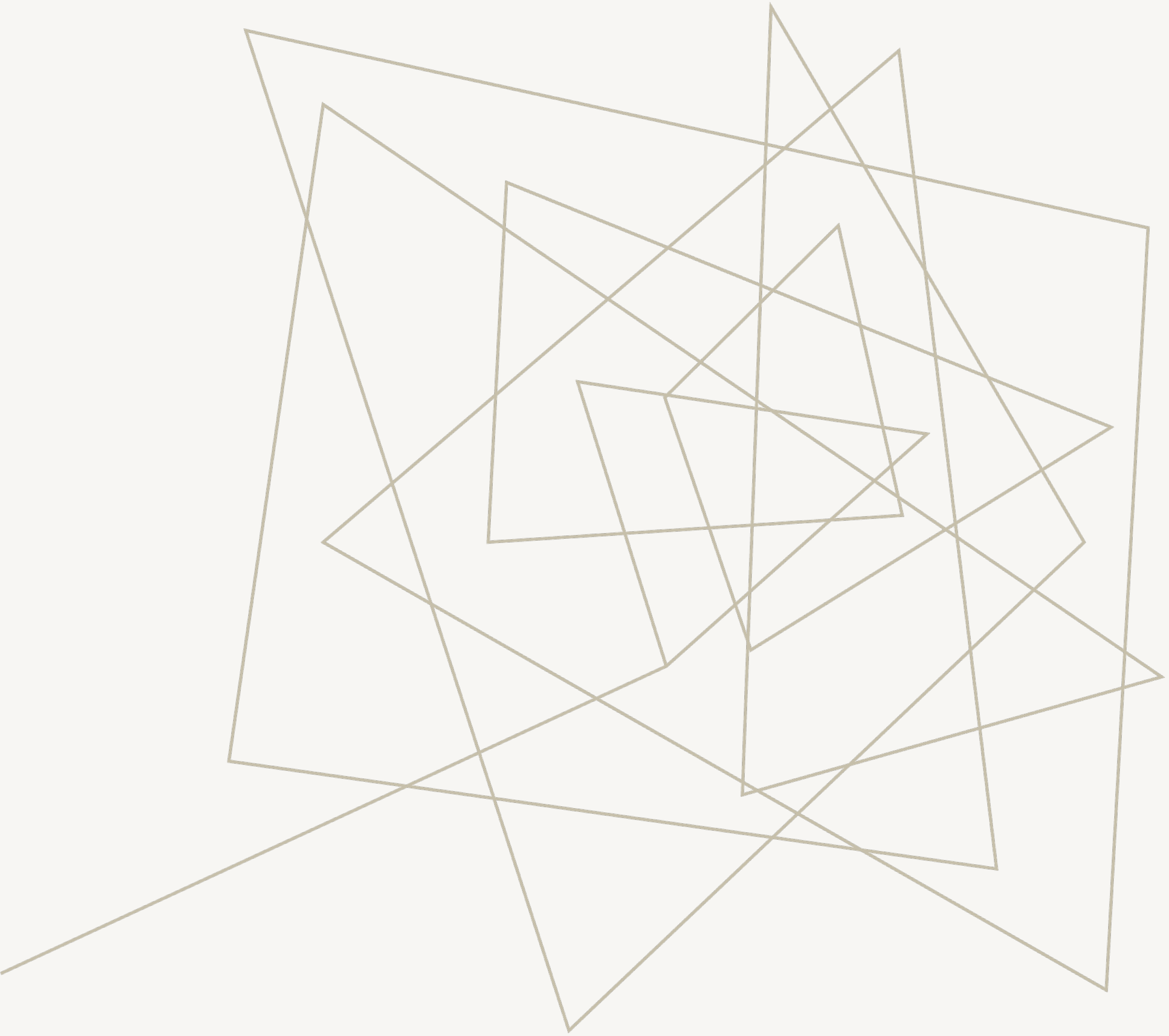
Decentralized Identity: Overview of Verification Method Revocation in Trustless Systems for DID Methods

Clare Nelson, CISSP, CIPP/E, AWS CCP
Executive Director
Decentralized Identity Foundation (DIF)
DIF website: https://identity.foundation/
Email: clare@identity.foundation

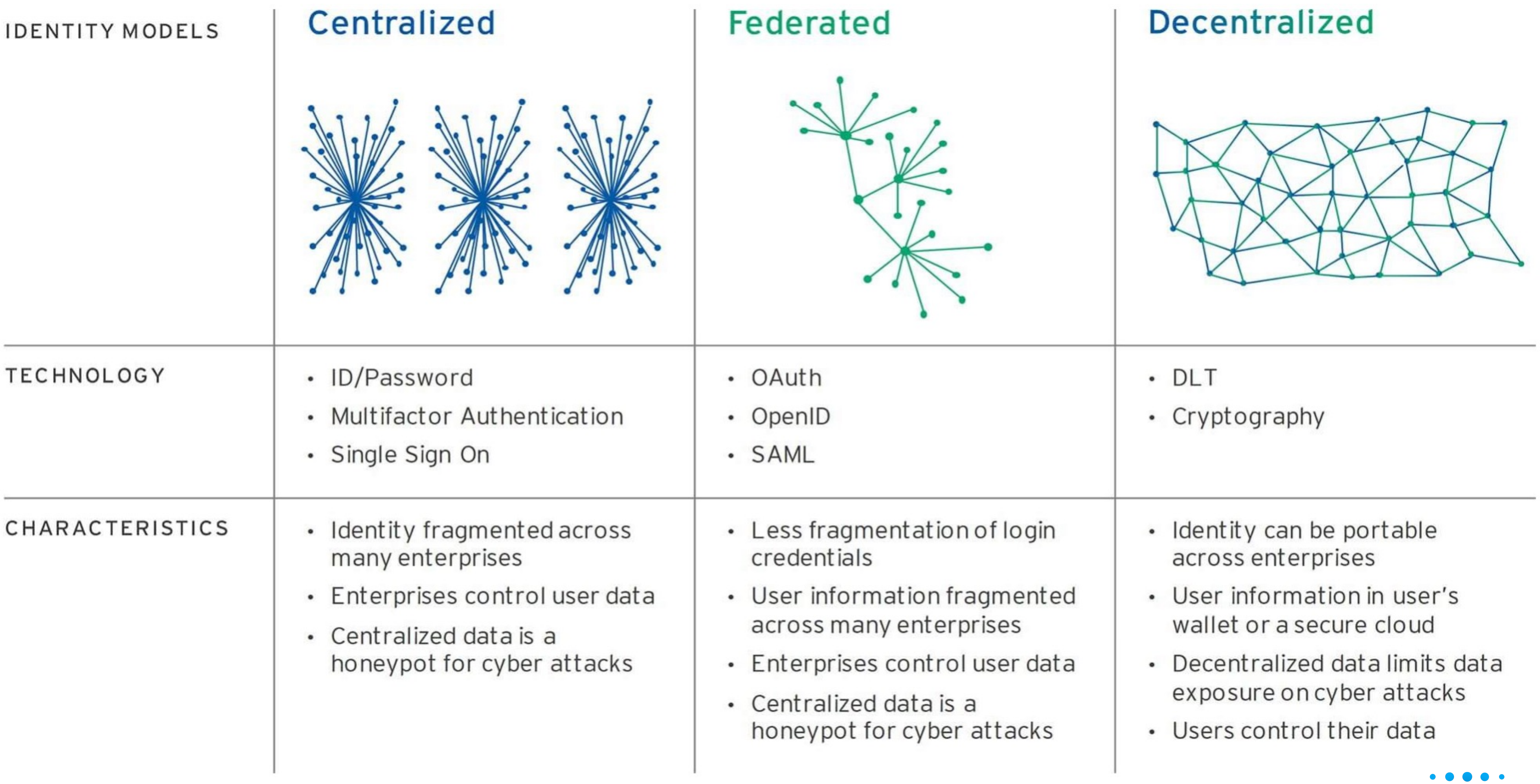
Contents

1. Introduction
   - Decentralized Identity
   - Scope
   - Decentralized Identifiers (DIDs)
   - Verifiable Credentials (VCs)
   - Verification Method (Keys)
   - Verification Method Revocation
   - Trustless Systems
2. W3C DID Recommendation, Section 9.8 Verification Method Revocation
3. Verification Method Revocation for Various DID Methods
4. Future Keys, Post Quantum
5. Conclusion
6. References



What is Decentralized Identity?

Decentralized Identity



IDENTITY MODELS	Centralized	Federated	Decentralized
TECHNOLOGY	• ID/Password • Multifactor Authentication • Single Sign On	• OAuth • OpenID • SAML	• DLT • Cryptography
CHARACTERISTICS	• Identity fragmented across many enterprises • Enterprises control user data • Centralized data is a honeypot for cyber attacks	• Less fragmentation of login credentials • User information fragmented across many enterprises • Enterprises control user data • Centralized data is a honeypot for cyber attacks	• Identity can be portable across enterprises • User information in user's wallet or a secure cloud • Decentralized data limits data exposure on cyber attacks • Users control their data

Source: https://www.citi.com/ventures/perspectives/opinion/digital-identity.html

Decentralized Identity

Gartner

"Decentralized identity is important for confirming user identities and securely storing them. It offers numerous advantages separate of the greater identity autonomy it delivers to customers."[1]

Deloitte.

"Individuals can own and manage their own tamper-proof credentials for applications such as personal health, education, and voting records in an encrypted digital wallet on their personal devices."[2]

wipro

"Utilizing DID improves the capabilities of anomaly detection systems. It will be easy to blend these systems with the existing ones to strengthen prevention processes and enhance privacy. The additional layer of security that DID will offer without compromising consumer privacy is invaluable."[3]

accenture

"Accenture has stellar capabilities to integrate a combined IAM and decentralized identity system with core organizational and business functions and cutting-edge technologies to create a holistic, future-forward solution to meet the needs of users and businesses, such as Blockchain, Biometrics, Analytics, AI, and more."[4]

Forbes

"…passkeys do not protect our *privacy* or give us complete control of our online identities. For that to happen, we need to look at self-sovereign identity (SSI)."[5]

[1] Source: Gartner, https://www.gartner.com/reviews/market/decentralized-identity-solutions

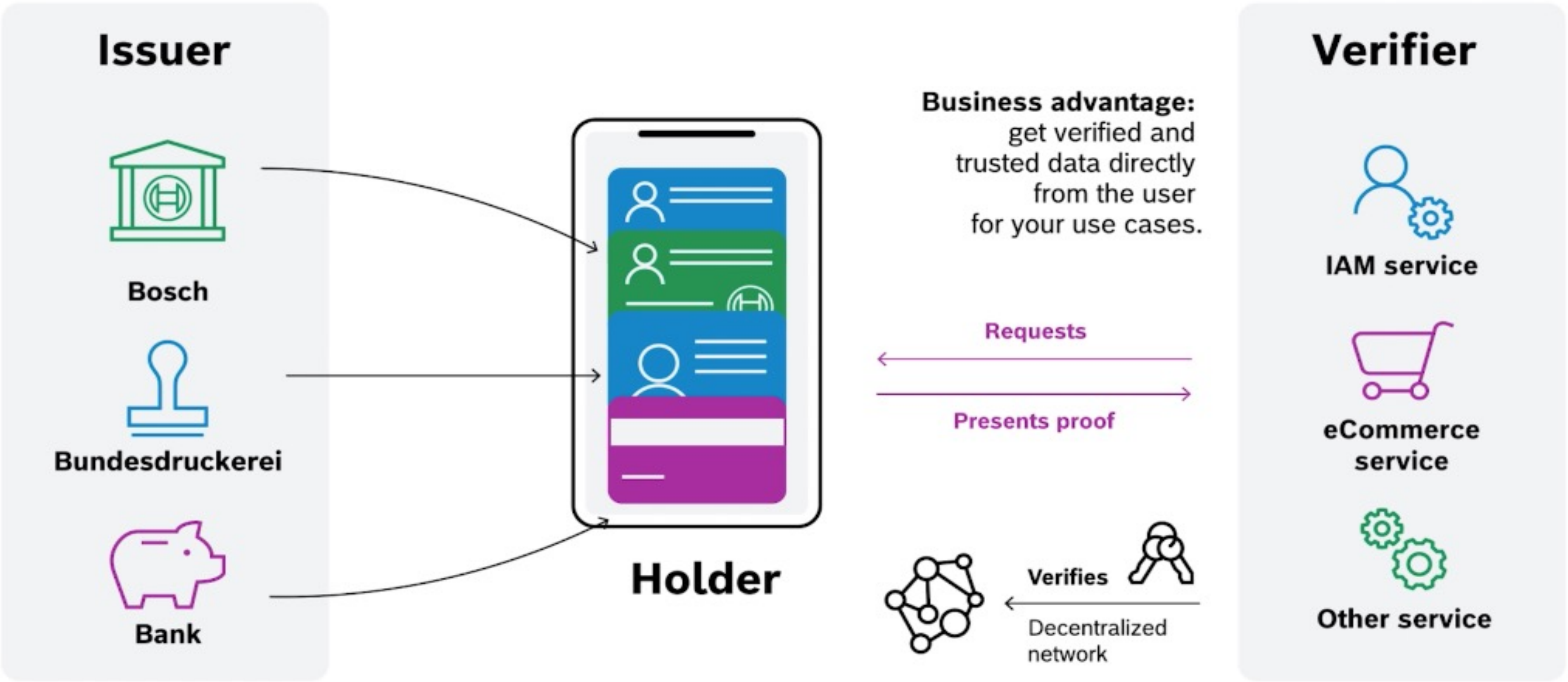
[2] Source: Deloitte, https://www2.deloitte.com/us/en/insights/focus/tech-trends/2023/trustless-blockchain-decentralized-internet.html

