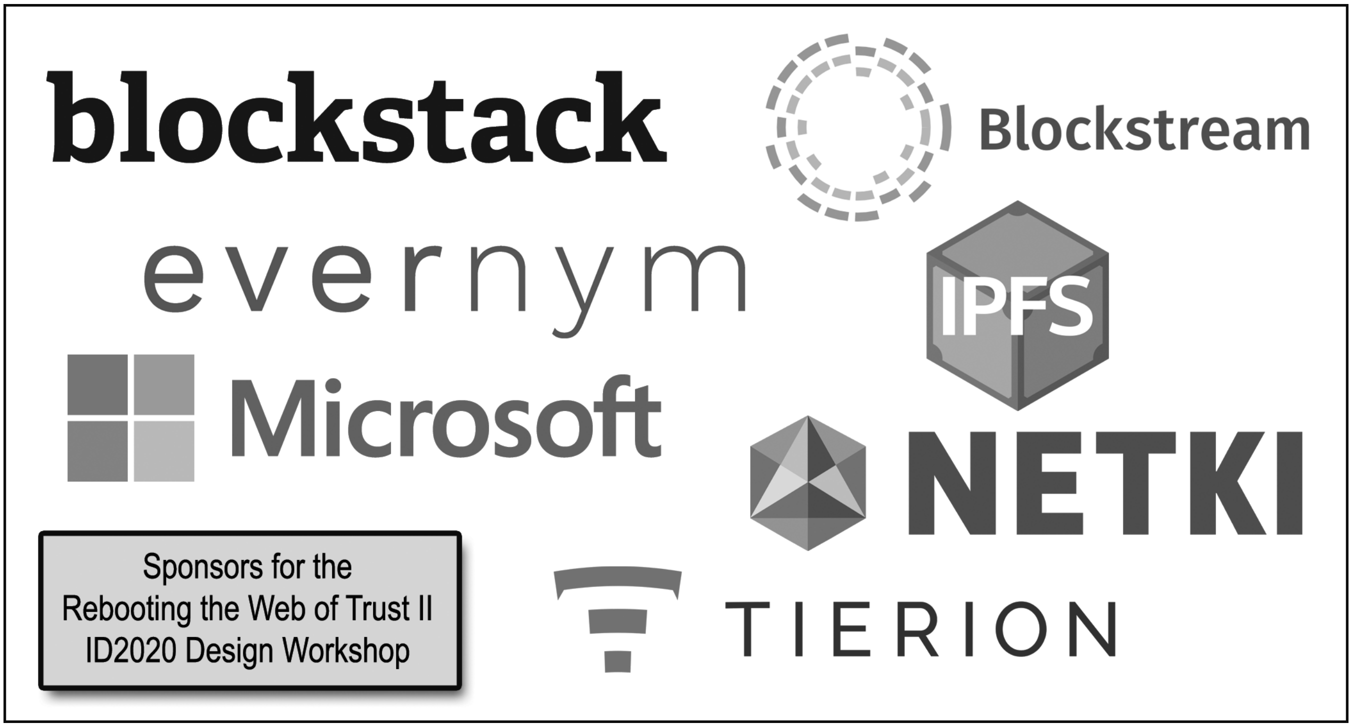
Requirements for DIDs (Decentralized Identifiers)

A Requirement from Rebooting the Web of Trust II: ID2020 Design Workshop

by Drummond Reed and Les Chasen, Respect Network



1. INTRODUCTION

Respect Network is conducting a research project for the U.S. Department of Homeland Security, HSHQDC-16-C-00061, to analyze the applicability of blockchain technologies to a decentralized identifier system. Our thesis is that blockchains, or more generically distributed ledgers, are a potentially powerful new tool for "identity roots" — the starting points for an Internet identity. However "blockchain identity" may not fully address the core security and privacy principles needed in a complete identity system. In this case DIDs — Decentralized Identifiers rooted on a distributed ledger — may end up being a foundational building block for higher level identity management solutions.

