

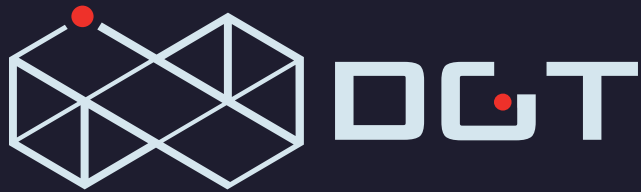


IDENTIFICATION

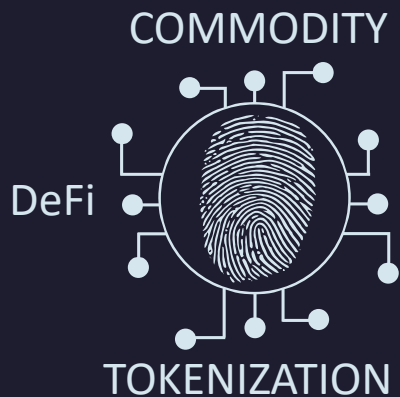
in decentralized hybrid networks

DGT Approach

WHAT IS



A decentralized platform based on F-BFT consensus providing a hybrid network for working with digital objects



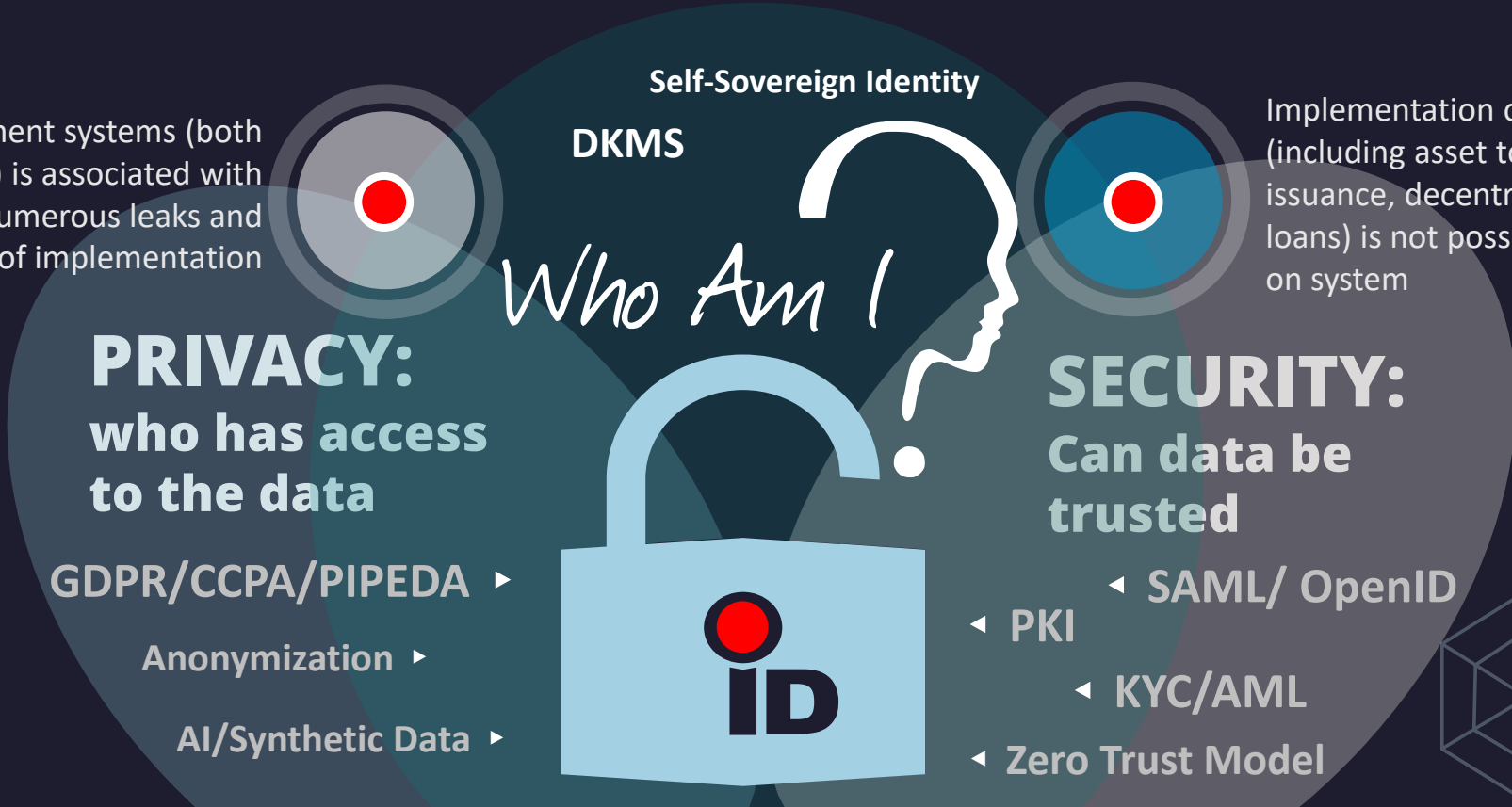
Practical application of modern business models in the blockchain requires support for decentralized identification

