Use Cases and Requirements for Decentralized Identifiers

W3C Editor's Draft 02 September 2020

This version:
https://w3c.github.io/did-use-cases/

Latest published version:
https://www.w3.org/TR/did-use-cases/

Latest editor's draft:
https://w3c.github.io/did-use-cases/

Editors:
Joe Andrieu (Invited Expert)
Phil Archer (GS1)

Authors:
Kim Hamilton-Duffy
Ryan Grant
Adrian Gropper

Participate:
GitHub w3c/did-use-cases
File a bug
Commit history
Pull requests

Abstract

This document sets out use cases and requirements for a new type of identifier that has 4 essential characteristics:

1. **decentralized**: there should be no central issuing agency;
2. **persistent**: the identifier should be inherently persistent, not requiring the continued operation of an underling organization;
3. **cryptographically verifiable**: it should be possible to prove control of the identifier cryptographically;
4. **resolvable**: it should be possible to discover metadata about the identifier.

Although existing identifiers may display some of these characteristics, none currently displays all four.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at https://www.w3.org/TR/.

This document was published by the Decentralized Identifier Working Group as an Editor's Draft.

GitHub Issues are preferred for discussion of this specification. Alternatively, you can send comments to our mailing list. Please send them to public-did-wg@w3.org (archives).

Publication as an Editor's Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

Table of Contents

1. Introduction §

The need for globally unique identifier schemes has been addressed many times. Globally unique ID schemes typically rely on a central authority controlling a 'root' space that is then delegated to local organizations who in turn delegate to further organizations who eventually add the final string to complete the identifier. Even if we restrict ourselves to online identifiers, there are many examples of this.

- IANA controls the root namespace of the Internet's Domain Name System; registrars operate the top level domains and then license specific domain names to their clients, and it's these clients who create the actual identifiers for online resources (URLs). The example below shows who controls what in a typical URL.

  EXAMPLE 1

  ```
                  IANA
                   ↓
  https://example.com/page.html
             ↑           ↑
          Registrar   Licensee
  ```

- DOIs have the form `10.{registrant}/{suffix}` where 'registrant' is defined by the DOI organization and the suffix by the registrant.
- Global Trade Item Numbers (GTINs) seen on many of the world's barcodes are managed in a similar way, as are Legal Entity Identifiers, ISBNs and more.

In all these cases, ultimately, there is a central authority on which the identifier system depends. Those central authorities go to significant efforts to make their identifiers persistent and resolvable, however, should they cease to exist, the long term integrity of the identifier is at least questionable to a greater or lesser extent. For as long as those organizations exist (and they are generally well established with no immediate threat to their survival), the way to assess whether a particular identifier is in some way 'valid' is to query the issuing authority.

These factors point to a need in some circumstances for a globally unique identifier that is 'self sovereign', that is, one that does not depend on any issuing authority. Universally unique identifiers (UUIDs) [RFC4122] fulfill this role, however, there is no way to *prove* control of a UUID.

This document sets out use cases and requirements for a new kind of identifier that meets all these basic requirements:

1. **decentralized**: there should be no central issuing agency;
2. **persistent**: the identifier should be inherently persistent, not requiring the continued operation of an underling organization;
3. **cryptographically verifiable**: it should be possible to prove control of the identifier cryptographically;
4. **resolvable**: it should be possible to discover metadata about the identifier.

Further desired features and their benefits are provided in a later section.

1.1 Existing Work §

The use cases and requirements set out below have not been created *a priori*. Substantial work has been done within W3C and elsewhere leading, in particular, to Decentralized Identifiers (DIDs) Data Model and Syntaxes published as a Community Group Report by the Credentials Community Group in August 2019. That work provides a framework — a set of concepts — that have proved to be useful when discussing DIDs and the problems they can solve (see below). Those concepts are used within this document to set out the detail of the problem that the Decentralized Identifier Working Group is chartered to solve. It is the nature of the standardization process that these terms may be modified within the standard itself and therefore, their use here should not be seen as authoritative.

1.2 Concepts of Decentralized Identity §

> ~~ISSUE 39~~: Relying Party terminology could confuse things PR in review
>
> Terminology in this opening prose is being discussed, in particular the term 'relying party' which has been changed to 'requesting party' in this version of the doc.

A decentralized system will enable several key actions by three distinct entities: the Controller, the Requesting Party, and the Subject.

Controllers create and control DIDs, while Requesting Parties rely on DIDs as an identifier for interactions related to the DID Subject.

The Subject is the entity referred to by the DID, which can be anything: a person, an organization, a device, a location, even a concept. Typically, the Subject is also the Controller, but in cases of guardianship, agents (human or software), and inanimate Subjects, this is not possible. As such, the Subject has no functional role. When the Subject is, in fact, the DID Controller, we consider the action to be taken by the Controller on their own behalf as the Subject. When the Subject is not the DID Controller, the Controller is said to be taking action on behalf of the Subject, such as when an employee manages a DID on behalf of their employer or a parent uses a DID on behalf of their child.

The DID Controller and Requesting Party may be individuals or interactive systems, but for simplicity in this document, we refer to both as if they were individual persons performing these actions.

Only a DID Controller can perform the actions that control a DID, however, anyone can act as a Requesting Party for any DID they know, including the DID Controller, should they wish to inspect or verify their own DID.

This use case document defines these actions in terms of the eventual systems we anticipate using the resultant specification.

Perhaps the most salient point about Decentralized Identifiers is that there are no "Identity Providers". Instead, this role is subsumed in the decentralized systems that Controllers use to manage DIDs and, in turn, Requesting Parties use to apply DIDs. These decentralized systems, which we refer to as DID registries, are designed to operate independently from any particular service provider and hence, free from any given platform authority. It is anticipated that DIDs will be registered using distributed ledger technology (DLT).

In practice, the definition and operation of all current decentralized systems retain some elements of centralized control. Depending on the criteria one uses to evaluate such systems — from who controls the most widely used code base to who controls the specification — where a system resides on the spectrum of centralized and decentralized varies. However, the design of any decentralized identity system is separate from the academic debate about how decentralized it may be in practice.

The use cases presented below make use of a number of high level concepts as follows.

> NOTE: Shared terminology
>
> This section is automatically synchronised with the terminology section in the DID Core specification.

This section defines the terms used in this specification and throughout decentralized identifier infrastructure. A link to these terms is included whenever they appear in this specification.

authenticate

Authentication is a process (typically some type of protocol) by which an entity can prove it has a specific attribute or controls a specific secret using one or more verification methods. With DIDs, a common example would be proving control of the private key associated with a public key published in a DID document.

binding

A concrete mechanism used by a caller to invoke a DID resolver or a DID URL dereferencer. This could be a local command line tool, a software library, or a network call such as an HTTPS request.

***decentralized identifier* (DID)**

A globally unique persistent identifier that does not require a centralized registration authority because it is generated and/or registered cryptographically. The generic format of a DID is defined in the DID Core specification. A specific DID scheme is defined in a DID method specification. Many—but not all—DID methods make use of distributed ledger technology (DLT) or some other form of decentralized network.

decentralized identity management

identity management that is based on the use of decentralized identifiers. Decentralized identity management extends authority for identifier generation, registration, and assignment beyond traditional roots of trust such as X.500 directory services, the Domain Name System, and most national ID systems.

