Decentralized Identity + Interoperability Workshop:

Connecting Credo (Aries Framework JavaScript) with Hyperledger Besu, Cardano, Cheqd, Hyperledger AnonCreds and OID4VC

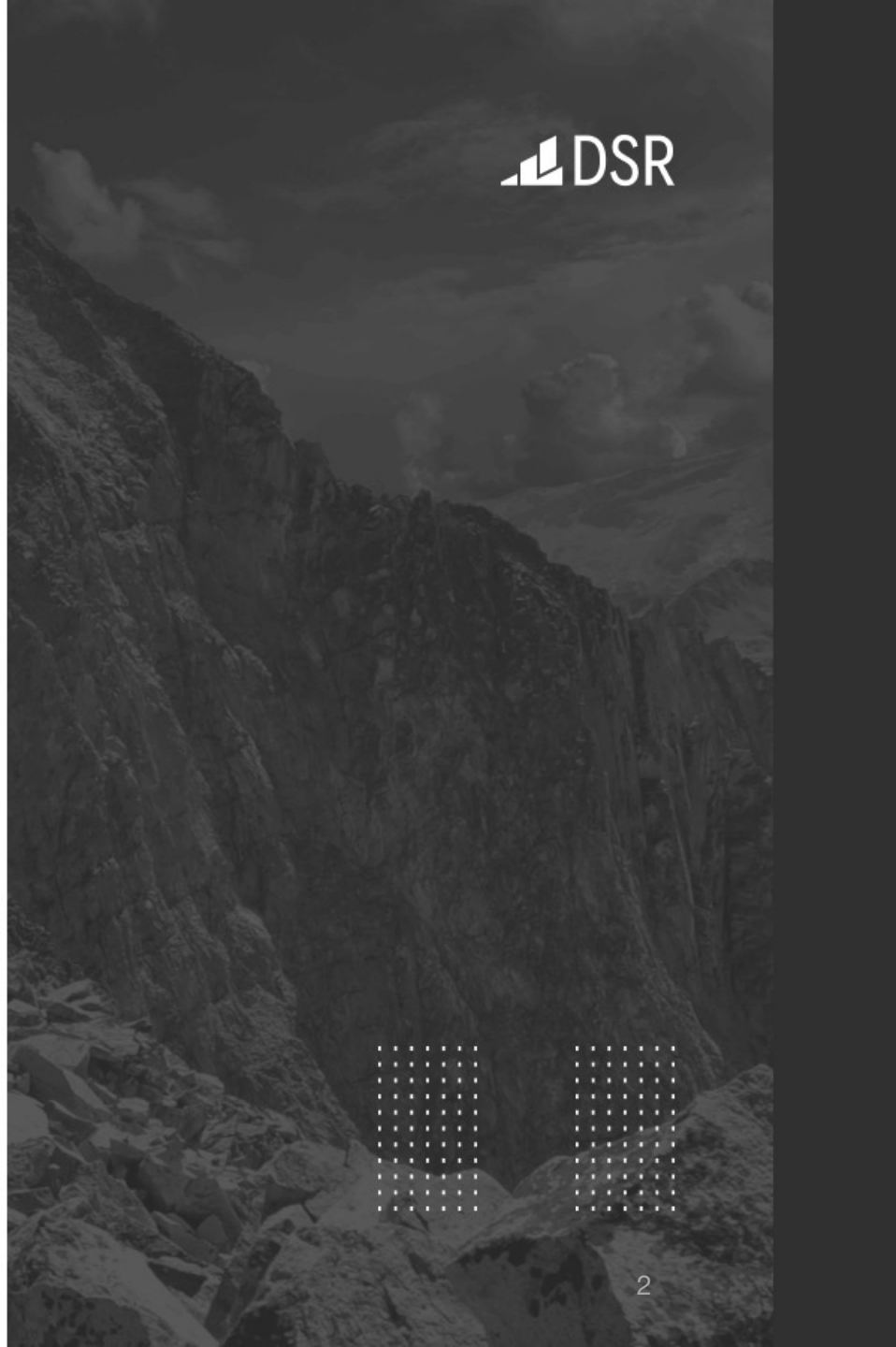
Alexander Shcherbakov
Renata Toktar
Artem Ivanov

DSR Corporation

Goals

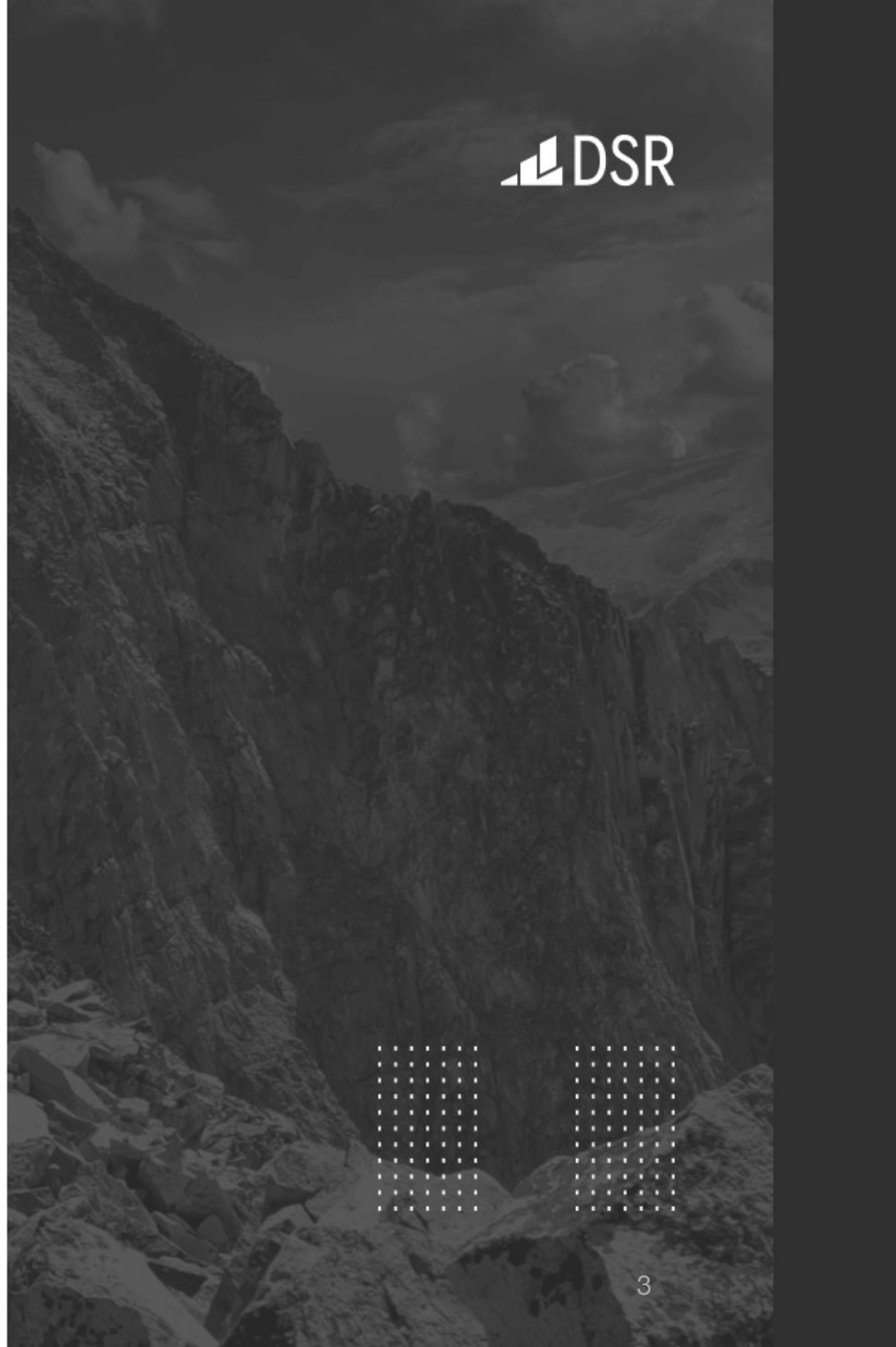
- SSI is not a framework/tool/library
- SSI is a concept/model for digital identity
- There are multiple specifications/protocols in SSI
- There are multiple frameworks/tools/libraries implementing SSI principles and protocols

- Goal 1: Summarize main approaches/specifications/profiles in SSI
- Goal 2: Show interoperability between some approaches, tools, frameworks



Agenda

1. About Self-sovereign Identity (SSI)
2. Interoperability Variables (Profiles)
   - VC Formats
   - VC Exchange Protocols
   - DID method / Verifiable Data Registry (VDR)
3. Interoperability Variables Values for the Demo Part
   - Credo (Aries Framework JavaScript)
   - VC Formats: Hyperledger AnonCreds, W3C VC
   - VC Exchange Protocols: Hyperledger Aries, OID4VC
   - VDR: Cardano, Cheqd, Hyperledger Besu
4. Demo
5. Hands-On



About the Speakers

Alexander Shcherbakov

Head of Decentralized Systems Department
DSR Corporation

- Ph.D. degree in Mathematics
- 13+ years of experience in software engineering and team management
- 7+ years of experience in Digital ID, Self-Sovereign Identity (SSI), Blockchain, Distributed Ledger Technology (DLT), Consensus protocols, and cryptography.
- One of the core maintainers and contributors of such projects as Hyperledger Indy, Hyperledger Aries, Hyperledger Ursa, etc.
- International conference speaker: Hyperledger Global Forum, Hyperledger Webinars/Workshops, IIW, CSA member meeting, etc.

About the Speakers

Renata Toktar

Lead Software Engineer
DSR Corporation

- Master degree in Mathematics
- 9+ years of experience in software engineering
- 6 years of experience in Digital ID, Self-Sovereign Identity (SSI), Blockchain, Distributed Ledger Technology (DLT), Consensus protocols, and cryptography.
- One of the core maintainers and contributors of such projects as Hyperledger Indy and Cheqd
- Used to lead blockchain startup
- International conference speaker on Hyperledger Global Forum

About DSR

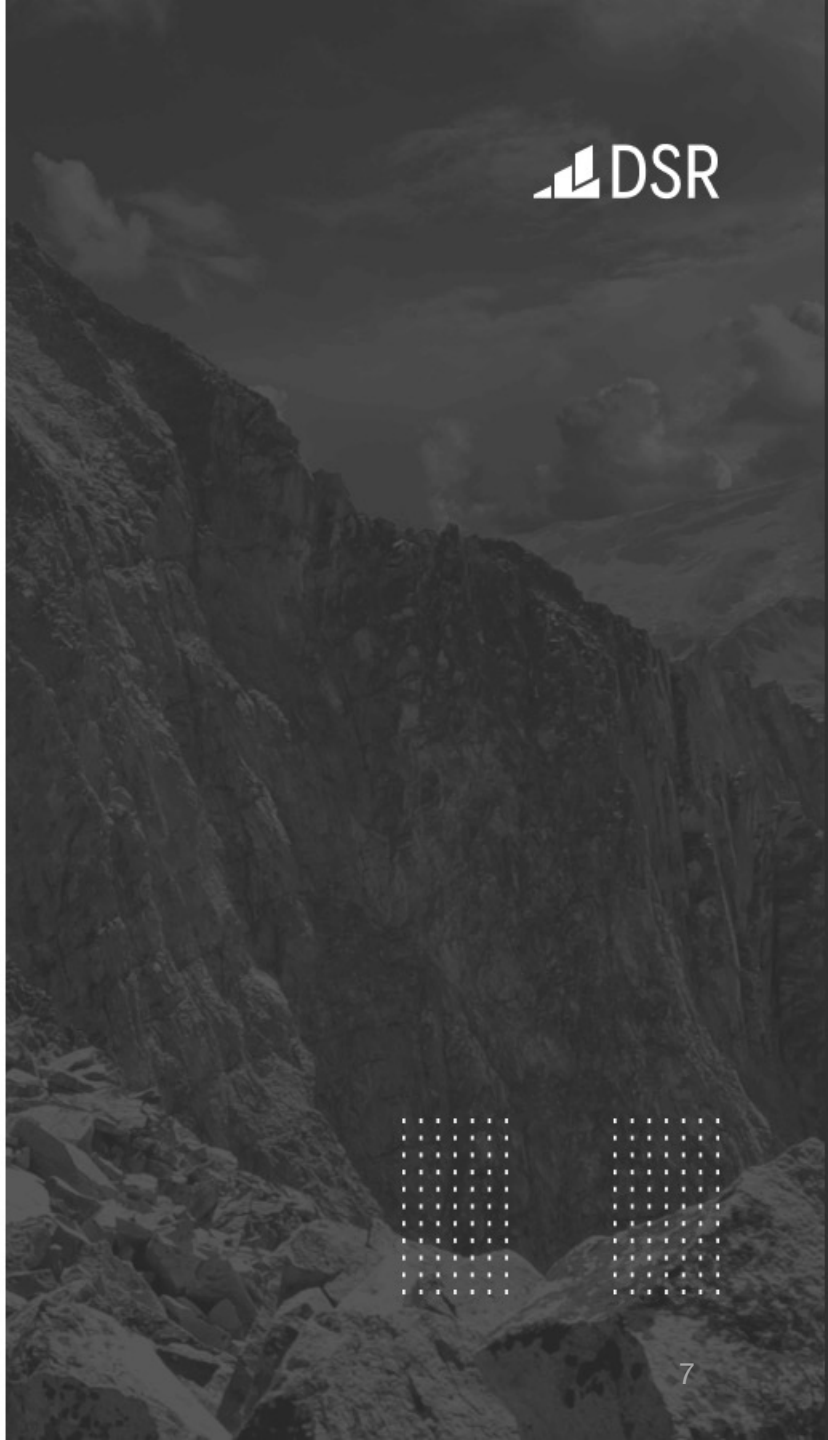
- Over 7 years of experience in Blockchain and Self-sovereign Identity
- Contributed to more than 50 open source projects
- One of the main contributors of:

HYPERLEDGER INDY
HYPERLEDGER ARIES
HYPERLEDGER URSA
sovrin
HYPERLEDGER ANONCREDS
cheqd

- A member of :

Hyperledger FOUNDATION
DIF
OpenWallet FOUNDATION
TRUST Over IP FOUNDATION
THE LINUX FOUNDATION

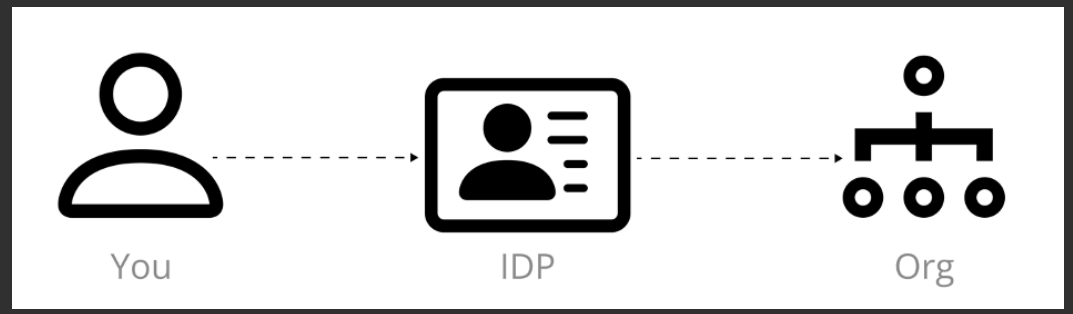
About SSI



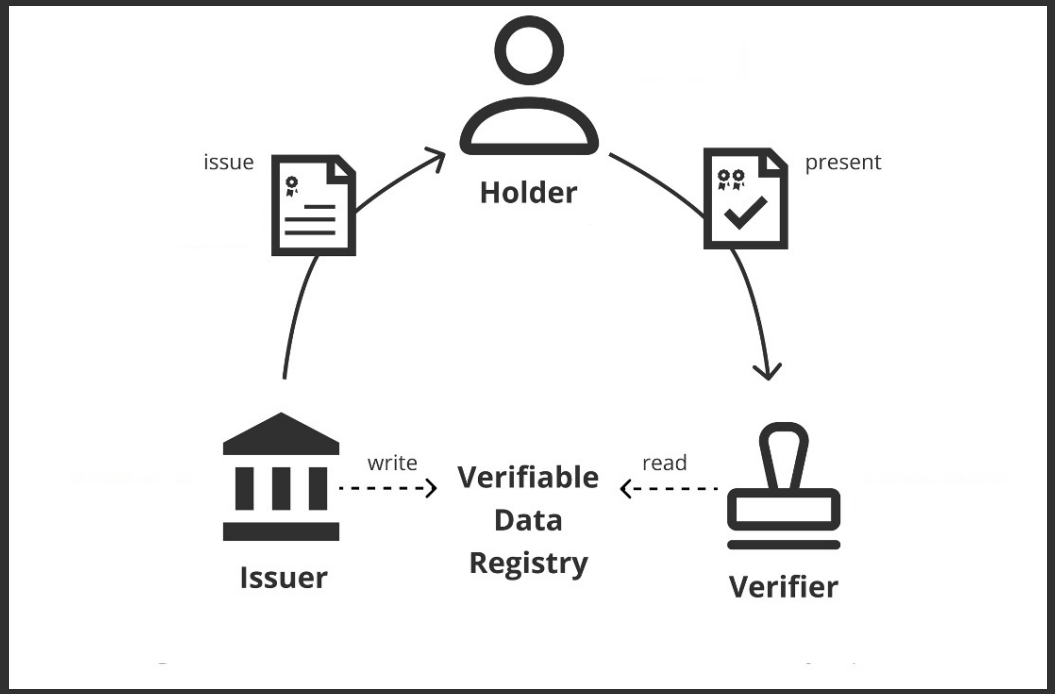
Digital Identity Models

#1: Siloed (Centralized) Identity

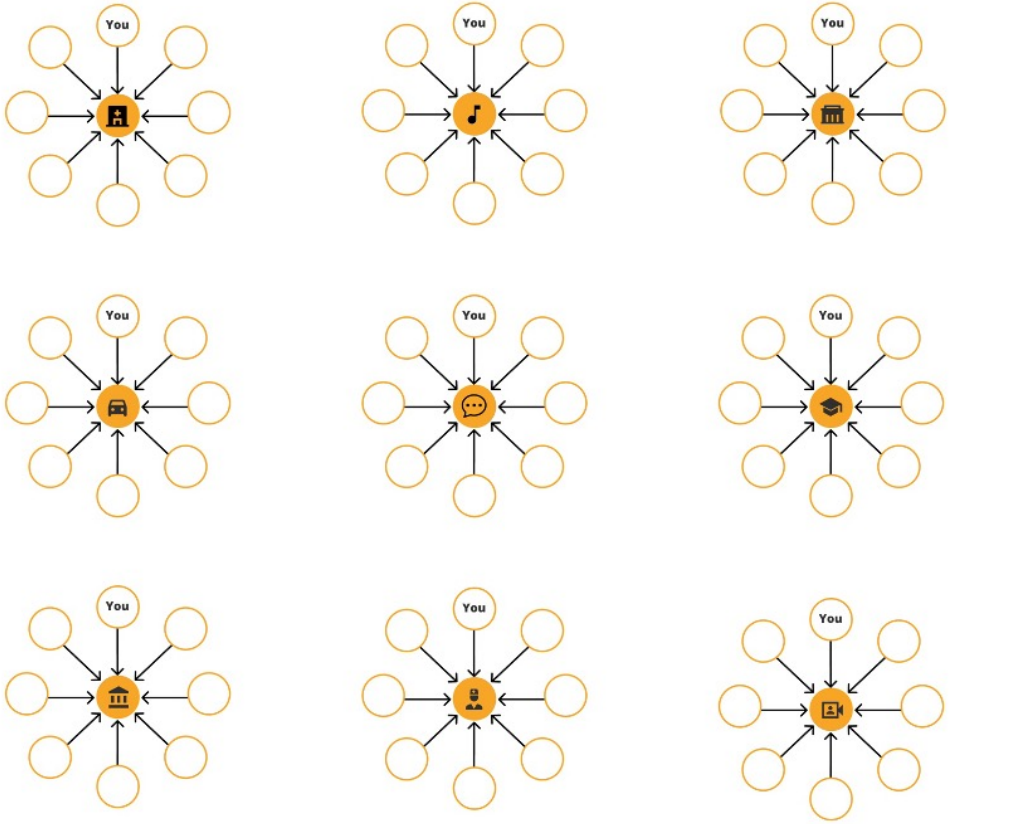
#2: Third-Party IDP (Federated) Identity



#3 Self-Sovereign Identity



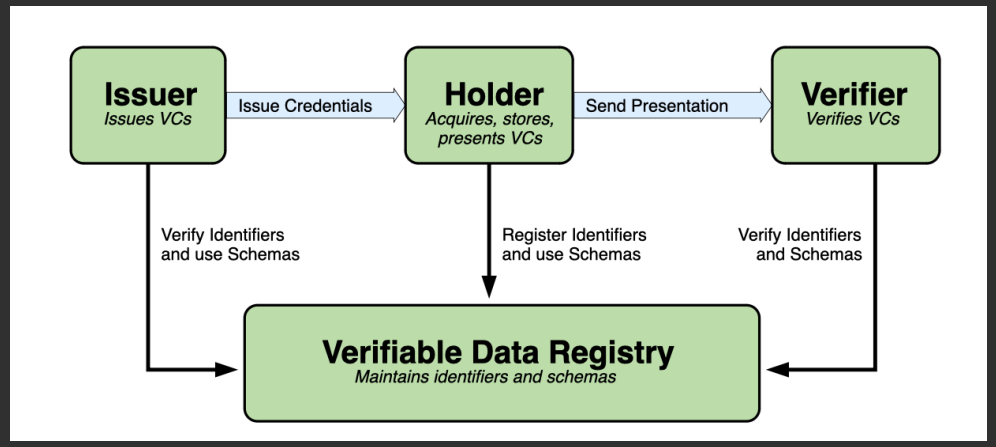
Non-SSI vs SSI



SSI Concepts

1. Verifiable Credentials (VC)

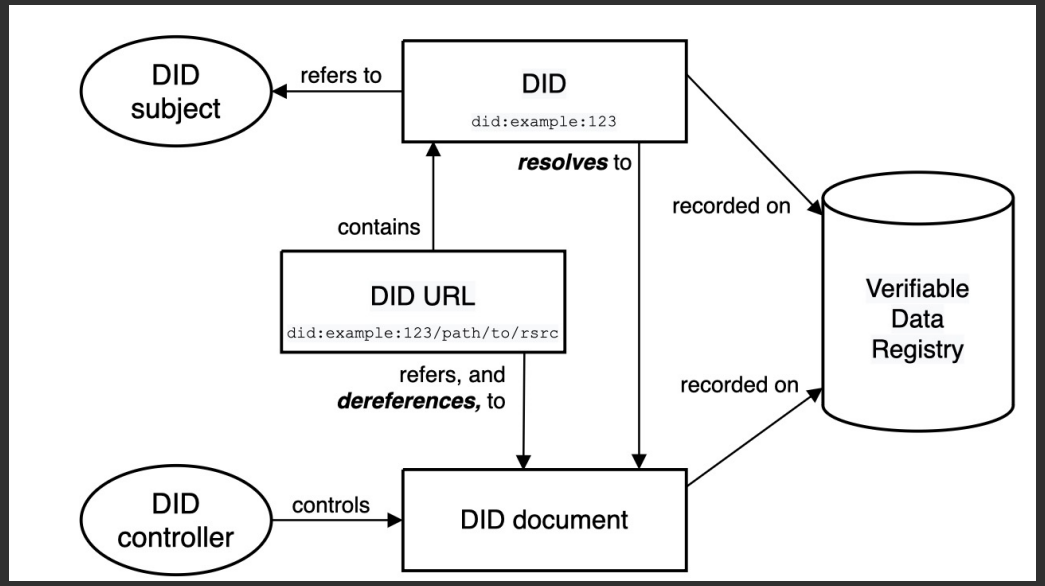
Verifiable Credentials Data Model v1.1,
W3C Recommendation 2022