IDENTIFICATION AS KEY PROBLEM

Blockchain systems such as Bitcoin have proven to be resilient to anonymous transfers. Classical blockchains access operations based on a pair of keys (private and public), leaving the body of the transaction open. An issue arises as to whether a public key remains outside of such systems. Private blockchains (such as Hyperledger Fabric) solve this problem by introducing X.509 certificates with a centralized PKI structure. However, such networks have scaling problems to the difficulty of free participants joining them. Implementing complex business models that integrate different subsystems and services requires key management, an analogue of PKI- DKMS (Decentralized Key Management Systems).

The crisis of identity management systems (both centralized and federated) is associated with their defragmentation, numerous leaks and complexity of implementation

Implementation of complex models (including asset tokenization, digital currency issuance, decentralized investments and loans) is not possible without a reliable go-on system



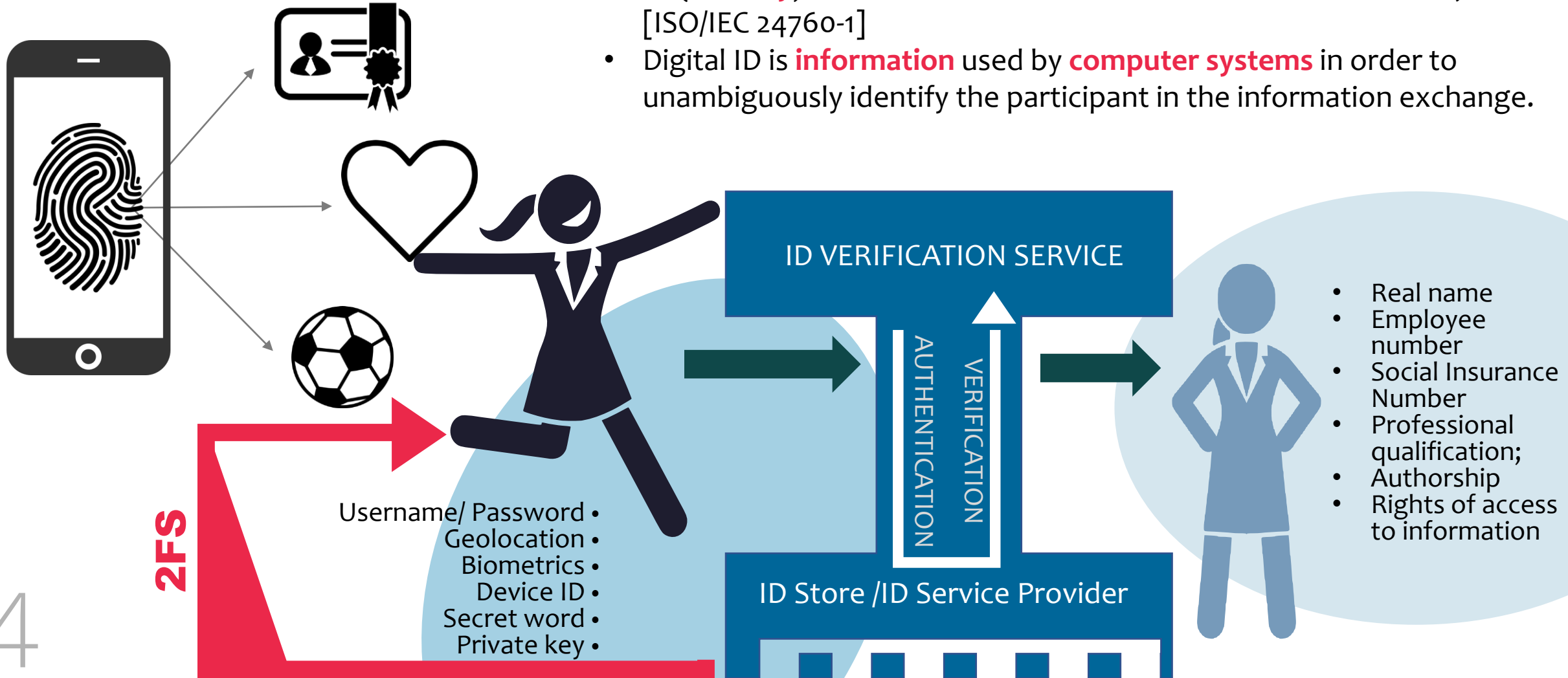
- GDPR/CCPA/PIPEDA ▶
- Anonymization ▶
- AI/Synthetic Data ▶

- ◀ SAML/ OpenID
- ◀ PKI
- ◀ KYC/AML
- ◀ Zero Trust Model

WHAT IS DIGITAL IDENTIFICATION?