***decentralized public key infrastructure* (DPKI)**

Public key infrastructure that does not rely on traditional certificate authorities because it uses decentralized identifiers and DID documents) to discover and verify public key descriptions.

DID controller

An entity that has the capability to make changes to a DID document. A DID may have more than one DID controller. The DID controller(s) can be denoted by the optional `controller` property at the top level of the DID document. Note that one DID controller may be the DID subject.

DID delegate

An entity to whom a DID controller has granted permission to use a verification method associated with a DID via a DID document. For example, a parent who controls a child's DID document might permit the child to use their personal device in order to authenticate. In this case, the child is the DID delegate. The child's personal device would contain the private cryptographic material enabling the child to authenticate using the DID. However the child may not be permitted to add other personal devices without the parent's permission.

DID document

A set of data describing the DID subject, including mechanisms, such as public keys and pseudonymous biometrics, that the DID subject or a DID delegate can use to authenticate itself and prove its association with the DID. A DID document may also contain other attributes or claims describing the DID subject. A DID document may have one or more different representations as defined in #core-representations or in the W3C DID Specification Registries [DID-SPEC-REGISTRIES].

DID fragment

The portion of a DID URL that follows the first hash sign character (`#`). DID fragment syntax is identical to URI fragment syntax.

DID method

A definition of how a specific DID scheme must be implemented to work with a specific verifiable data registry. A DID method is defined by a DID method specification, which must specify the precise operations by which DIDs are created, resolved and deactivated and DID documents are written and updated. See #methods.

DID path

The portion of a DID URL that begins with and includes the first forward slash (`/`) character and ends with either a question mark (`?`) character or a fragment hash sign (`#`) character (or the end of the DID URL). DID path syntax is identical to URI path syntax. See #path.

DID query

The portion of a DID URL that follows and includes the first question mark character (`?`). DID query syntax is identical to URI query syntax. See #query.

DID resolution

The function that takes as its input a DID and a set of input metadata and returns a DID document in a conforming representation plus additional metadata. This function relies on the "Read" operation of the applicable DID method. The inputs and outputs of this function are defined in #resolution.

DID resolver

A DID resolver is a software and/or hardware component that performs the DID resolution function by taking a DID as input and producing a conforming DID document as output.

DID scheme

The formal syntax of a decentralized identifier. The generic DID scheme begins with the prefix `did:` as defined in the section of the DID Core specification. Each DID method specification must define a specific DID scheme that works with that specific DID method. In a specific DID method scheme, the DID method name must follow the first colon and terminate with the second colon, e.g., `did:example:`

DID subject

The entity identified by a DID and described by a DID document. A DID has exactly one DID subject. Anything can be a DID subject: person, group, organization, physical thing, digital thing, logical thing, etc.

DID URL

A DID plus any additional syntactic component that conforms to the definition in #did-url-syntax. This includes an optional DID path, optional DID query (and its leading `?` character), and optional DID fragment (and its leading `#` character).

DID URL dereferencing

The function that takes as its input a DID URL, a DID document, plus a set of dereferencing options, and returns a resource. This resource may be a DID document plus additional metadata, or it may be a secondary resource contained within the DID document, or it may be a resource entirely external to the DID document. If the function begins with a DID URL, it use the DID resolution function to fetch a DID document indicated by the DID contained within the DID URL. The dereferencing function then can perform additional processing on the DID document to return the dereferenced resource indicated by the DID URL. The inputs and outputs of this function are defined in #did-url-dereferencing.

DID URL dereferencer

A software and/or hardware system that performs the DID URL dereferencing function for a given DID URL or DID document.

***distributed ledger* (DLT)**

A distributed database in which the various nodes use a consensus protocol to maintain a shared ledger in which each transaction is cryptographically signed and chained to the previous transaction.

proof purpose

A property of a DID document that communicates the purpose for which the DID controller included a specific type of proof. It acts as a safeguard to prevent the proof from being misused for a purpose other than the one it was intended for.

public key description

A data object contained inside a DID document that contains all the metadata necessary to use a public key or verification key.

resource

As defined by [RFC3986]: "...the term 'resource' is used in a general sense for whatever might be identified by a URI." Similarly, any resource may serve as a DID subject identified by a DID.

representation

As defined for HTTP by [RFC7231]: "information that is intended to reflect a past, current, or desired state of a given resource, in a format that can be readily communicated via the protocol, and that consists of a set of representation metadata and a potentially unbounded stream of representation data." A DID document is a representation of information describing a DID subject. The #core-representations section of the DID Core specification defines several representation formats for a DID document.

services

Means of communicating or interacting with the DID subject or associated entities via one or more service endpoints. Examples include discovery services, agent services, social networking services, file storage services, and verifiable credential repository services.

service endpoint

A network address (such as an HTTP URL) at which services operate on behalf of a DID subject.

***Uniform Resource Identifier* (URI)**

The standard identifier format for all resources on the World Wide Web as defined by [RFC3986]. A DID is a type of URI scheme.

verifiable credential

A standard data model and representation format for cryptographically-verifiable digital credentials as defined by the W3C [VC-DATA-MODEL].

verifiable data registry

A system that facilitates the creation, verification, updating, and/or deactivation of decentralized identifiers and DID documents. A verifiable data registry may also be used for other cryptographically-verifiable data structures such as verifiable credentials. For more information, see [VC-DATA-MODEL].

verifiable timestamp

A verifiable timestamp enables a third-party to verify that a data object existed at a specific moment in time and that it has not been modified or corrupted since that moment in time. If the data integrity could reasonably have modified or corrupted since that moment in time, the timestamp is not verifiable.

verification method

A set of parameters that can be used together with a process or protocol to independently verify a proof. For example, a public key can be used as a verification method with respect to a digital signature; in such usage, it verifies that the signer possessed the associated private key.

"Verification" and "proof" in this definition are intended to apply broadly. For example, a public key might be used during Diffie-Hellman key exchange to negotiate a shared symmetric key for encryption. This guarantees the integrity of the key agreement process. It is thus another type of verification method, even though descriptions of the process might not use the words "verification" or "proof."

verification relationship

An expression of the relationship between the DID subject and a verification method. An example of a verification relationship is #authentication.

***Universally Unique Identifier* (UUID)**

A type of globally unique identifier defined by [RFC4122]. UUIDs are similar to DIDs in that they do not require a centralized registration authority. UUIDs differ from DIDs in that they are not resolvable or cryptographically-verifiable.

> ~~ISSUE 14~~: What does it mean for a DID to be "recorded in a registry"? Needs-WG-Discussion
>
> The term DID registry is under discussion within the Working Group. A particular point to bear in mind is that not all DID methods require DIDs to be registered to be functional.

When we refer to *methods* and *registries*, we mean DID methods and DID registries. A working assumption for the use cases is that all DIDs resolve to DID Documents. DID Documents contain the cryptographic material to perform the functions related to that particular DID, including associated proof methods and any service endpoints, that is, services that can make use of the DID.

2. Use Cases §

2.1 Online shopper §

Traditionally, a shopper frequents a trusted retailer and can physically hold the products they wish to purchase. The product and the information about it is trusted because it is put there by the brand, and the shopper trusts that the retailer has received the product through trusted supply chain partners. Today, there is a multitude of channels and platforms for selling and buying products. The internet has changed consumer purchasing behavior as more and more commerce is conducted digitally. This introduces new challenges for brands, retailers and consumers, as the relationship is not as direct as the traditional mode of shopping.