[3] Source: Wipro, https://www.wipro.com/innovation/improve-detection-of-online-frauds-using-decentralized-identity-management/

[4] Source: Accenture, https://www.accenture.com/_acnmedia/PDF-173/Accenture-Decentralize-Digital-Identity.pdf

[5] Source: Forbes, https://www.forbes.com/sites/forbestechcouncil/2022/09/26/self-sovereign-identity-taking-control-over-your-digital-identity/?sh=6918b35364e0

Decentralized Identity, Example



Issuer

Bosch

Bundesdruckerei

Bank

Holder

Business advantage: get verified and trusted data directly from the user for your use cases.

Requests

Presents proof

Verifies

Decentralized network

Verifier

IAM service

eCommerce service

Other service

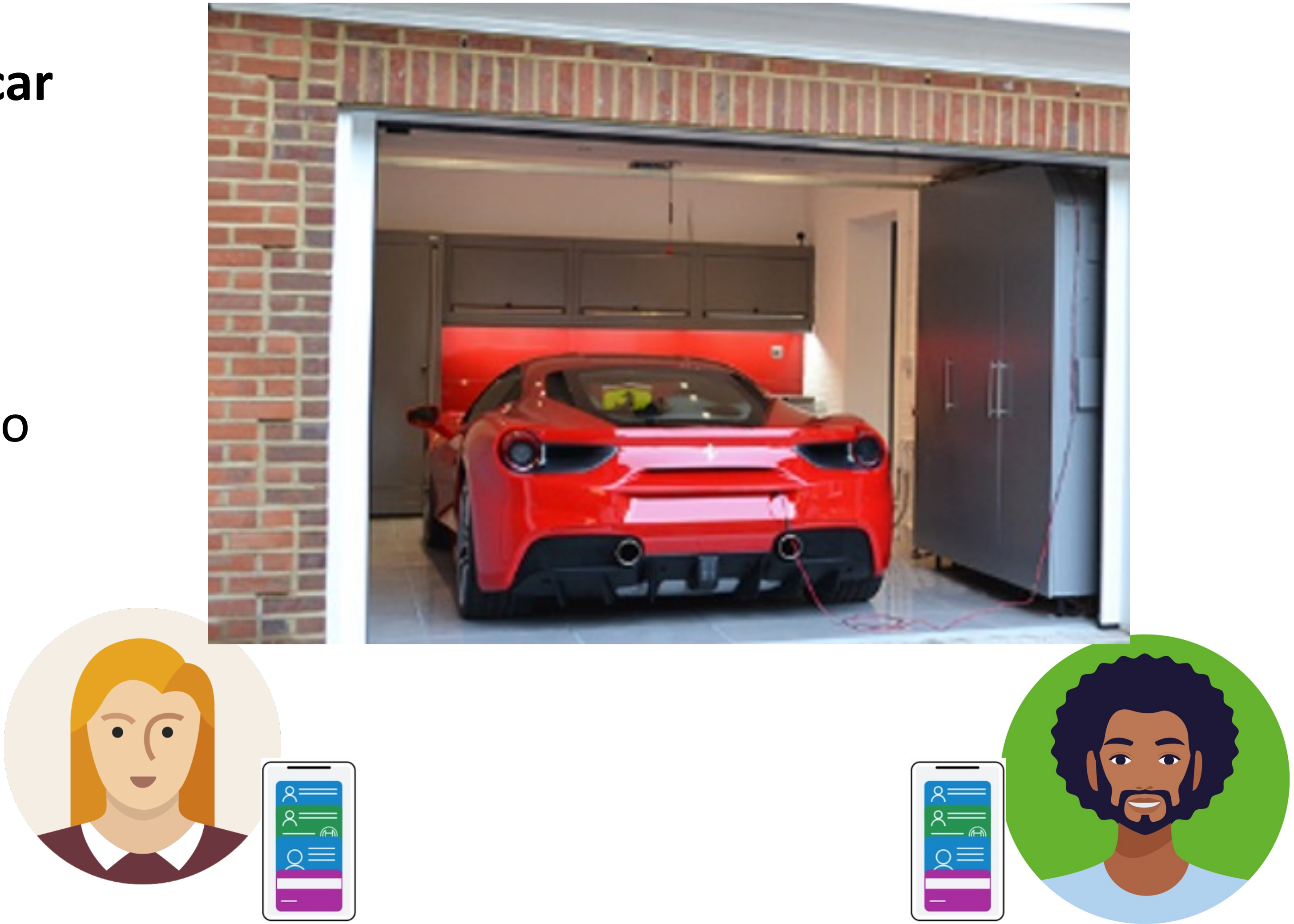
Source: https://www.bosch.com/stories/self-sovereign-identities/

Decentralized Identity, Example

Give your neighbor access to your car

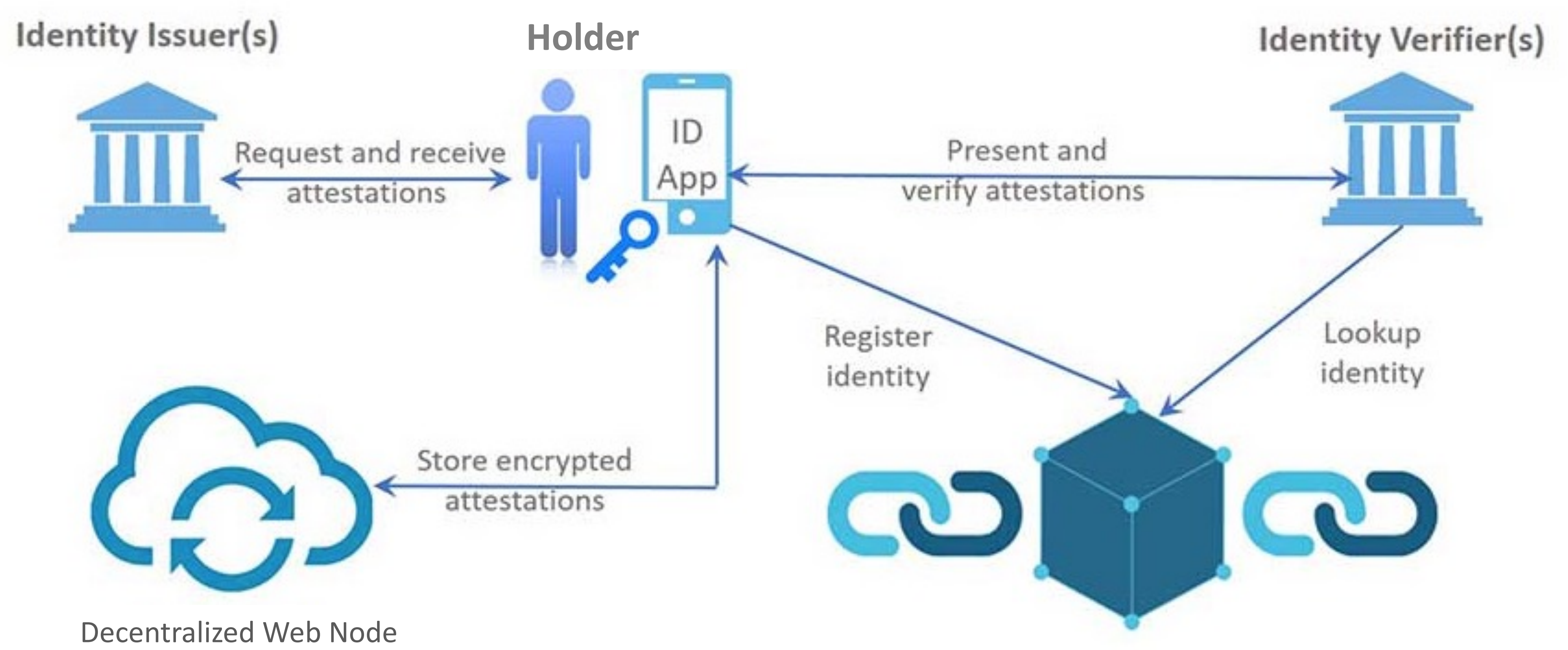
- You are issuer
- Neighbor is a holder
- Garage door, car are verifiers

Peer-to-peer, no central authority, no Identity Provider (IdP)



Source: https://www.bosch.com/stories/self-sovereign-identities/
Figures: https://www.vecteezy.com/free-vector/black-man
Figures: https://www.iconfinder.com/icons/403023/avatar_user_woman_female_person_young_icon

Decentralized Identity, Example



Does Decentralized Identity have to be implemented on a blockchain or digital ledger?

Source: https://www.businessworld.in/article/The-Future-Of-Identity-Is-Self-Sovereign-And-Enabled-By-Blockchain/29-10-2018-163096/

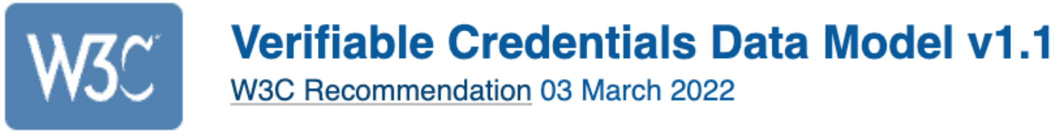
What is Our Scope?

Decentralized Identity: Scope

Decentralized Identifiers (DIDs) v1.0

Core architecture, data model, and representations

W3C Recommendation 19 July 2022



Verifiable Credentials Data Model v1.1

W3C Recommendation 03 March 2022

Decentralized Identifier (DID)

- Globally unique identifier that is designed to provide decentralized control over the entity's identity and personal data

Verifiable Credential (VC)

- Digitally signed and tamper-evident set of claims about an entity's identity, attributes, or qualifications

By including a DID in a verifiable credential, the credential holder can control who can access their data and how it is used

Source: https://www.w3.org/TR/did-core/
Source: https://www.w3.org/TR/vc-data-model/

Verification Method (Key) Revocation

	Term	Explanation
ø	**Credential Revocation**	Revoking a Verifiable Credential (VC)
ø	**DID Rotation**	In DIDComm, DIDs may be rotated for a number of reasons[1]
ø	**DID Recovery**	Recovery is a reactive security measure whereby a controller that has lost the ability to perform DID operations, such as through the loss of a device, is able to regain the ability to perform DID operations
ø	**Verification Method Rotation**	Rotating a DID Verification Method, e.g., cryptographic key, typically a proactive process
✓	**Verification Method Revocation**	Revoking a DID Verification Method, e.g., cryptographic key, typically a reactive process

[1] Source: https://didcomm.org/book/v2/didrotation

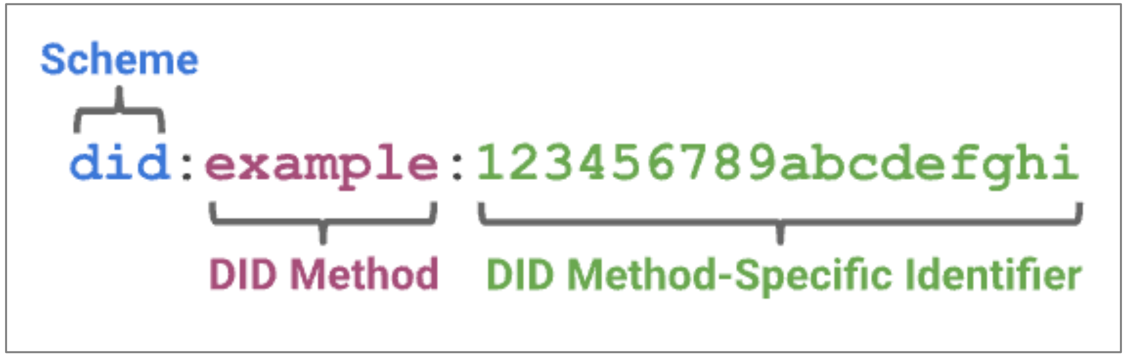
What are Decentralized Identifiers (DIDs)?

