A SURVEY ON SSI INTEROPERABILITY

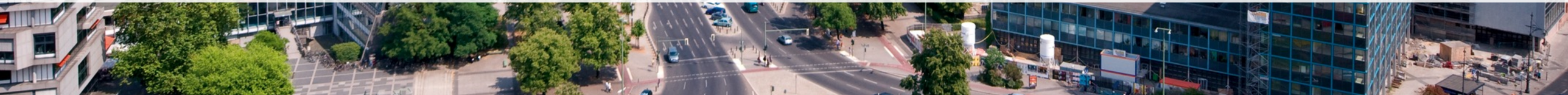
PREAMBLE, SSI INTEROPERABILITY, LAYERS OF INTEROPERABILITY, INTEROP EFFORTS, CONCLUSION

Hakan Yildiz | *Service-centric Networking*

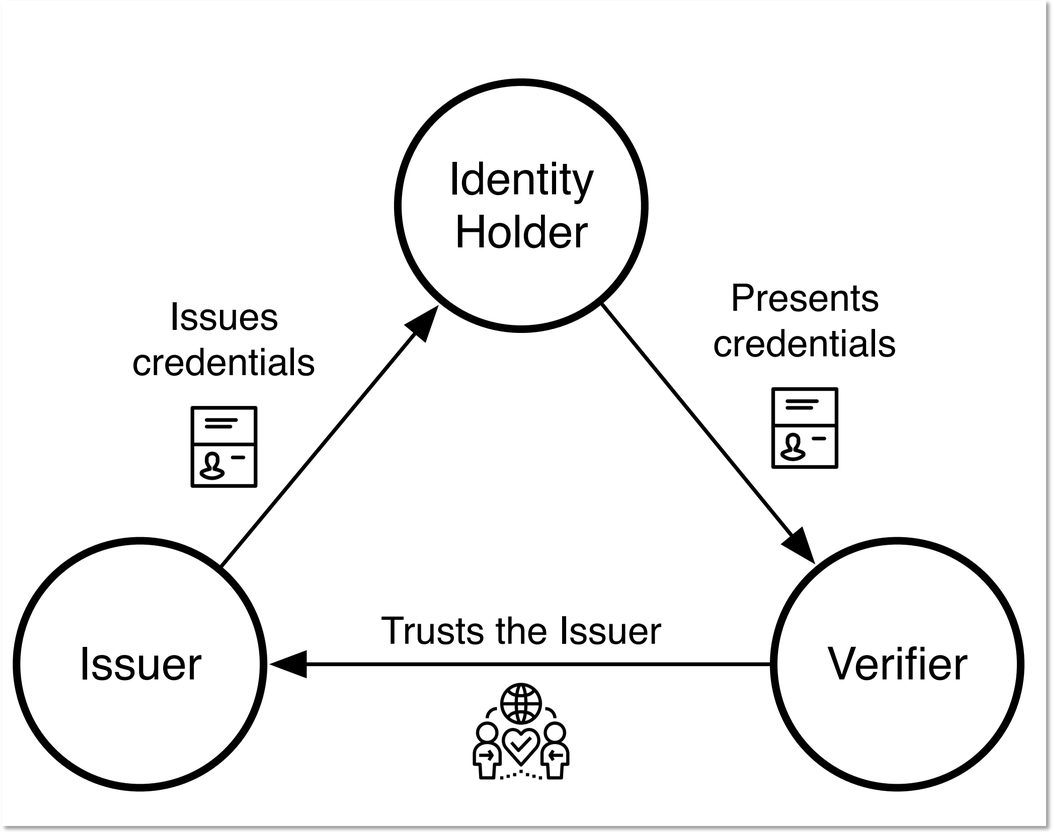
Technische Universität Berlin

SNET SERVICE-CENTRIC NETWORKING

T-LABS Innovative solutions for the future



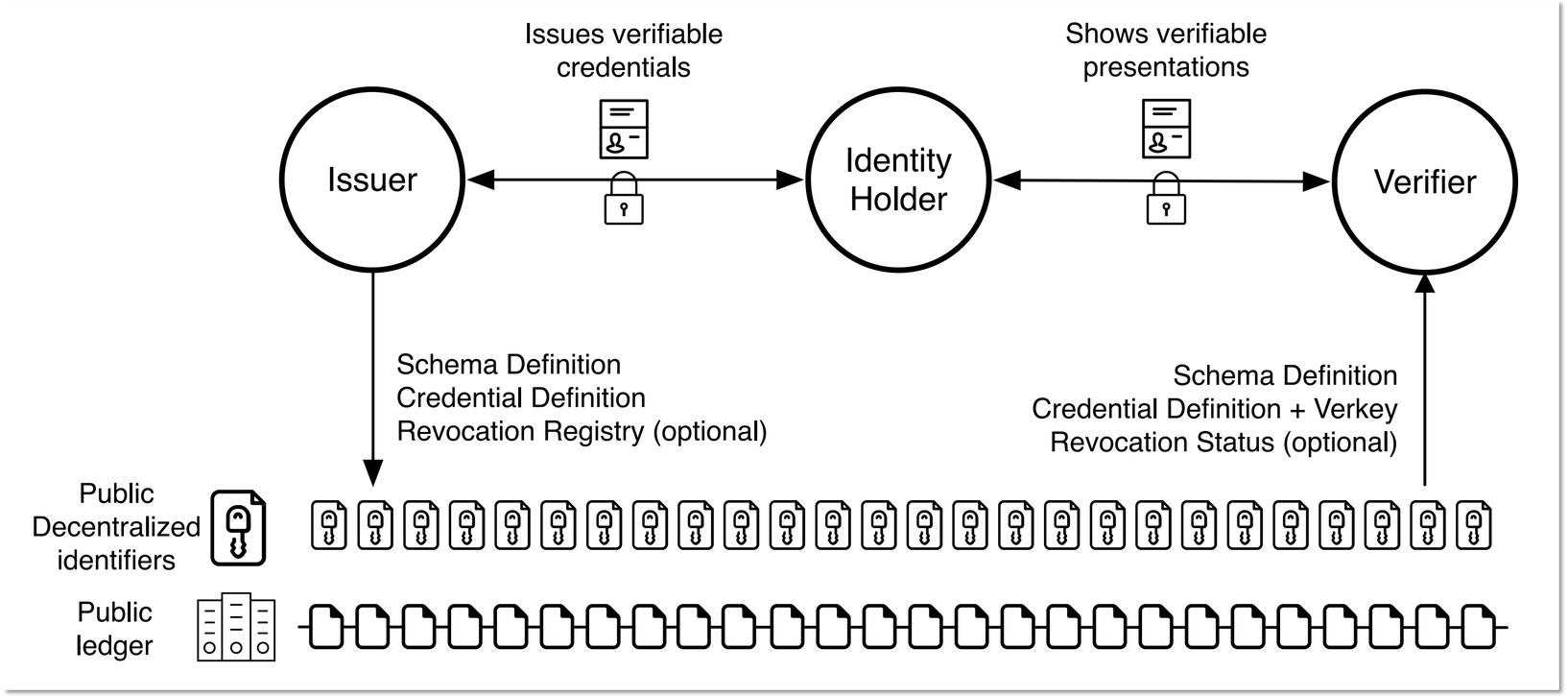
PREAMBLE
IDENTITY TRUST TRIANGLE



Roles

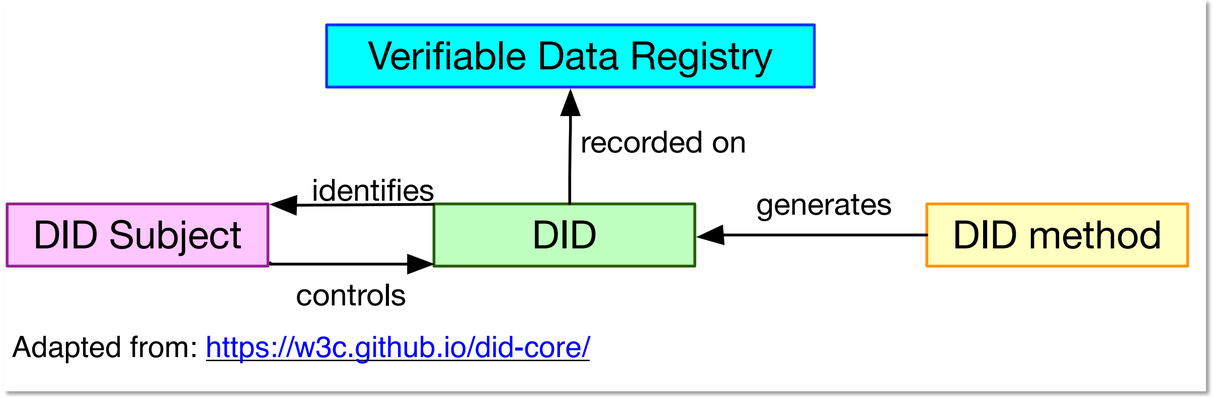
- Issuer, issuing credentials
- Identity holder, keeping credentials
- Verifier, validating credentials from identity holder
- Verifier trusts the issuer

PREAMBLE
IDENTITY TRUST TRIANGLE TRANSLATED TO SSI



- Based on Hyperledger Indy based SSI solution. Different flavors may differ. However, the result stays the same.
- We will discuss the differences in the following slides

PREAMBLE
DECENTRALIZED IDENTIFIER (DID)



Adapted from: https://w3c.github.io/did-core/

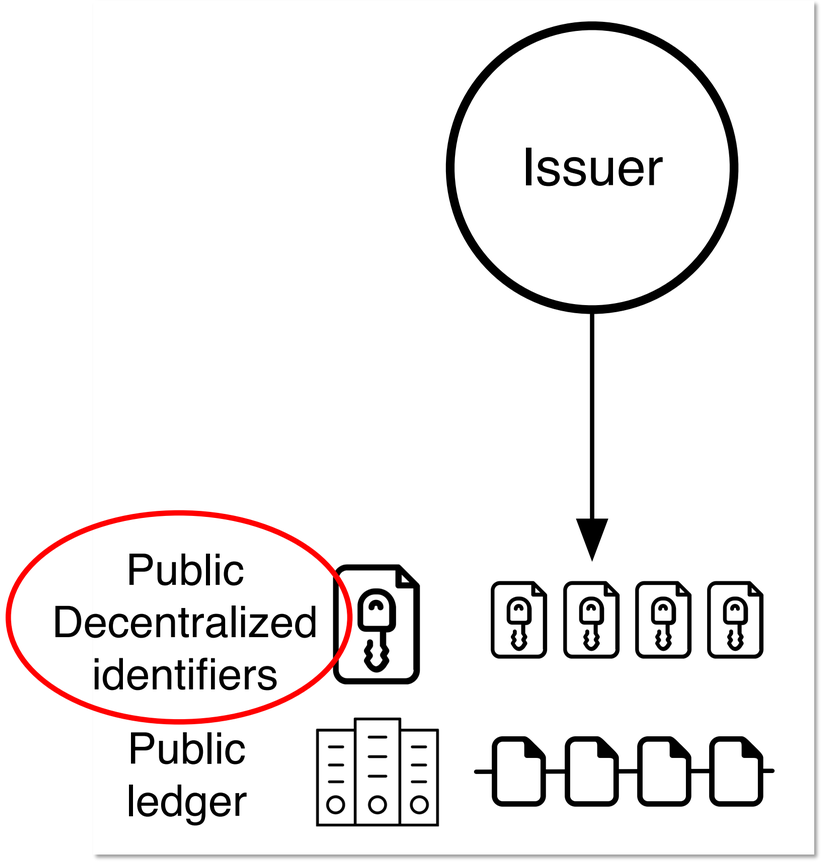
Description

Exists of three main parts:

