

# DID Resolution and DID URL Dereferencing v1.0

Markus Sabadello, Danube Tech  
<https://danubetech.com/>



W3C DID WG Weekly Meeting  
25<sup>th</sup> July 2024



# Contents

- What does DID Core v1.0 define about DID Resolution?
- What does DID Core v1.0 NOT define about DID Resolution?
- Topics covered by DID Resolution spec
  - “Decentralized Identifier (DID) Resolution and DID URL Dereferencing v1.0”
  - Adopted from CCG Work Item “DID Resolution v0.3”  
<https://w3c.github.io/did-resolution/>
- Proposed Roadmap

# What does DID Core v1.0 define about DID Resolution?

## 3. Identifier

- 3.1 DID Syntax
- 3.2 DID URL Syntax
  - 3.2.1 DID Parameters
  - 3.2.2 Relative DID URLs

## 7. Resolution

- 7.1 DID Resolution
  - 7.1.1 DID Resolution Options
  - 7.1.2 DID Resolution Metadata
  - 7.1.3 DID Document Metadata
- 7.2 DID URL Dereferencing
  - 7.2.1 DID URL Dereferencing Options
  - 7.2.2 DID URL Dereferencing Metadata
- 7.3 Metadata Structure

**DID Parameters:** service, relativeRef, versionId, versionTime, hl

```
resolve(did, resolutionOptions) →  
  « didResolutionMetadata, didDocument, didDocumentMetadata »  
  
resolveRepresentation(did, resolutionOptions) →  
  « didResolutionMetadata, didDocumentStream, didDocumentMetadata »  
  
dereference(didUrl, dereferenceOptions) →  
  « dereferencingMetadata, contentStream, contentMetadata »
```

```
resolutionOptions: accept  
  
didResolutionMetadata: contentType, error  
  
didDocumentMetadata: created, updated, deactivated, nextUpdate,  
  versionId, nextVersionId, equivalentId, canonicalId  
  
dereferenceOptions: accept  
  
dereferencingMetadata: contentType, error  
  
contentMetadata
```

# What does DID Core v1.0 NOT define about DID Resolution?

- DID Resolution and DID URL Dereferencing Algorithms
  - E.g. what exactly happens when you dereference this:  
`did:ex:1234?service=AgentService` or `did:ex:1234#key-1`
- DID Resolution architectures
  - Local API call, remote REST API, etc.
- Serializations of DID Resolution Result and DID URL Dereferencing Result
  - E.g. how can you serialize the following in a file or network payload:  
« `didResolutionMetadata`, `didDocument`, `didDocumentMetadata` »
- Interface bindings
  - E.g. how can you call a resolver API via HTTP(S) or DIDComm, etc.?

# Topics covered by DID Resolution spec

## ■ DID Resolution and DID URL Dereferencing Algorithms

2) Determine if the DID method of the *input DID* is supported by the [DID resolver](#) that implements this algorithm. If not, the [DID resolver](#) *MUST* return the following result:

**2.1) didResolutionMetadata:** «[ "error" → "methodNotSupported" ]»

**2.2) didDocument:** null

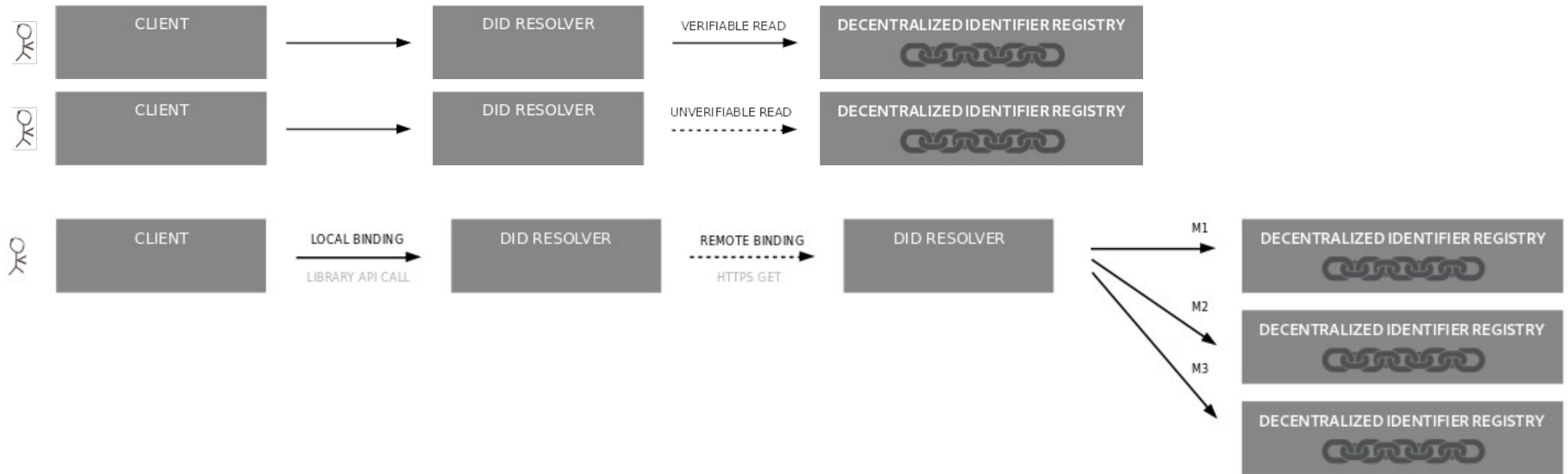
**2.3) didDocumentMetadata:** «[ ]»

**4.1)** If the *input DID URL* contains the [DID parameter](#) service and optionally the relativeRef DID parameter:

**4.1)** If the result of 4.1.1 [Dereferencing the Primary Resource](#) is a *resolved DID document* with media type `application/did+ld+json`, and the *input DID URL* contains a [DID fragment](#):

# Topics covered by DID Resolution spec

- DID Resolution architectures
  - Local vs. remote, proxies, client-side dereferencing, etc.



# Topics covered by DID Resolution spec

- Serializations of DID Resolution Result and DID URL Dereferencing Result

```
{
  "@context": "https://w3id.org/did-resolution/v1",
  "didDocument": {
    "@context": "https://www.w3.org/ns/did/v1",
    "id": "did:example:123456789abcdefghi",
    ...
  },
  "didResolutionMetadata": {
    "contentType": "application/did+ld+json",
    "retrieved": "2024-06-01T19:73:24Z"
  },
  "didDocumentMetadata": {
    "created": "2019-03-23T06:35:22Z",
    "updated": "2023-08-10T13:40:06Z"
  }
}
```

# Topics covered by DID Resolution spec

## ■ Interface bindings

```
curl -H "Accept: application/did+ld+json" \  
-X GET "https://dev.uniresolver.io/1.0/identifiers/did:sov:WRfXPg8dantKVubE3HX8pw"
```

```
curl -H "Authorization: Bearer b082c420-df67-4b06-899c-b7c51d75fba0" \  
-H "Accept: application/did+ld+json" \  
-X GET  
"https://api.godiddy.com/0.1.0/universal-resolver/identifiers/did%3Asov%3AWRfXPg8dantKVubE3HX8pw%23key-1"
```

```
curl -H "Authorization: Bearer b082c420-df67-4b06-899c-b7c51d75fba0" \  
-H "Accept: text/uri-list" \  
-X GET "https://api.godiddy.com/0.1.0/universal-resolver/identifiers/did:web:danubetech.com?  
service=github&relativeRef=did-method-dns"
```



# Topics covered by DID Resolution spec

- Additional topics:
  - Parameters
  - Error codes
  - Caching
  - Versioning
  - Trust in DID Resolution
  - Authentication, encryption, selective disclosure
  - Some others...

# Proposed Roadmap

1. Discuss if we agree on high-level sections/scope of the DID Resolution spec
  - <https://w3c.github.io/did-resolution/>
2. Decide on DID Core questions that affect DID Resolution
  - Abstract data model?
  - Controller document?
  - Move anything from DID Core to DID Resolution or vice versa?
3. Work through all the major and minor topics, align with DID Core, process Github issues and PRs
  - 44 open Github issues, 1 open PR