagrams based on the ABNF to help guide what tasks a resolver performs.

TOPIC: Scope of a DID URL and How to Dereference It.


Decentralized Identifer Registry
dereference( )
resolve( )
did:xyz:1234
DID
DID Document
1
2
1
3
2
3
did:xyz:1234;service=agent/profile?query#frag
https://example.com/myagent/profile?query#frag
DID
DID URL
Service Endpoint URI
INPUT
RESULT

DID Document

```
{
  "@context": "https://w3id.org/did/v1",
  "id": "did:xyz:1234",
  "publicKey": [{
    "id": "did:xyz:1234#keys-1",
    "type": "RsaVerificationKey2018",
    "publicKeyPem": "-----BEGIN PUB...0101010..END PUB -----\r\n"
  }],
  "service": [{
    "id": "did:xyz:1234;service=agent",
    "type": "ExampleService",
    "serviceEndpoint": "https://example.com/myagent"
  }]
}
```

Markus: Diagram one shows a bare DID and a DID URL and how to resolve and dereference.

Jonny: There are security issues when the resolver has to apply a fragment.

Markus: I think it still works because you end up parsing the entire transformed service endpoint URI.

Jonny: I have to parse through the service array to find a corresponding id that ends with a matching "service=agent" as shown in the example.

Markus: I think that adding the key/value pair makes a more readily resolvable URL.

Markus: I will upload the slide link.

Drummond: I love Markus' diagrams. They add clarity.

Manu: +1

Drummond: Thanks to Manu for capturing the essence of the discussion started on the email thread. The section labeled 1 in the diagram is part of the authority portion of the URL. We have been following a principle of letting the DID URL author control the path, query, and fragment. These should be appended to the transformed authority portion of the URL.

… There are 3 different proposals for how the "magic" happens.

Markus: There are options to consider regarding what happens in the resolver vs. client. That can be discussed separately.

… Here we should focus on the syntax. The other items are related but can be deferred.

Drummond: Next topic?

TOPIC: Syntax/Grammar/ABNF Discussion

Manu: Colon Delimited Proposal is workable.

… The Matrix parameter is also workable. The proposal based on OData syntax is more complicated.

“The more microsyntax you have, the more bullets you put in the gun pointed at your foot.”

Manu: the simpler you keep that DSL (domain-specific language), the less there is to mess up. Problems can propagate. Why aren't matrix parameter-based solutions more widely deployed today? The colon-delimited solution could reserve the other delimiters for future use. The goal of the colon path was to address the use cases of today without complicating the syntax.

… The other question I had was regarding matrix parameters out of order. Do implementors know they can be out of order or what are the rules for extensions? If the parameters are under the control of the DID spec, then the spec would need to be extended to introduce a new one.

Drummond: We would specify a small universal set, but allow methods to extend but not redefine the universal ones.

Manu: That feels dangerous because it invites interop problems.

Drummond: You shouldn't be able to change how resolution happens. You have a new way to address immutable content. It should change nothing about resolution.

Manu: What happens when there is a conflict? Now we need a registry to prevent conflicting implementations stomping on each other. I think we should avoid that. The hardcoding of the matrix parameter syntax reduces potential conflict. Let's be stricter in the early stages to avoid misinterpretation. If the arises a large number of new syntaxes, then we can support that.

Manu: “Be more conservative about the types of extensions that we support.”

Markus: For resolution, there can be “out of band” parameters that are not part of the DID, such as caching. These can be included in headers. Michael Herman suggested that the DID says what, but not how resolution occurs. Those parameters could be more method specific. There is some debate on how strict we should be.

Jonny: Elliptic curves can be slow; limiting the record count to only three might be a useful parameter. It does introduce security risks. This can be method specific and not hard-coded into the ABNF.

Manu: There are a number of different parameters that can be passed to the resolver. We can avoid putting these in the DID URL.

Jonny: What gateway do you trust?

Markus: There are potentially many ways to perform the resolution. It is not just one.

Jonny: Is there an easy way out to say that the semi-colon syntax is the default and allow did-method specific syntax?