- URI scheme identifier (did)
- Identifier from the DID method
- DID method-specific identifier (base58char)
- eg: **did:sov:idu:Cf1Y171S4uVtnjnYCSEFJM**

Properties

- Resolvable with high availability,
- E.g., stored in a Verifiable Data Registry
- Cryptographically verifiable
- Fully under control of the DID Subject
- Independent of any centralized registry, identity provider, or CA

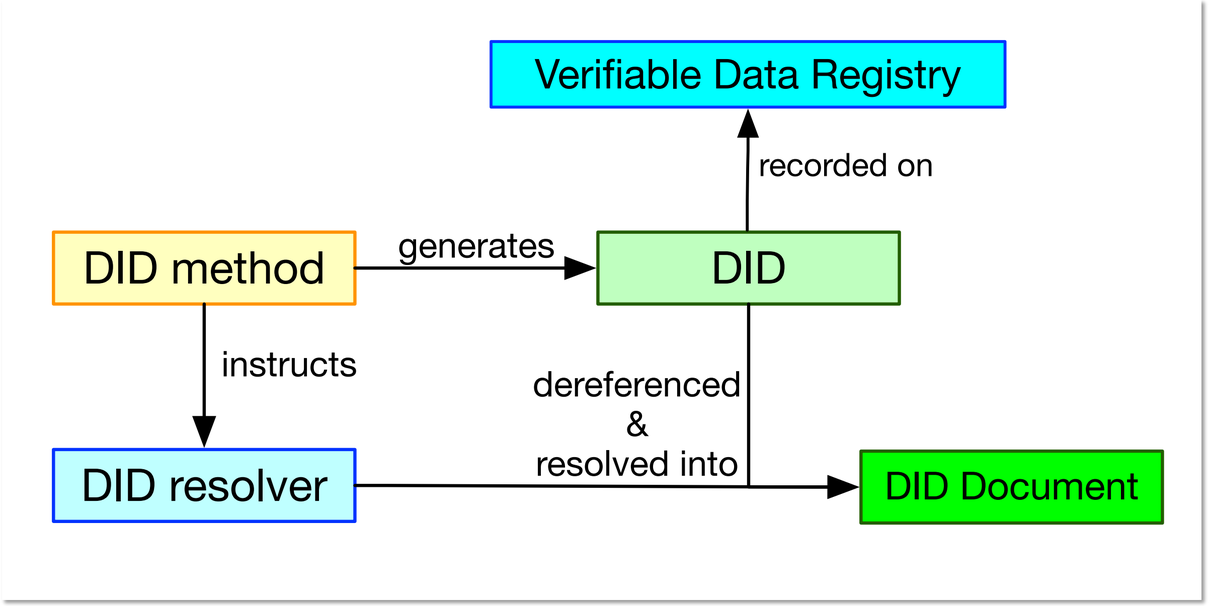


Example of the use of a public DID

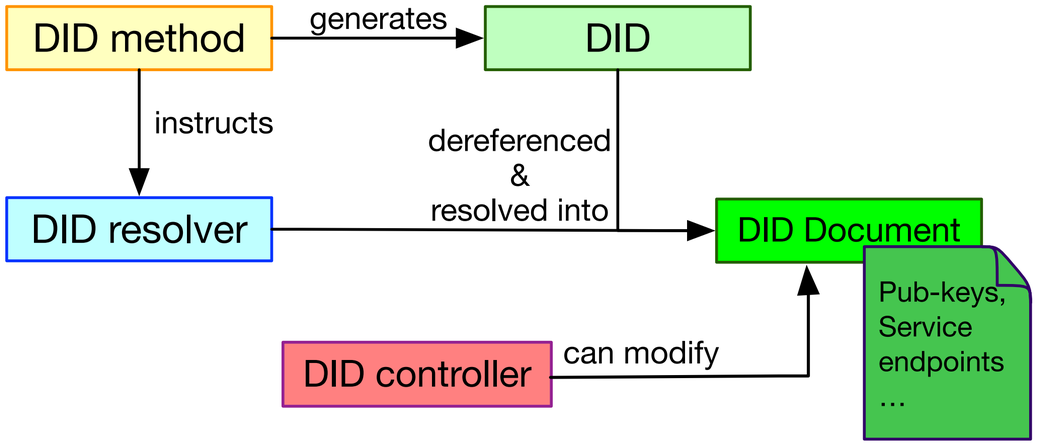
PREAMBLE
DID METHOD

Description

- Describes the functionality of DID with the storage it uses (Verifiable Data Registry)
- Specify a protocol to resolve a DID Document (DDO) from a DID
- DID method -specific operations are at least the following:
- Create
- Read (Resolve) DID resolver
- Update
- Delete (Revoke)



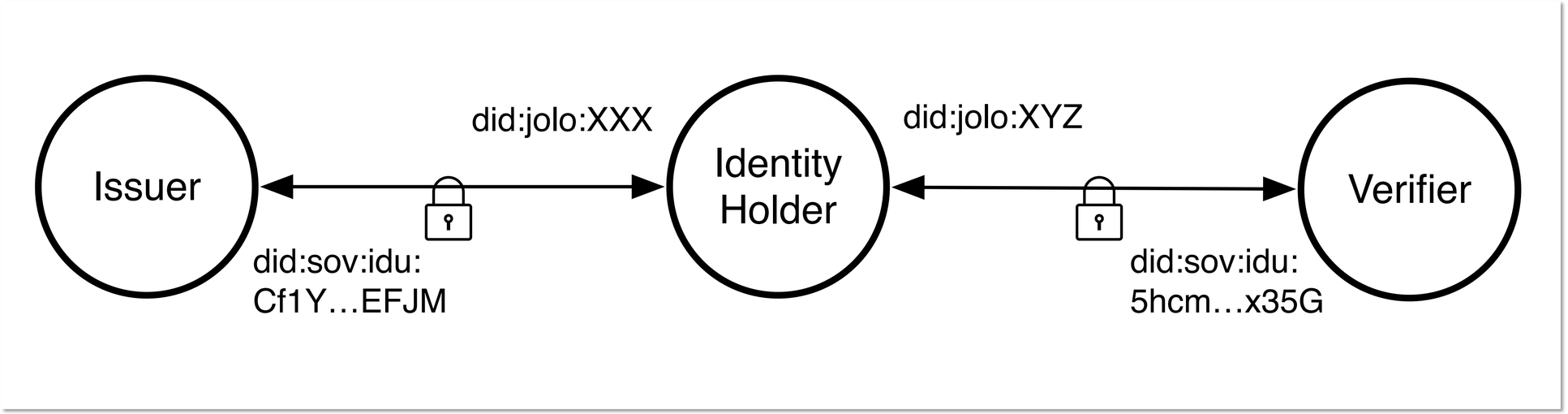
PREAMBLE
DID DOCUMENT (DDO)



DID Document (DDO)

- Resolves from DID with read operation
- 99% in JSON-LD Format
- Contains information about Auth
- Contains information about service
- Contains no personal or private information about the DID Subject
- Can be stored on-chain (HL Indy), off-chain (Jolocom) or partly on-/off-chain.
- A DID controller except DID subject CAN modify DID Document