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

2. Decentralized Identifiers (DID)

Decentralized Identifiers (DIDs) v1.0,
W3C Recommendation 2022



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

SSI Concepts: Verifiable Credential

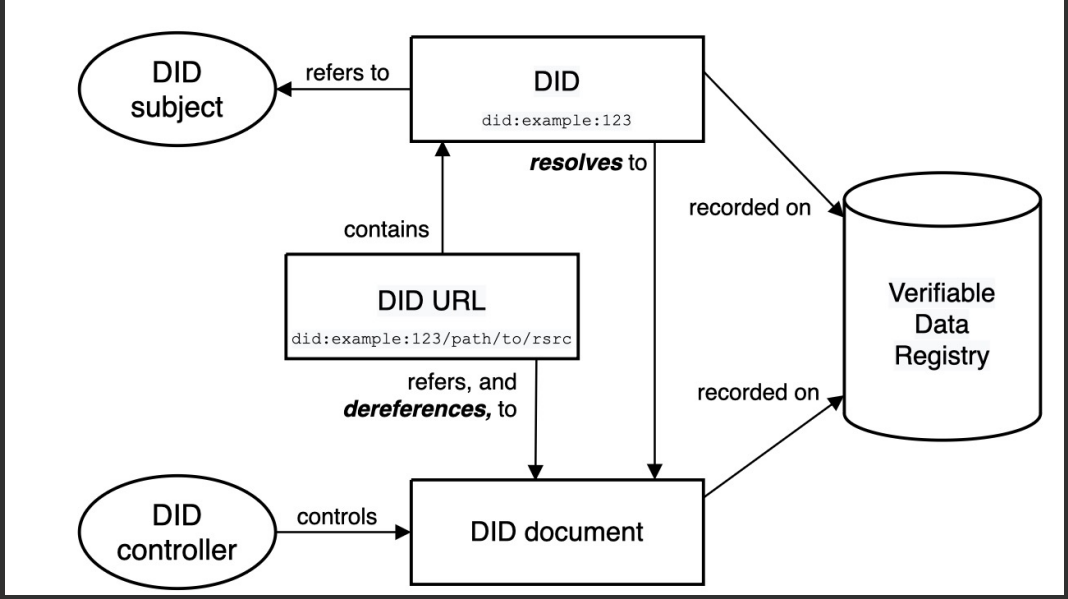
- A credential is a set of one or more claims made by an issuer. Typically the claims describe some properties of the credential holder.
- A verifiable credential is a tamper-evident credential that has authorship that can be cryptographically verified. Verifiable credentials can be used to build verifiable presentations, which can also be cryptographically verified. The claims in a credential can be about different subjects.

Verifiable Credentials Data Model v1.1, W3C Recommendation 2022

JUST LIKE TODAY, BUT DIGITAL, MORE SECURE, AND GLOBAL

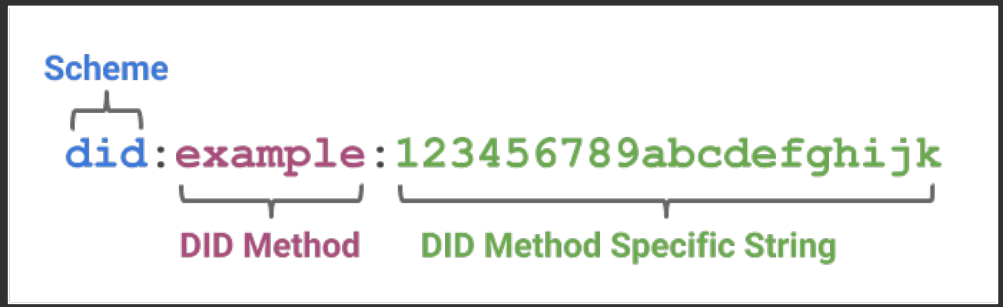
SSI Concepts: DID

- A Decentralized Identifier (DID) refers to any subject (e.g., a person, organization, thing, data model, abstract entity, etc.).
- In contrast to typical, federated identifiers, DIDs may be decoupled from centralized registries, identity providers, and certificate authorities.



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

Decentralized Identifiers (DIDs) v1.0, W3C Recommendation 2022



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

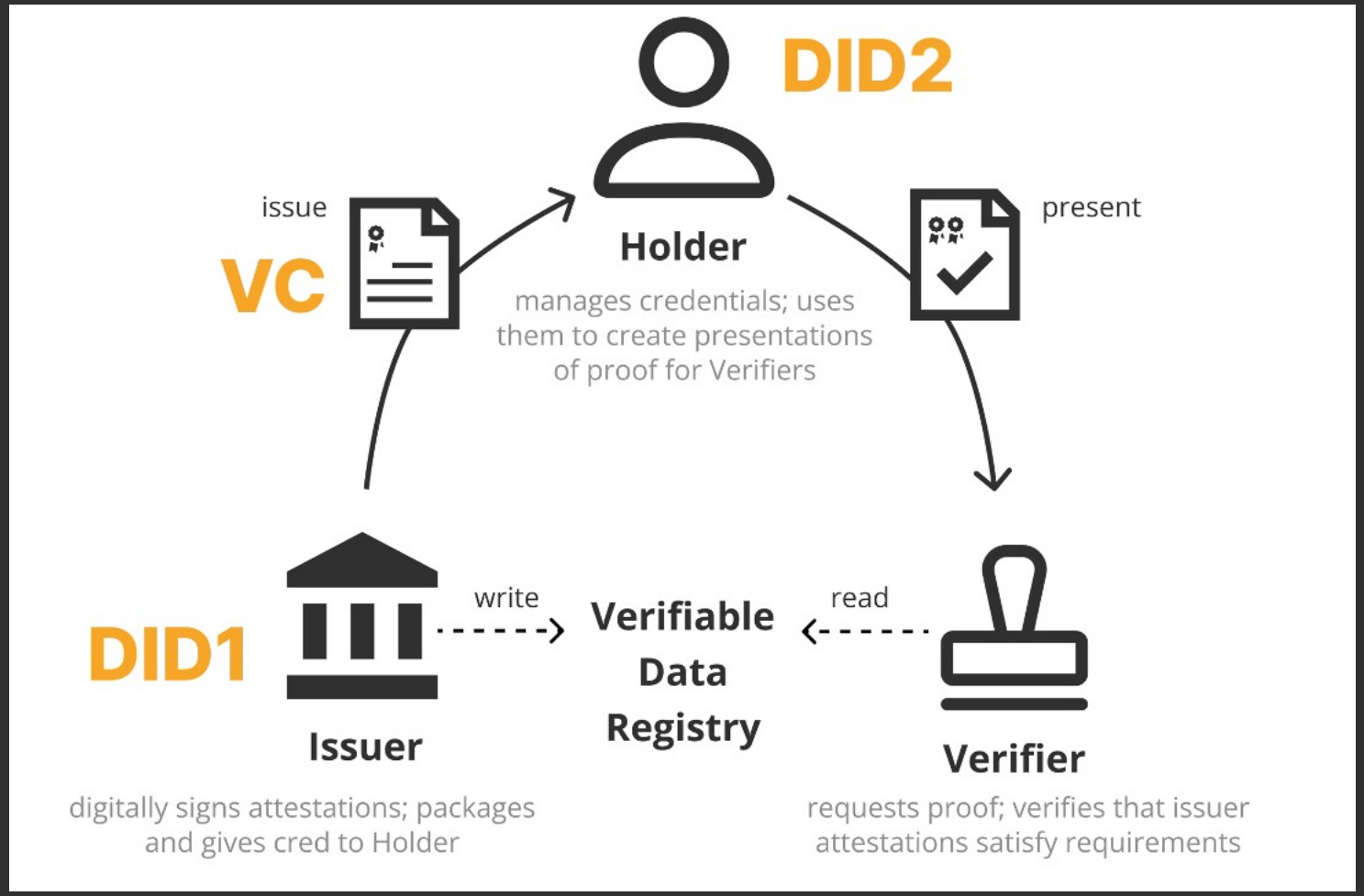
SSI Concepts: DID and DID DOC

```
{
  "@context": ["https://www.w3.org/ns/did/v1","https://identity.foundation/.well-known/did-
  "id": "did:example:123",
  "verificationMethod": [{
    "id": "did:example:123#456",
    "type": "JsonWebKey2020",
    "controller": "did:example:123",
    "publicKeyJwk": {
      "kty": "OKP",
      "crv": "Ed25519",
      "x": "VCpo2LMLhn6iWku8MKvSLg2ZAoC-nlOyPVQaO3FxVeQ"
    }
  }],
  "service": [
    {
      "id":"did:example:123#foo",
      "type": "LinkedDomains",
      "serviceEndpoint": {
        "origins": ["https://foo.example.com", "https://identity.foundation"]
      }
    },
    {
      "id":"did:example:123#bar",
      "type": "LinkedDomains",
      "serviceEndpoint": "https://bar.example.com"
    }
  ]
}
```

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

Property	Required?
id	yes
alsoKnownAs	no
controller	no
verificationMethod	no
authentication	no
assertionMethod	no
keyAgreement	no
capabilityInvocation	no
capabilityDelegation	no
service	no

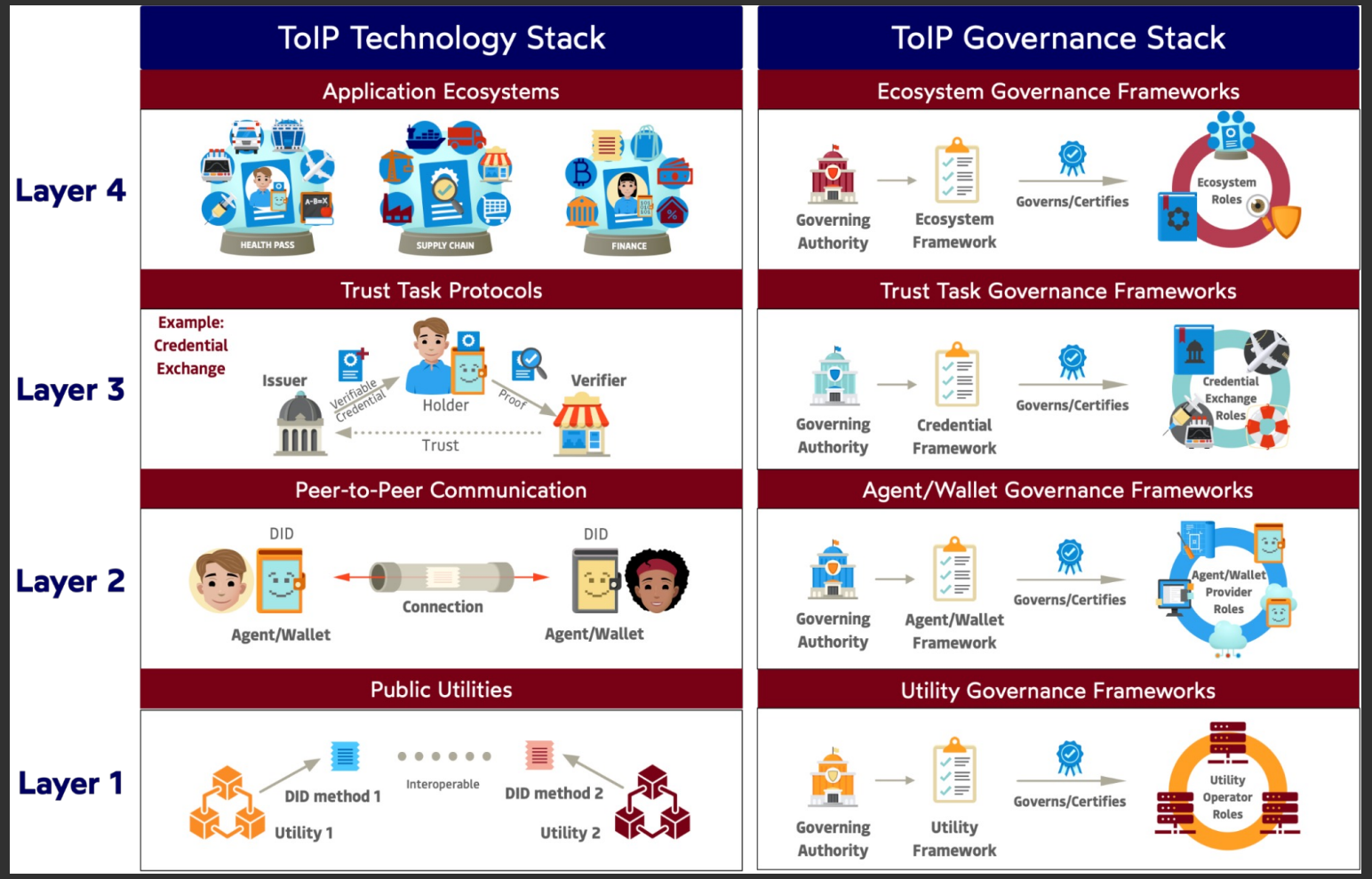
SSI Concepts In Action



By Daniel Hardman licenced under CC BY-SA 4.0

- Issuer's DID and DIDDoc (Public Key)
  - Sign VC
  - Public, usually on a Blockchain or a trusted Web service (did:web, did:indy, did:cheqd, etc.)
- Holder's DID
  - Associates a VC with a DID
  - Prove ownership (signature) of the Holder during presentation
  - Either Public or Private (did:key, did:peer, long form of did:ion, etc.)