Online shopping, especially through 3rd party marketplaces, creates a proliferation of digital records about that product across platforms. Unlike a physical product, a consumer cannot be assured that the record (and the information presented about that product) came from the brand or other authoritative source. Product identification and information and the source of the product itself is less reliable, and introduces trust issues with representations of products bought and sold online. Additionally, unique identification is critical to business processes, but also to online purchasing. Very often two different products share the same identifier across the supply chain, and so what a consumer purchases and what ultimately is received may be different.

Mechanisms are required for the following to provide trust in the digital representation of a product across platforms:

1. validation of the legitimacy of an online listing — in particular on third party marketplaces;
2. assurance of identification uniqueness and key identifying data element accuracy.

Requirements: Authentication/proof of control, Decentralized/self issued, Inter-jurisdictional, Can't be administratively denied, No call home, Survives issuing organization mortality, Survives deployment end-of-life, Survives relationship with service provider

Contributed by GS1

2.2 Vehicle assemblies §

Manufactured goods are usually the output of multiple processes carried out by multiple actors. Think of a vehicle (be it a ship, a train, a car or a plane) with components and assemblies of components from multiple suppliers that are then added to the emerging vehicle as it makes its way along the production line. During the lifetime of that vehicle, components will be replaced and serviced, and this work can be carried out by any number of different agents.

A mechanism that allows each agent to make independent assertions about their work – the components created or replaced, the tests carried out, which part was married up to which other part and so on – would allow the complete manufacturing and service history of the vehicle to be computed.

The independent aspect is important. A decentralized identifier created and applied to each component or assembly can be cryptographically verified without the need to call home to a supply company that may or may not continue to exist. Moreover, there is no requirement for the overall vehicle manufacturer to maintain an aggregated database concerning a vehicle that, in a traditional system, is not only a single point of failure but that is immediately out of date as soon as the vehicle goes in for its first service. A decentralized system allows each component of the vehicle to have its own history, its own credentials and its own lifecycle, independent of other components and the vehicle – or vehicles – of which it is a part over time. At the end of its life, the identifier may continue to be useful as a reference point for recycling and safe disposal information, even if the original manufacturer is no longer in business.

The vehicle itself has its own identifier too. This would allow that particular vehicle to be a reference for the components and any licensing, insurance, servicing etc. Again, the decentralized nature of DIDs promises *subsequent owners* control over the identifier without dependency on any central agency allowing the maintenance record(s) to transition smoothly throughout the entire lifecycle of the vehicle.

Requirements: Authentication/proof of control, Decentralized/self issued, Inter-jurisdictional, Can't be administratively denied, No call home, Survives issuing organization mortality, Survives deployment end-of-life, Survives relationship with service provider

Contributed by Spherity

2.3 Encrypted Data Vault §

Data stored in the cloud is typically visible to the cloud platform operator, even when flagged as being 'only accessible to you.' For data to be stored in the cloud – in other words, on someone else's computer – and for it to be only accessible to the data owner, it must be encrypted. A Decentralized Identifier that leads to information with which a user can prove control of the identifier without resort to a centralized authority could enable substitutable encrypted cloud storage. That is, the data owner would be able to change cloud storage provider, take their encrypted data with them, and still offer fine-grained access control to specific parties or their delegates. Associating cryptographic information with the DID would allow the data owner to change their keys and *still* be able to control access without having to decrypt and re-encrypt it.

Requirements: Authentication/proof of control, Associated cryptographic material, Streamlined key rotation, No vendor lock in, Privacy preserving

Contributed by Digital Bazaar & Transmute

2.4 Accessing Master Data of Entities §

Decentralized identifiers allow one to discover the location of an authoritative public master data record of an entity. This mechanism can be used for organizations as well as things. The authoritative master data record could be retrieved from a designated service endpoint listed in the DID document. The record may be self-certifying, i.e. verifiable with a key listed in the DID document or third party attested represented as a verifiable presentation.

The third party attesting master data of an organization might be a chamber of commerce, while the third party attesting the master data of a thing might be its manufacturer. The decentralized nature of the identifier is important in particular for the device as the DID can act as an entry to that master data even if the manufacturer goes out of business or stops the service.

Requirements: Decentralized/self issued, Guaranteed unique identifier, Authentication/proof of control, Service endpoint discovery, Associated cryptographic material

Contributed by Robert Bosch GmbH

2.5 Identifiers in an ecosystem of verifiable credentials (VCs) §

Remembering the 4 essential characteristics of a decentralized identifier, there are situations where DIDs offer tangible benefits over existing identifier schemes. For example:

- Verifiable credential issuers publishing the list of subject attributes that they are authoritative for (their authoritative nature can be proven).
- In similar vein, Verifiable Credential issuers may publish authoritative schemas for the types of credential that they issue.
- Terms of use or other policy constraints that might apply to issued VCs.
- A published list of the public keys used to sign credentials becomes useful in a chain of trust if that list can itself be verified.
- The controller of a decentralized identifier can rotate (update) their cryptographic keys, either to overcome their existing keys being compromised or the development of superior technology. This does not affect the validity or provability of any verifiable credentials associated with that identifier.

Requirements: Authentication/proof of control, Guaranteed unique identifier, Associated cryptographic material, Streamlined key rotation, Cryptographic future proof

Contributed by David Chadwick, University of Kent

2.6 Sharing opted-in information across platforms §

Individuals are concerned that their online activity is tracked and analysed by many websites and third party services without their full knowledge. Consent for the storing of cookies is routinely given without any real care being taken, and service terms and conditions rarely read before being agreed to. A solution to this problem might have the following characteristics:

1. The ability to create an identifier for oneself without having to provide any personally identifiable information (PII) to any third party.
2. The ability to prove control of that identifier, again, without revealing any PII.
3. The ability to associate information likely to be of commercial interest with that identifier, such as purchase history, age, gender or social status, again, without revealing any PII.

Such an identifier could be shared across multiple platforms that would be able to use it to provide a more customized service, but without gaining any PII and therefore without being able to correlate with other services that did depend the individual’s identity, such as their health records. A single individual would be well able to maintain multiple online personas in this way and thereby have good control over what they did and didn't share with different services.

Requirements: Decentralized/self issued, Privacy preserving, Preempt/limit trackable data trails

Suggested by Airgrid

2.7 Collecting payments for work conducted anonymously §

Before becoming a full time journalist, Sahar worked many odd jobs and uncredited gigs in the publishing and movie industries: she was a ghostwriter, a scriptfixer, various kinds of editor, a translator, and more. Some of these gigs still pay her residuals years later in the form of tiny micropayments. When these now-classic works are reprinted or syndicated, the payments are automatically sent using information associated with Sahar's DID without anyone involved thinking much about who exactly the contract refers to as “anonymous ghostwriter X” or “translator of appendix C”. Using a DID as her identifier, rather than her name, allows Sahar to retain anonymity but provide a means for others to pay the right person.