PREAMBLE
PAIRWISE DIDS & DIDCOMM



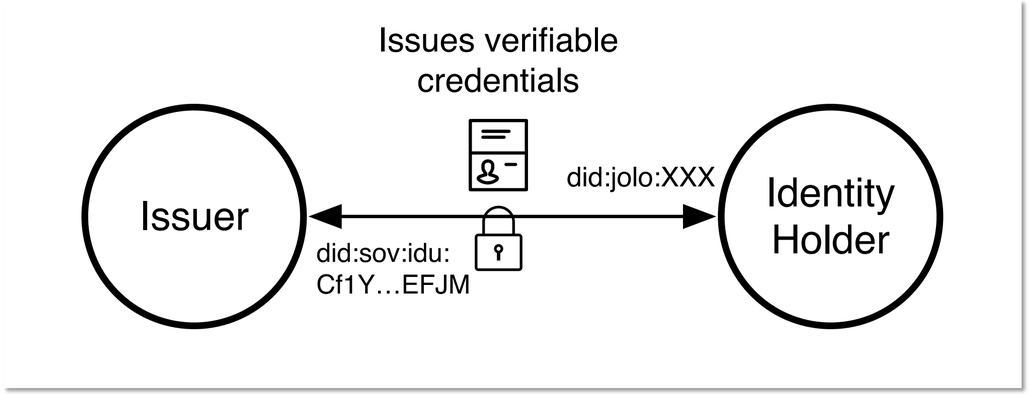
Pairwise DIDs

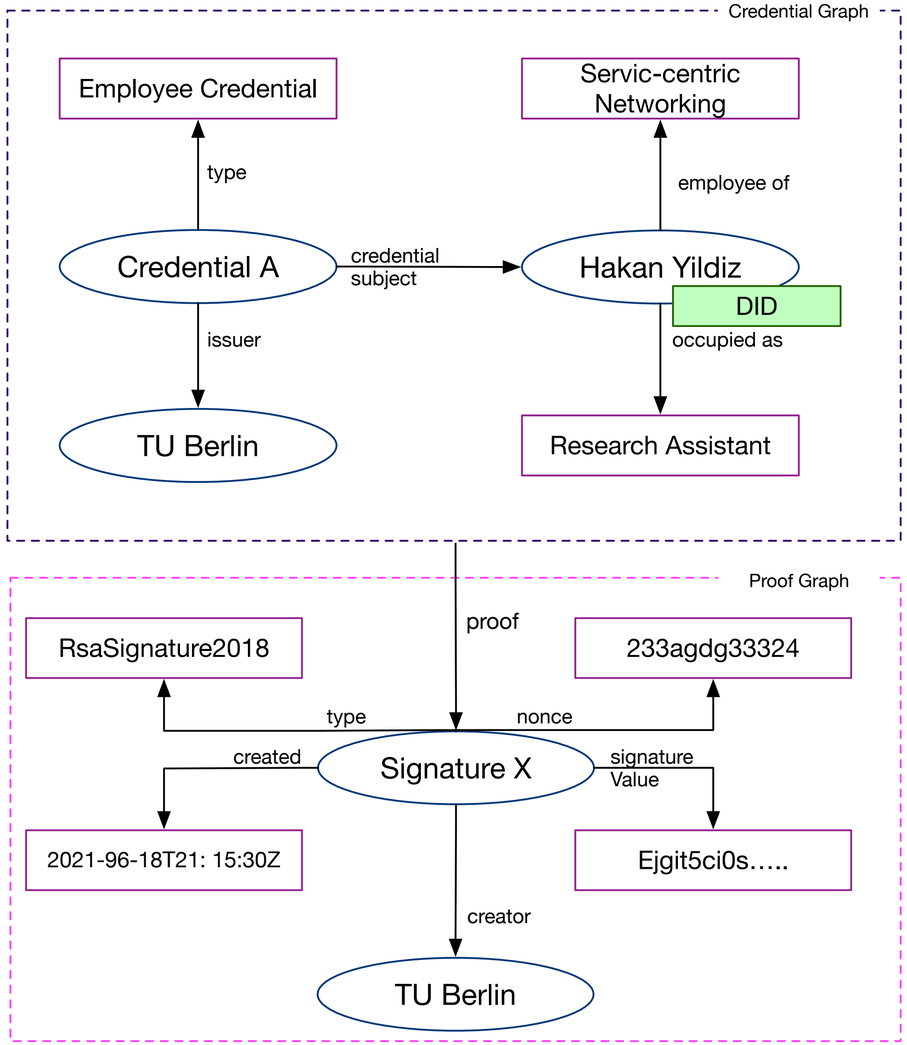
- One time use DID from Identity holder side
- DID communicated to issuer or verifier by the identity holder
- None of the DIDs from identity holder goes to the ledger
- Can be generated as many as needed

DIDComm

- Provides secure connection between parties
- Asymmetric messaging protocol which is E2E encrypted
- Can be used with any transport protocol (http(s), Bluetooth, NFC, QR codes etc.)

PREAMBLE
VERIFIABLE CREDENTIALS



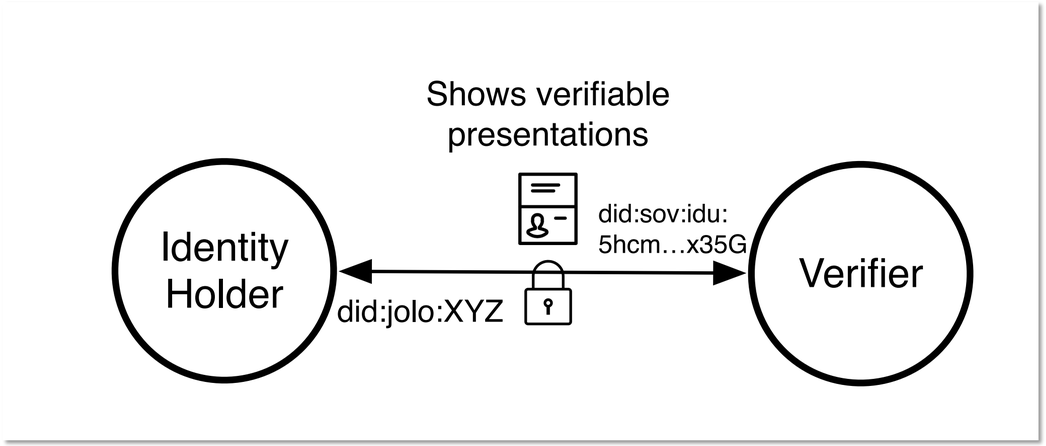


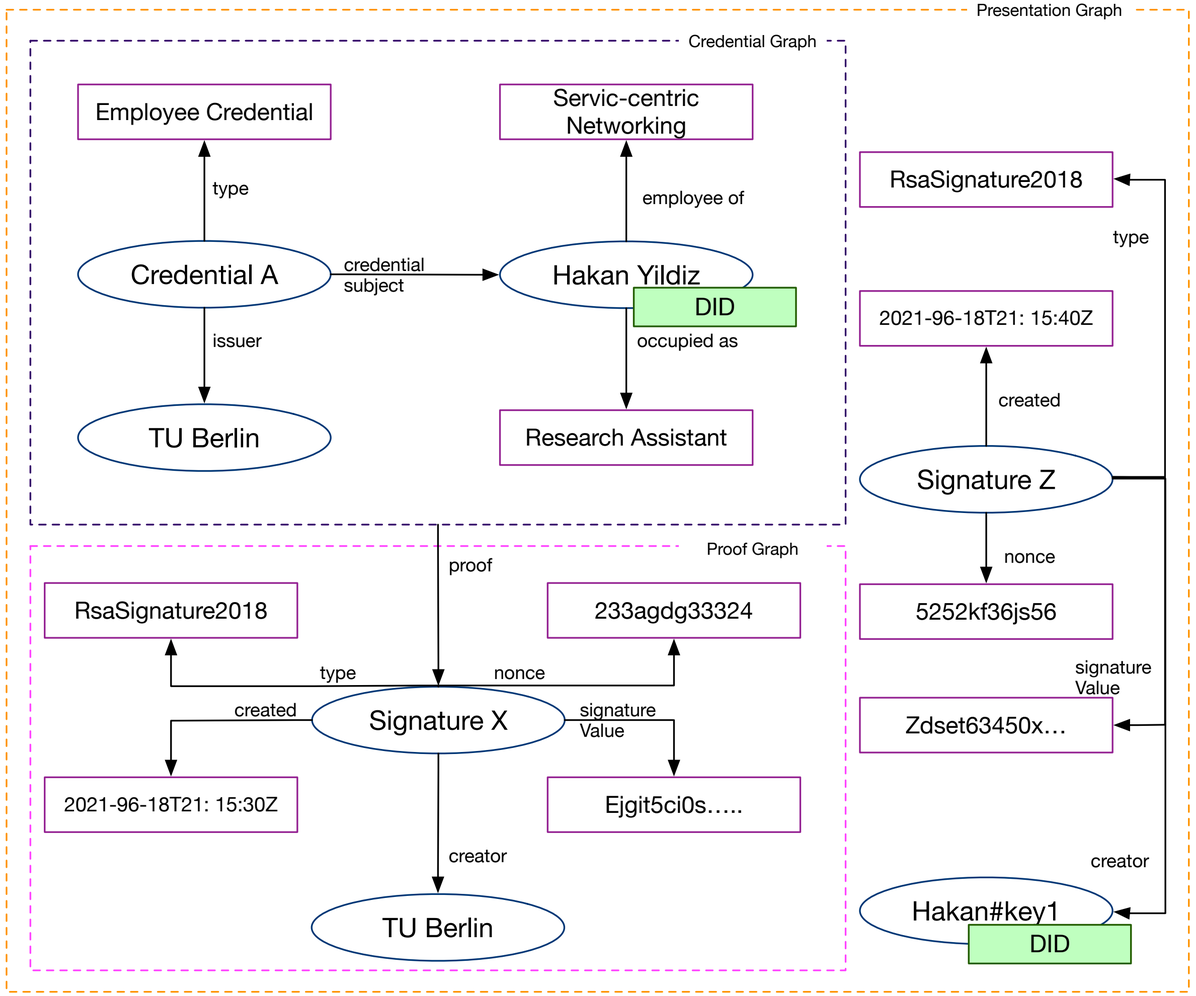
Description

- A set of claims,
- That are tamper-resistant,
- whose origins are cryptographically verifiable
- Has four different flavors
- Credential binding comes also in two flavors

Adapted from: https://w3c.github.io/vc-data-model/

PREAMBLE
VERIFIABLE PRESENTATIONS





Description

- Contains partial or complete information about the verifiable credential
- with additional proof that:
- the presented credential is really bound to the identity holder
- with or without showing to what the credentials are bound to

PREAMBLE
TYPES OF INTEROPERABILITY

SSI Interoperability

Scope of TU Berlin - SNET / This Presentation

Scope of TUB - SNET

Technical Interoperability

Ability to communicate between agents (issuer, wallet, verifier) and exchange information between various SSI solutions

Semantic Interoperability

Ability to understand the exchanged information betwen agents (issuer, wallet, verifier)

Economic Interoprability

Ability to create value chain between different SSI ecosystems

Legal Interoperability

Ability to choose to communicate and exchange information between different SSI ecosystems based on their legal compliance

Interoperability with existing IAM Solutions

Ability to use SSI based credentials for authentication while keeping existing federated IAM workflows

SSI INTEROPERABILITY
WHY IS THIS EVEN AN ISSUE?

Technische Universität Berlin

Standardization as a Guideline

- DID, DIDComm, VC all standardized
- DID → Great standardization
- DIDComm → Flavors of DIDComm are incompatible with each other
- VC → Flavors of VCs are incompatible with each other