Trust Over IP Stack



From https://trustoverip.org/toip-model/

- Layer 3:
  - Verifiable Credential issued and signed by the Issuer
  - Verifiable Presentation created by the Holder and sent to the Verifier
  - The exact mechanism (protocol) how to exchange credentials and presentations
- Layer 2:
  - Secure “connection” to share the credentials and presentations (such as DIDComm)
- Layer 1:
  - Issue/Sign: Issuer puts an Identifier (DID) and associated Public Key and Metadata (DIDDoc) on the VDR (such as Blockchain).
  - Hold/Present: Holder ***may*** put an Identifier (DID) to be associated with a credential on the VDR (such as Blockchain). Proof of possession.

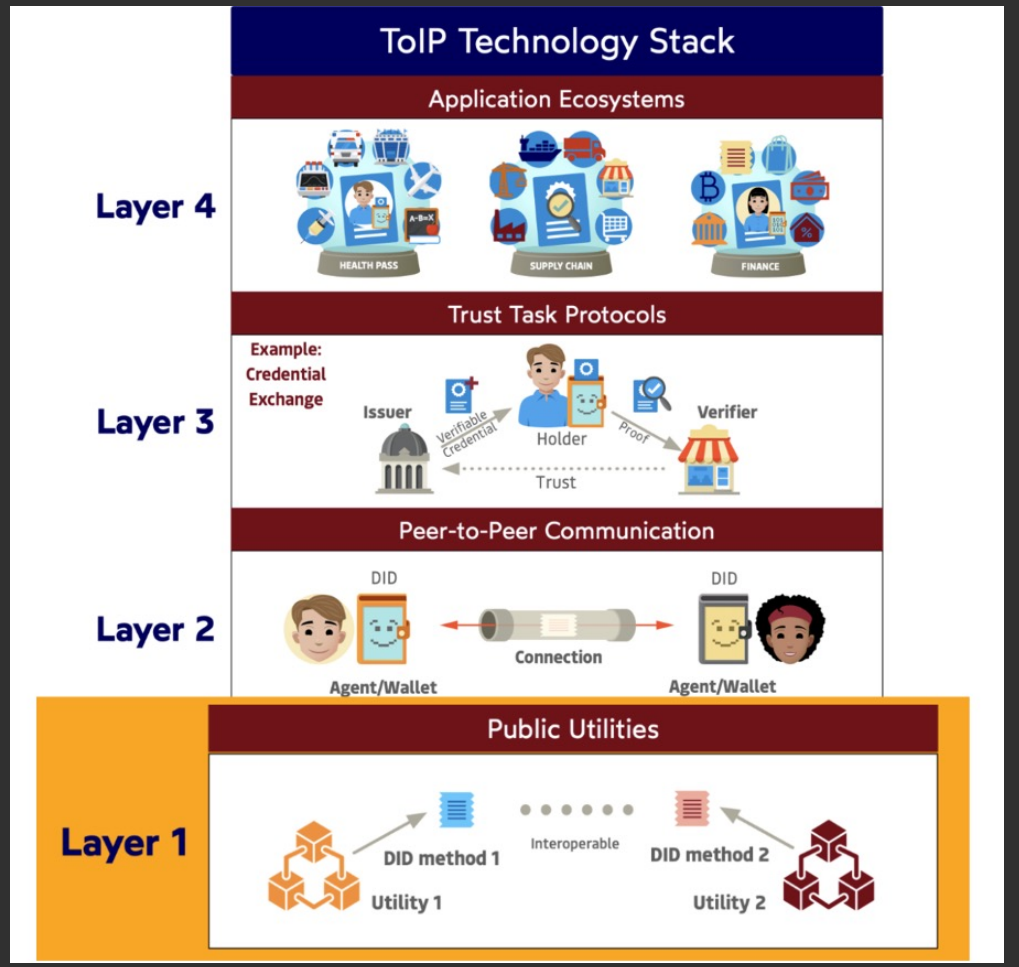
SSI and Blockchain

- Blockchain is optional in SSI cases
- Blockchain may appear on Layer 1 only as one of the options for DID's Verifiable Data Registry (VDR)

- What is **usually** stored on the Blockchain:
  - Issuer's DID/DIDDoc and Public Keys
  - Revocation registries
  - Credential Schemas

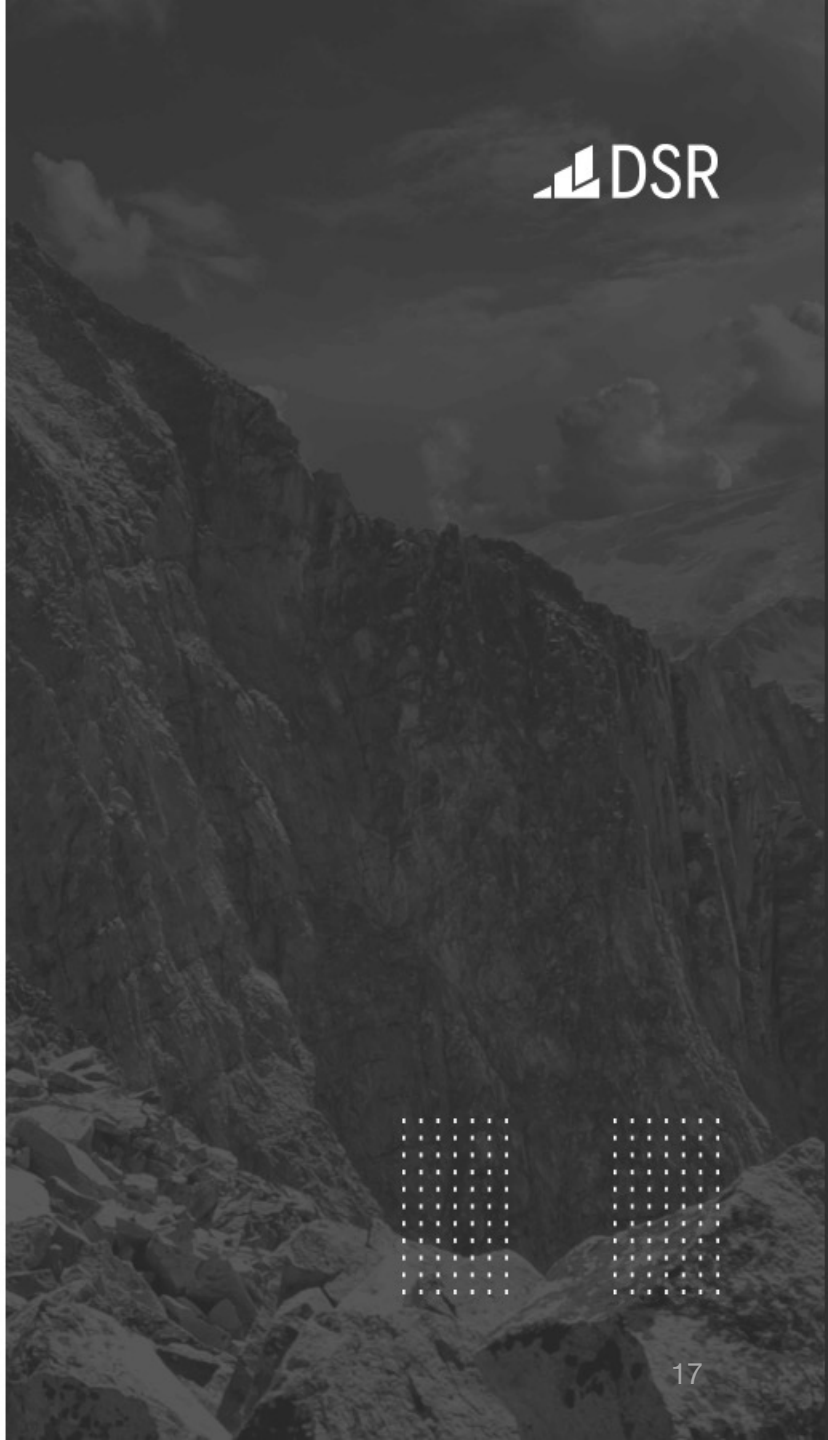
- What **may be** stored on the Blockchain:
  - Holder's DID/DIDDoc and Public Keys

- What is **never** stored on the Blockchain:
  - Verifiable Credentials
  - Private keys

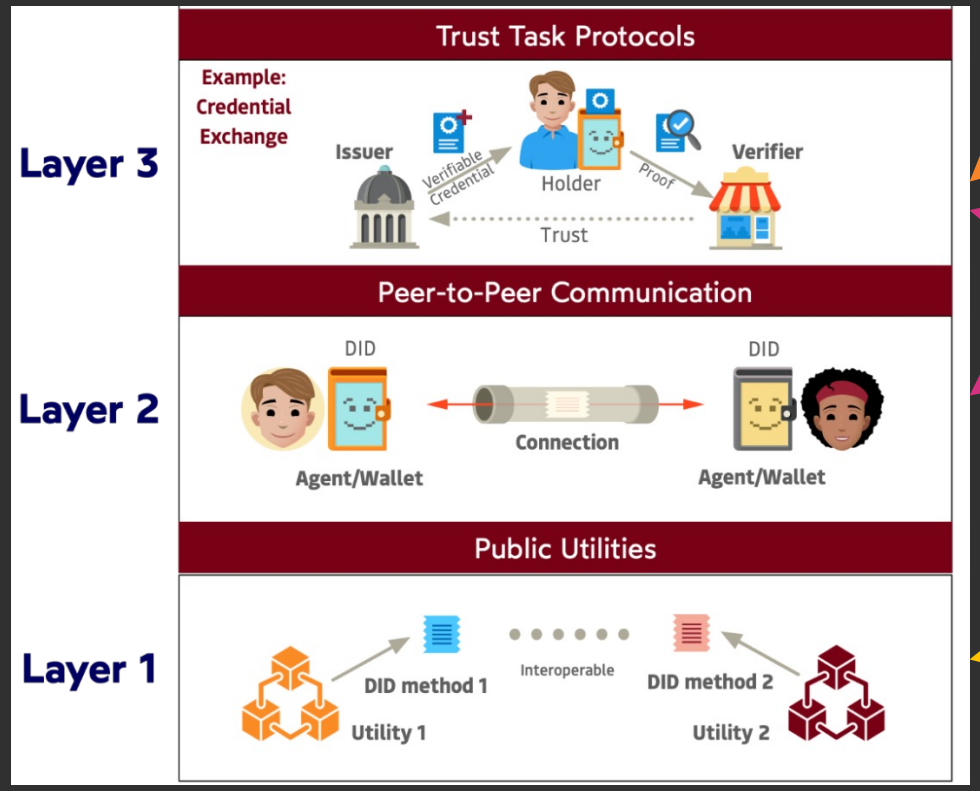


From https://trustoverip.org/toip-model/

Interoperability Variables (Profiles)



Interoperability Variables



From https://trustoverip.org/toip-model/

1. What: VC Format
   - Hyperledger AnonCreds
   - W3C VC (JSON-LD, JWT, SD-JWT)
   - ISO mDL
   - etc.
2. How: VC Exchange Protocols
   - Hyperledger Aries (DIDComm based)
   - DIF WACI (DIDComm based)
   - OID4VC
   - W3C CHAPI
   - W3C VC API
   - ISO mDL
   - etc.
3. Where: DID methods / VDR
   - Hyperledger Indy Ledger (did:indy, did:sov)
   - Cheqd blockchain (did:cheqd)
   - Cardano blockchain (did:prism)
   - Hyperledger Besu, Indy-Besu (did:indy, did:indy2)
   - Self-resolving (did:key)
   - DNS (did:web)
   - etc.