Requirements: Privacy preserving, Minimized rents, Survives relationship with service provider

Contributed by Juan Caballero

2.8 Anonymity within a supply chain §

Modern supply chains are highly complex, typically involving multiple companies taking ownership and/or custody of shipments from the point of origin through to the final destination, be that a retailer or end consumer. Furthermore, since most mass-produced products are made to stock rather than bespoke for a specific customer, the manufacturer typically does not know which final retailer or retail store will receive each specific product instance; the individual path through the supply chain network is not predefined by the manufacturer but instead emerges over time as a set of intermediate distributors and wholesalers distribute product in response to orders from their customers. Data concerning that supply chain can be commercially sensitive. For example, a transport company may not wish their customer to know that they sub contracted part of the journey to another partner. It can be a security risk too. The locations of warehouses and distribution centers where large stocks of particular items are held can be a target for the insertion of counterfeit goods into legitimate supplies.

A downstream party such as a retailer, pharmacy or end consumer may have a legitimate reason or in some cases a legal requirement to know that each product is genuine and can be traced back to the genuine manufacturer through an unbroken chain of custody, without any gaps or inconsistencies in the traceability data.

A downstream party may also want to check whether the product ID and serial number were actually issued by the manufacturer. However, the manufacturer may not want to provide such information about valid serial numbers to its competitors or counterfeiters, so there may be a need to control access to such services.

For these reasons and more, identifiers for business partners, locations and transactions often need to be obfuscated, and the data exchanged between partners is typically restricted both in terms of its content and in terms of access to it. This is especially true when a manufacturer does not want its competitors to know the actual source of the ingredients or components in its products. Different parties will be interested in different things too. For example, a retailer may be interested in knowing where a particular shipment is and when it is expected to arrive; a consumer of a food product is more likely to be concerned about the original source of the item and the total number of food miles in the product.

Decentralized Identifiers could be generated by sending and receiving parties at each stage in the chain. This would preserve anonymity yet prove control of the identified shipment at each stage and, via the DID Document, provide access to data (subject to authorization) pertaining to each transaction without enabling correlation across multiple shipments or other sensitive information.

Requirements: Authentication/proof of control, Decentralized/self issued, Service endpoint discovery, Privacy preserving Preempt/limit trackable data trails, Survives relationship with service provider

Contributed by GS1

2.9 Digital Permanent Resident Card §

Sam is a long term immigrant to the United States who just received notice of Permanent Resident status from the United States Customs and Immigration Services (USCIS). Along with his notice is directions for downloading and using a digital version of his physical card, including a one-time activation code. After getting a digital wallet, he visits the USCIS website, signs in, and uses his activation code to get a digital credential. His wallet provides a DID to the website and demonstrates control over the DID to prove to USCIS that the identifier is under Sam's control. USCIS issues a newly minted digital credential with the subject identifier set to the provided DID.

Now, Sam can use that digital credential anywhere by demonstrating the same proof of control to provide a specific level of identity assurance, anchored in the cryptography of the proof-of-control ceremony. Verifiers of that credential can cryptographically verify both the authenticity and origin of the credential itself—it can be proven that it was issued by USCIS and unchanged since then—AND it can verify that the presenter of the credential still controls the identifier.

Requirements: Authentication/proof of control, No call home, Associated cryptographic material, Minimized rents

Suggested by DHS SVIP

2.10 Importing retro toys §

Jodie is operations manager at ImportersRUs, a boutique importer specializing in small run retro products manufactured in Asia. Last year she got her big break when she licensed the right to make "Retro" Lima Bean Babies, a huge collector phenomenon in the 1990s. As long as her products had a distinct, visible tag with "R" on one side and "Retro" on the other, she could manufacture anything from the Lima Bean Babies catalog.

To handle fulfilment, Jodie works in small batches with a large number of different products and frequently changes manufacturers to stay on top of shifting materials, transportation, and tariff costs.

Jodie registers a DID with US Customs and Border Protection (US CBP) for ImportersRUs (demonstrating proof-of-control as she does) and secures a digital license from Tie Enterprises (owners of the Lima Bean Babies trademark and designs) issued to her DID and signed using a DID that Tie Enterprises has also registered with US CBP. For her manufacturers, Jodie cryptographically delegates the digital license to individual manufacturers using the DID (which is both the subject of the license *and* known to US CBP). Those manufacturers are responsible for getting the product to the port of Los Angeles. They, in turn, cryptographically delegate their licenses to logistics firms to actually move the product from their facility to Los Angeles, while maintaining cryptographic verifiability of both the original license, and the chain of delegations.

When her products reach the United States, CBP can automatically review the digital manifest filed for entry and each of the delegations to verify that this given batch of products are, in fact, officially licensed Lima Bean Baby products.

This digitally verifiable provenance of her license streamlines access through US Customs, regardless of her supply chain, without reducing the effectiveness of CBP at fighting the illegal import of pirated product.

Requirements: Authentication/proof of control, Associated cryptographic material, Delegation of control, Inter-jurisdictional

Suggested by DHS SVIP

3. Requirements §

The short use cases in the previous section, and the focal use cases towards the end of the document, point to a set of requirements that Decentralized Identifiers must fulfil. The use cases and their requirements are tabulated below, followed by a definition of each requirement.

	Requirements																			
Use case	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Online shopper	X	X		X						X	X					X	X	X		
Vehicle assemblies	X	X		X						X	X					X	X	X		
Encrypted Data Vault	X				X	X		X					X							
Accessing Master Data of Entities	X	X	X		X		X													
Identifiers in an ecosystem of verifiable credentials (VCs)	X		X		X	X									X					
Sharing opted-in information across platforms		X						X						X						
Collecting payments for work conducted anonymously								X				X						X		
Anonymity within a supply chain	X	X					X	X						X				X		
Digital Permanent Resident Card	X			X	X							X								
Importing retro toys	X				X				X	X										
Decentralized Corporate Identifiers (enterprise)	X	X	X	X	X	X	X		X	X	X		X		X	X	X	X	X	X
Life-long, recipient-managed Credentials (education)	X	X	X	X	X	X	X			X	X	X	X	X	X	X	X	X	X	X
Prescriptions (healthcare)	X	X	X		X		X	X						X					X	
Digital Executor (law)	X	X	X		X	X	X	X	X	X	X	X	X		X		X	X	X	
Single Sign On (security)	X	X	X		X							X			X					

1. Authentication/proof of control
It is possible to prove that the entity claiming control over the identifier is indeed its controller

2. Decentralized/self issued
These identifiers are created and managed by the subject of the identifier. They are not assigned, given, sold, or rented to the individual using them. The party relying on the identifier for identification, authentication, and authorization, does not need to manage the creation, update, and recovery of these identifiers.

3. Guaranteed unique identifier
identifiers are globaly unique with no possibility of duplication, however unlikely that may be.

4. No call home
When using these identifiers, there is no need to contact the issuer of the identifier to verify it. Verification and authentication can occur without further communication with the issuer.

5. Associated cryptographic material
The identifier is tightly coupled with cryptographic material that can be used to prove control over that identifier.

6. Streamlined key rotation
When authentication materials need to be updated, these identifiers can update without direct intervention with requesting parties and with minimal individual interaction.

7. Service endpoint discovery
These identifiers allow requesting parties to look up available service endpoints for interacting with the subject of the identifier.

8. Privacy preserving

Use of the identifier does not, of itself, reveal any information about the subject

9. Delegation of control

The controller of the identifier is able to delegate that control to a third party.

10. Inter-jurisdictional

Inter-jurisdictional identifiers do not depend on the legal jurisdiction in which they are issued. They are valid for uses anywhere without loss of fidelity or reliability. No jurisdiction is directly able to prevent their use.

11. Can't be administratively denied

These identifiers can't be denied or taken away by administrative function. This prevents shifting politics and bad actors from interfering.