During this phase of our work, we interviewed industry experts in identity and blockchain technology, both in informal discussions at major identity conferences as well as in-depth interviews. This included a series of sessions on DIDs at the ID2020 Design Workshop held in New York City on May 21-22. Attendees included Drummond Reed, Les Chasen, Christopher Allen, Manu Sporny, Markus Sabadello, Juan Bennet, Ryan Shea, Christian lundkvist, Rouven Heck, and Greg Slepak.

From those sessions and our interviews, we have gathered the following set of requirements for DIDs.

2. SUMMARY OF REQUIREMENTS

2.1 Principles

The following principles were originally drafted by the XDI.org Registry Working Group (XRWG) to describe their requirements for an XDI registry. Our discussions have lead us to believe they apply to DID as well, so we have included them as guiding principles for design.

- **Maximum Interoperability.** DID infrastructure must conform to open standards and should enable any set of users and communities to discover and interoperate with each other.
- **Maximum Decentralization.** DID infrastructure should be designed to minimize central points of control and attack.
- **Critical Infrastructure.** DID infrastructure must provide for a high level of reliability, stability, scalability, security, sustainability and other requirements typical of critical internet infrastructure.
- **Sovereign Identity.** DID infrastructure should enable any principal (person or organization) to fully administer the principal's own set of DIDs without the need to rely on an external administrative authority.
- **Neutrality.** DID infrastructure should be available to all members of the public and should not discriminate against any party that wishes to use it for any lawful purpose.

2.2 Design Goals

At the ID2020 Design Workshop, there was broad consensus on the following two major goals for the DID specifications:

1. **DID:** Define the structure of a universally unique system-independent discoverable identifier (a decentralized identifier) that can serve as the key to a value

2. **DID Object:** Define the structure of a value for that key that can meet four requirements:

    1. Provide cryptographic proof of:

        1. Ownership of the DID

        2. Permission to update the DID object

    2. Provide pointers to:

        1. Other sources of claims

        2. Other peer DIDs

The combination of a DID and its associated DID object is called a **DID Registration**. From the standpoint of *claims-based identity*, a DID registration is "the genesis claim" for an identity.

2.3 Public and Private DIDs

In developing DID architecture, it has become clear that DIDs will exist in two different contexts: public and private. A **Public DID** is discoverable and enables a high degree of correlation (i.e., it is associated with "public" identity, or what Kim Cameron in the *Laws of Identity* calls an **Omnidirectional Identifier**).

A **private DID** is a DID registration that is shared just between two parties, or a limited number of parties. It is intentionally not discoverable, or discoverable only in a limited context, so that correlation can be controlled (i.e., it's what Kim Cameron in the *Laws of Identity* calls a **Unidirectional Identifier**).

3. REGISTRATION AND DISCOVERY ARCHITECTURE

At the ID2020 Design Workshop, there was considerable exploration and debate about the most desirable way to perform registration and discovery of DID objects via blockchains, distributed ledgers, or DHTs. For the purposes of this document, these different DID database options will be referred to as "ledgers". Discussion revolved around five specific ledgers:

- Bitcoin
- Ethereum
- Stellar
- IPFS
- Sovrin

Following is a summary of four major approaches discussed by the group.

3.1 Approach 1: Peer Ledgers

Approach

A DID conforming to standard DID specifications may be registered in any qualified ledger (i.e., any ledger capable of accepting DID registration). The structure of a DID and DID object would be logically the same on any qualified ledger. Each qualified ledger would define its own way of registering a DID and storing a DID object so a client can look up and return the associated DID object and perform updates as governed by the access control policy in the DID object.

Advantages

- DIDs are atomic and portable.
- Principals can register DIDs in the ledger of their choice.
- No consensus is required across ledgers.
- Each ledger can optimize the implementation of a DID registration for its format/protocol.
- Lookups can be very efficient on each specific ledger.

Disadvantages

- Discovery of a DID requires searching across all possible ledgers; there is no known starting point.
- There is no clear source of authority if the same DID registration is registered on multiple ledgers.
- DID impersonation attacks are difficult or impossible to prevent.