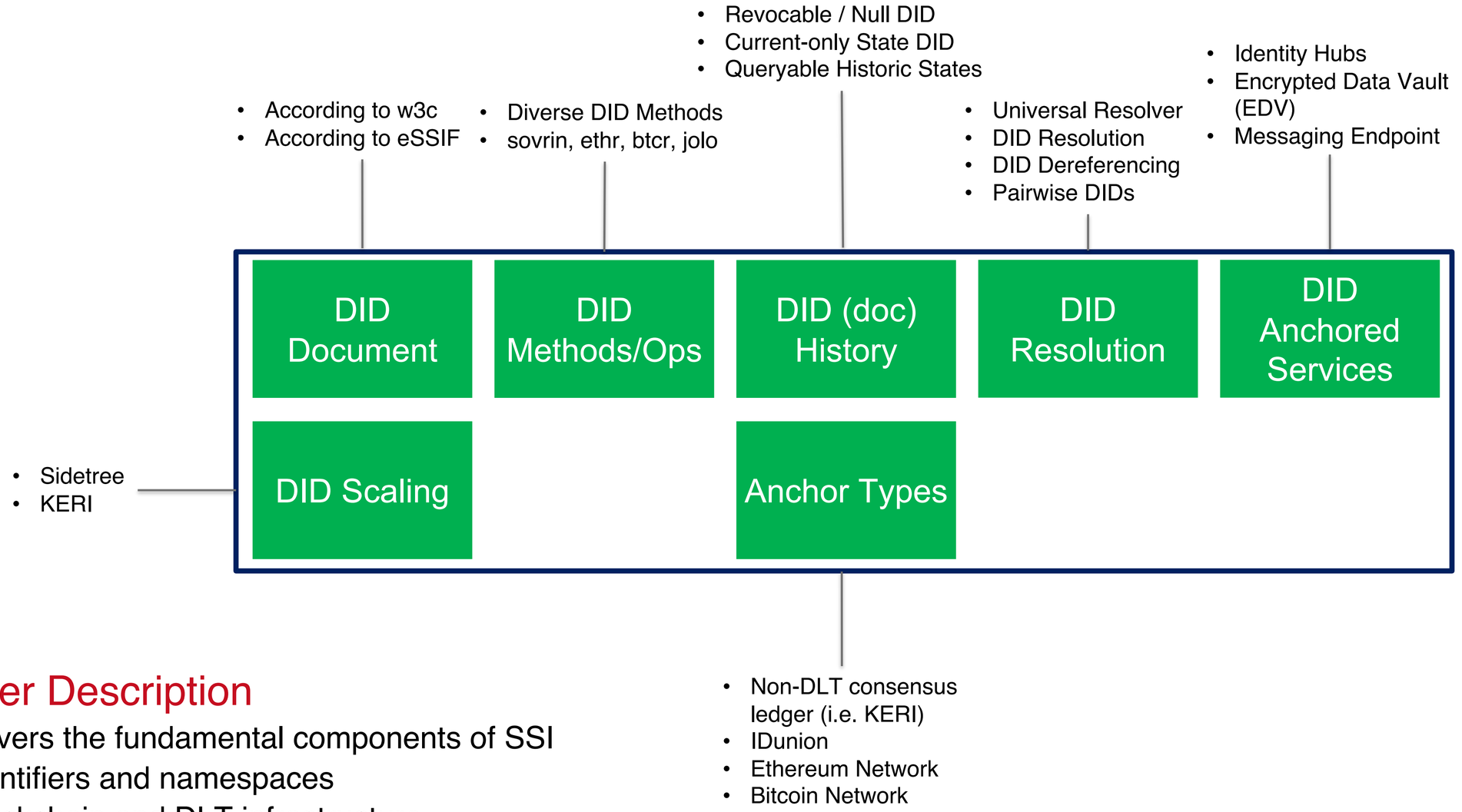
Semantics

- There are semantic guidelines
- VCs don’t have to follow these guidelines
- Understanding of claims are sometimes an issue
- Linked Data is useful, but it’s not used in every type of credentials

Components

- To create an SSI solution and infrastructure, there are many components to take upon
- Some of these components are not compatible with each other
- Components are categorized in 5 Layers
- L1: Public Trust Layer
- L2: Agent Layer
- L3: Credential Layer
- L4: Application Layer (Use case Layer)
- L5: Vertical (Cross-cutting) Layer

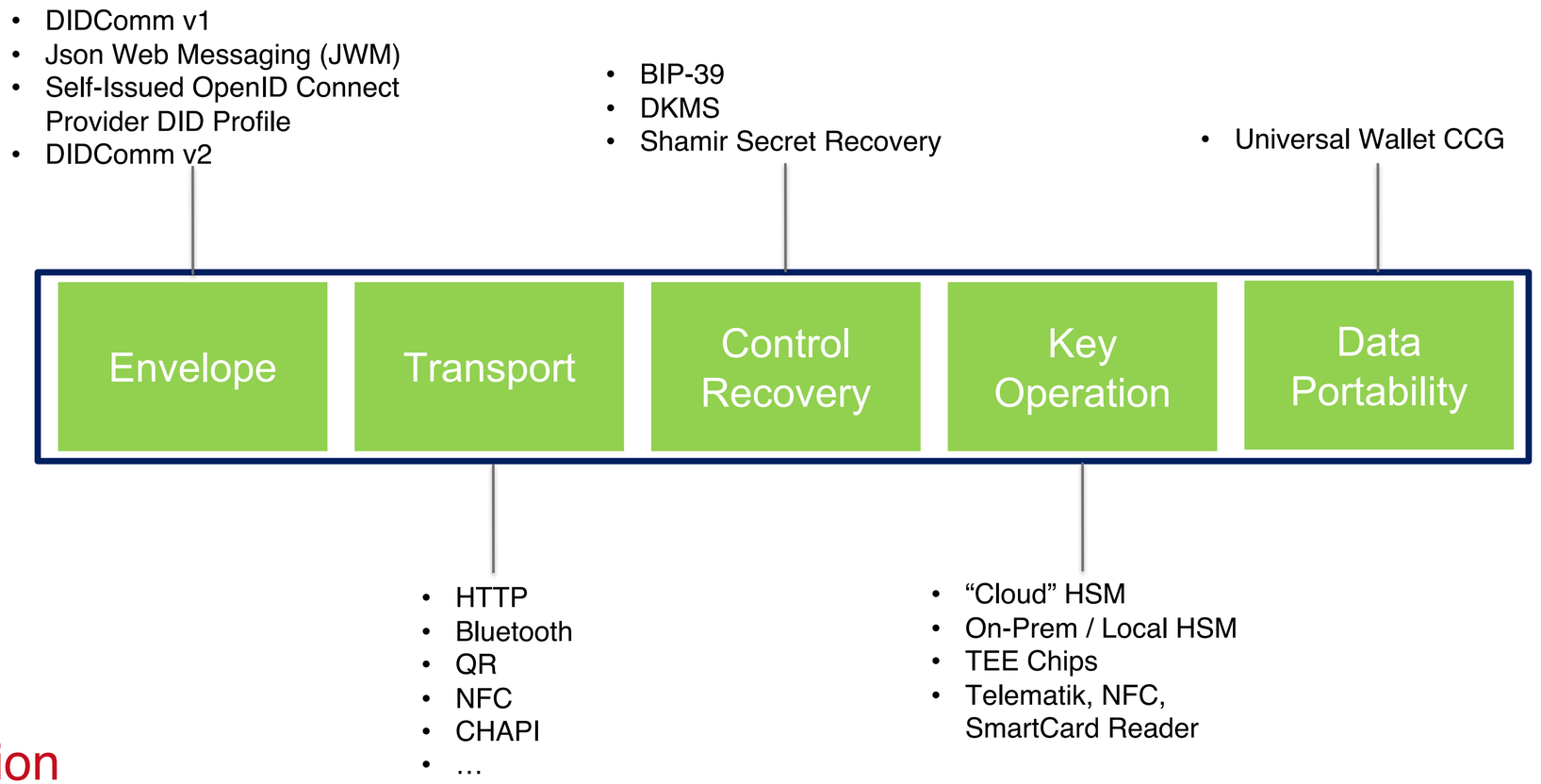
LAYERS OF INTEROPERABILITY
PUBLIC TRUST LAYER



Layer Description

- Covers the fundamental components of SSI
- Identifiers and namespaces
- Blockchain and DLT infrastructure

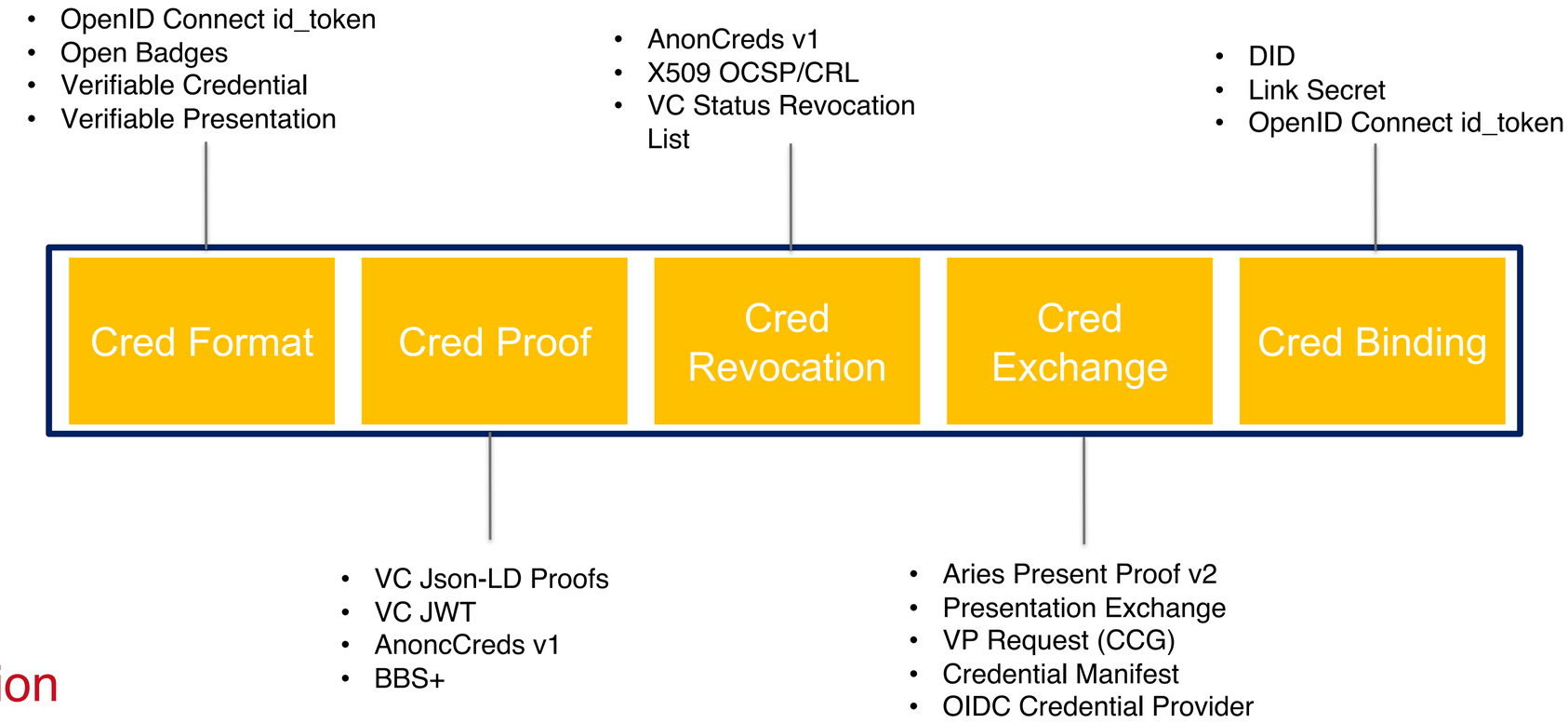
LAYERS OF INTEROPERABILITY
AGENT LAYER



Layer Description

- Agent Communication
- Control Recovery
- Personal Data Storage

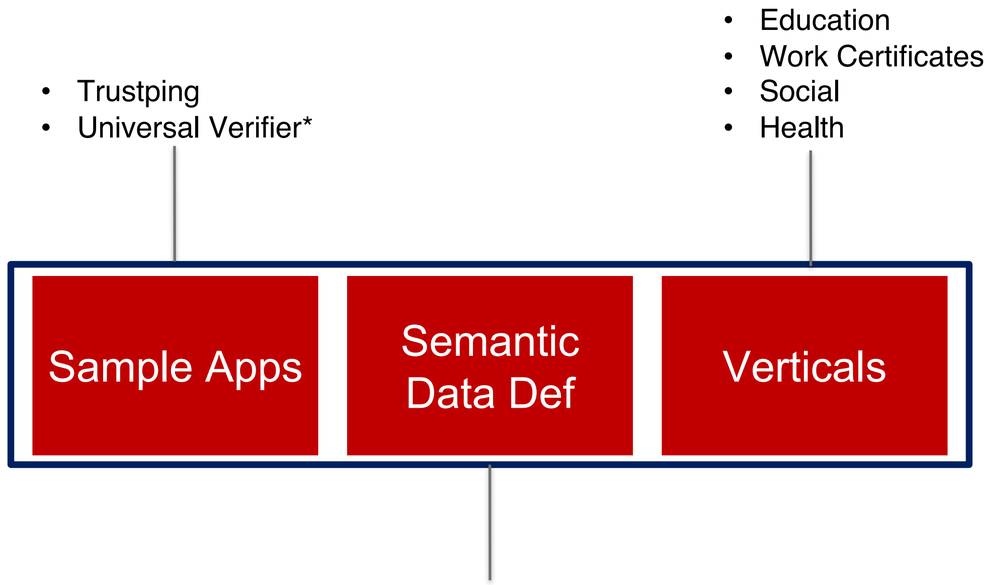
LAYERS OF INTEROPERABILITY
CREDENTIAL LAYER



Layer Description

- Covers the identity related information
- Verifiable Credential
- Verifiable Presentation
- Credential Revocation