12. Minimized rents

These identifiers don't incur ongoing expenses if unused nor on a per transaction basis. Fees based on updates—which incurs network and computational costs to verify—are considered "minimal".

13. No vendor lock in

These identifiers are not dependent on any given vendor. Vendor-specific identifiers restrict usage to that which is acceptable to the vendor and may allow vendors to extract disproportionate rents for usage.

14. Preempt/limit trackable data trails

As cookies and other session/state-tracking mechanisms were gradually turned into scaffolding for unwanted or collusive tracking in the evolution of the web stack, so too might any new data exchange or communication systems unwittingly facilitate unwanted tracking based on new data trails. Resistance to these kinds of surveillance exploits need to be designed into new systems.

15. Cryptographic future proof

These identifiers are capable of being updated as technology evolves. Current cryptography techniques are known to be susceptible to quantum computational attacks. Future-proofed identifiers provide a means to continue using the same identifier with updated, advanced authentication and authorization technologies.

16. Survives issuing organization mortality

These identifiers survive the demise of the organization that issued them. They usefulness of these identifiers survive organizations going out of business, being purchased (and potentially losing domain names or root credentials), and even the internal decay of an organization that no longer has the ability to verify the authenticity of records they once issued.

17. Survives deployment end-of-life

These identifiers are usable even after the systems deployed by requesting parties move past their useful lifetime. They are robust against technology fads and can seamlessly work with both legacy and next-generation systems.

18. Survives relationship with service provider

These identifiers are not dependent on the tenure of the relationship with a service provider. This contrasts with identifiers like service-centric emails, e.g., joe@example.com, which when used as identifiers in other systems can cause problems when individuals no longer use the service provider.

19. Cryptographic authentication and communication

These identifiers enable cryptographic techniques to authenticate individuals and to secure communications with the subject of the identifier, typically using public-private key pairs.

20. Registry agnostic

These identifiers are free to reside on any registry implementing a compatible interface. They are not beholden to any particular technology or vendor.

4. Features and Benefits §

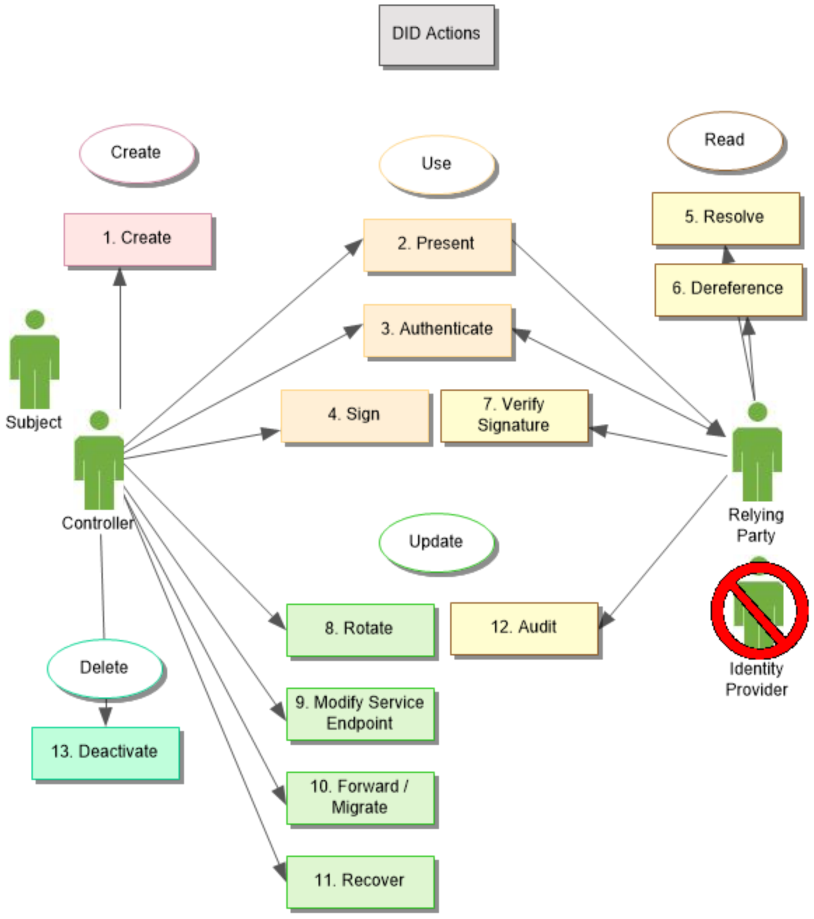
In collecting and evaluating potential use cases and requirements it is posisble to identify the key features of DIDs that provide benefits in the areas of anti-censorship, anti-exploitation, ease of use, privacy, and sustainability. The requirements and their associated benefits can be seen in the following grid.

	Feature				
Requirement	**Anti-censor**	**Anti-explotation**	**Ease of use**	**Privacy**	**Sustainability**
Authentication/proof of control		X			X

Decentralized/self issued	X	X		X	X
Guaranteed unique identifier			X		X
No call home	X	X		X	X
Associated cryptographic material	X	X	X		
Streamlined key rotation	X	X	X		X
Service endpoint discovery			X		
Privacy preserving	X	X		X	
Delegation of control			X		
Inter-jurisdictional	X				X
Can't be administratively denied	X				
Minimized rents					X
No vendor lock in		X			X
Preempt/limit trackable data trails	X	X		X	
Cryptographic future proof		X			X
Survives issuing organization mortality					X
Survives deployment end-of-life	X				X
Survives relationship with service provider		X			X
Cryptographic authentication and communication	X	X			
Registry agnostic	X	X			X

5. DID Actions §

Here are the thirteen (13) actions envisioned in earlier work by Credentials Community Group as being necessarily supported by DIDs. In the diagram, actions have been grouped by Create, Use, Read, Update, and Delete.



5.1 Create §

~~ISSUE 14~~: What does it mean for a DID to be "recorded in a registry"? Needs-WG-Discussion

This section refers to recording in a registry. I've changed it to say 'may be' but this can only be finalized when Issue-14 is resolved. (2020-04-01 - do we mean Issue-14? Doesn't look relevant)

Controllers create DIDs, uniquely binding cryptographic proofs with the identifier, typically using public-private key-pairs. These DIDs may be recorded in a registry in such a manner as to be able to resolve to a DID Document. The DID Document may be dynamically and deterministically generated through resolution or it may be explicitly constructed as a stand-alone resource and either stored or referenced in the registry. In this scenario, the process will need access to any registry, ideally a decentralized system, and like the rest of the DID actions, it should be possible to create the DID without interaction with any particular authority.

5.2 Use §

5.2.1 Present §

DIDs are URIs, which is to say a string of characters. As such, they may be presented in the same manner as URIs by simply transmitting or presenting that string of characters. There is no requirement, however, that DIDs be human readable. Thus they may contain long, complex numbers represented in various formats. For ease of use, implementations may rely on data carriers such as QR codes [QR] for ease of capture using a camera-enabled device such as a smart phone.

5.2.2 Authenticate §

Requesting Parties may wish to prove that the individual presenting a DID is in fact its DID Controller or specified as a Controller for a particular service endpoint. This authentication process should use the cryptographic material in the DID Document to test if the claimed Controller can, in fact, prove control, typically through some sort of challenge-response. DID Documents and methods may allow for separate proofs for different service endpoints, distinct from update and delete actions. This separation would support transactional proofs that are expected to be used frequently, while controlling proofs are expected to be used rarely.