3.2 Approach 2: Addressable Ledgers

Approach

Unify all qualified ledgers into a global namespace in which each qualified ledger is addressable using a prefix. There are two options for the prefix:

Option #1: Use a URI or URN scheme:

```
urn:btc:<ledger-specific-address>
```

Option #2: Use a meta-ledger that assigns DIDs to other ledgers. DIDs become hierarchical pairs in the format:

```
<ledger-location-did><ledger-specific-address>
did:33ad7beb-1abc-4a26-b892-
466df4379a51/<ledger-specific-address>
```

Advantages

- Principals can make DID registrations in the ledger of their choice.
- DID impersonation attacks are impossible because any DID hierarchical pair is unique and only registered once.
- No direct consensus is required across ledgers.
- Each ledger can optimize the implementation of a DID registration for its format.
- Lookups can efficiently follow a two-level path, the first leg being pre-cached and the second leg being very efficient on each specific ledger.

Disadvantages

- DIDs become location-specific so portability must be provided using semantic equivalence statements.

- If a meta-ledger is used, community consensus must be reached on implementing and maintaining the meta ledger.

- If a meta-ledger is not used, all relying parties must keep track of all ledger prefix addresses.

3.3 Approach 3: Meta-Consensus Protocol

Approach

All participating ledgers implement a meta-consensus protocol that ensures uniqueness of a DID across all participating ledgers.

Advantages

- DIDs are atomic and portable.

- Principals can register DIDs in the ledger of their choice.

- DID impersonation attacks are impossible because a DID can only be registered once.

- Each ledger can optimize the implementation of DID for its format.

- Lookups can be very efficient on each specific ledger.

Disadvantages

- DID transactions require direct consensus across all participating ledgers — a nearly impossible burden for each of the ledgers.

- Discovery of a DID requires searching across all possible ledgers.

3.4 Approach 4: Index Ledger aka "DID Registry"

Approach

Create a new DID index ledger that maps a unique DID to any ledger or repository that can store a DID object. The DID itself is a UUID or similar identifier and the value is either:

1. A DID object, OR

2. A pointer to a DID object on another ledger or repository.

Advantages

- DIDs are atomic and portable.

- Principals can register DID objects in the ledger or repository of their choice.

- DID impersonation attacks are impossible because a DID can only be registered once.

- Each ledger can optimize the implementation of a DID object for its format.

- Discovery is easy and authoritative because it always starts with the index ledger.

- Lookups are efficient due to following a two-level path, the first leg being optimized for the index ledger and the second leg optimized for each specific ledger or repository.

Disadvantages

- Community consensus must be reached on implementing and maintaining an index ledger.

Additional Credits

Authors: Drummond Reed and Les Chasen, Respect Network

About Rebooting the Web of Trust

This paper was produced as part of the **Rebooting the Web of Trust II** design workshop. On May 21st and May 22nd, 2016, over 40 tech visionaries came together in Manhattan, New York following the ID2020 Summit at the UN to talk about the future of decentralized trust on the internet with the goal of writing 3-5 white papers and specs. This is one of them.

Workshop Sponsors: Blockstack, Blockstream, Evernym, IPFS, Microsoft, Netki, Tierion, ID2020

Workshop Producer: Christopher Allen

Workshop Facilitators: Christopher Allen with graphic facilitation by Sue Shea, additional paper editorial & layout by Shannon Appelcline, and additional support by Kiara Robles.

What's Next?

The design workshop and this paper are just starting points for Rebooting the Web of Trust. If you have any comments, thoughts, or expansions on this paper, please post them to our GitHub issues page:

https://github.com/WebOfTrustInfo/ID2020DesignWorkshop/issues

The next Rebooting the Web of Trust design workshop is scheduled for October 19th-21st in San Francisco, California. If you'd like to be involved or would like to help sponsor these events, email:

ChristopherA@LifeWithAlacrity.com