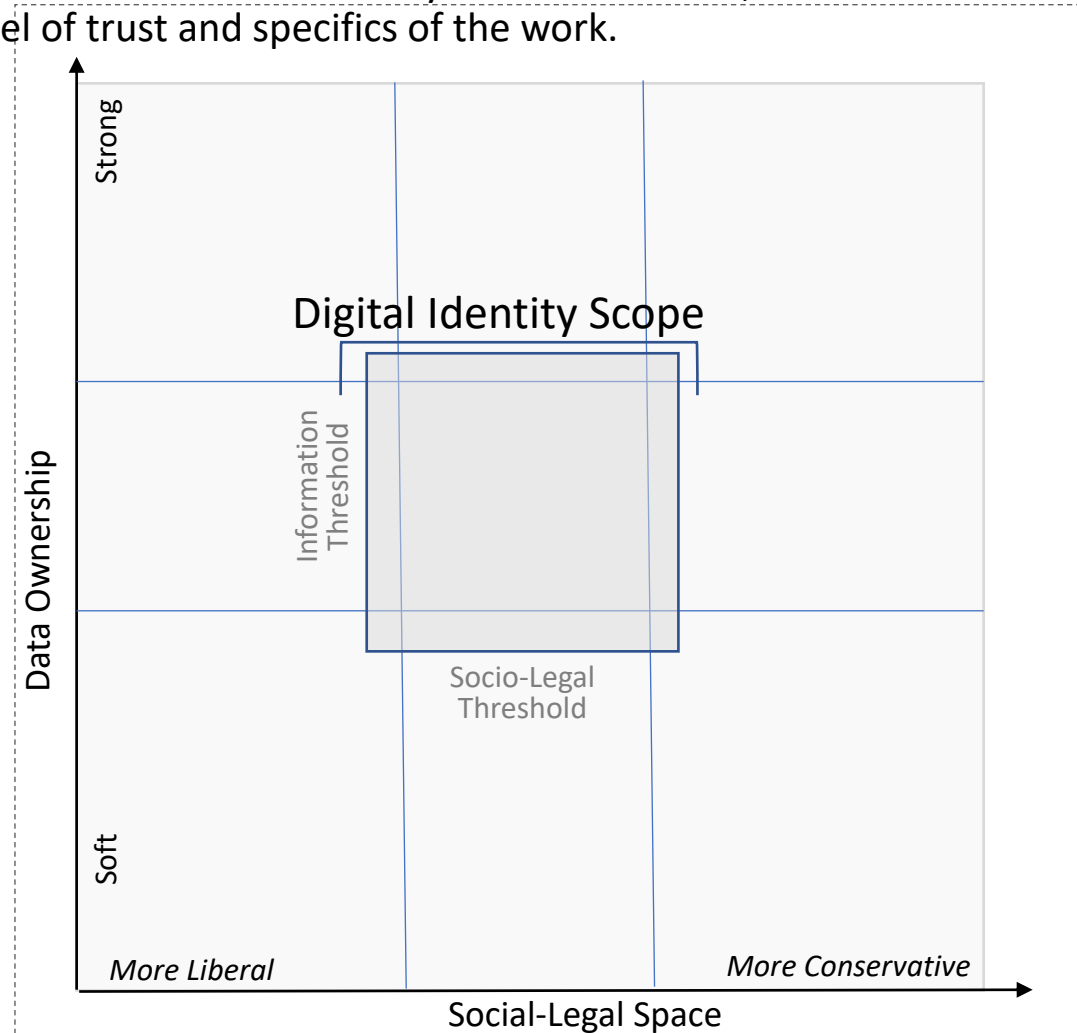
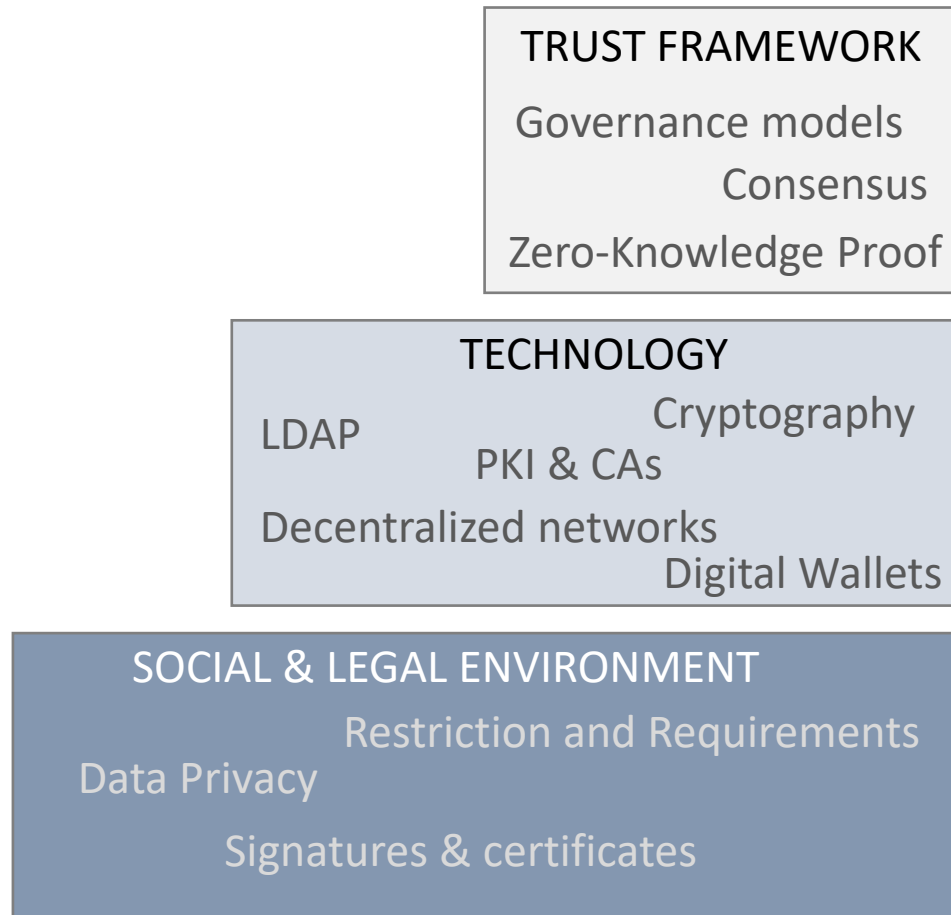
... **the most important problem** in cyberspace is determining **who is interacting with whom**...