5.2.3 Sign §

Using cryptographic material associated with that found in a DID Document, DID Controllers may sign digital assets or documents. This signature can later be verified to demonstrate the authenticity of the asset. In this way, it should be possible to refer to the asset as "signed by the DID".

5.3 Read §

5.3.1 Resolve §

The first step in using a DID for anything other than presentation is to resolve the DID to a specific DID Document, to reveal the cryptographic material and service endpoints associated with that DID. How this occurs should be method-specific and is out of scope for the DID Working Group.

5.3.2 Dereference §

Dereferencing a DID uses the material in its DID Document to return a resource. The expectation is that, by default, dereferencing a DID without a reference to a service endpoint will return the DID Document itself. When a DID is combined with a `service` parameter (forming a DID URL), dereferencing will return the resource pointed to from the named service endpoint, which was discovered by resolving the DID to its DID Document and looking up the endpoint by name. In this way, a Requesting Party may dynamically discover and interact with the current service endpoints for a given DID. Services can therefore be given persistent identifiers that do not change even when the underlying service endpoints change.

5.3.3 Verify Signature §

Given a digital asset signed by a DID, a Requesting Party may use the cryptographic material in the DID Document to verify the signature.

5.3.4 Audit §

Some methods may provide an explicit audit trail of all actions on that DID, including a timestamp for when the actions took place. For distributed ledger-based registries, this audit trail is fundamental to the way the ledgers record transactions. This would allow Requesting Parties to see, for example, how recently a DID was rotated or its service endpoints updated, which may inform certain analytics regarding the reliability of the DID's cryptographic material.

5.4 Update §

5.4.1 Rotate §

Controllers may rotate (that is, update) the cryptographic material for a DID by updating the DID Document as recorded in its registry. Different methods should be able to handle this differently, but the result would be an update to the core cryptographic proof required to prove control of the DID and the DID Document.

5.4.2 Modify Service Endpoint §

DID Controllers should be able to change service endpoints associated with a DID, including the proof mechanism for authenticating as the Subject for any given endpoint. The process for doing this is method specific, but is designed to allow Controllers to make these changes without necessarily changing the primary proof mechanism for control of the DID itself.

5.4.3 Forward / Migrate §

To support interoperability, some methods may provide a way for DID Controllers to record in their registry (by updating the DID Document), that the DID should be redirected to another DID, which now has full authority to represent the originating DID. This mechanism would allow DID Controllers to migrate a DID from one method or registry to another.

5.4.4 Recover §

Some methods may provide a means for recovering control of a DID if its existing private cryptographic material is lost. These means will vary by method but can include social recovery, multi-signature, Shamir sharing, or pre-rotated keys. In general, recovery triggers a rotation to a new proof, allowing the DID Controller of that new proof to recover control of the DID without interacting with any Requesting Parties.

5.5 Delete §

5.5.1 Deactivate §

Instead of deleting a DID, Controllers should be able to deactivate a DID such that downstream processes like authentication and dereferencing are no longer functional. Most decentralized systems cannot guarantee actual deletion of a record. Indeed, distributed ledgers are often touted as "immutable". Methods should define deactivation processes to achieve the same effect as deletion. The mechanisms for deactivation will vary based on the method.

6. Features and Benefits §

In collecting and evaluating potential use cases, we have identified fifteen (15) key features of DIDs that provide benefits in the areas of anti-censorship, anti-exploitation, ease of use, privacy, and sustainability.

The features and their associated benefits can be seen in the following grid. A brief definition of each feature follows.

Feature	Anti-censor	Anti-exploitation	Ease of Use	Privacy	Susta
1. Inter-jurisdictional	X				
2. Can't be administratively denied	X				
3. Minimized rents		X			
4. No vendor lock in		X			
5. Self-issued, self-managed	X	X		X	
6. Streamlined rotation			X		
7. No phone home				X	
8. Preempt/limit trackable data trails		X		X	
9. Cryptographic future proof					
10. Survives issuing organization mortality			X		
11. Survives deployment end-of-life					
12. Survives relationship with service provider			X		
13. Cryptographic authentication and communication	X	X		X	
14. Service Discovery			X		
15. Registry agnostic	X	X		X	

1. Inter-jurisdictional

Inter-jurisdictional identifiers do not depend on the legal jurisdiction in which they are issued. They are valid for uses anywhere without loss of fidelity or reliability. No jurisdiction is directly able to prevent their use. (Anti-censorship and Sustainable).

2. Can't be administratively denied

These identifiers can't be denied or taken away by administrative function. This prevents shifting politics and bad actors from interfering. (Anti-censorship).

3. Minimized rents

These identifiers don't incur ongoing expenses if unused nor on a per transaction basis. Fees based on updates—which incurs network and computational costs to verify—are considered "minimal". (Anti-exploitation and Sustainable).

4. No vendor lock in

These identifiers are not dependent on any given vendor. Vendor-specific identifiers restrict usage to that which is acceptable to the vendor and may allow vendors to extract disproportionate rents for usage. (Anti-exploitation, Privacy, and Sustainable).

5. Self-issued, self-managed

These identifiers are created and managed by the subject of the identifier. They are not assigned, given, sold, or rented to the individual using them. The party relying on the identifier for identification, authentication, and authorization, does not need to manage the creation, update, and recovery of these identifiers. (Anti-censorship, Ease of use, and Privacy)

6. Streamlined rotation

When authentication materials need to be updated, these identifiers can update without direct intervention with requesting parties and with minimal individual interaction. (Ease of use)

7. No phone home

When using these identifiers, there is no need to contact the issuer of the identifier to verify it. Verification and authentication can occur without further communication with the issuer. (Privacy)

8. Preempt/limit trackable data trails

As “cookie” and other session/state-tracking mechanisms were gradually turned into scaffolding for unwanted or collusive tracking in the evolution of the web stack, so too might any new data exchange or communication systems unwittingly facilitate unwanted tracking based on new data trails. Resistance to these kinds of surveillance exploits need to be designed into new systems.

9. Cryptographic future proofing

These identifiers are capable of being updated as technology evolves. Current cryptography techniques are known to be susceptible to quantum computational attacks. Future-proofed identifiers provide a means to continue using the same identifier with updated, advanced authentication and authorization technologies. (Sustainable)

10. Survives organizational mortality

These identifiers survive the demise of the organization that issued them. They usefulness of these identifiers survive organizations going out of business, being purchased (and potentially losing domain names or root credentials), and even the internal decay of an organization that no longer has the ability to verify the authenticity of records they once issued.

11. Survives deployment end-of-life

These identifiers are usable even after the systems deployed by requesting parties move past their useful lifetime. They are robust against technology fads and can seamlessly work with both legacy and next-generation systems.

12. Survives relationship with service provider

These identifiers are not dependent on the tenure of the relationship with a service provider. This contrasts with identifiers like service-centric emails, e.g., joe@example.com, which when used as identifiers in other systems can cause problems when individuals no longer use the service provider.

13. Cryptographic authentication and communication

These identifiers enable cryptographic techniques to authenticate individuals and to secure communications with the subject of the identifier, typically using public-private key pairs.