Drummond: DID-spec parameters should be universal. DID-method extensions should be done in a way to interoperate without conflicts. Innovation can then be enabled within the DID-method. Hopefully this will result in maximum interop with maximum innovation.

Manu: Do we believe that as soon as we hit the delimiter (";" or other delimiter), the well defined thing that happens after it is either universally declared in the DID spec or within the DID method spec?

Drummond: It must either be in the DID spec (done) or in the DID-method spec. That's what the resolver needs. Or produce an error.

Jonny: Priority?

Drummond: The DID spec universal definitions trump other definitions.

Drummond: Objections?

Chris: Concrete example?

Drummond: Service endpoint. See 1 in the diagram.

Jonny: It's easy to enforce. For example "records-required=3". It's not in the DID spec.

Dmitri?: Why json schema?

Jonny: ???

Chris: The thing that that points to, will I always get back the same type of object?

Drummond: Yes. Universal parameters specified at the DID spec level would always return the same type of object, if it exists.

Drummond: Is there anyone in favor of the OData approach at this point?

Jonny: It is not intuitive and overwhelming.

Drummond: That leaves us with two proposals. Any others?

Markus: We are mixing topics between calls.

Dmitri: The OData syntax is too complicated in my experience.

Drummond: It feels enterprise-centric and heavy for our use case.

Dmitri: I prefer ":" delimited over the Matrix approach. I expect that there will be a small number of extensions; therefore I would select the colon-delimited. If there’s going to be a *lot* of keywords (and especially if order does not matter), the semicolon approach is preferred.

Manu: You might want to match on "type". Why not have just one type? and select from an array. Is there a use-case where I have two services with the same name?

Jonny: I might have two different kinds of keys.

Manu: Are you saying that you have elliptic curve services vs RSA based services?

Jonny: Right now we have TLS for auth; ideally that would be done using elliptic curves and avoid RSA.

Manu: That is a valid use case, but in the case of the colon-delimited, :path:"service identifier" would imply search all the services and if you don't find one, such all the types. The key difference is colon-based is simpler, but if you have to express multiple things, Matrix would be simpler.

Markus: I have a use-case with multiple parameters.

Manu: The gating factor is multiple parameters to get to a need for the Matrix parameters.

Drummond: BC has a use case for this. My first reaction to the colon-delimited syntax was, "this is simpler". Colon-delimited is already allowed in the syntax. If you hit a delimiter, such as "!" you know you have crossed the boundary. Colon-delimited support now implies colon-delimited parameters forever.

Manu: If we have both, it is because we arrived at a point where we needed it. It think that it is backward compatible.

Drummond: It doubles the burden of the resolvers to support both.

Drummond: one of the use cases we talked about in Victoria with the BC Gov team was extensibility of content addressing, i.e.: type=”creddef”;id=”1234”

Manu: Can we gather the examples and evaluate them using both methods side-by-side.

Dmitri: Can we look at Markus' example?

Drummond: How many open issues?

Manu: DID Spec: 52. I expect some of these can be handled editorially. Real issues can have appropriate discussion. The spec being in better condition can ease the resolution of the remaining issues.

Drummond: Can we have a dedicated call this week to plan the remaining work?

Manu: Let's try to figure it out.

Justin: The colon is now treated at two different levels in order to process it. One pass from the beginning and another one from the end backwards. This may introduce the need for escaping some of the colons. I don't like the fact that colon means two different things depending on where it appears in the URL.

Ken: Asking Manu if he is “stuck on colons all the way through”?

Manu: I'm trying to be “ultra-minimalist and see if we can get away with it”.

Ken: A single second different character doesn’t go all the way to the complexity and micro-syntax of OData but it may result in fewer parsing errors.

Justin: Most developers will just use standard tools, like regex, that enable simple string manipulation to work with it. That could result in interpretation errors.

Manu: How easy it is to screw it up, especially with regex.

Markus: If you want to approach it with a regex, then matrix parameters would be easier and less error-prone.

Justin: A test should be: If you throw away your DID library code, can a developer write code to parse the DID URLs.

# ACTION ITEM: Drummond and Markus to work together to propose merging the two calls starting next week.

Markus ended the first recording.