- The combination is sometimes called a **Profile**
- More variables can be defined (such as Revocation, etc.)

Interoperability Variables: More than Indy

DSR

1. What: VC Format
   - **Hyperledger AnonCreds**
   - W3C VC (JSON-LD, JWT, SD-JWT)
   - ISO mDL
   - etc.
2. How: VC Exchange Protocols
   - **Hyperledger Aries (DIDComm based)**
   - DIF WACI (DIDComm based)
   - OID4VC
   - W3C CHAPI
   - W3C VC API
   - ISO mDL
   - etc.
3. Where: DID methods / VDR
   - **Hyperledger Indy Ledger (did:indy, did:sov)**
   - Cheqd blockchain (did:cheqd)
   - Cardano blockchain (did:prism)
   - Hyperledger Besu, Indy-Besu (did:indy, did:indy2)
   - Self-resolving (did:key)
   - DNS (did:web)
   - etc.

Traditional / First Versions

Interoperability Variables: More than Indy

DSR

1. What: VC Format
   - Hyperledger AnonCreds
   - W3C VC (JSON-LD, JWT, SD-JWT)
   - ISO mDL
   - etc.
2. How: VC Exchange Protocols
   - Hyperledger Aries (DIDComm based)
   - DIF WACI (DIDComm based)
   - OID4VC
   - W3C CHAPI
   - W3C VC API
   - ISO mDL
   - etc.
3. Where: DID methods / VDR
   - Hyperledger Indy Ledger (did:indy, did:sov)
   - Cheqd blockchain (did:cheqd)
   - Cardano blockchain (did:prism)
   - Hyperledger Besu, Indy-Besu (did:indy, did:indy2)
   - Self-resolving (did:key)
   - DNS (did:web)
   - etc.

There are much more options / profiles

Interoperability Variables: More than Indy

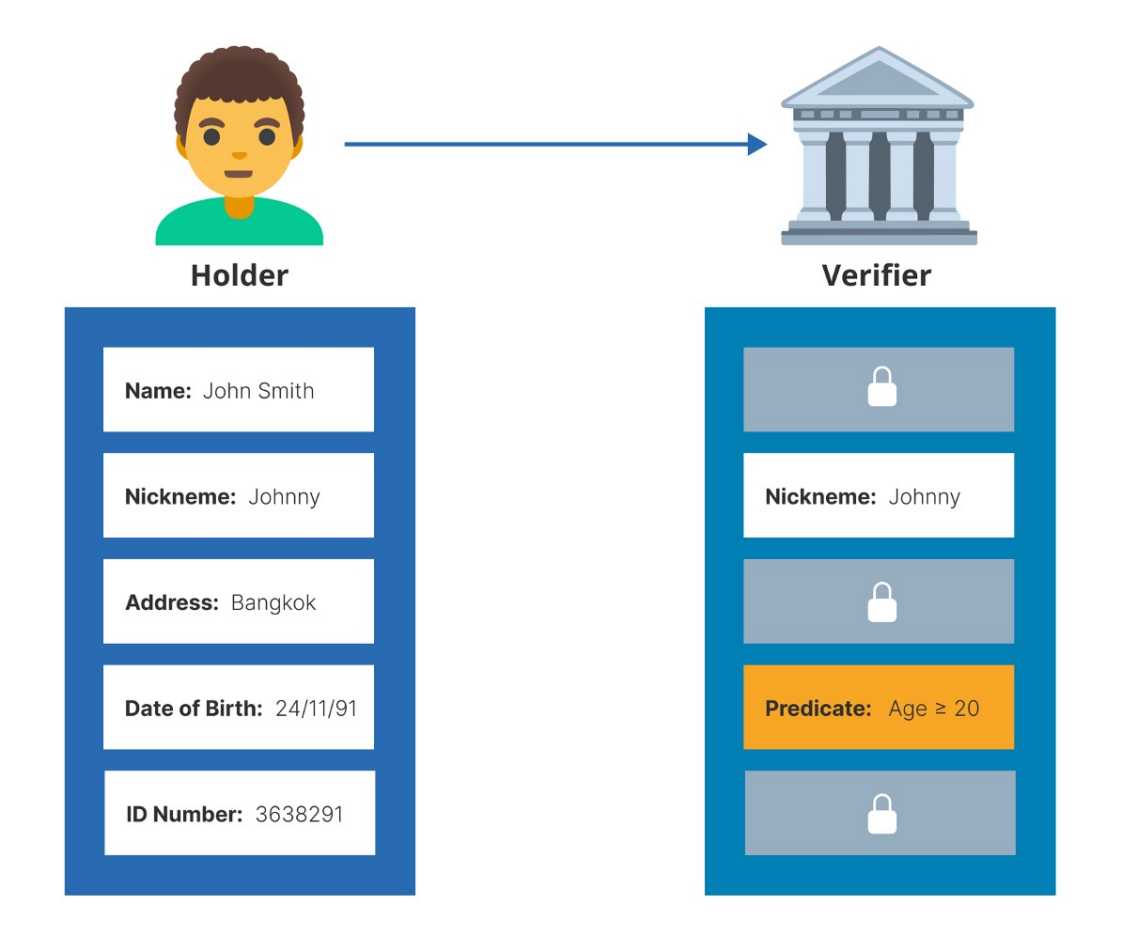
DSR

1. What: VC Format
   - **Hyperledger AnonCreds**
   - **W3C VC (JSON-LD, JWT, SD-JWT)**
   - ISO mDL
   - etc.
2. How: VC Exchange Protocols
   - **Hyperledger Aries (DIDComm based)**
   - **DIF WACI (DIDComm based)**
   - **OID4VC**
   - W3C CHAPI
   - W3C VC API
   - ISO mDL
   - etc.
3. Where: DID methods / VDR
   - **Hyperledger Indy Ledger (did:indy, did:sov)**
   - **Cheqd blockchain (did:cheqd)**
   - **Cardano blockchain (did:prism)**
   - **Hyperledger Besu, Indy-Besu (did:indy, did:indy2)**
   - **Self-resolving (did:key)**
   - **DNS (did:web)**
   - etc.

HYPERLEDGER ARIES

There are much more options / profiles

What: Verifiable Credential Formats Selective Disclosure and Predicates



What: Verifiable Credential Formats

DSR

VC Format	Standard	Selective Disclosure	Predicates (Ex.: proof over 18)	Serialization	Proof Format, Signing Algorithm
W3C VC / JSON-LD	W3C VC	No	No	JSON-LD	Data Integrity Proofs ECDSA, RSA, EdDSA (Ed25519)
W3C VC / JSON-LD BBS+		Yes	No		Data Integrity Proofs BBS+
W3C VC / JWT		No	No	JSON	JWT/JWS: ECDSA, RSA, EdDSA (Ed25519)
W3C VC / SD-JWT		Yes	No	JSON	JWT/JWS: ECDSA, RSA, EdDSA (Ed25519) Selective disclosure JWT (SD-JWT)
Hyperledger AnonCreds (Indy)	Hyperledger AnonCreds	Yes	Yes	JSON	CL AnonCreds (ZKP)
ISO mDL	ISO 18013-5	Yes	No	CBOR	COSE (ECDSA)

How: Verifiable Credential Exchange Protocols

Protocol	Specification	VC Format	Transport	Offline/Online
OpenID for VC	• OpenID for Verifiable Credential Issuance (OID4VCI) • OpenID for Verifiable Presentations (OID4VP) • Self-Issued OpenID Provider v2 (SIOPv2)	Any	HTTP(s), Bluetooth (work in progress)	Online, work in progress for offline
DIDComm based	• Hyperledger Aries • DIF WACI	Any, but usually Hyperledger AnonCreds	Any	Offline and online
ISO mDL Protocols	• ISO/IEC WD TS 23220-3 • ISO/IEC CD TS 18013-7	ISO mDL	NFC, Bluetooth, WiFi, HTTP(s)	Offline and online
W3C Credential Handler API (Browser specific)	W3C Credential Handler API (CHAPI)	W3C VC	Browser API, HTTP(s)	Online
W3C Verifiable Credentials API (REST)	W3C VC API	W3C VC	HTTP(s)	Online

Where: DID methods / VDR

https://www.w3.org/TR/did-spec-registries/#did-methods

- **did:key** - Self-resolving and ledger-independent
- **did:peer** - Ledger-independent; partially self-resolving
- **did:web** - Web domain's existing reputation; resolved through DNS (no Ledger)
- **did:webs** - Web based; KERI instead of DNS for trust
- **did:keri** - Ledger agnostic VDR
- **did:ion** - resolved through blockchain-agnostic Sidetree protocol on top of Bitcoin; self-resolving option
- **did:indy** / **did:sov** – HyperIdger Indy Ledger as VDR
- **did:ethr** - Ethereum as VDR
- **did:cheqd** - Cosmos-sdk based cheqd ledger as VDR
- etc.

Existing Profiles(1)(2)

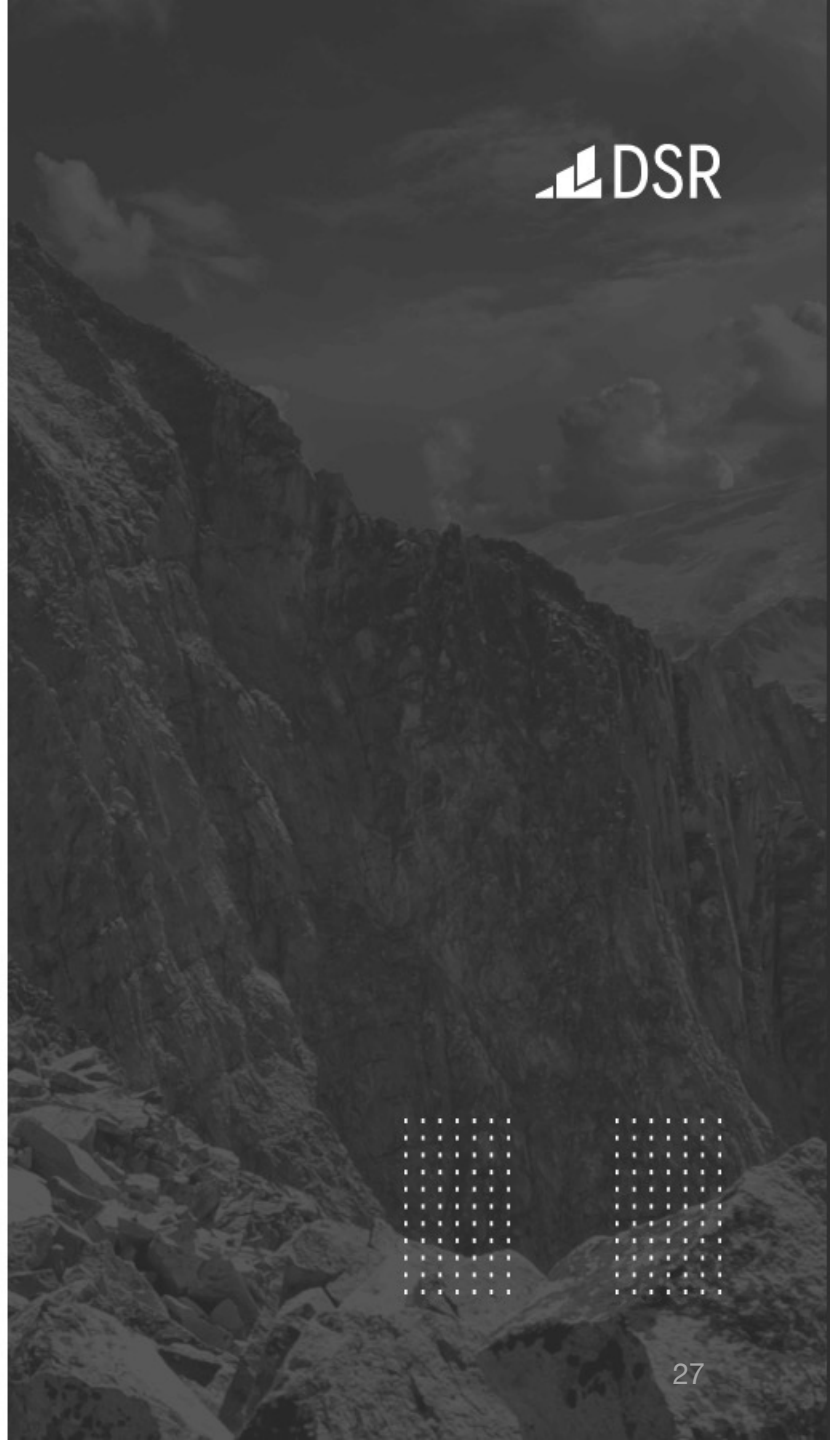
https://openwallet-foundation.github.io/credential-format-comparison-sig