What are Decentralized Identifiers (DIDs)?

A DID refers to any subject (person, organization, thing, data model, abstract entity) as determined by the controller of the DID

- Decoupled from centralized registries, identity providers, and certificate authorities
- Controller of a DID can prove control over it without requiring permission from any other party

DIDs are URIs that associate a DID Subject with a DID Document allowing trustable interactions associated with that subject

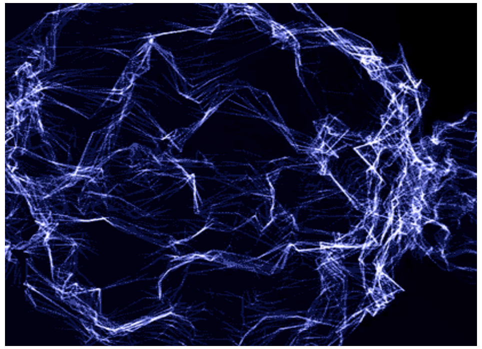

Scheme
did:example:123456789abcdefghi
DID Method
DID Method-Specific Identifier

Person

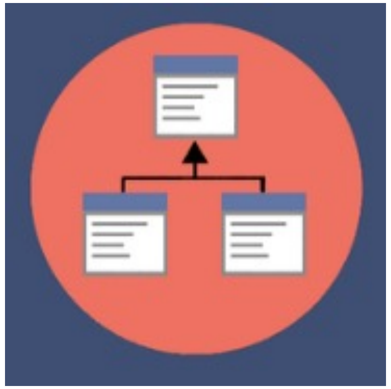
Organization (ACM headquarters)

IoT

Basketball with sensor



Abstract entity



Data Model

Source: https://www.w3.org/TR/did-core/

Graphics: https://adioma.com/icons/walking-person, https://en.wikipedia.org/wiki/Association_for_Computing_Machinery, , https://www.livescience.com/43410-94fifty-smart-sensor-basketball-review.html, https://dataedo.com/blog/basic-data-modeling-techniques

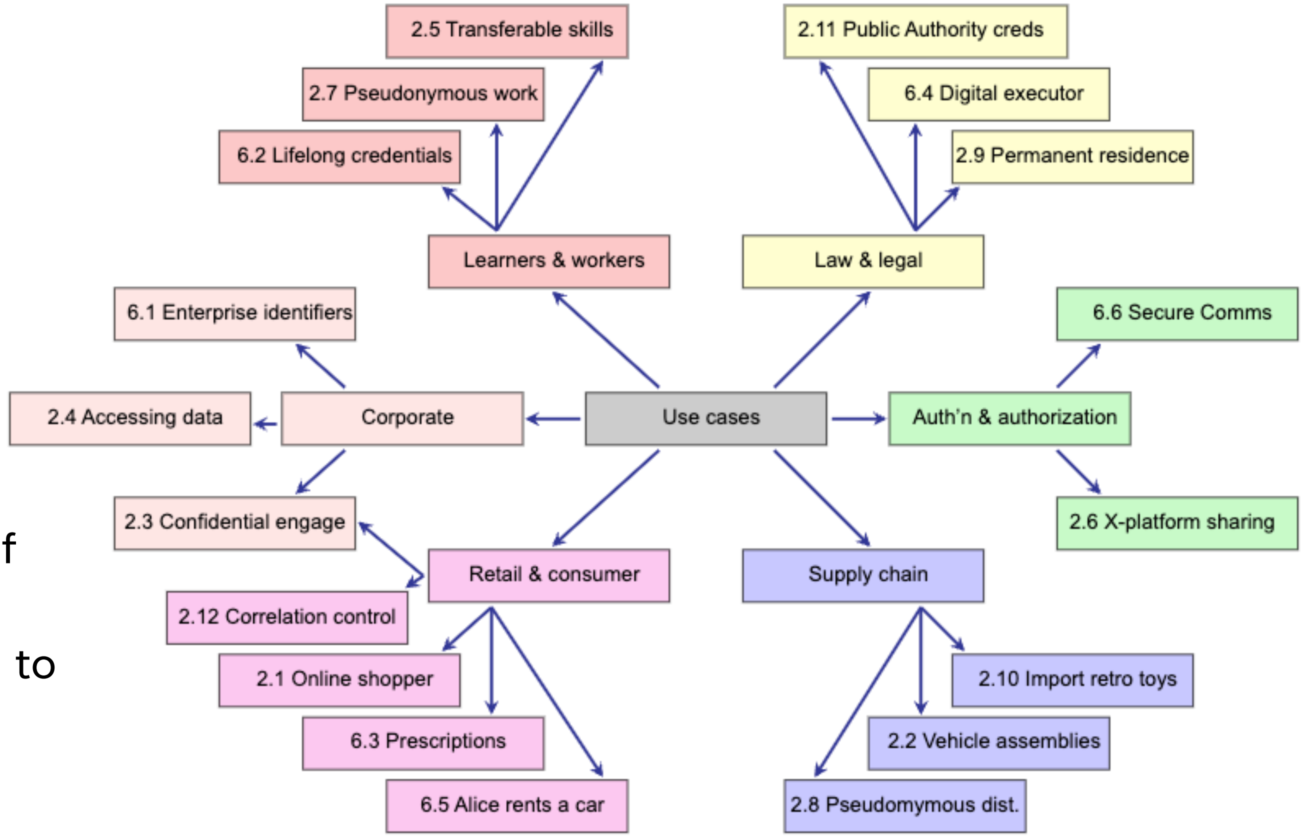
Decentralized Identifiers (DIDs)

Use Cases and Requirements for Decentralized Identifiers

W3C Working Group Note 17 March 2021

DID Requirements, four essential characteristics

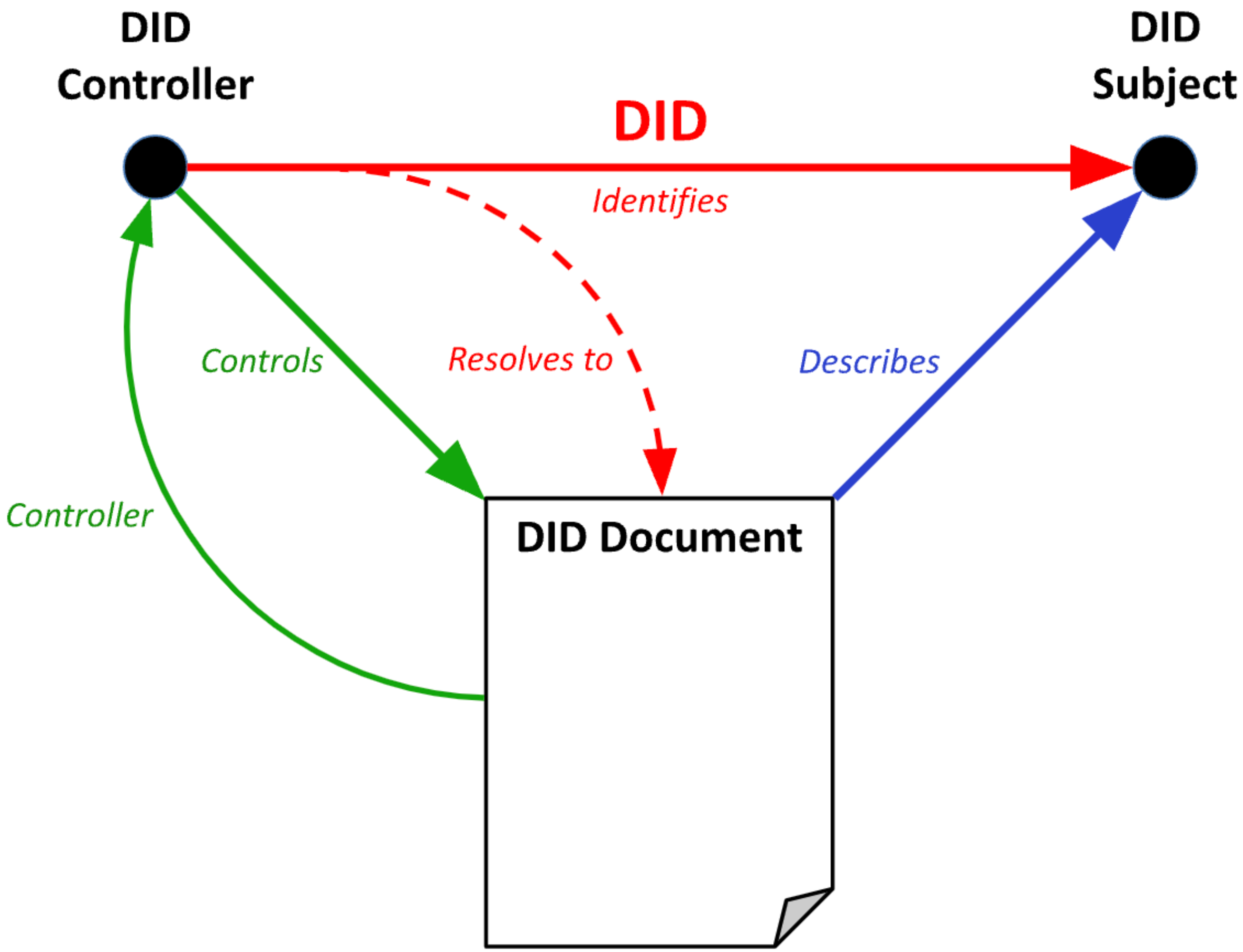
1. **Decentralized**: there should be no central issuing agency
2. **Persistent**: the identifier should be inherently persistent, not requiring the continued operation of an underling organization
3. **Cryptographically verifiable**: it should be possible to prove control of the identifier cryptographically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