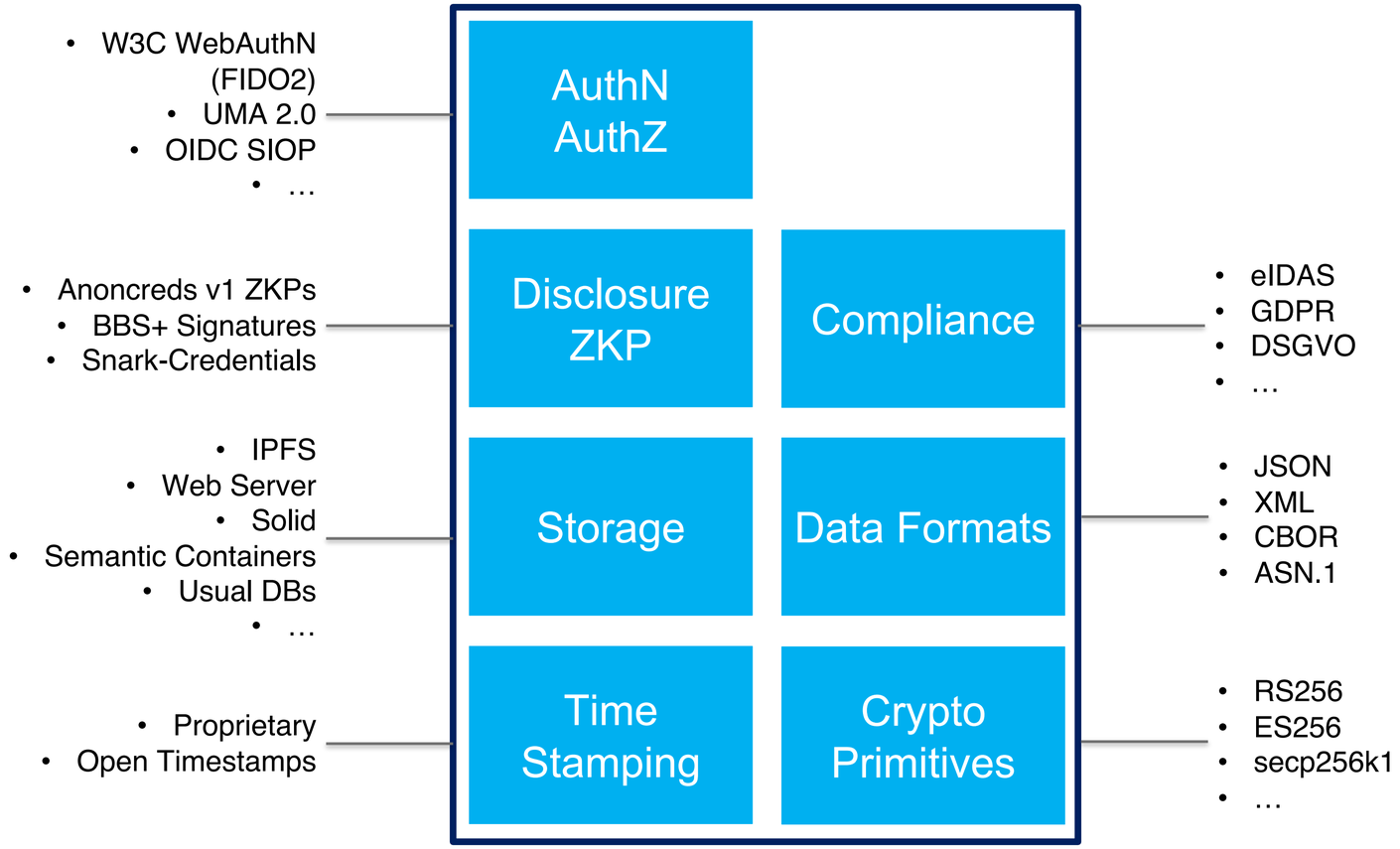
LAYERS OF INTEROPERABILITY
APPLICATION LAYER (USE CASE)



Layer Description

- Highest Layer
- Contains the use case areas
- Necessary semantic data definitions for understanding credentials from usecases coming from different ecosystems