- HAIP - High Assurance Interoperability Profile (OID4VC, SD-JWT-VC, raw keys)
- DIIP - Decentralized Identity Interop Profile (OID4VC, JWT-VC, did:web/did:jwk)
- ISO mDL
- Hyperledger Indy AnonCreds – Hyperledger AnonCreds, Hyperledger Aries, Hyperledger Indy (did:indy, did:sov)

(1) *The notion of a 'Profile' is not fully established yet and may vary across organizations and working groups*

(2) Not all profiles have formal specification

Interoperability Variables Values for the Demo Part



Demo Scenarios: Interoperability Variables

DSR

		VC Format	VC Exchange Protocol	VDR
Demo Part	1. AnonCreds + Cardano	Hyperledger AnonCreds	Hyperledger Aries	Cardano (as AnonCreds VDR) + did:key
	2. AnonCreds + cheqd	Hyperledger AnonCreds	Hyperledger Aries	Cheqd (as AnonCreds VDR + did:cheqd)
	3. W3C VC + cheqd	W3C VC (JSON-LD / Ed25519)	Hyperledger Aries	Cheqd (did:cheqd)
	4. W3C VC + OID4VC	W3C VC (JWT / Ed25519)	OID4VC	did:key
Hands-on Part	5. AnonCreds + Besu	Hyperledger AnonCreds	Hyperledger Aries	Hyperledger Besu (Indy-Besu)
	6. W3C VC + Besu	W3C VC (JSON-LD / Ed25519)	Hyperledger Aries	Hyperledger Besu (Indy-Besu)

Demo Scenarios: Frameworks

DSR

		VC Format	VC Exchange Protocol	VDR
Demo Part	1. AnonCreds + Cardano	Credo (AFJ)	Credo (AFJ)	• Cardano (Test Net) • AFJ extention https://github.com/roots-id/cardano-anoncreds
	2. AnonCreds + cheqd			• Cheqd (Test Net) • AFJ module https://github.com/openwallet-foundation/credo-ts/tree/main/packages/cheqd
	3. W3C VC + cheqd			
	4. W3C VC + OID4VC			Credo (AFJ) (did:key)
Hands-on Part	5. AnonCreds + Besu			• Hyperledger Besu (Indy-Besu) local setup (Docker) • AFJ extention link
	6. W3C VC + Besu			

Credo (Aries Framework JavaScript) - 1

https://github.com/openwallet-foundation/credo-ts
(Ex. https://github.com/hyperledger/aries-framework-javascript)

- JavaScript / TypeScript
- BE Web Servers (Node.JS) or Mobile Apps (React Native)
  - A basis for Aries Bifold
- SDK/API to be integrated into Web or Mobile apps
  - There are extensions providing REST API on top of it

- Used to be one of Hyperledger Aries projects since 2019
- Moved to Open Wallet Foundation at the end of 2023
- Renamed from Aries Framework JavaScript (AFJ) to Credo

Credo (Aries Framework JavaScript) - 2

https://github.com/openwallet-foundation/credo-ts
(Ex. https://github.com/hyperledger/aries-framework-javascript)

- Wallet:
  - Hyperledger Aries Ascar
  - Hyperledger Indy SDK
- VC Formats:
  - Hyperledger AnonCreds
  - W3C VC JSON-LD, W3C VC JWT,
  - W3C VC SD-JWT
  - W3C VC BBS+
- VC Exchange Protocols:
  - Aries v1/v2
  - OID4VC (WIP)
- DID methods / VDRs
  - did:sov, did:key, did:peer, did:cheqd
  - Extenentions (DID method, AnonCreds registries)

VC Format: Hyperledger AnonCreds

https://github.com/hyperledger/anoncreds-spec

https://github.com/hyperledger/anoncreds-rs

- The most privacy preserving VC Format
  - Predicates, selective disclosure, ZKP, anonymous revocation
- Main adoption historically – Hyperledger Indy, Hyperledger Aries
- Implemented and adopted before W3C VC standard was finalized
- Custom format (JSON), ZKP based signature (CL)
- Recent W3C VC Representation Support (link)
- Entities (published to a VDR, usually a ledger):
  - Credential Schema
  - Credential Definition (Issuer PubKey)
  - Revocation registry

VC Format: Hyperledger AnonCreds History

DSR

2017: Hyperledger Indy, Sovrin Main Net
- indy-crypto, indy-sdk, Indy Ledger

2018: Hyperldger Ursa
- replaced indy-crypto

2019: Hyperledger Aries
- used indy-sdk, Indy Ledger

2022: Hyperledger AnonCreds
- AnonCreds spec, anoncreds-rs, ledger-agnostic AnonCreds

VC Format: W3C VC

https://www.w3.org/TR/vc-data-model/ (W3C Recommendation 2022)

https://www.w3.org/TR/vc-data-model-2.0 (W3C Working Draft)

- Serialization:
  - JSON
  - JSON-LD (Linked Data, Context, Semantic)
- Proof Format:
  - Data Integrity Proofs
  - JWT / SD-JWT (selective disclosure)
- Crypto/signatures:
  - ECDSA
  - RSA
  - EdDSA (Ed25519)
  - BBS+ (selective disclosure)

VC Format:
W3C VC JSON-LD + Data Integrity Proofs + EdDSA

DSR

Verifiable Credential (VC)

```
{
  "@context": ["https://www.w3.org/2018/credentials/v1", …. ],
  "type": ["VerifiableCredential", ”ComcastCredential"],
  ……

  // (1) ID of Credential Issuer.
  // Resolved to machine-readable info about the issuer, e.g. Public Keys
  "issuer": ”did:example:565049",
   …

  // (2) claims about the subject of the credential (holder)
  "credentialSubject": {
    // ID of credential subject (holder)
    // Resolved to machine-readable info about the holder, e.g. Public Keys
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    ”name": “John”,
    “contractNum”: 123456,
    ……
  },

  // (3) digital proof that makes the credential tamper-evident
  "proof": {
    "type": "Ed25519Signature2020",
    "created": "2022-02-25T14:58:42Z",
    "verificationMethod": ”did:example:565049#key-1",
    "proofPurpose": "assertionMethod",
    "proofValue": "z3FjecWufY46yg5LhxhueARiKBk9czhSePTFehP…”
  }
}
```

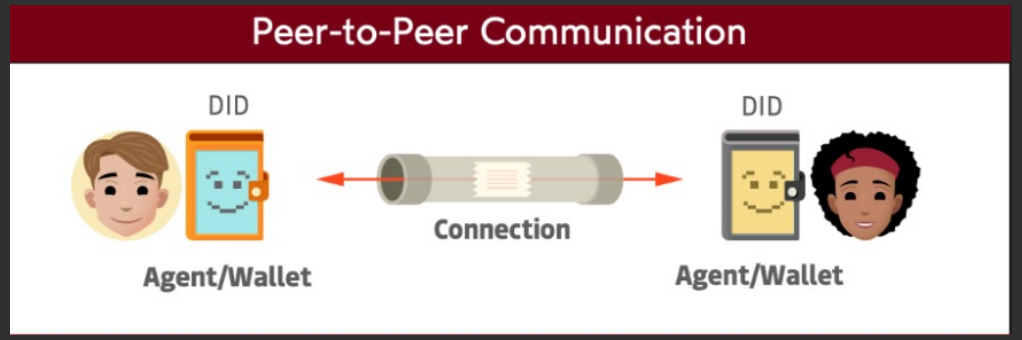
Verifiable Presentation (VP)

```
{
  "@context": ["https://www.w3.org/2018/credentials/v1", …. ],
  "type": "VerifiablePresentation",

  // (1) Verifiable Credential, see example ay the left.
  "verifiableCredential": [{
      "issuer": ” did:example:565049",
      ….
      "credentialSubject": {
        "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
        ”name": “John”,
        “contractNum”: 123456,
        ……
      },
      "proof": { …. }
  }],


  // (2) digital signature by the credential holder
  // (proof of key possession)
  "proof": {
    "type": "Ed25519Signature2020",
    "created": "2022-03-25T16:37:21Z",
    "verificationMethod": ”did:example:ebfeb1f712ebc6f1c276e12ec21#key-1",
    "proofPurpose": "assertionMethod",
    "proofValue": ”f3FjecWufY46yg5Lhxhued244Jsdfs,PsdfgIO2d…”
  }
}
```

VC Exchange Protocol: Hyperledger Aries

https://github.com/hyperledger/aries-rfcs

- DIDComm based
  - A secure, private communication methodology built atop the decentralized design of DIDs.
  - DIDComm v1: https://github.com/hyperledger/aries-rfcs/tree/main/concepts/0005-didcomm
  - DIDComm v2: https://identity.foundation/didcomm-messaging/spec/v2.0/
- Protocols (state machine) running on top of it
  - VC Exchange Protocols
  - Any other (custom) protocol
  - Composition of protocols
  - https://didcomm.org – list of protocols



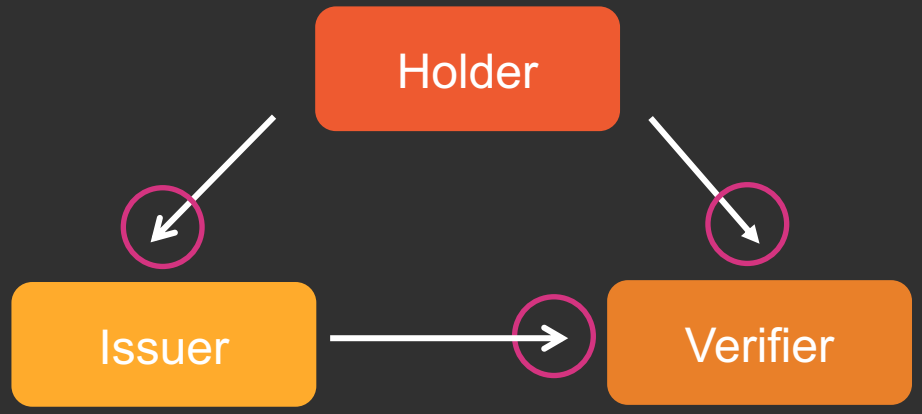
From https://trustoverip.org/toip-model/