Use Cases

Source: https://www.w3.org/TR/did-use-cases/

Decentralized Identifiers (DIDs): Public Key

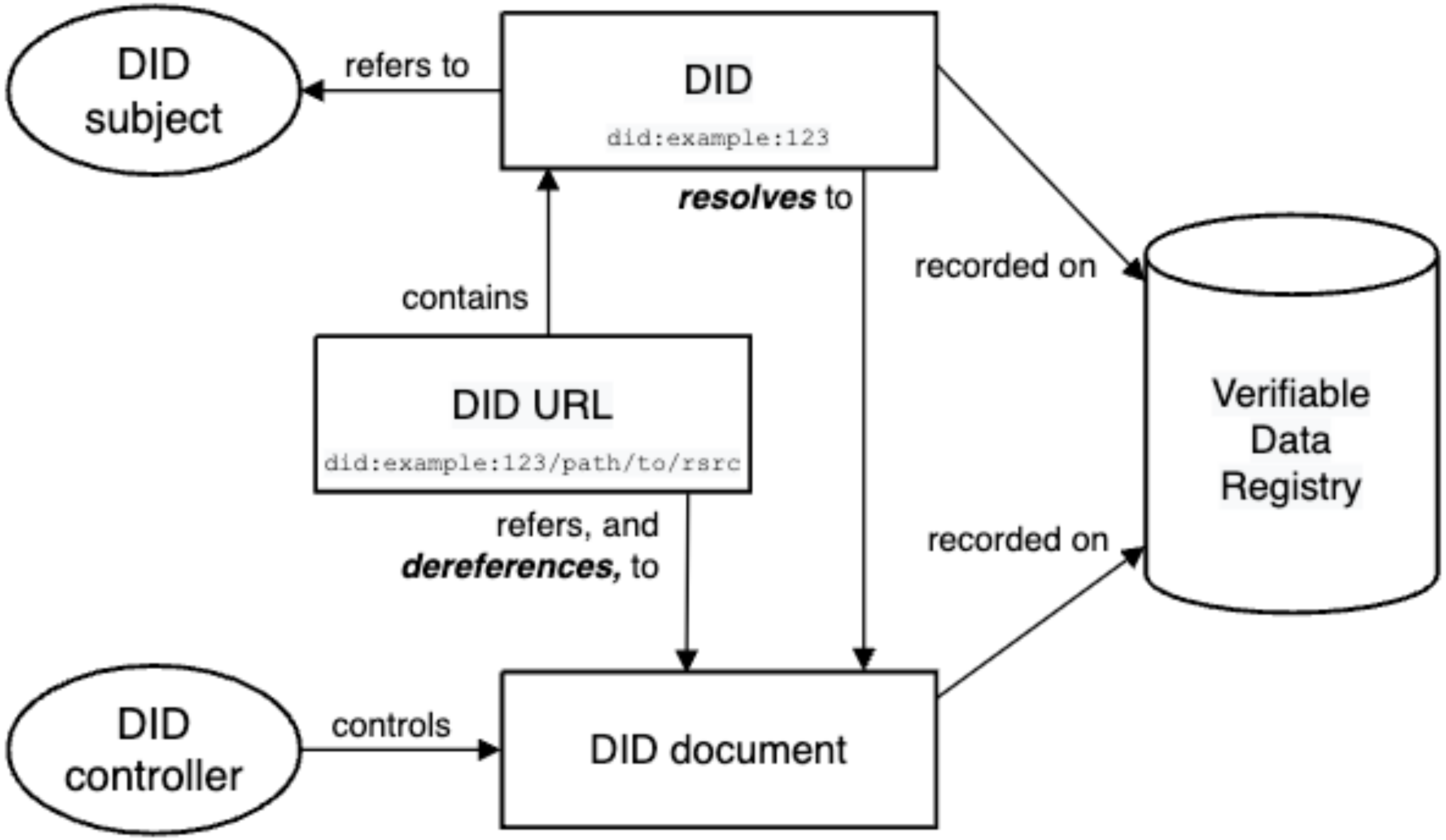


DID creation generates key pairs

The DID Document includes the Public Key element that describes the public keys associated with the DID

Source: https://www.w3.org/TR/did-core/

DID Architecture



DIDs

- Typically recorded on an underlying system or network
- Resolvable to DID Documents

Verifiable Data Registry

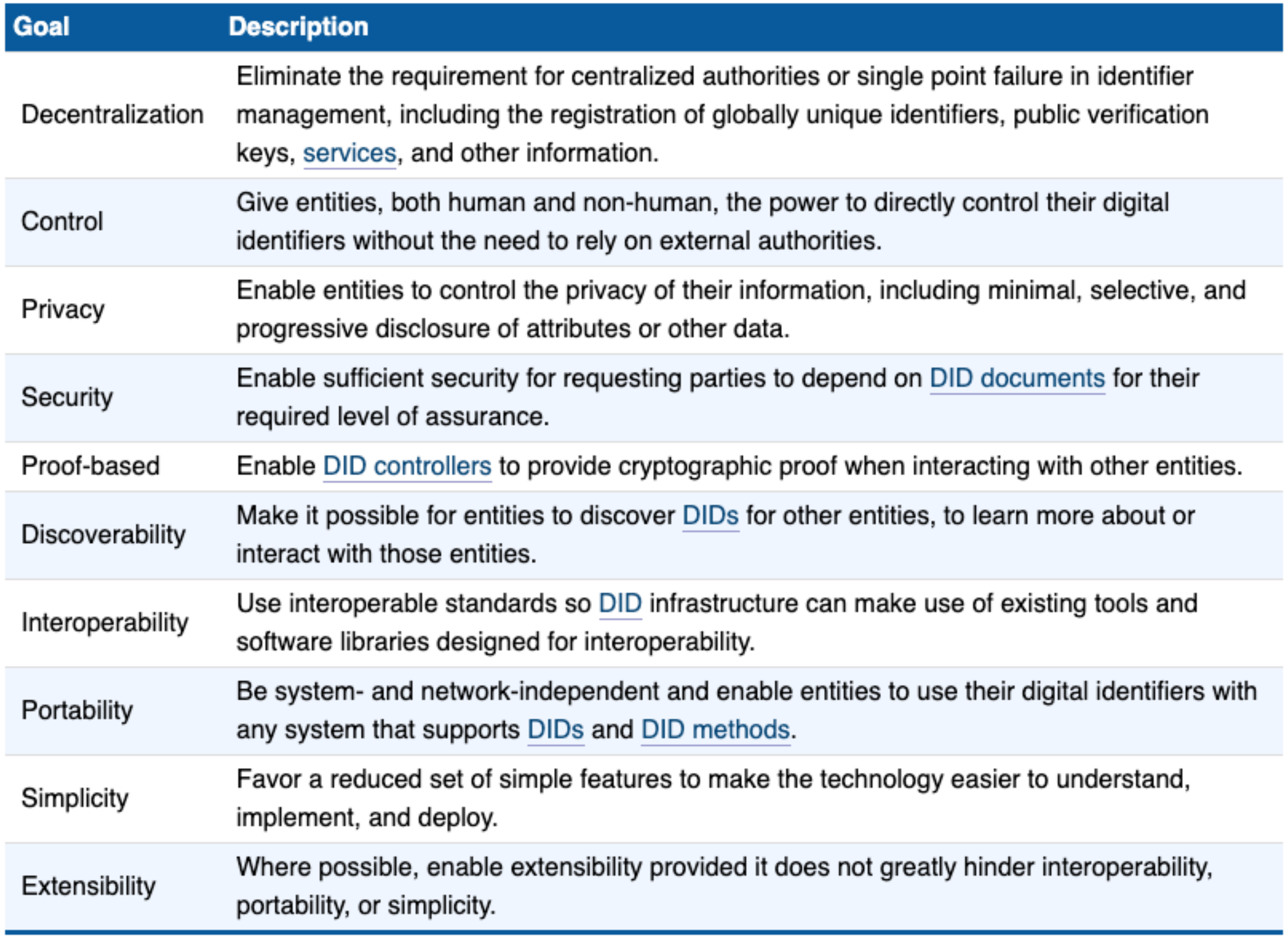
Supports recording DIDs and returning data necessary to produce DID Documents

- Distributed ledger
- Decentralized file system
- Database
- Peer-to-peer network
- Trusted data storage

[1]Source: https://www.w3.org/TR/did-core/

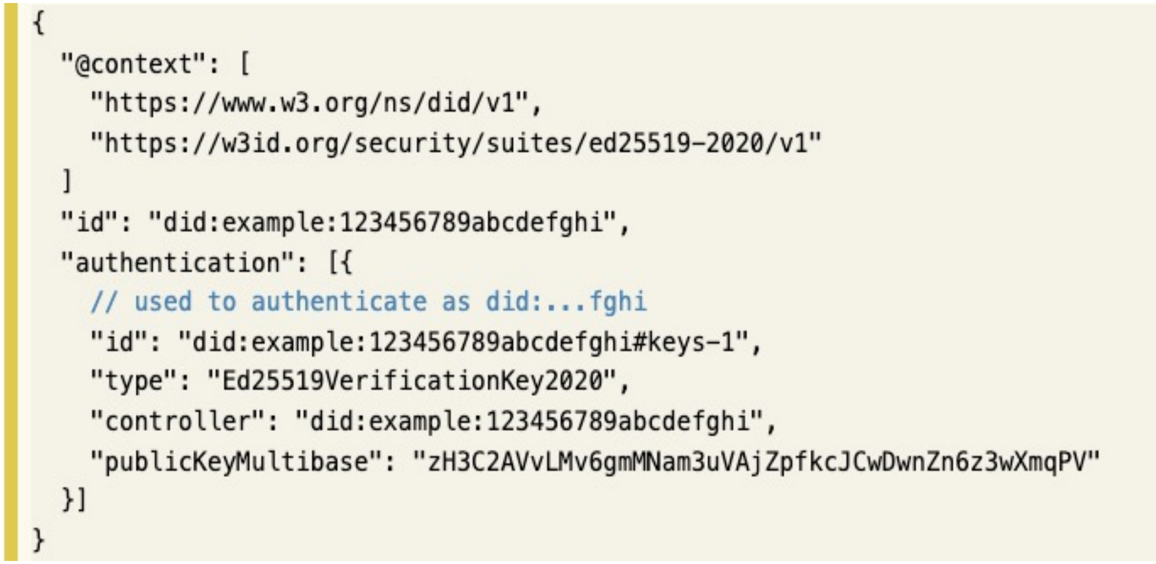
Decentralized Identifier (DID) Design Goal

DID Design Goals



Goal	Description
Decentralization	Eliminate the requirement for centralized authorities or single point failure in identifier management, including the registration of globally unique identifiers, public verification keys, services, and other information.
Control	Give entities, both human and non-human, the power to directly control their digital identifiers without the need to rely on external authorities.
Privacy	Enable entities to control the privacy of their information, including minimal, selective, and progressive disclosure of attributes or other data.
Security	Enable sufficient security for requesting parties to depend on DID documents for their required level of assurance.
Proof-based	Enable DID controllers to provide cryptographic proof when interacting with other entities.
Discoverability	Make it possible for entities to discover DIDs for other entities, to learn more about or interact with those entities.
Interoperability	Use interoperable standards so DID infrastructure can make use of existing tools and software libraries designed for interoperability.
Portability	Be system- and network-independent and enable entities to use their digital identifiers with any system that supports DIDs and DID methods.
Simplicity	Favor a reduced set of simple features to make the technology easier to understand, implement, and deploy.
Extensibility	Where possible, enable extensibility provided it does not greatly hinder interoperability, portability, or simplicity.

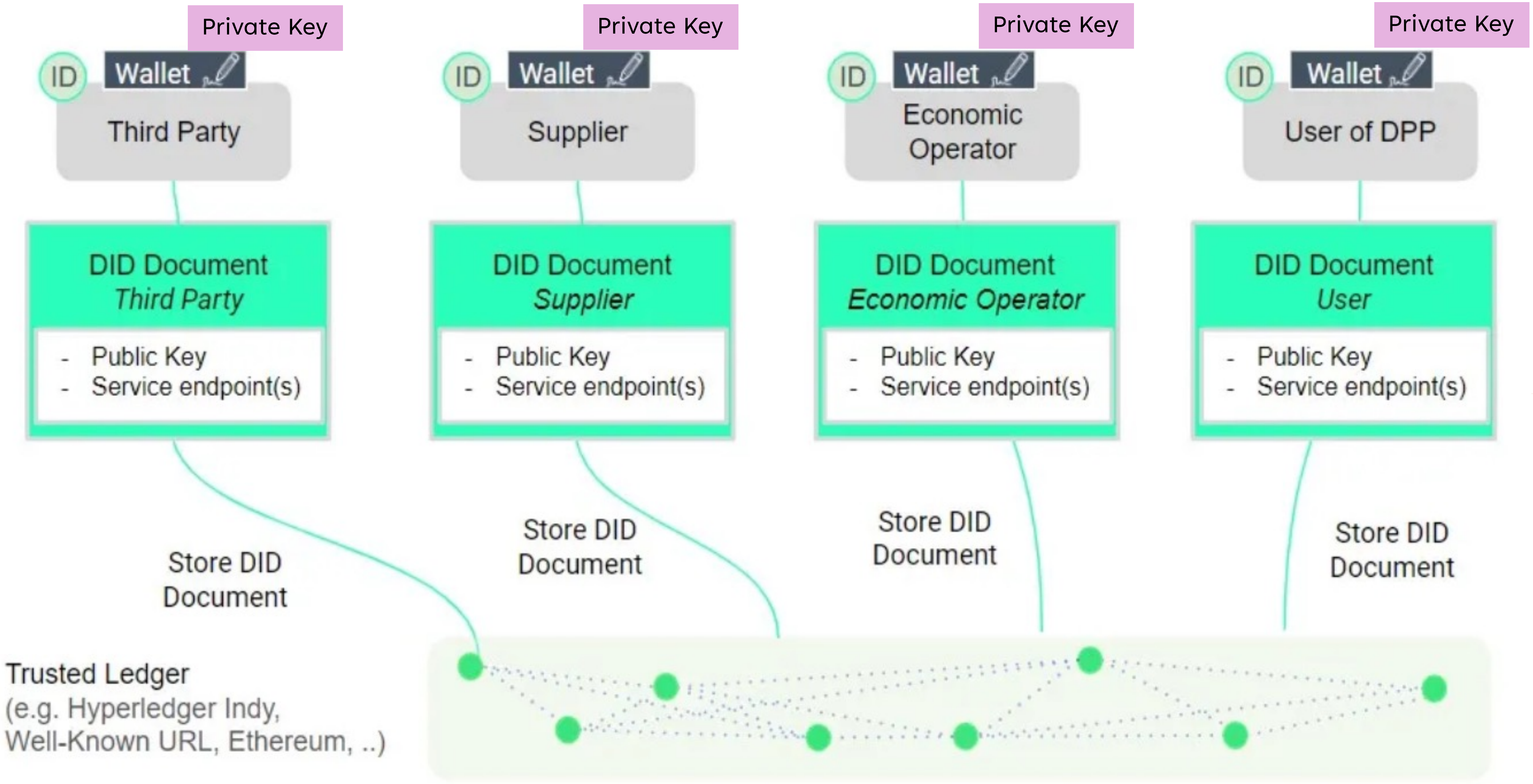
A simple DID Document



```
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/suites/ed25519-2020/v1"
  ]
  "id": "did:example:123456789abcdefghi",
  "authentication": [{
    // used to authenticate as did:...fghi
    "id": "did:example:123456789abcdefghi#keys-1",
    "type": "Ed25519VerificationKey2020",
    "controller": "did:example:123456789abcdefghi",
    "publicKeyMultibase": "zH3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV"
  }]
}
```