14. Service discovery

These identifiers allow requesting parties to look up available service endpoints for interacting with the subject of the identifier. (Ease of use and Sustainable)

15. Registry agnostic

These identifiers are free to reside on any registry implementing a compatible interface. They are not beholden to any particular technology or vendor.

7. Feature Coverage §

Not all use cases illustrate each feature, and not all DID methods support all features. The following chart shows which features are explicitly illustrated in the Focal Use Cases.

Feature	Corporate	Educational credentials	Prescriptions	Digital Executor
1. Inter-jurisdictional	X	X		X
2. Can't be administratively denied	X		X	X
3. Minimized rents			X	
4. No vendor lock in	X	X	X	X
5. Self-issued, self-managed	X	X	X	X
6. Streamlined rotation		X		X
7. No phone home		X	X	
8. Preempt/limit trackable data trails			X	
9. Cryptographic future proof	X	X		
10. Survives issuing organization mortality		X		
11. Survives deployment end-of-life		X		
12. Survives relationship with service provider	X	X		

13. Cryptographic authentication and communication	X	X	X	X
14. Service Discovery				
15. Registry agnostic				

8. Focal Use Cases §

8.1 Decentralized Corporate Identifiers (enterprise) §

8.1.1 Background §

There are many types of identifiers that corporations use today including tax identification numbers (e.g. 238-42-3893), Legal Entity Identifiers (e.g. 5493000IBP32UQZ0KL24), Data Universal Numbering System identifiers (a.k.a. DUNS Number, e.g. 150483782), Global Location Number (e.g. 9501101020016), and many more that communicate the unique identity of an organization. None of these numbers enable an organization to self-issue an identifier or to use the number to cryptographically authenticate or digitally sign agreements. Business to business and business to customer transactions might be conducted with more efficiency and with greater assurance of the validity of the transaction if a mechanism to self-issue cryptographic identifiers were created.

8.1.2 Description §

A North American government would like to ensure that the supply chain that feeds electronic products into the country is secure. As a result, a new method of submitting digital documentation to Customs is enabled that requires that all documentation is provided as machine-readable digitally signed data with proof of provenance from supply chain partners whose identities are themselves known to a high degree of certainty. Digitally signed documentation is collected at each stage of the manufacturing, packaging, and shipping process. This documentation is then submitted to Customs upon the product's entry into the country where all digital signatures are verified on the documentation. Some aspects of the signed documentation, such as firmware hashes and checksums, are then used by Customs and downstream customers to verify that the products have not been tampered with after leaving the manufacturing facility.

Decentralized Identifiers should ensure 1) low management overhead for the government, 2) self-management of identifiers and cryptographic key material, and 3) a competitive marketplace.

8.1.3 Challenges §

The requirement of downstream customers to use the same documentation and digital signature mechanisms that were provided to Customs is potentially problematic in this scenario. Governments often create ad-hoc solutions for their import solutions, which make securing the global supply chain difficult as each government has their own method of securing the supply chain and identifying corporations that downstream customers need to integrate with. If you are a global company, that means integrating with many supply chain systems (each with different capabilities). As such, any securing of the supply chain with downstream customers must then depend on the country-specific corporate identification and PKI solution, which leads to ad-hoc solutions that drive up the cost of doing business across borders.

A supply chain identifier solution that is simple, self-administered, built on global standards, is flexible in the cryptographic mechanisms used to authenticate, and can be used by governments and downstream customers with little to no modification to the regional government or corporate systems does not exist today.

8.1.4 Distinctions §

Many Decentralized Identifier use cases focus on Self-Sovereign Identity and individuals. This use case focuses on organizations and their departments as entities that would also benefit from Decentralized Identifiers.

Requirements: Authentication/proof of control, Decentralized/self issued, Guaranteed unique identifier, No call home, Associated cryptographic material, Streamlined key rotation, Service endpoint discovery, Delegation of control, Inter-jurisdictional, Can't be administratively denied, No vendor lock in, Cryptographic future proof, Survives issuing organization mortality, Survives deployment end-of-life, Survives relationship with service provider, Cryptographic authentication and communication, Registry agnostic

8.2 Life-long, recipient-managed Credentials (education) §

Contributed by Kim Hamilton-Duffy

8.2.1 Background §

Educational Verifiable Credentials [VC-DATA-MODEL] offer benefits over traditional educational credentials in that the recipient is able to store and share their credentials, and a third party may independently verify the credential (including authenticating the identity of the recipient), without necessarily consulting the issuer, and without dependence on centuries old treaty-based bureaucratic process for the international verification of credentials. This provides the promise of recipient-owned long-lived credentials that the recipient may use in any country, even if the issuing institution goes out of business.

However, traditional public-private key pair-based identifiers present challenges for rotating keys, especially if the identifier in a credential is simply the public key (with the private key used for authentication).

1. With the public key embedded in the digitally signed credential, it is literally impossible to update the signing key; a recipient must contact the issuer and request re-issuing with a new identifier. If the issuer is not reachable, or is unwilling or unable to issue a new credential, the recipient cannot update the cryptographic material.
2. If the credential relies on a centralized key registry or authority for managing rotation, then that registry becomes the centralized point of failure. This may or may not be an improvement, especially for longer held credentials.
3. If the credential's cryptographic technology becomes outdated, there is no way to update the credential to use a more robust technology; a recipient must contact the issuer for reissuance.

The key rotation is particularly problematic for credentials expected to last a lifetime. It should be anticipated that a given individual will change their key management strategy and systems several times over the course of their life, e.g. relying on a cloud wallet, a mobile wallet, or a dedicated hardware wallet, as their needs change.

By issuing an educational credential to a recipient's DID, the recipient has the ability to prove ownership of a credential even if the cryptographic material used for authenticating changes over time.

8.2.2 Description §

When Sally earned her master’s degree at the University of Oxford, she received a digital diploma that contained a decentralized identifier she provided. This digital diploma is signed using a decentralized identifier which has been published and verified by the University of Oxford. Over time, she updates the cryptographic material associated with that DID to use her latest hardware wallet, with biometric protections and a quantum resistant algorithm. A decade after graduation, she applies for a job in Japan, for which she provides her digital diploma by uploading it to the prospective employee’s website. To verify she is the actual recipient of that degree, she uses the decentralized identifier to authenticate, using her current hardware wallet (with rotated keys). In addition to the fact that her name matches the name on the diploma, the cryptographic authentication provides a robust verification of her claim, allowing the employer to rely on Sally’s assertion that she earned a master’s degree from the stated university without having to contact the university directly.

8.2.3 Challenges §

Rotating keys without invalidating Sally's educational credentials and providing acceptable proof of education internationally.

8.2.4 Distinction §

Oxford had no need to provide services for resetting or updating Sally's username or password; they had no role in managing Sally's changes to her authentication credentials. The potential employer did not need to contact Oxford to verify Sally's claim of a master's degree; they were able to verify the credential and authenticate Sally's identity with information retrieved over the Internet.

Requirements: Authentication/proof of control, Decentralized/self issued, Guaranteed unique identifier, No call home, Associated cryptographic material, Streamlined key rotation, Service endpoint discovery, Inter-jurisdictional, Can't be administratively denied, Minimized rents, No vendor lock in, Preempt/limit trackable data trails, Cryptographic future proof, Survives issuing organization mortality, Survives deployment end-of-life, Survives relationship with service provider, Cryptographic authentication and communication, Registry agnostic

8.3 Prescriptions (healthcare) §

8.3.1 Background §

Alicia wants help with her urinary tract infection (UTI) and is a bit touchy about her privacy. In the old days, she would have to make an appointment in-person and get a paper prescription to take to a pharmacy. She wants to save money and have peace of mind.