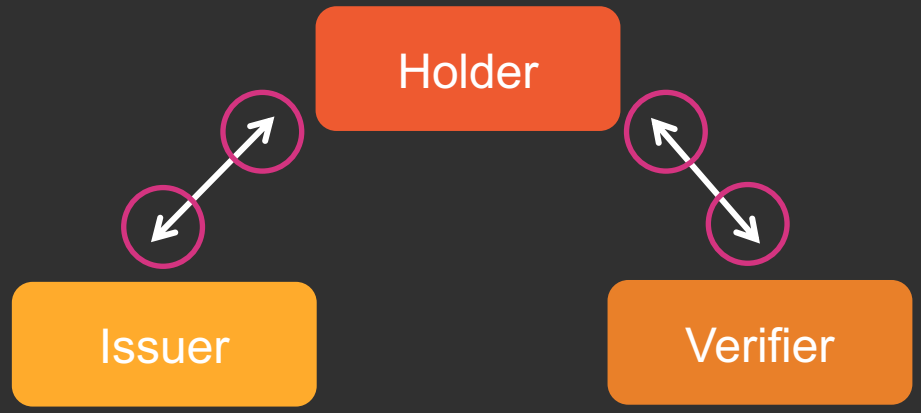
VC Exchange Protocol: Hyperledger Aries

https://github.com/hyperledger/aries-rfcs

- Aries Interop Profile v1
  - Establish connections (Connection protocol)
  - Exchange credentials and presentations via connections (Issuance and Presentation protocols)
  - Complete a connection-less proof-request/proof transaction
- Aries Interop Profile v2
  - Establish connections (DID Exchange protocol)
  - Exchange credentials and presentations via connections (Issuance and Presentation protocols)
  - Complete a connection-less proof-request/proof transaction
  - Reuse connections (out-of-band protocol)
  - Improved UX
  - Transition to DIDComm v2
  - Multiple ledger types and verifiable credential formats
  - Standard mediator coordination capabilities for mobile agents and multi-tenant agencies

VC Exchange Protocol: OID4VC
OIDC vs OID4VC





OAuth and OpenID Connect (OIDC)

Protocol that support authorization-based credential exchange where the holder authorizes a verifier (client) to access information on her behalf.

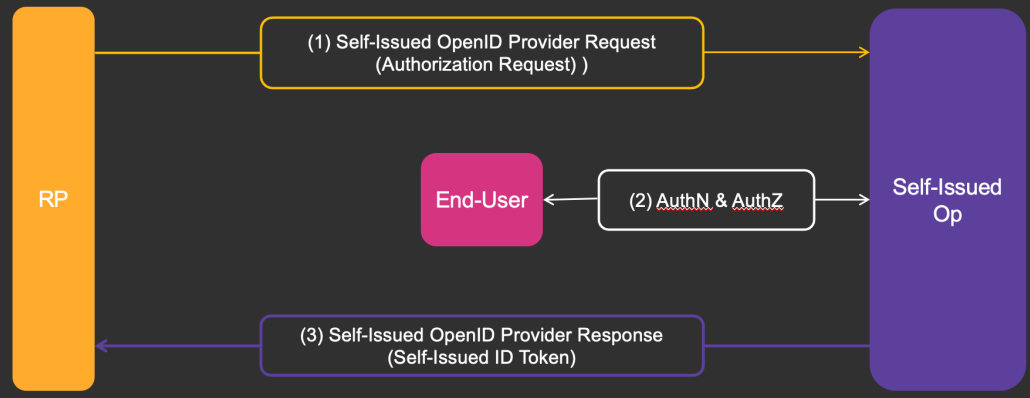
OpenID for VCs (OID4VC)

Protocol that supports self-sovereign credential exchange where the holder can autonomously control the exchange of credentials with any verifier she wants.

VC Exchange Protocol: OID4VC Standards

1.1 Authentication: SIOPv2

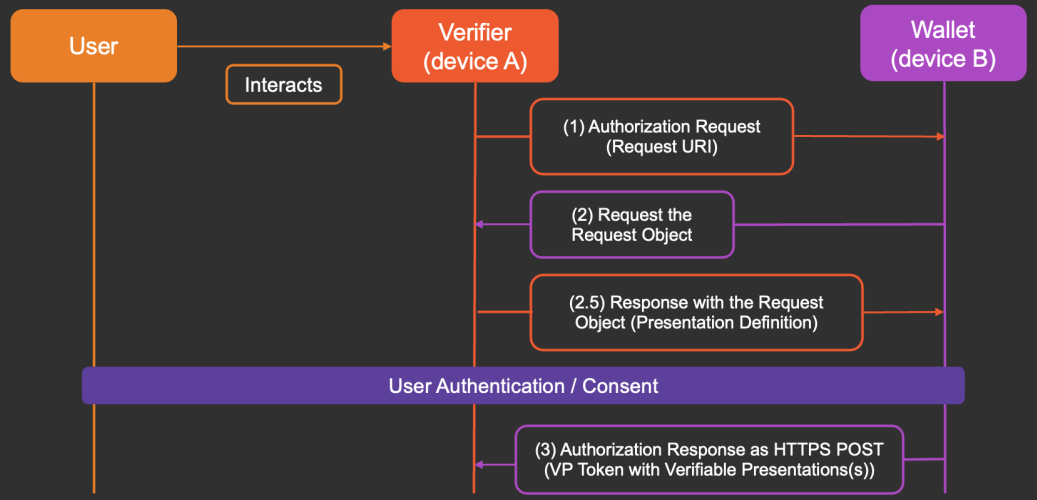
Defines how holders can authenticate in a self-sovereign way with any actor
(Self-issued ID Token)



From https://openid.github.io/SIOPv2/openid-connect-self-issued-v2-wg-draft.html

1.2 Presentation: OID4VP

Defines mechanisms on top of SIOPv2 to allow the presentation of claims in the form of Verifiable Credentials
(complements self-issued ID token with cryptographically verifiable claims – VP token)

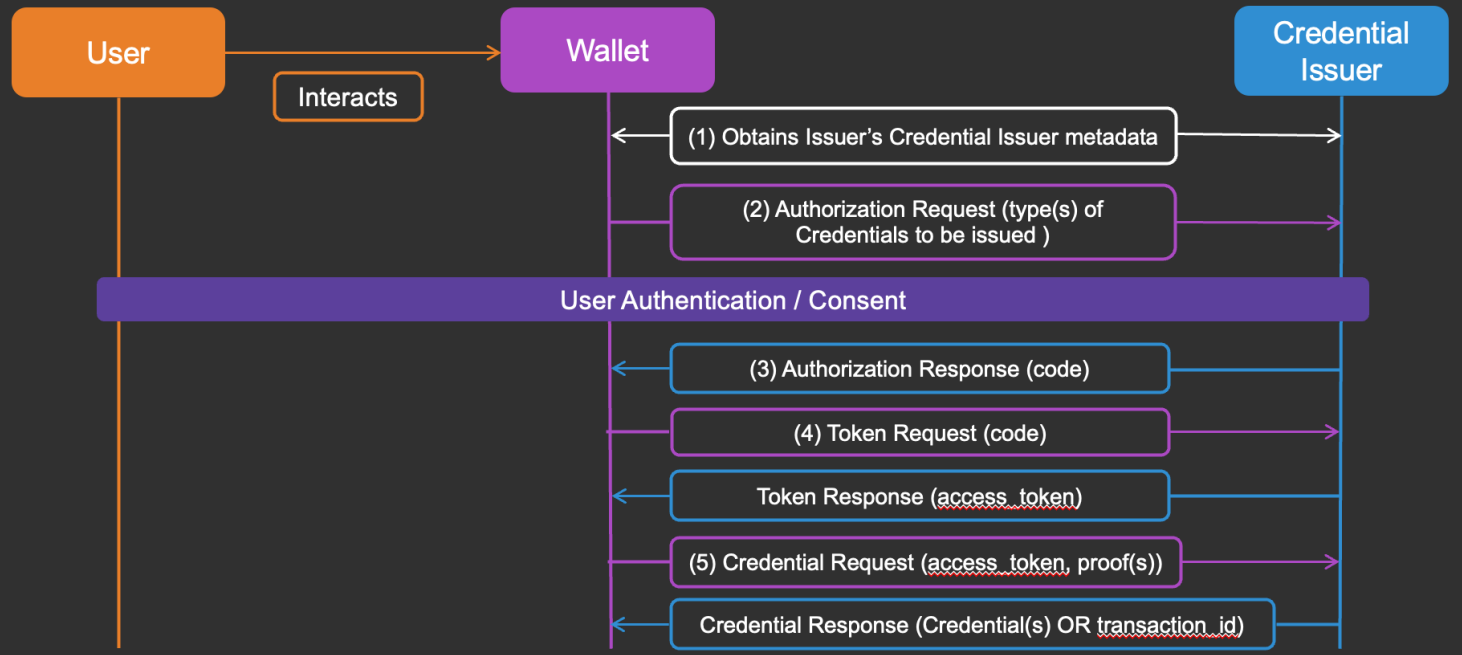


From https://openid.github.io/OpenID4VP/openid-4-verifiable-presentations-wg-draft.html

VC Exchange Protocol: OID4VC Standards

2. Issuance: OID4VCI

Defines APIs and the corresponding OAuth2-based authorisation mechanisms for the issuance of Verifiable Credentials



From https://openid.github.io/OpenID4VCI/openid-4-verifiable-credential-issuance-wg-draft.html

VDR: Cardano

https://github.com/IntersectMBO/cardano-node

Public Permissionless Blockchain

- Proof-of-stake consensus protocol Ouroboros
  - Peer-reviewed, verifiably secure
  - One of the pioneers of the proof-of-stake approach
- Applications: Smart Contracts, DApps, DeFi, DAO, NFT, SSI
- Main Net since 2017

Cardano and SSI

- AFJ extention: Cardano as AnonCreds Registry to publish Schema, CredDef, etc. https://github.com/roots-id/cardano-anoncreds
- Atala Prism, now open sourced as Hyperledger Labs Open Enterpise Agent
- VC Formats: W3C VC, Hyperledger AnonCreds

VDR: cheqd

https://github.com/cheqd/cheqd-node

- Public Permissionless Blockchain
- Application specific Blockchain: Decentralized Identity / SSI
- Built using the Cosmos SDK blockchain framework
- Proof-of-stake consensus protocol(Tendermint / Cosmos)
- Main Net since 2021
- SSI Features:
  - did:cheqd
  - Cheqd as AnonCreds registry
  - Integrated into AFJ as a core module: https://github.com/openwallet-foundation/credo-ts/tree/main/packages/cheqd
  - VC Formats: W3C VC, Hyperledger AnonCreds
  - DID-linked resources
  - Universal Resolver