[1] Source: https://www.w3.org/TR/did-core/

Example: EU Digital Product Passport (DPP)



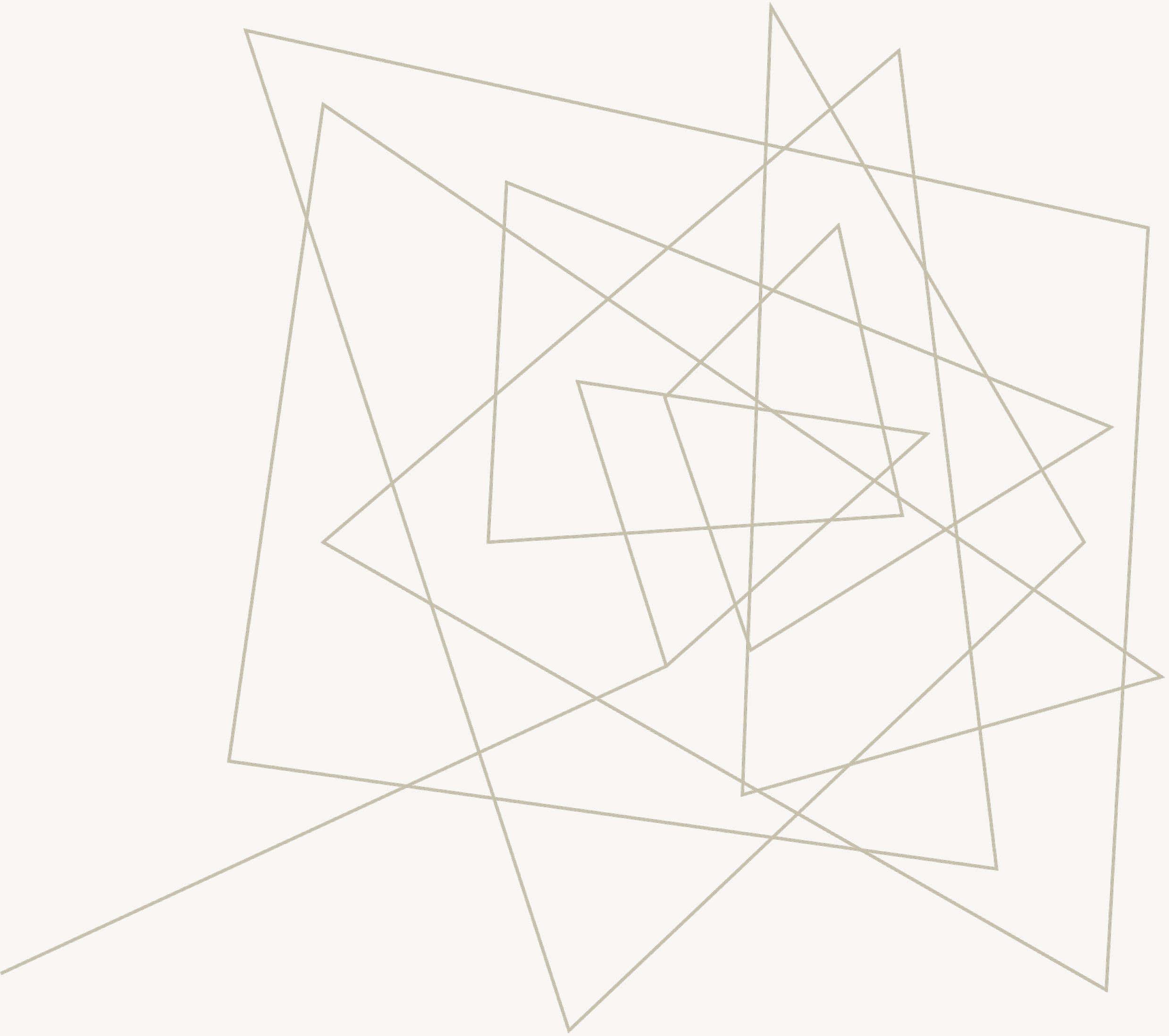
Source: https://medium.com/@susi.guth/implementing-digital-product-passports-using-decentralized-identity-standards-f1102c452020

Decentralized Identifiers (DIDs): Private Key

The private key associated with a DID is used to prove ownership and control of the DID

The storage location of the private key depends on:

- Security and privacy requirements
- Specific DID method being used



What are Verifiable Credentials?

Verifiable Credentials (VCs)

Verifiable Credentials provide a mechanism to express credentials on the Web

- Cryptographically secure
- Privacy respecting
- Machine-verifiable

República de Panamá
AUTORIDAD DEL TRANSITO
Y TRANSPORTE TERRESTRE
LICENCIA DE CONDUCIR
C
TIPO
8-345-321
María del Carmen
Cedeño Sánchez
NACIONALIDAD
PANAMEÑA
EXPEDICIÓN
05 ENERO 2006
FECHA DE NACIMIENTO
15 FEBRERO 1986
RESTRICCIONES
NINGUNA
TIPO DE SANGRE
ORH-
EXPIRACIÓN
ENERO 2010

SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

DIPLOME
MATHEMATIQUES

Vu le procès verbal de l'examen établi le 6 janvier 2011
par le président du jury, examinateur, ayant autorité sur les diplomes en ligne
Le Diplome Mathematiques
du site www.mon-diplome.fr
est conféré à KPADONOU Noudéhouénou François

pour en jouir avec les droits et prérogatives qui y sont attachés

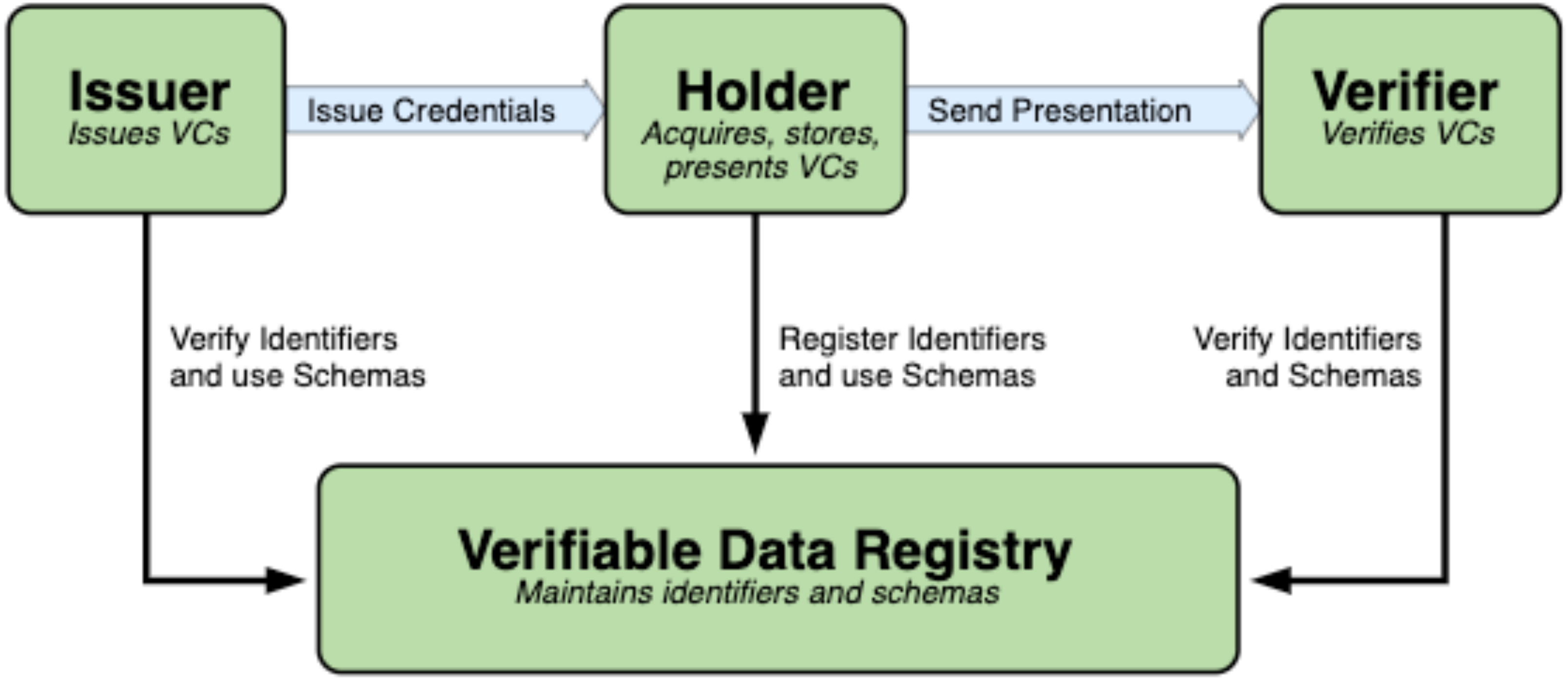
Signature du titulaire
Ministère de l'enseignement supérieur et de la recherche scientifique
www.mon-diplome.fr
N° 0000000017645

Source: https://www.w3.org/TR/vc-data-model/
Graphics: https://en.wikipedia.org/wiki/Australian_passport, https://ispwelcomesyou.wordpress.com/2016/04/12/how-to-obtain-panamanian-drivers-license/, https://www.ruerude.com/2011/10/st%C3%A9phane-a-guy-i-know-is-studying-to-be-a-translator-i-asked-him-why-and-he-said-i-was-bored-with-teaching-math-in-hig.html, https://usacustomsclearance.com/process/what-is-a-shipping-manifest/, https://www.biogro.co.nz/organic-faqs

Verifiable Credentials (VCs) and Decentralized Identifiers (DIDs)