8.3.2 Description §

Alicia is in a state that allows an online service to diagnose and prescribe medication. She uses the identity wallet on her smartphone to register with the online medical practice. She tells the online practice her name is Althea (a pseudonym) with password-less authentication and a verified driver's license credential to prove that she's a resident of the state. The remote physician, Barkley, is licensed by the state Board of Medicine and credentialed by the online service. He's securely signed in using the identity wallet on his smartphone. Barkley issues Alicia a digital prescription in the form of a verifiable credential and allows Alicia to download it however she pleases. Alicia is a librarian and trusts her local public library to erase their logs as allowed by law. She uses one of their computers to sign-in and do all of this. She snaps a picture of the QR code that is the prescription to take to the pharmacy. Connor, the licensed pharmacist, scans the prescription QR code and fills the prescription. Alicia pays cash.

8.3.3 Challenges §

The challenge of this particular use-case is that only Barkley and Connor are verified identities and accountable for their interaction with Alicia. Alicia can be anonymous or pairwise-pseudonymous with both Barkley and Connor and everything just works. Alicia, Barkley, and Connor all keep separate and legally authentic copies of the records of their interaction in case of dispute.

8.3.4 Distinction §

The Prescription use-case is a common and high-value example of privacy engineering as we shift to convenient and cost-effective online commerce among licensed and unlicensed individuals as peers. Barkley and Connor benefit by reducing or even eliminating the influence of their respective institutions or employers and therefore make more money. They pass some savings to Alicia who also gets increased peace of mind.

Requirements: Authentication/proof of control, Decentralized/self issued, Guaranteed unique identifier, Associated cryptographic material, Service endpoint discovery, Privacy preserving, Preempt/limit trackable data trails, Cryptographic authentication and communication

8.4 Digital Executor (law) §

8.4.1 Background §

Today, when people die, there are no standard technologies for heirs, executors, or probate courts to properly take control of an individual's online accounts and digital assets. With a DID linked to accounts and assets, a DID owner could define a trigger for a third party to assume control over the DID Document. Ideally, this trigger would specify (a) an oracle (how to know the death/incapacity occurred), (b) a means for the new owner to assert control, and (c) appropriate checks and accountability.

8.4.2 Description §

Kathy uses DIDs to manage her authentications to various services. As part of her estate planning, she generates a unique credential that she gives to her attorney, Gloria, with provisions specified in her will, which initially lists Mike as the digital executor. With appropriate obfuscation, that credential is specified in multiple DID documents as a probate authority, with the authorization to change the master key in case of death, which shall be recorded publicly, on chain, as a notarized invocation of the probate authority. As it happens, Kathy had a falling out with Mike and notified Gloria just two weeks before her death that her friend Miyake should now be her digital executor. Upon Kathy's death, Gloria uses the probate credential to publicly record the assertion of probate and to replace the DID's master key with a new key, controlled by Miyake, who lives in Japan (Kathy, Gloria, and Mike live in the United States). Now, any system using Kathy's DIDs for authentication can programmatically recognized Miyake's authority *and* specifically know that Kathy's credentials were modified under a assertion of probate.

8.4.3 Challenges §

The late date change in digital executorship from Mike to Miyake could be problematic if Kathy had directly listed Mike's credential in the DID Document. Because she instead chose to rely on her attorney, Kathy has a more flexible way to direct her wishes, while still leveraging the collective control over her authenticated logins to various services. In addition, Miyake's geographic location could make it hard for them to travel to the United States and may make it difficult to provide proof of identity traditionally used by U.S. courts. Also, because Gloria invokes the probate mechanism, Miyake need only provide a suitable credential at that time; he did not need to create and maintain a credential over a long period of time (as would be the case if Gloria weren't involved).

8.4.4 Distinction §

Multiple DIDs with a common, blinded authority for probate assumption of control. The legal selection of the new owner is mediated through a trusted fiduciary (an attorney of record). Cross-border transfer of ownership.

Requirements: Authentication/proof of control, Decentralized/self issued, Guaranteed unique identifier, Associated cryptographic material, Streamlined key rotation, Service endpoint discovery, Privacy preserving, Delegation of control, Inter-jurisdictional, Can't be administratively denied, Minimized rents, No vendor lock in, Cryptographic future proof, Survives deployment end-of-life, Survives relationship with service provider, Cryptographic authentication and communication

8.5 Single Sign On (security) §

8.5.1 Background §

Passwords are notoriously misused ("123456"), stolen from the supposedly-secure database on the server-side, easy to forget when sufficiently secure, and never the last word in authentication for forgotten password situations. Proving control of a DID can replace storage and retrieval of a shared secret.

8.5.2 Description §

EDITOR'S NOTE

This section in particular needs review to ensure it carries information relevant to the WG's work.

Use a DID as a single-sign-on to a Web site, for example between a Web page and a Web browser with a mobile identity app. When desirable, the relationship can add a shared secret for two-factor authentication (2FA).

Detailed aspects of this use case are out of scope for the Decentralized Identifier Working Group but they have been explored elsewhere [DID-Auth].

8.5.3 Challenges §

Transfer sign-on capability from control of a password to control of the DID.

8.5.4 Distinction §

This use case describes the most common authentication action for people on the Internet.

Requirements: Authentication/proof of control, Decentralized/self issued, Guaranteed unique identifier, Associated cryptographic material, Minimized rents, Cryptographic future proof

A. References §

A.1 Informative references §

[DID-Auth]
Introduction to DID Auth. Markus Sabadello; Kyle Den Hartog; Christian Lundkvist; Cedric Franz; Alberto Elias; Andrew Hughes; John Jordan; Dmitri Zagidulin. URL: https://github.com/WebOfTrustInfo/rwot6-santabarbara/blob/master/final-documents/did-auth.md

[DID-SPEC-REGISTRIES]
DID Specification Registries. Orie Steele; Manu Sporny. W3C. 1 October 2020. W3C Note. URL: https://www.w3.org/TR/did-spec-registries/

[QR]
Information technology — Automatic identification and data capture techniques — QR Code bar code symbology specification. ISO/IEC JTC 1/SC 31 Automatic identification and data capture techniques. February 2015. ISO/IEC standard. URL: https://www.iso.org/standard/62021.html

[RFC3986]
Uniform Resource Identifier (URI): Generic Syntax. T. Berners-Lee; R. Fielding; L. Masinter. IETF. January 2005. Internet Standard. URL: https://tools.ietf.org/html/rfc3986

[RFC4122]
A Universally Unique IDentifier (UUID) URN Namespace. P. Leach; M. Mealling; R. Salz. IETF. July 2005. Proposed Standard. URL: https://tools.ietf.org/html/rfc4122

[RFC7231]
Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. R. Fielding, Ed.; J. Reschke, Ed.. IETF. June 2014. Proposed Standard. URL: https://httpwg.org/specs/rfc7231.html

[VC-DATA-MODEL]
Verifiable Credentials Data Model 1.0. Manu Sporny; Grant Noble; Dave Longley; Daniel Burnett; Brent Zundel. W3C. 19 November 2019. W3C Recommendation. URL: https://www.w3.org/TR/vc-data-model/

↑