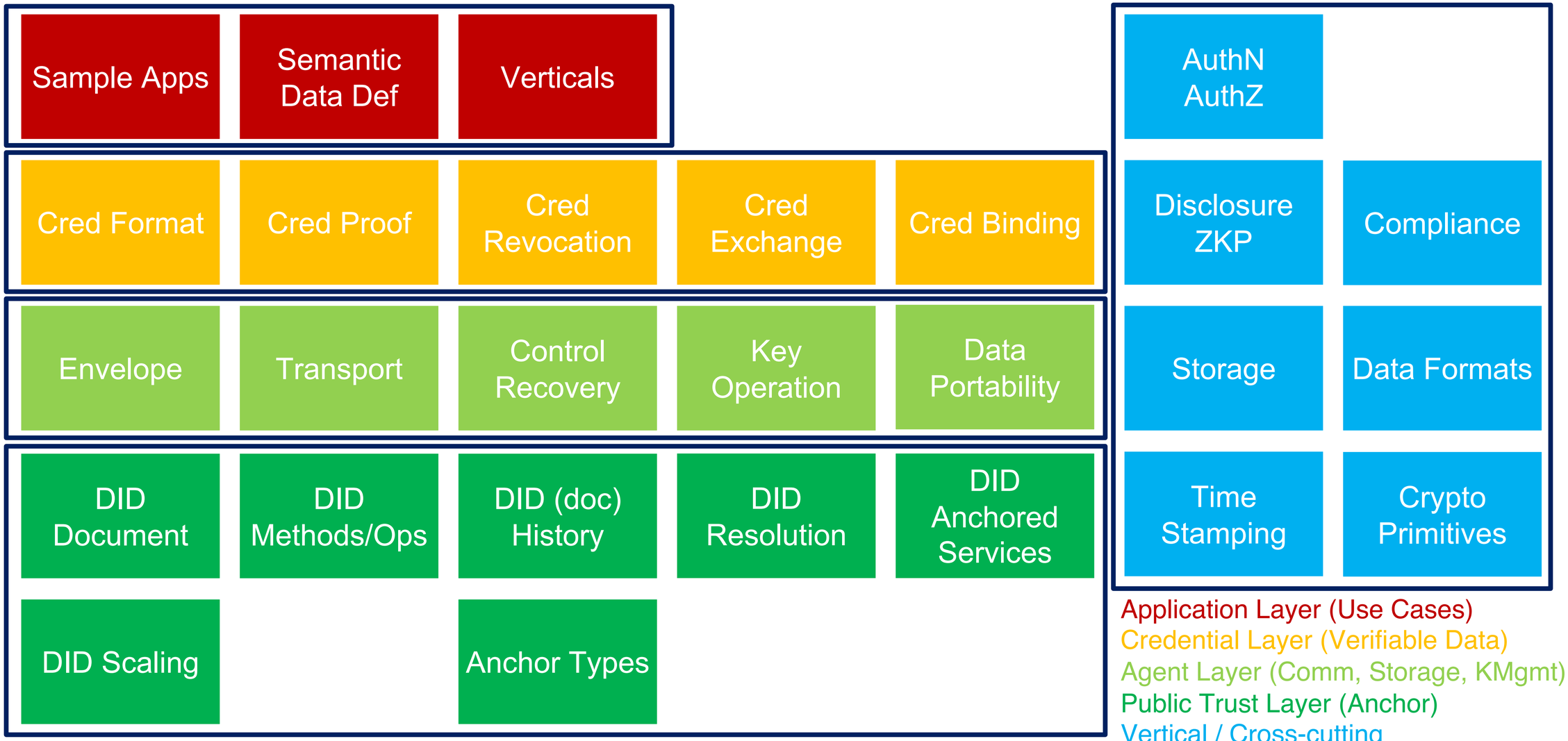
LAYERS OF SSI INTEROPERABILITY
VERTICAL / CROSS-CUTTING LAYER



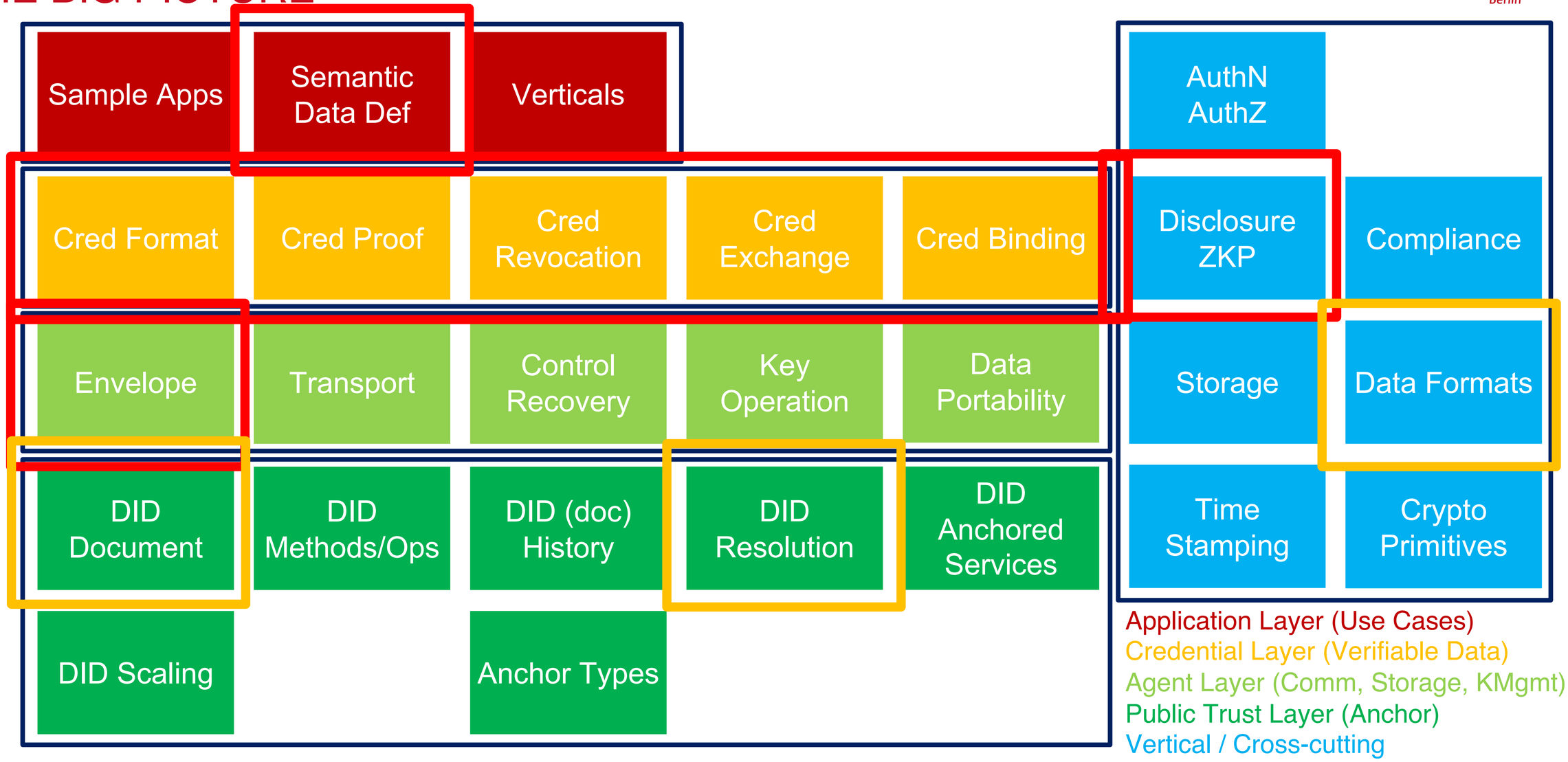
Layer Description

- The layer that has dependencies with all other four layers

LAYERS OF SSI INTEROPERABILITY
THE BIG PICTURE



LAYERS OF SSI INTEROPERABILITY
THE BIG PICTURE



INTEROPERABILITY
LOW LEVEL INCOMPATIBILITY

Technische Universität Berlin

DID Document

- According to w3c
- According to eSSIF

Issue

- There are two different DDO Specification
- The one from eSSIF needs to align with the larger standardization
- Or incompatibility between eSSIF and the rest of the world

DID Resolution

- Universal Resolver
- DID Resolution
- DID Dereferencing
- Pairwise DIDs

Issue

- There are hundreds of DID methods and resolvers as of now
- DID itself is not capable of showing where it is stored.
- For resolving DID, DID method is needed (location)
- DIF Universal Resolver solves this

INTEROPERABILITY
HIGH LEVEL INCOMPATIBILITY

Technische Universität Berlin

Envelope

- DIDComm v1
- Json Web Messaging (JWM)
- Self-Issued OpenID Connect Provider DID Profile
- DIDComm v2

Semantic Data Def

- Activity Pub
- schema.org
- Open Badges
- Learner Record
- FHIR
- Europass Learning Model

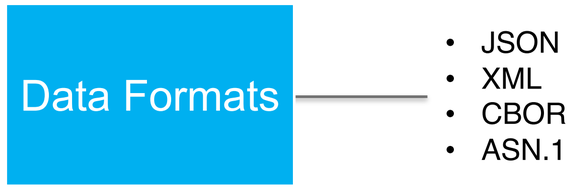
Issue

- None of these above can communicate with each other
- DIDComm v2 is still in draft and will have some level of compatibility
- Agent to agent communication supporting different envelopes is not possible
- Moving everything to one envelope would help.

Issue

- There are many ways to ensure semantic interoperability by using external sources like schema.org
- There is no way to enforce it while creating a schema or issue a VC
- JSON-LD Format is significantly better
- JSON not so much

INTEROPERABILITY
HIGH LEVEL INCOMPATIBILITY



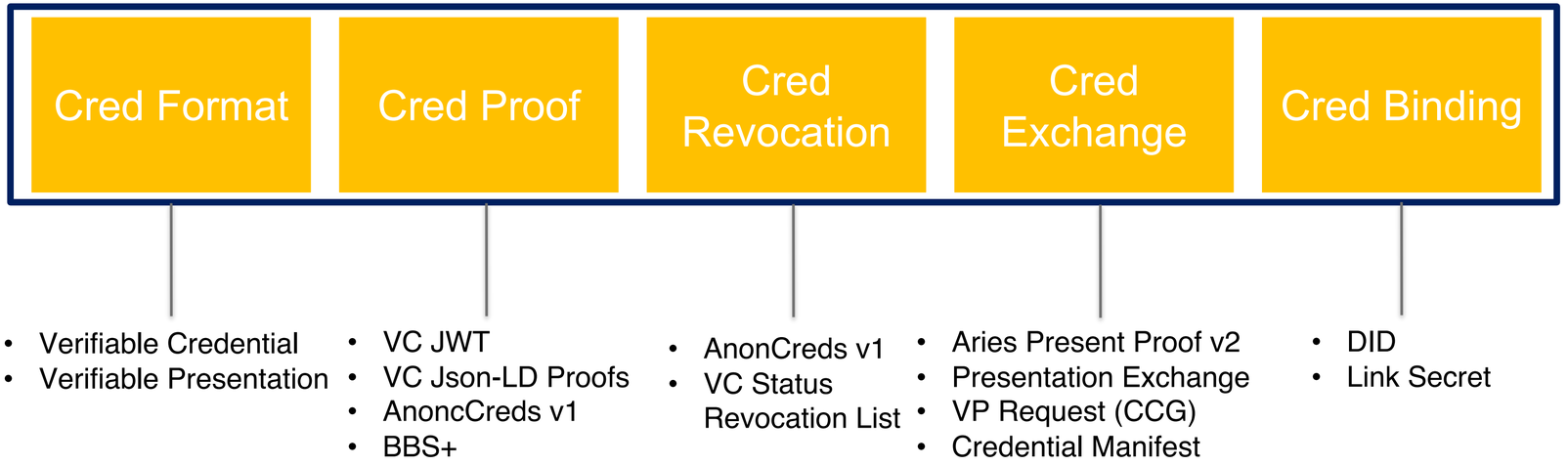
Issue

- Different ZKP based credentials → Different signatures
- Verification of these signatures are also different
- None of them are compatible with each other and they won't be in the future as well.

Issue

- Occurance: JSON >>>> XML > CBOR > ASN.1
- This compatibility can be solved by implementing libraries or compilers

INTEROPERABILITY
HIGH LEVEL INCOMPATIBILITY



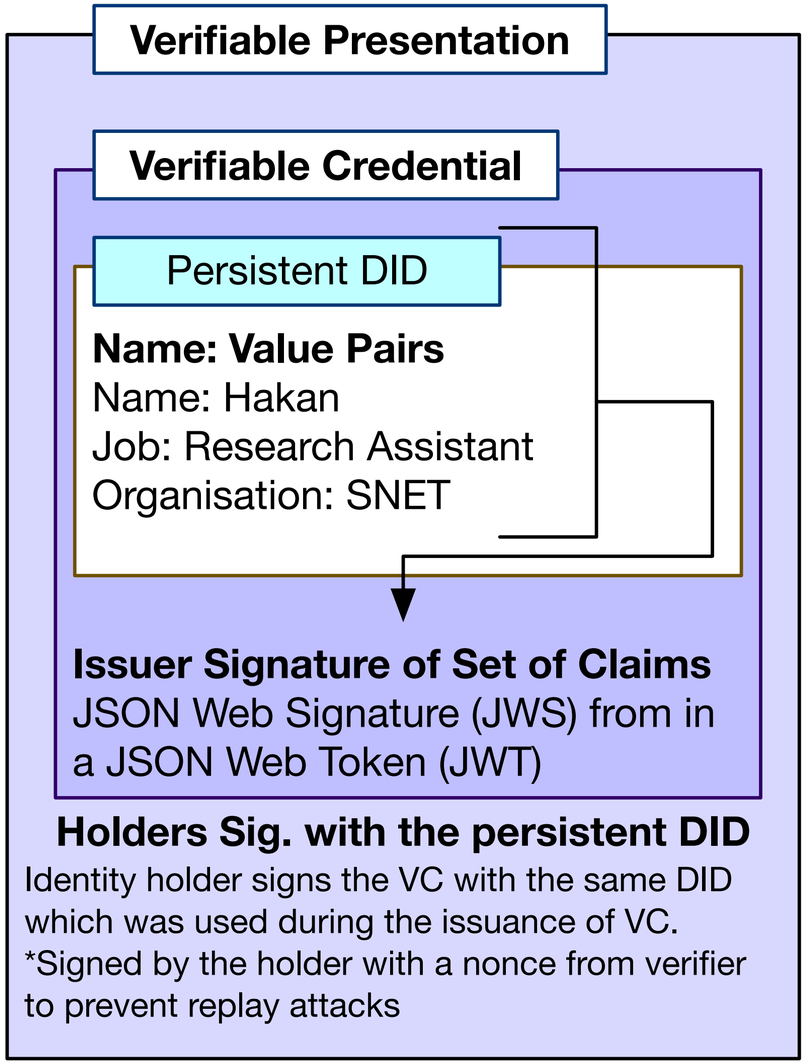
Issue

- There are four different flavors of Verifiable Credentials
- They come with four different credential proof types
- JWT Signatures
- LD Signatures
- CL Signatures
- BBS+ Signatures

Issue – Continued

- Revocation type depends on the credential proof types
- Same as Credential Exchange protocol
- And credential binding type
- Last type of credentials are actually a solution to the incompatibility of the first three credential proof types