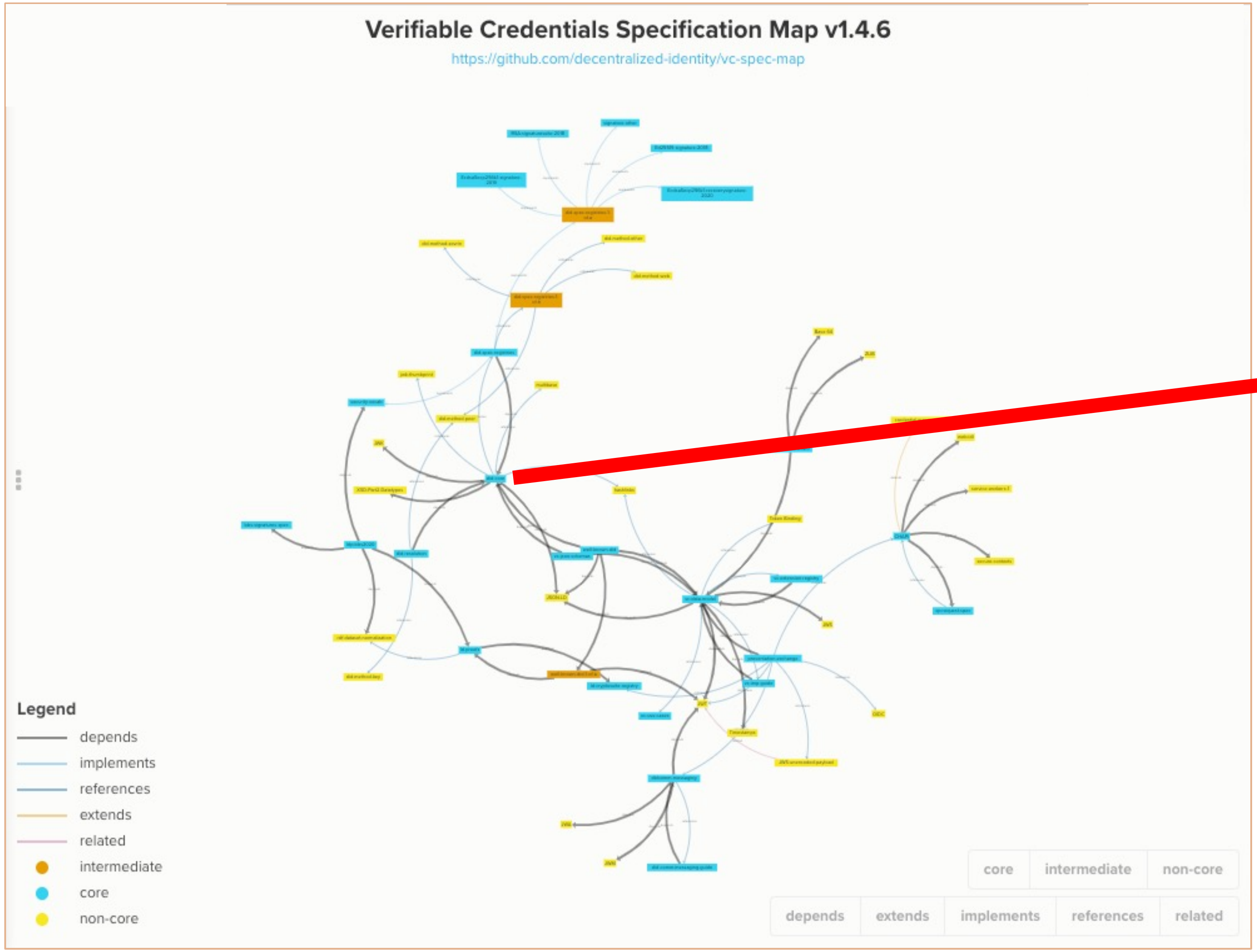
Decentralized Identifiers (DIDs)

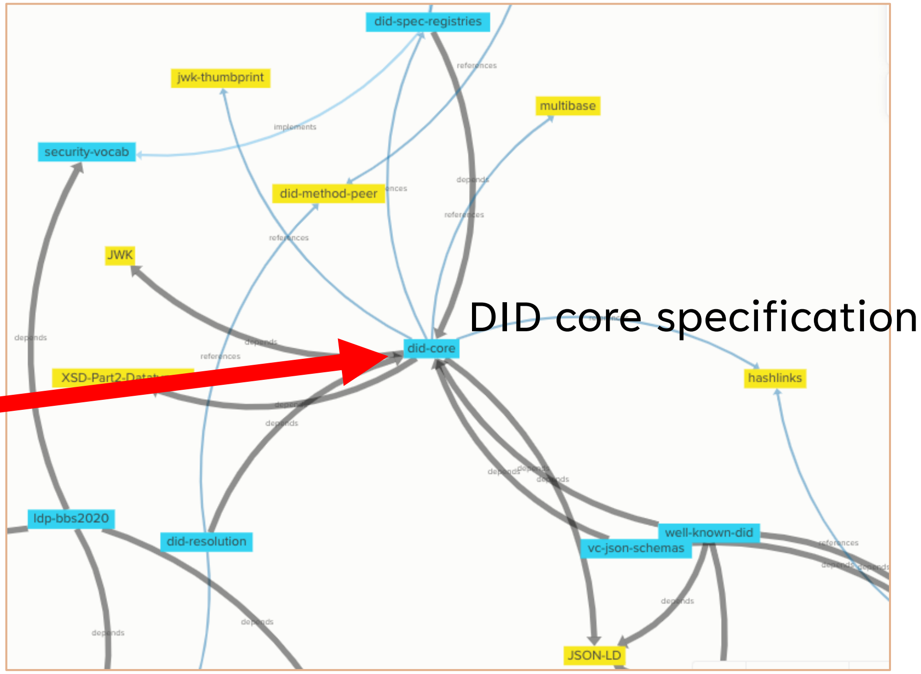
- Often used in a Verifiable Credential (VC)
- A VC can easily be ported from one repository to another without the need to reissue the credential


Issuer
Issues VCs
Issue Credentials
Holder
Acquires, stores, presents VCs
Send Presentation
Verifier
Verifies VCs
Verify Identifiers and use Schemas
Register Identifiers and use Schemas
Verify Identifiers and use Schemas
Verifiable Data Registry
Maintains identifiers and schemas

Source: https://www.w3.org/TR/vc-data-model/

VC and DID Specifications Map





DID core specification

- Pre-Requisite Knowledge: JSON, JSON-LD, JWT, JWS, JWK, JWA, and sometimes CBOR.
- Decentralized Identifiers: DID-Core, DID-Resolution, DID-Spec, DID Use-Cases.
- Verifiable Credentials: VC-Data Model, VC Use-Cases, and VC-Implementors Guide
- Transport: HTMl, DID-Comm
- Credential Presentation: Presentation Exchange, Credential Manifest
- Optional: Well-known-did
- Other Data Formats: Open Badges
  - Independent DID Methods: DID-method-key, DID-method-peer, DID-method-web
- Categorizing Verifiable Credentials – Evernym Not all verifiable credentials are created the same. This post examines the categories of credentials and the architectural choices driving this variation.

Source: https://www.kumu.io/michaelruminer/vc-spec-map

What is a Verification Method?

Definition of Verification Method

Verification Method
A set of parameters that can be used together with a process to independently verify a proof

- Cryptographic public key can be used as a verification method with respect to a digital signature
- Verifies the signer possessed the associated cryptographic private key

Each DID Method specification is expected to detail how revocation is performed and tracked

Source: https://www.w3.org/TR/did-core/

What is a Trustless System?

Trustless System

Section 9.8 Verification Method Revocation

Revocation in Trustless Systems

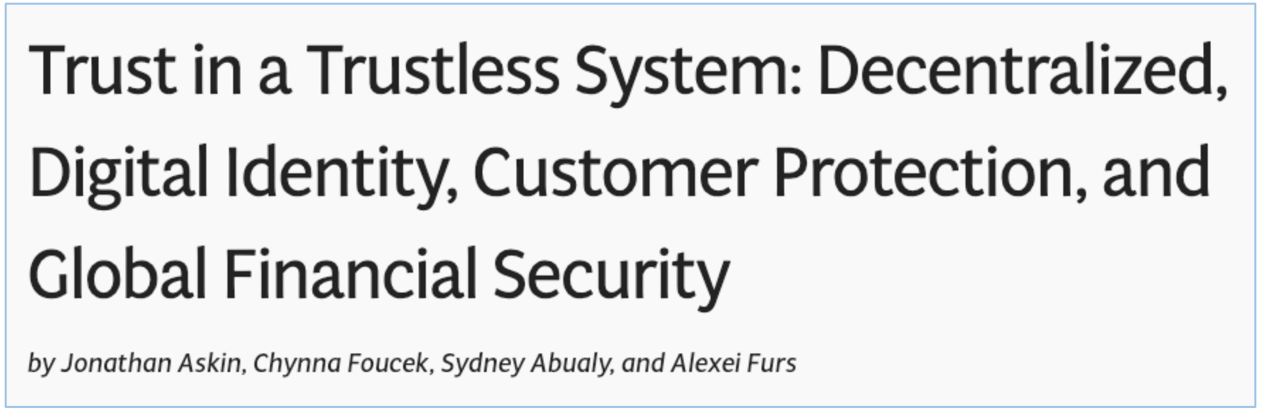
Trustless systems: all trust is derived from cryptographically provable assertions

- No metadata outside of the cryptographic system is factored into the determination of trust in the system

Source: https://www.w3.org/TR/did-core/

Trust in a Trustless System

Financial Services Security



Trust in a Trustless System: Decentralized, Digital Identity, Customer Protection, and Global Financial Security

by Jonathan Askin, Chynna Foucek, Sydney Abualy, and Alexei Furs

…decentralized identity uses a distributed ledger to provide a robust public key infrastructure and allow users to prove their identity using digital signatures without a centralized authority

[1]Source: https://law.mit.edu/pub/trustinatrustlesssystem/release/1

Excerpts from the Decentralized Identifier (DID) v1.0 W3C Recommendation

Section 9.8

Decentralized Identifiers (DIDs) v1.0

Core architecture, data model, and representations
W3C Recommendation, 19 July 2022

Section 9.8 Verification Method Revocation

Deactivate verification method [key]

- Ceases to be a valid form of creating new proofs of digital signatures

Useful mechanism for reacting to a verification method compromise

Perform revocation immediately after rotation

- For verification methods for short-lived verifications
- Encrypting messages and authentication*

*Authentication in this context: verifying the owner of a DID, or entry associated with a DID

Source: https://www.w3.org/TR/did-core/
Graphic: https://reciprocity.com/blog/analyzing-cybercriminal-reconnaissance-to-improve-your-strategic-planning/

Verification Method Revocation

Section 9.8 Verification Method Revocation

Verification Method [Key] compromise may allow attackers to use them

- Might be indistinguishable from the legitimate use
- Vulnerable from time key was registered, to time it was revoked

Source: https://www.w3.org/TR/did-core/
Graphic: https://reciprocity.com/blog/analyzing-cybercriminal-reconnaissance-to-improve-your-strategic-planning/

Verification Method Revocation Considerations (1 of 2)

Section 9.8 Verification Method Revocation
10 Considerations

1. Verification method revocation is a reactive security measure.
2. It is considered a best practice to support key revocation.
3. A controller is expected to immediately revoke any verification method that is known to be compromised.
4. Verification method revocation can only be embodied in changes to the latest version of a DID document; it cannot retroactively adjust previous versions.
5. Absence of a verification method is the only form of revocation that applies to all DID methods that support revocation.
6. If a verification method is no longer exclusively accessible to the controller or parties trusted to act on behalf of the controller, revoke immediately
   - Reduce risk of compromises (masquerading, theft, fraud)

Source: https://www.w3.org/TR/did-core/

Verification Method Revocation Considerations (2 of 2)

Section 9.8 Verification Method Revocation Considerations

7. Revocation: proofs or signatures associated with a revoked verification method should be treated as invalid.
    - Might have been created by an attacker
    - Verifiers may choose to accept or reject proofs or signatures at their own discretion
8. DID operations include **update** and **deactivate**, which might be used to remove a verification method from a **DID document**.
9. Not all DID methods support verification method revocation.
10. Even if a verification method is present in a DID document, additional information, such as a public key revocation certificate, or an external allow or deny list, could be used to determine whether a verification method has been revoked.