VDR: Hyperledger Besu

https://github.com/hyperledger/besu

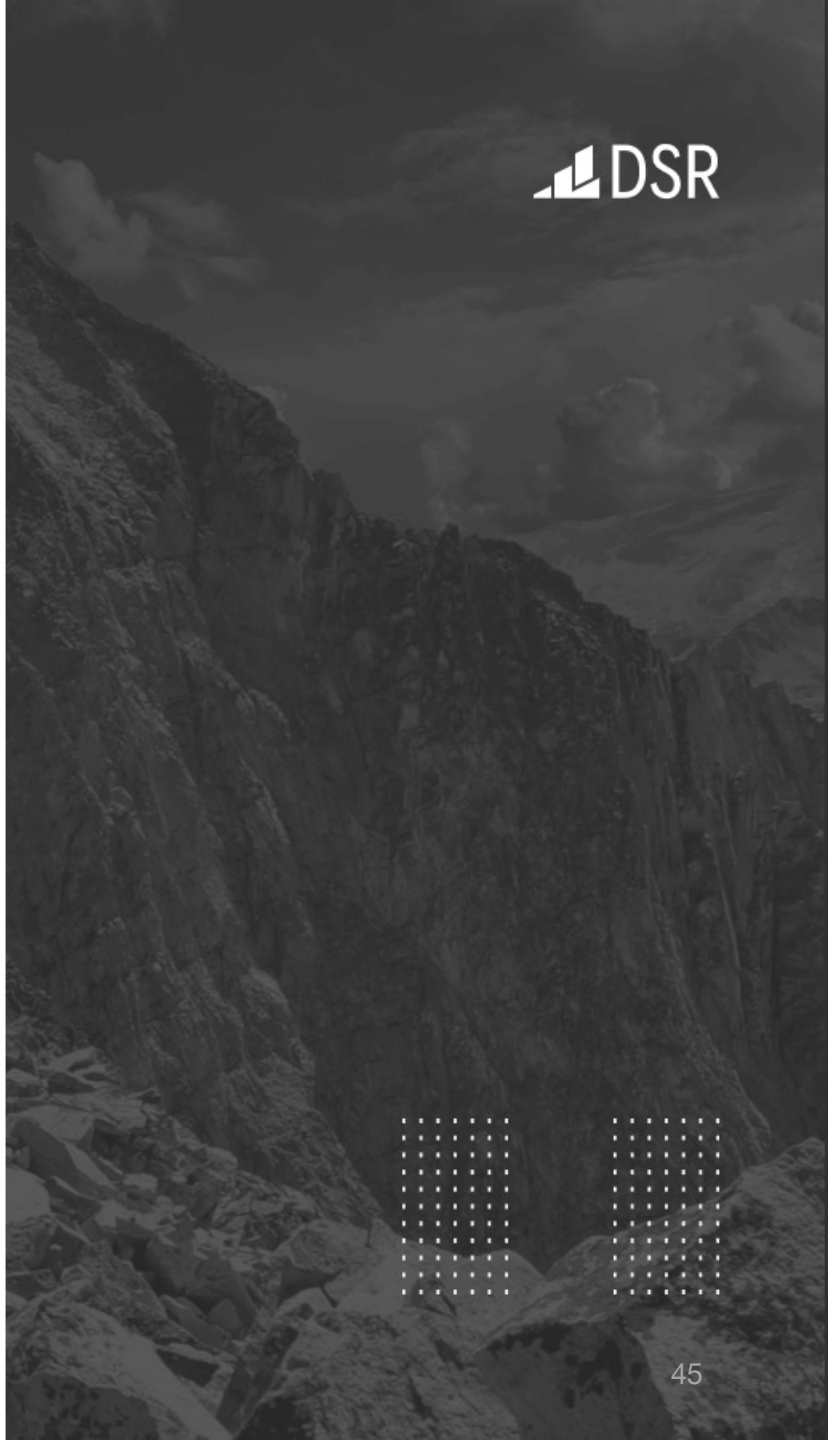
- Graduated/Active Hyperledger project since 2020
- Ethereum client written in Java
- Two use cases:
  - Public networks (such as public Ethereum nets)
  - Private Permissioned networks (such as enterprise or supply chain ledgers)
- Different approach for public and private cases (consensus, features, etc)
- Includes several consensus algorithms: PoS, PoW, PoA (IBFT 2.0, QBFT, Clique)
- Business logic as Solidity/Ethereum Smart Contracts
- Pluggable Architecture
- Private Permissioned Use Cases: CBDC, Supply Chain, Enterprise ledgers, SSI

VDR: Indy-Besu

https://github.com/hyperledger/indy-besu

- Public permissioned Ledger for SSI
- Indy (did:sov, did:indy) compatible
- Based on Hypereldger Besu, replaces Indy Plenum
  - QBFT consensus protocol instead of RBFT
- SSI business logic (DID and AnonCreds registry) as Solidity/Ethereum smart contracts
- Client library; integration with Credo (Aries frameworks JavaScript)
- SSI Features:
  - did:ethr
  - did:indybesu (WIP)
  - AnonCreds registry
  - VC Formats: W3C VC, Hyperledger AnonCreds
- Status:
  - Sep 2023: DSR proposes initiative at Indy Summit meeting
  - Nov 2023: DSR delivers PoC
  - Jan 2023: Code moved to a separate Hyperledger Indy (indy-besu)
  - Ongoing: MVP work

Demo

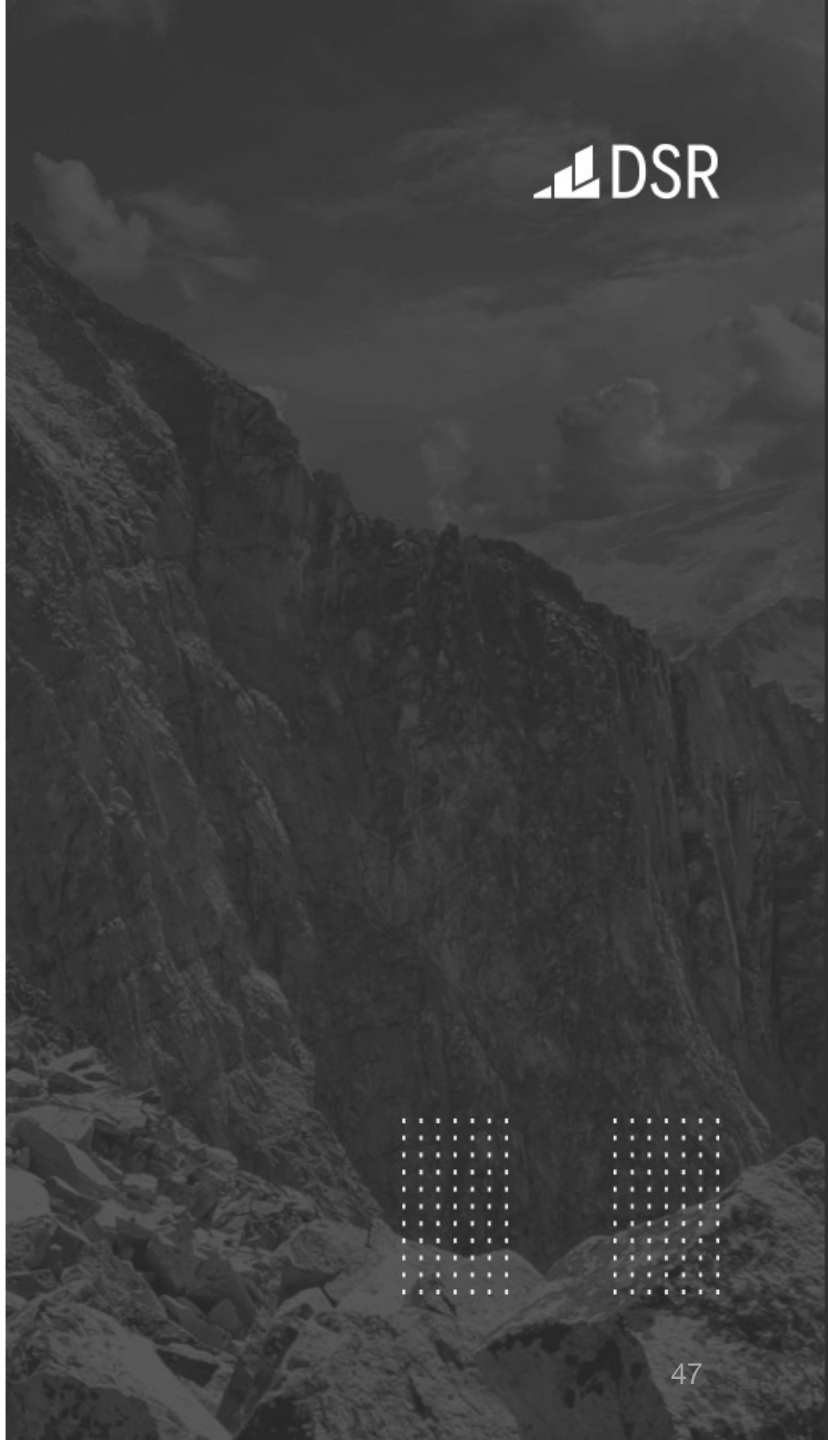


Demo Scenarios

DSR

	VC Format	VC Exchange Protocol	VDR
1. AnonCreds + Cardano	Hyperledger AnonCreds	Hyperledger Aries	Cardano (as AnonCreds VDR) + did:key
2. AnonCreds + cheqd	Hyperledger AnonCreds	Hyperledger Aries	Cheqd (as AnonCreds VDR + did:cheqd)
3. W3C VC + cheqd	W3C VC (JSON-LD / Ed25519)	Hyperledger Aries	Cheqd (did:cheqd)
4. W3C VC + OID4VC	W3C VC (JWT / Ed25519)	OID4VC	did:key

Hands-On



Demo Scenarios

	VC Formats	VC Exchange Protocol	VDR
5. AnonCreds + Besu	Hyperledger AnonCreds	Hyperledger Aries	Hyperledger Besu (Indy-Besu)
6. W3C VC + Besu	W3C VC (JSON-LD / Ed25519)	Hyperledger Aries	Hyperledger Besu (Indy-Besu)

Pre-Requisites

DSR

For the hands-on, we are going to use a Gitpod profile, which you can log into through your GitHub account.

- We ask all participants to make sure that they have at least a few hours of workspace usage in their Gitpod profile to participate in the hands-on. You can check this using the following link.
- Recommended browsers: Chrome or Firefox.

Part 1: Write the Code
Store Indy-Besu DID Document

DSR

```
const assertKey = await this.agent.wallet.createKey({ keyType: KeyType.Ed25519 })

const createdDid = await this.agent.dids.create<IndyBesuDidCreateOptions>({
  method: 'ethr',
  options: {
    verificationKeys: [
      {
        type: VerificationKeyType.Ed25519VerificationKey2018,
        key: assertKey,
        purpose: VerificationKeyPurpose.AssertionMethod,
      },
    ],
  },
})
```

Part 2: Write the Code
Issue AnoncCreds Credential

DSR

Create credential

```
const credential = {
      attributes: [
        {
          name: 'name',
          value: 'Alice Smith',
        },
        {
          name: 'degree',
          value: 'Computer Science',
        },
        {
          name: 'date',
          value: '01/01/2022',
        },
      ],
      credentialDefinitionId:
this.credentialDefinition.credentialDefinitionId,
    }
```

Offer credential

```
   const record = await this.agent.credentials.offerCredential({
      connectionId: connectionRecord.id,
      protocolVersion: 'v2',
      credentialFormats: {
        anoncreds: credential,
      },
    })
```

Wait for accept

```
await this.waitForAcceptCredential(record.id)
```

Part 3: Write the Code
Issue AnoncCreds Credential

DSR

Create credential

```
const credential = {
      '@context': [CREDENTIALS_CONTEXT_V1_URL,
'https://www.w3.org/2018/credentials/examples/v1'],
      type: ['VerifiableCredential',
'FaberCollege'],
      issuer: this.issuerId,
      issuanceDate: '2023-12-07T12:23:48Z',
      credentialSubject: {
        name: 'Alice Smith',
        degree: 'Computer Science',
      },
    }
```

Offer credential

```
const record = await this.agent.credentials.offerCredential({
     connectionId: connectionRecord.id,
     protocolVersion: 'v2',
     credentialFormats: {
       jsonld: {
         credential: credential,
         options: {
           proofType: 'Ed25519Signature2018',
           proofPurpose: 'assertionMethod',
         },
       },
     },
   })
```

Wait for accept

```
await this.waitForAcceptCredential(record.id)
```


issueJsonLdCredential

Run the Demo
AnonCreds + Besu

DSR

1. Select credential type: Hyperledger *AnonCreds*
2. Create DID
3. Register Schema
4. Register Credential Definition
5. Establish connection
    1. Issuer (Faber): *Create connection invitation*
    2. Holder (Alice): *Receive connection invitation*
6. Offer credential
    1. Issuer (Faber): *Offer credential*
    2. Holder (Alice): *Accept credential*
7. Proof credential
    1. Issuer (Faber): *Request proof*
    2. Holder (Alice): *Accept Credential*

Run the Demo
W3C JSON-LD + Besu

DSR

1. Select credential type: W3C JSON-LD
2. Create DID
3. Register Schema
4. Establish connection
    1. Issuer (Faber): *Create connection invitation*
    2. Holder (Alice): *Receive connection invitation*
5. Offer credential
    1. Issuer (Faber): *Offer credential*
    2. Holder (Alice): *Accept credential*

www.dsr-corporation.com • www.dsr-iot.com

DSR

DOING SOFTWARE RIGHT

Full Stack Web | Embedded | Wireless

Analytics / Big Data | Scalable Databases | Digital Media

System Software | Mobile | Blockchain | CV / Machine Learning