FLAVORS OF CREDENTIALS
JSON - JWT

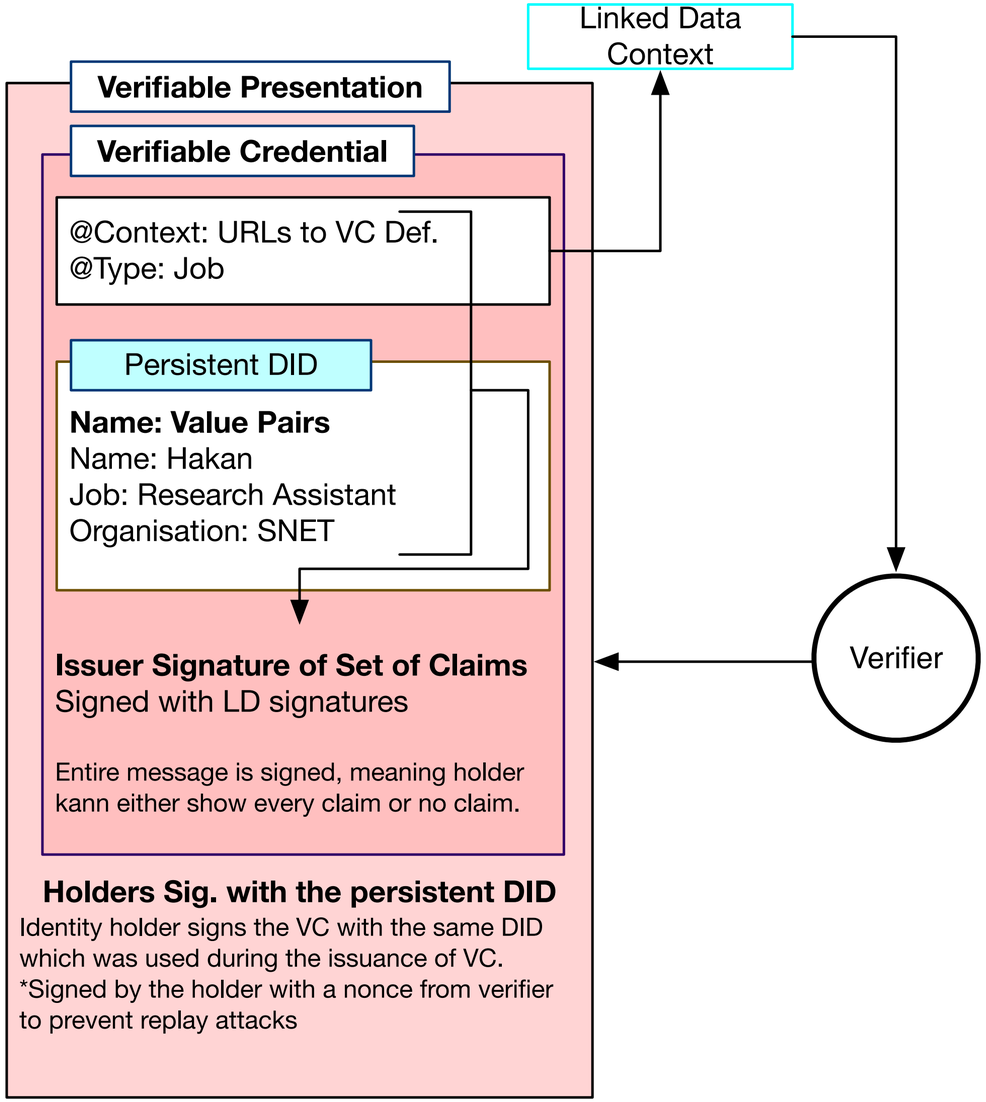


Properties

- **Simplicity:** Simplest
- **Privacy preserving:** No
- **Selective Disclosure:** No
- **Zero Knowledge Proof:** No
- **Need to reveal persistent identifier:** Yes
- **Semantic Disambiguation:** No
- **Examples:** MSFT ION

Adapted from Kaliya Young's work: https://www.lfph.io/wp-content/uploads/2021/04/Verifiable-Credentials-Flavors-Explained-Infographic.pdf

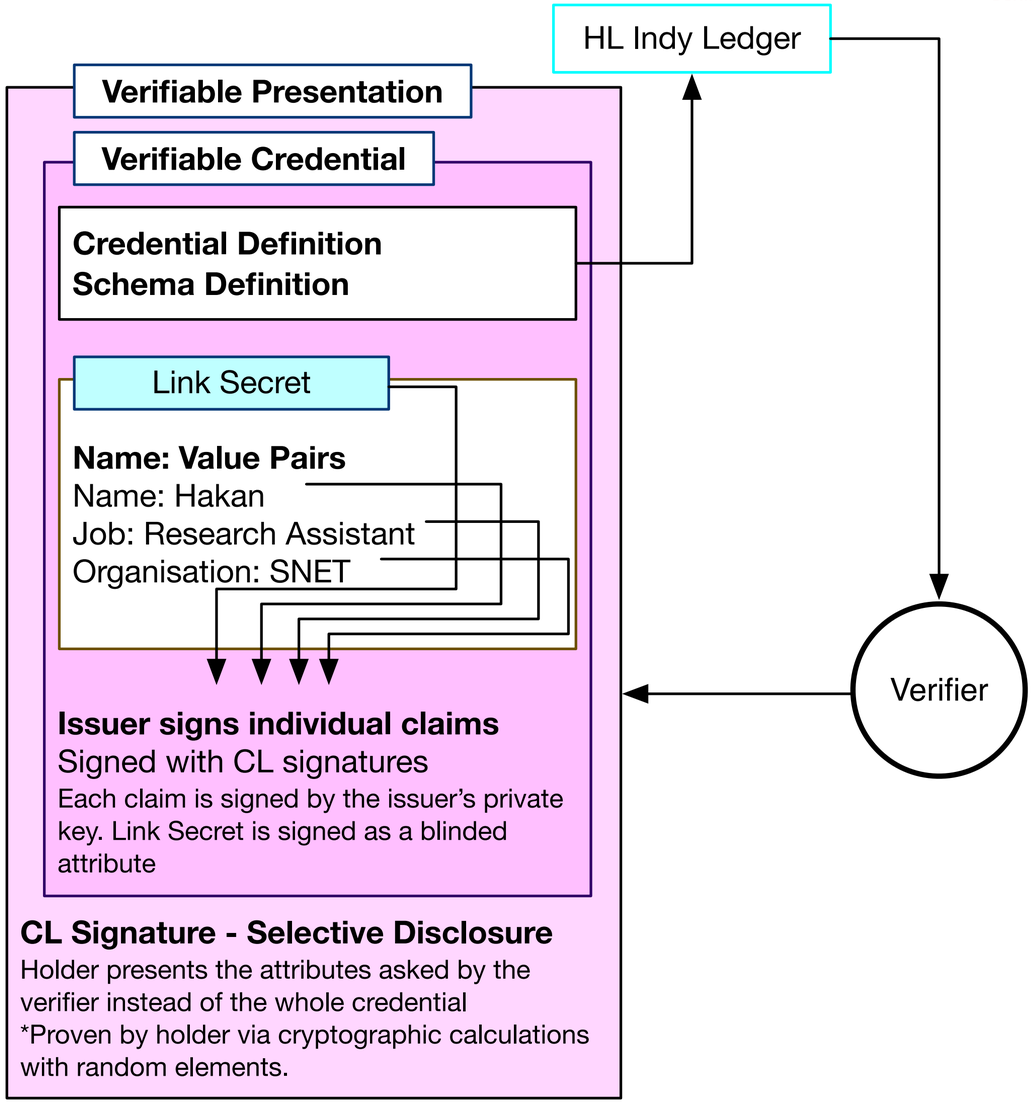
FLAVORS OF CREDENTIALS
JSON-LD WITH LD SIGNATURES



Properties

- **Simplicity:** Simple-ish
- **Privacy preserving:** No
- **Selective Disclosure:** No
- **Zero Knowledge Proof:** No
- **Need to reveal persistent identifier:** Yes
- **Semantic Disambiguation:** Yes
- **Examples:** Jolocom

FLAVORS OF CREDENTIALS
ANONCRED ZKP WITH CL-SIGNATURES

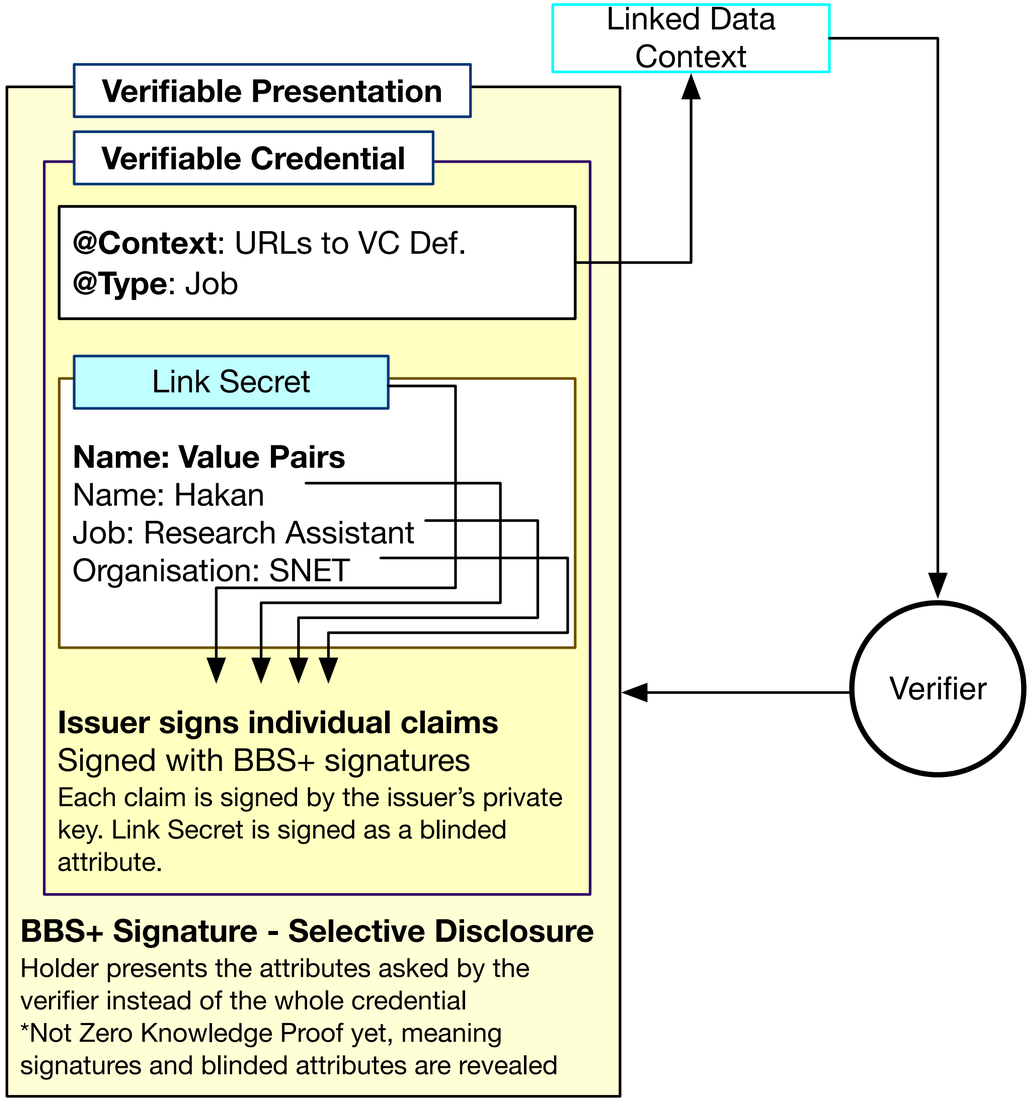


Properties

- **Simplicity:** It's complicated…
- **Privacy preserving:** Yes
- **Selective Disclosure:** Yes
- **Zero Knowledge Proof:** Yes
- **Need to reveal persistent identifier:** No
- **Semantic Disambiguation:** No
- **Examples:** Hyperledger Indy / Aries