Source: https://www.w3.org/TR/did-core/

Verification Method Revocation Semantics (1 of 2)

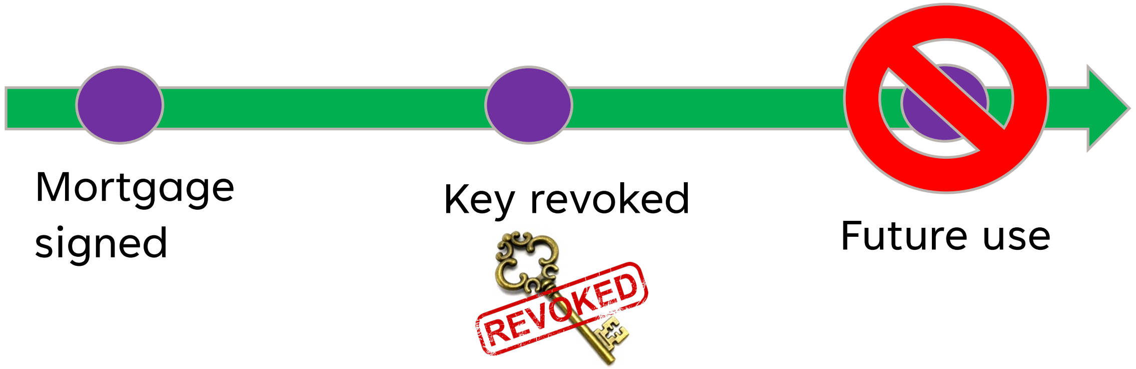
Section 9.8 Verification Method Revocation Semantics

Verifiers might choose not to accept proofs or signatures from a revoked verification method

- ***Knowing whether a verification was made with a revoked verification method is trickier than it might seem***

Some DID methods provide the ability to look back at the state of a DID at a point in time

- DIDs can be used to make binding commitments
- Revocation is not retroactive
- Only nullifies future use of the method.



Source: https://www.w3.org/TR/did-core/
Graphics: https://www.arcolinuxiso.com/create-a-key-to-sign-your-packages/, https://blog.nucypher.com/what-exactly-do-we-mean-by-access-revocation/

Verification Method Revocation Semantics (2 of 2)

Section 9.8 Verification Method Revocation Semantics

Important to know state of the DID document at the time assertion was made

- Someone could discover a revoked key and use it to make cryptographically verifiable statements with a simulated date in the past

Finds revoked key, simulates date from past

Source: https://www.w3.org/TR/did-core/

Verification Method Revocation in Trustless Systems

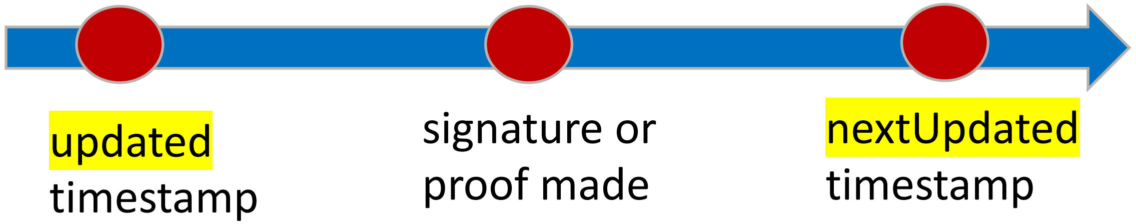
Section 9.8 Verification Method Revocation in Trustless Systems

To verify a signature of proof for a verification method which has been revoked in a trustless system, the DID method needs to support **DID document metadata**:

- Either or both of the versionId or versionTime
- Both updated and nextUpdate

A verifier can validate a signature or proof of a revoked key if and only if all of the following are true:

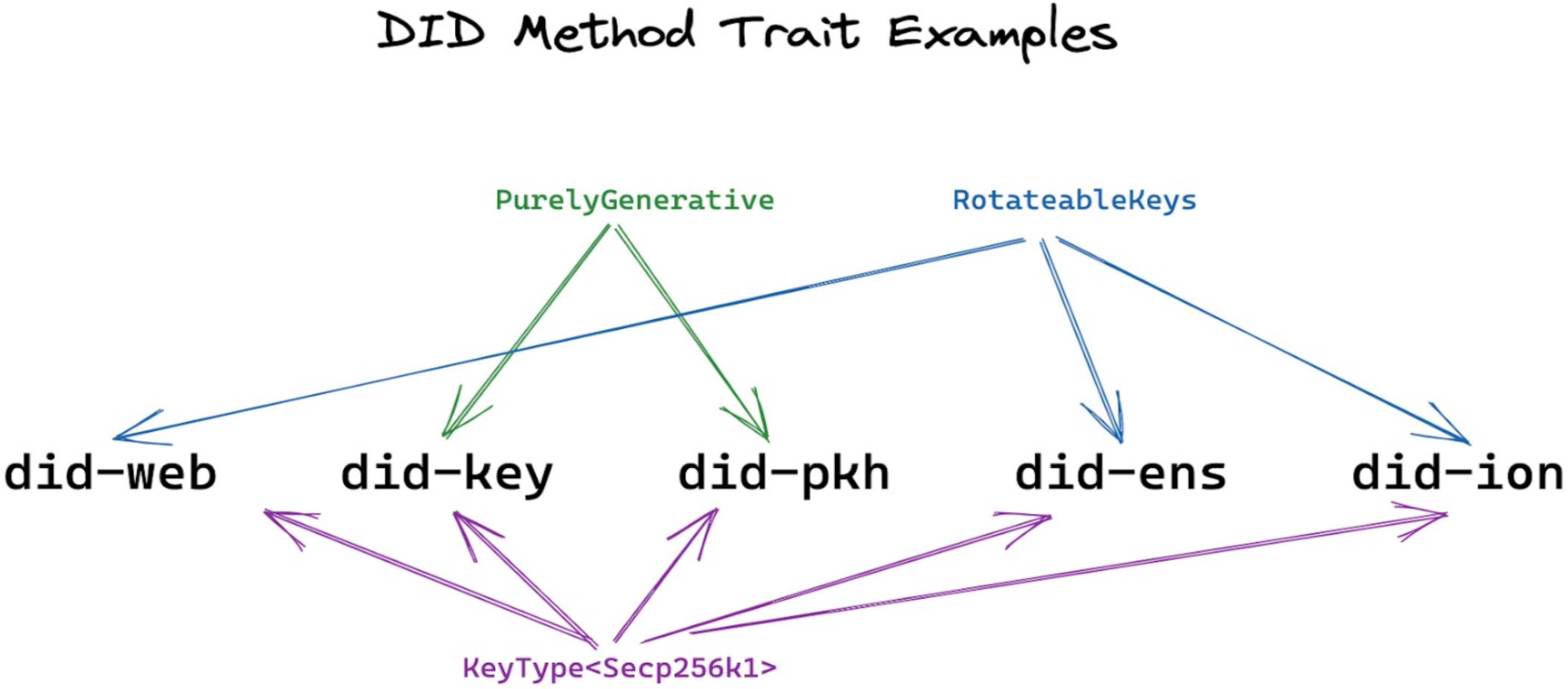
- The proof or signature includes the versionId or versionTime of the DID document that was used at the point the signature or proof was made
- The verifier can determine the point in time the signature or proof was made (e.g., anchored on a blockchain)
- The updated timestamp is before, and the nextUpdate timestamp is after, the signature or proof was made



Source: https://www.w3.org/TR/did-core/
Figure: not from the DID core specification, created by Clare Nelson

Verification Method Revocation Varies by DID Method

Do All DID Methods Support Verification Method Revocation?



Source: https://blog.spruceid.com/upgradeable-decentralized-identity/

Key Revocation Support by DID Method

Many DID methods are drafts

DID Method	Revocation Supported?	Description
did:key	N/A	Non-registry based DID Method, expands a cryptographic public key into a DID Document.
did:jwk	N/A	Deterministic transformation of a JWK into a DID Document
did:erc725	Yes	Revoking the DID can be supported by executing a selfdestruct() operation that is part of the smart contract. This will remove the smart contract's storage and code from the Ethereum state, effectively marking the DID as revoked.
did:indy	Yes	DID controller creates new version of DID document with revoked property set to true for public key being revoked. The new version of the DID document must then be signed and stored on the ledger. The revoked key is considered invalid and should not be used for any further transactions.
did:ion	Yes	DID controller could publish a new DID Document on a peer-to-peer network that revokes a specific public key, or use an external protocol to communicate the key revocation event to relevant parties.
did:keri	Yes	Use Key Event Receipt (KER), DID controller generates a KER that indicates the revocation of the key. The KER is then propagated through the distributed ledger system to all relevant parties, informing them that the key has been revoked.
did:lac	Yes	Set DID controller to 0x0. Although, 0x0 is a valid Ethereum address, this will indicate the identity has no controller, and is invalid.
did:peer	Yes	DID controller could publish a new DID Document on a peer-to-peer network that revokes a specific public key, or use an external protocol to communicate the key revocation event to relevant parties.
did:sov	Yes	Set verification key to null, permanently terminates the identity's ability to operate on the network because there is no key that the identity can use to authenticate itself--even to submit a new key rotation request. It is irreversible.
did:web	Yes	To delete the DID document, the did.json has to be removed or has to be no longer publicly available due to any other means.

Source: https://www.w3.org/TR/did-spec-registries/, follow links to DID Method description

did:sov Supports Key Revocation

Deleting or revoking a verification key is not to be confused with temporary suspension or rotation

Deletion sets an identity's verification key to null

- Permanently terminates the identity's ability to operate on the network
- There is no key that the identity can use to authenticate itself--even to submit a new key rotation request
- It is irreversible

Revocation may be appropriate when a person dies or a business is legally dissolved

- It does not remove any record or history of the identity
- Prevents any new history from accruing

Guarantees that no malicious actor can recover and reactivate an identity that is dead

- Prevents an entity from reusing the old DID that has been terminated

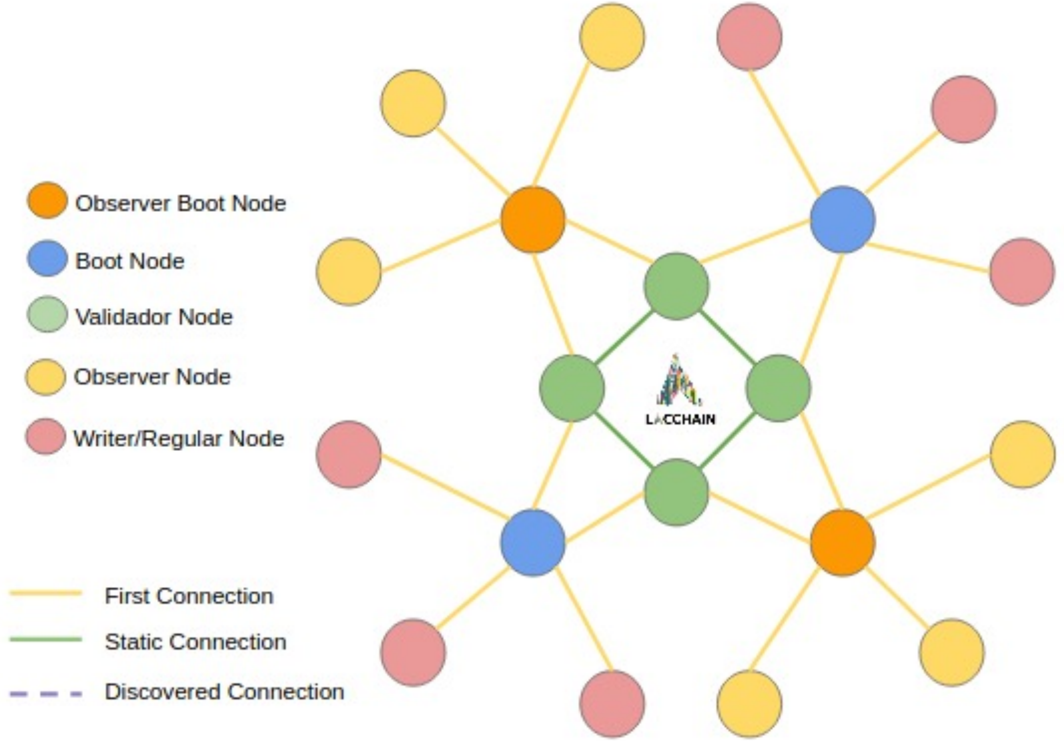
Source: https://sovrin-foundation.github.io/sovrin/spec/did-method-spec-template.html

did:lac supports Automatic Key Rotation, Revocation

did:lac

LACChain, fully on-chain DID method based on the ERC-1056 standard

- Compliant with latest DID model v1.0 specs by W3C
- Listed by DIF
- Fully on-chain DID method
- Based on the ERC-1056, originally implemented in “ethr” method by uPort
- Support for multiple controllers
- Automatic Key Rotation
- On-chain Key Recovery
- Public Keys -> Verification Methods
- Introduce a new concept: Verification Relationships
- Added blockchainAccountId as a special type of Public Key
- Definition of controller for each Verification Method
- Included new relationship: Invocation Capability



LACChain Topology

ERC-1056 = Ethereum Lightweight Identity

Source: https://lacnet.lacchain.net/decentralized-identifiers/

Future Keys

DID Method, Working on Post-Quantum

did:dyne

Public Keys for:

- Secp256k1 ECDSA, widely used for single signatures
- ED25519 EDDSA widely used for single signatures
- BLS381 [“Reflow”](https://dyne.org/W3C-DID/) [REFLOW], for multisignature and advanced zero-knowledge proof operations
- Dilithium2, for post-quantum signatures
- Ethereum public addresses (“blockchainAccountId”), following the eip155 standard

Source: https://dyne.org/W3C-DID/
Figure: https://physicsworld.com/a/what-you-need-to-know-before-investing-in-quantum-technology/

Conclusion

Conclusion

Decentralized Identity Systems

- Do not rely on centralized authorities or intermediaries

Trustless systems

- All trust is derived from cryptographically provable assertions

Verification Method Revocation in Trustless Systems for DID Methods

- Directly impacts the security and trustworthiness of digital identities

Before you select a DID Method, investigate how key rotation, key revocation are implemented

- Not all DID Methods 'need' verification method revocation
- Not all DID Methods support verification method revocation
- ***Knowing whether a verification was made with a revoked verification method is trickier than it might seem[1]***
- How to perform rotation, revocation is not defined in the W3C DID Recommendation, it's an implementation consideration left to the developer
- Each DID Method specification is expected to detail how revocation is performed and tracked

[1] Source: https://www.w3.org/TR/did-core/

Graphic: https://www.trustlesscomputing.org/

Thank You

clare@identity.foundation

References

- Allen, Christopher; Brock, Arthur; Buter, Vitalik; Callas, Jon; Dorje, Duke; Lundkvist, Christian; Kravchenko, Pavel; Nelson, Jude; Reed, Drummond; Sabadello, Markus; Slepak, Greg; Thorp, Noahy; Wood, Harlan T. *Decentralized Public Key Infrastructure* (December 2015), https://github.com/WebOfTrustInfo/rwot1-sf/blob/master/final-documents/dpki.pdf
- Askin, Jonathan; Foucek, Chynna; Abualy, Sydney; Furs, Alexei. *Trust in a Trustless System: Decentralized, Digital Identity, Customer Protection, and Global Financial Security* (January 2022), https://law.mit.edu/pub/trustinatrustlesssystem/release/1
- Baya, Vinod. *Digital Identity, Moving to a Decentralized Future*, (October 2019), https://www.citi.com/ventures/perspectives/opinion/digital-identity.html
- Bicacki, Kemal; Crispo, Bruno; Tanenbaum, Andrew. *How to Incorporate Revocation Status Information into the Trust Metrics for Public-Key Certification* (March 2005), https://dl.acm.org/doi/pdf/10.1145/1066677.1067037 (Requires ACM Membership)
- Boneh, Fan. *DeFi Lecture 11: Decentralized Identity* (November 2021), https://www.youtube.com/watch?v=3FL-1HMKvYA
- Chang, Wayne. *Upgradeable Decentralized Identity – DID Method Traits* (July 2022), https://blog.spruceid.com/upgradeable-decentralized-identity/
- Chang, Wayne. *Sign-in with Ethereum* (January 2023), https://www.youtube.com/watch?v=VHwzE6mVm_s
- Collins, Benjamin. *Beyond Blockchain: How Decentralized Identifiers Work* (February 2023), https://medium.com/transmute-techtalk/beyond-blockchain-how-decentralized-identifiers-dids-work-20bb199d038
- Cooper, David A., Computer Security Division, NIST, *A Closer Look at Revocation and Key Compromise in Public Key Infrastructures, https://csrc.nist.gov/csrc/media/publications/conference-paper/1998/10/08/proceedings-of-the-21st-nissc-1998/documents/paperg2.pdf*
- DIDComm V2 Guidebook (2022), https://didcomm.org/book/v2/didrotation
- DID the Decentralized Identifier, https://decentralized-id.com/web-standards/w3c/wg/did/decentralized-identifier/
- Fdhila, Walid; Stifter, Nicholas; Kostal, Kristian; Saglam, Cihan; Sabadello, Markus. *Methods for Decentralized Identities: Evaluation and Insights*, http://eprints.cs.univie.ac.at/7094/1/2021-1087.pdf

References

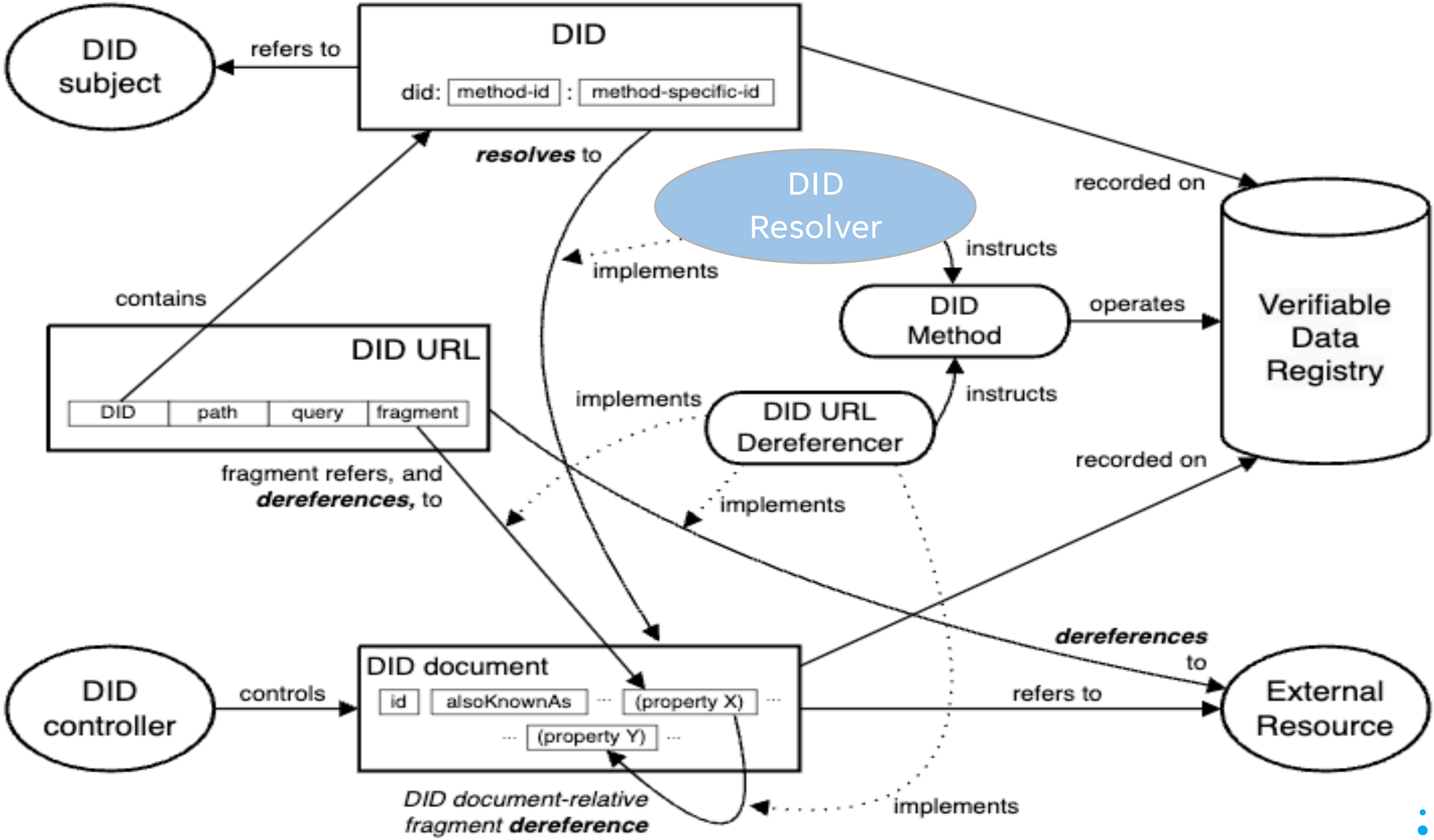
- Fernandes, Bruno Miguel Gomes. Self-Sovereign Identity Decentralized Identifiers, Claims and Credentials using non Decentralized Ledger Technology (November 2021), https://repositorium.sdum.uminho.pt/bitstream/1822/82791/1/Bruno%20Miguel%20Gomes%20Fernandes.pdf?utm_source=substack&utm_medium=email
- Genise, Nick; Balenson T., David. *Cryptography Review of W3C Verifiable Credentials Data Model (VCDM) and Decentralized Identifiers (DIDs) Standards and Cryptography Implementation Recommendations* (October 2021), http://www.csl.sri.com/papers/vcdm-did-crypto-recs/crypto-review-and-recs-for-VCDM-and-DIDs-implems-FINAL-20211015.pdf
- Guth-Orlowski, Susanne; Ebert, Johannes; Thiermann, Ricky. *Implementing Digital Product Passports using decentralized identity standards* (April 2023), https://medium.com/@susi.guth/implementing-digital-product-passports-using-decentralized-identity-standards-f1102c452020
- Jacques, Samuel; Lodder, Michael; Montgomery, Hart. *ALLOSAUR: Accumulator with Low-Latency Oblivious Sublinear Anonymous credential Updates with Revocations* (October 2022), https://eprint.iacr.org/2022/1362.pdf
- Brunner, Clemens; Gallersdörfer, Ulrich; Knirsch, Fabian; Engel, Dominik; Matthes, Florian. *DID and VC: Untangling Decentralized Identifiers and Verifiable Credentials for the Web of Trust* (December 2020), https://dl.acm.org/doi/fullHtml/10.1145/3446983.3446992
- MATTR, *Create a Static Key DID*, https://learn.mattr.global/tutorials/dids/did-key
- Park Chang-Seop; Nam, Hye-Min. *A New Approach to Constructing Decentralized Identifier for Secure and Flexible Key Rotation* (October 2021), https://ieeexplore.ieee.org/abstract/document/9583584
- Pope, Nick; Tabor, Michał; Barreira, Iñigo; Nicholas Dunha; Granc, Franziska; Thiell, Christoph; Fiedler, Arno (ENISA). *Digital Identity: Leveraging the SSI Concept to Build Trust* (January 2022), https://www.enisa.europa.eu/publications/digital-identity-leveraging-the-ssi-concept-to-build-trust
- Thorstensson, Joel. *Key Revocation in Self-Certifying Protocols* (April 2022), https://blog.ceramic.network/key-revocation-in-self-certifying-protocols/#:~:text=Key%20revocation%20%2D%20a%20function%20in,the%20new%20key%20becomes%20active

References

- Rocco, Gregory. *Decentralized Identity and Web3* (August 2022), https://blog.spruceid.com/decentralized-identity-and-web3/
- Sabadello, Markus. *The Power of DIDs #2: Creating DIDs* (April 2023), https://www.linkedin.com/posts/danube-tech_the-power-of-dids-2-creating-dids-activity-7051844692723789824-2UjD?utm_source=share&utm_medium=member_desktop
- Smith, Samuel. *Key Event Receipt Infrastructure (KERI): A secure identifier overlay for the inter*net (May 2020), https://www.youtube.com/watch?v=izNZ20XSXR0
- Sporny, Manu. *Verifiable Credentials and DIDs* (September 2022), https://www.youtube.com/watch?v=Nk8Ey0MC528
- W3C Recommendation, *Decentralized Identifiers (DIDs) v1.0, Core architecture, data model, and representations* (July 2022), https://www.w3.org/TR/did-core/
- Weston, Georgia. *Self Sovereign Identity & Decentralized Identity – An Unlimited Guide* (July 2022), https://101blockchains.com/self-sovereign-identity-and-decentralized-identity/
- Windley, Philip. *Digital Identity Design, Deploy, and Manage Identity Architectures* (February 2023), O'Reilly Media

DID Architecture

DID architecture and relationship of the basic components



Source: https://www.w3.org/TR/did-core/