FLAVORS OF CREDENTIALS: INTEROP EFFORTS

JSON-LD ZKP WITH BBS+ SIGNATURES



Properties

- **Simplicity:** Complex
- **Privacy preserving:** Yes
- **Selective Disclosure:** Yes
- **Zero Knowledge Proof:** Not yet
- **Need to reveal persistent identifier:** No
- **Semantic Disambiguation:** Yes
- **Examples:** Hyperledger Aries (Ongoing)

INTEROP EFFORTS
HYPERLEDGER ARIES

Technische Universität Berlin

Aries Cloud Agent (ACA-Py)
- Meant to operate on cloud agents (issuer / verifier)
- Native libraries for Hyperledger Indy connection
- Uses DIDComm messaging and Aries protocols

Aries Framework JavaScript (AFJ)
- Framework for communication between SSI Agents (issuer / wallet / verifier)
- “Interoperable” with other RFCs (ACA-Py, .NET etc.)
- Uses JavaScript wrapper, still in early development

Aries Framework .NET (AF-.NET)
- Implementation for using Aries protocols
- Used for building SSI application for cloud, mobile and IoT stack
- Also uses DIDComm messaging and Aries protocols
- Uses .NET libraries

Aries Framework Go (AF-Go)
- Most interoperable framework of all, for all agents
- DLT agnostic
- Uses DIDComm messaging
- Incomplete…

INTEROP EFFORTS
ARIES INTEROPERABILITY

Technische Universität Berlin

Latest Interoperability Results

Test Agent	Scope	Exceptions	ACA-Py	AF-Go	AFJ	AF-.NET
ACA-Py	AIP 1, 2	None	**118 / 139 84%**	0 / 10 0%	37 / 39 94%	44 / 54 81%
AF-Go	AIP 2	None	0 / 10 0%	**3 / 13 23%**	0 / 0 0%	0 / 0 0%
AFJ	AIP 1	Revocation	37 / 39 94%	0 / 0 0%	**73 / 83 87%**	30 / 39 76%
AF-.NET	AIP 1	Proof Proposal	44 / 54 81%	0 / 0 0%	30 / 39 76%	**76 / 93 81%**

Source: https://aries-interop.info/

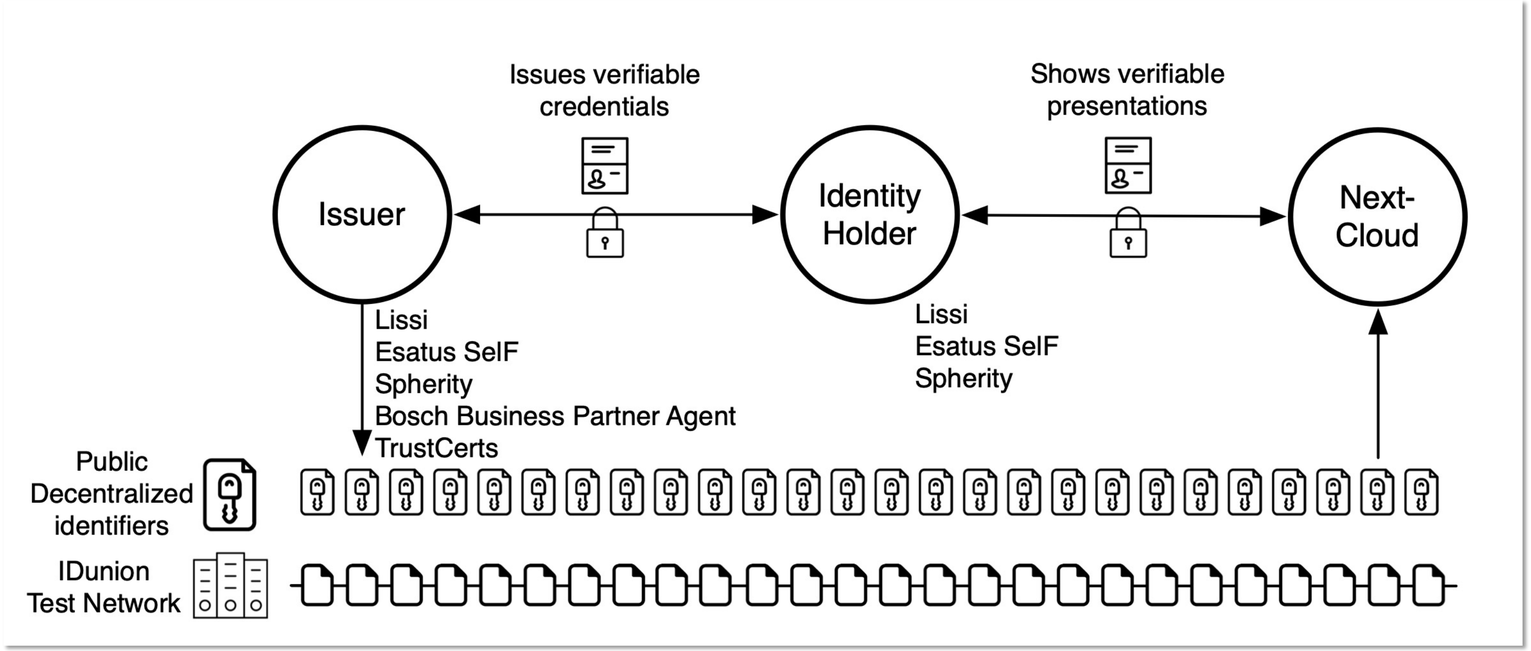
Aries Agent Test Harness (AATH)

- Test execution engine
- Set of tests for evaluating interoperability between Aries Agents and Aries Frameworks
- Results are collected and put under Aries interoperability test suite

Aries Interoperability Test Suite

- Test suite to create interoperability within different Aries RFCs
- But also between Aries RFCs
- Based on Aries Interop Profile 1 or AIP2

INTEROP EFFORTS
INTEROP PLUG FEST



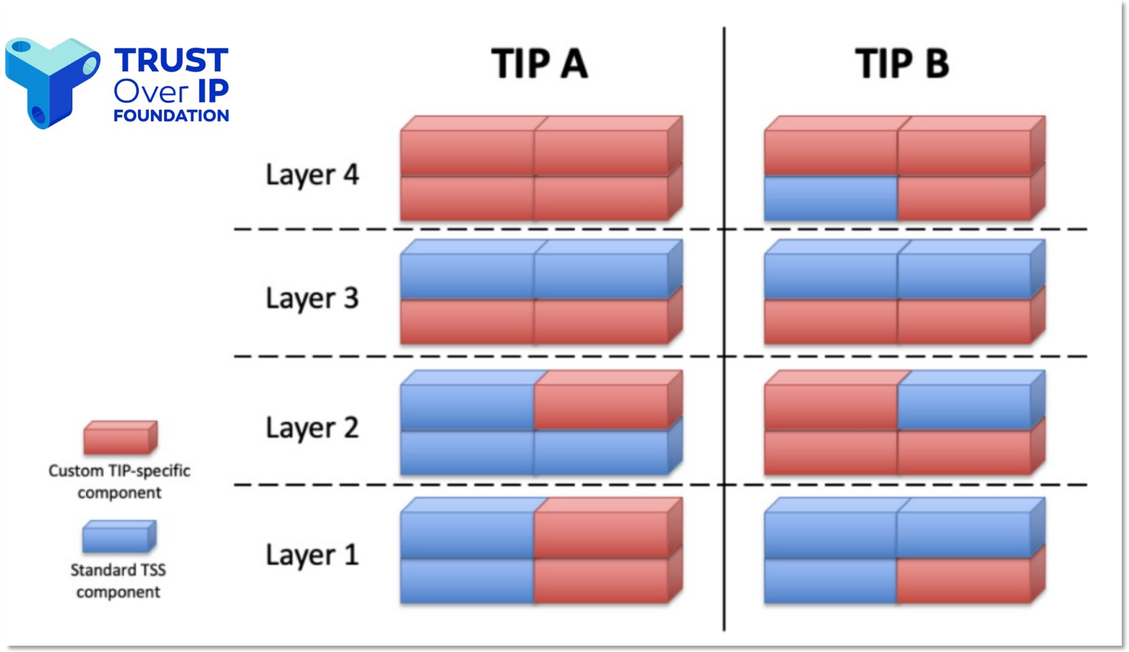
Description

- A couple of SSI solutions come together to work on a pre-defined use-case
- Issuer / Holder / Verifier agents from different solutions try to communicate with each other
- If there are any incompatibilities, they are recorded, and solution providers work on solving those incompatibilities

IDunion Use-Case

- Nextcloud Login via SSI-based credentials
- Lissi, Esatus SelF, Bosch Business Agent, TrustCerts and Spherity as SSI solution provider
- All five solutions will test their compatibility based on the use-case mentioned above

INTEROP EFFORTS
SATURN V

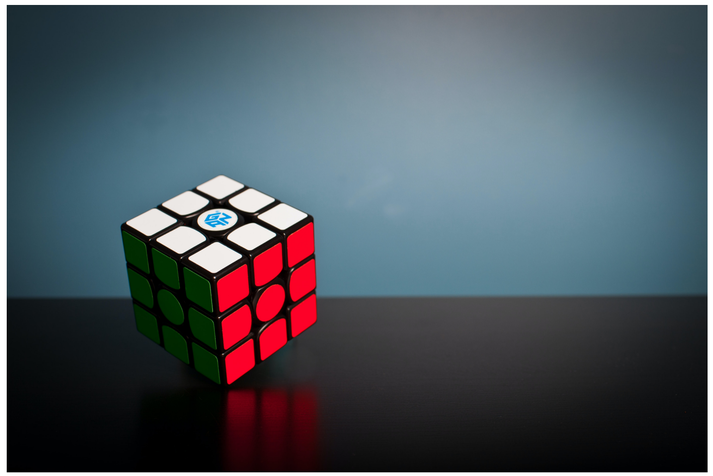


Description

- An interop approach initiated by Trust over IP
- A ToIP Interop Profile (TIP) is being used for testing (similar to Aries Interop Profiles)
- In theory based on a use case, similar to plug fest, different vendor software communicating with other vendor software (issuer, holder, agent software)
- Testing based on ToIP – Layers
- Lissi, esatus SeLF, Trinsic, Evernym, IBM and idRamp are participating
- Divided into different missions and constantly evolving

More info at: https://docs.google.com/presentation/d/1WkqSpFERc8now-f-Pz7PsRg9NMywSiZb92rTqJx5y00/edit#slide=id.g735a895e13_0_0

CONCLUSION
A RECAP AND FUTURE AHEAD



Status Quo

- There are as of now limited interoperability between different SSI solutions within the same stack
- There is no interoperability between different SSI ecosystems
- BBS+ Signatures look promising to bridge between Indy and non-Indy solutions
- There are however many other incompatibilities that need to be addressed
- Among others envelope, semantics, resolution and ZKP

Hakan Yildiz

TU Berlin | Telekom Innovation Laboratories | *Service-centric Networking*

hakan.yildiz@tu-berlin.de

https://www.linkedin.com/in/h-yildiz/

http://www.snet.tu-berlin.de/