- ID (**identity**) is defined as "a set of attributes that define an object" [ISO/IEC 24760-1]
- Digital ID is **information** used by **computer systems** in order to unambiguously identify the participant in the information exchange.



IDENTIFICATION SCOPE

Identification depends significantly on the context, in the broad sense of the word - on society and its freedoms, and in the narrow sense - on a specific subject area that determines its attributes, model of trust and specifics of the work.

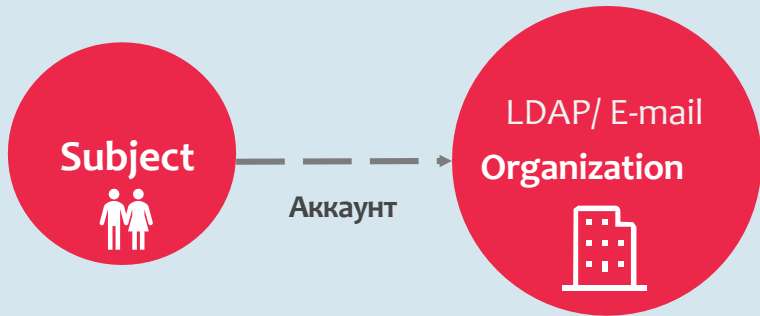


Adapted from Andrej J. Zwitter and etc. "Digital Identity and the Blockchain: Universal Identity Management and the Concept of the "Self-Sovereign" Individual"

EVOLUTION OF IDENTIFICATION SYSTEMS

MODEL 1 TRADITIONAL CENTRALIZED

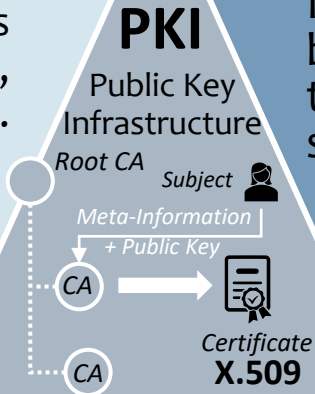
The traditional system is based on an account, implies access only to the information that is inside the perimeter



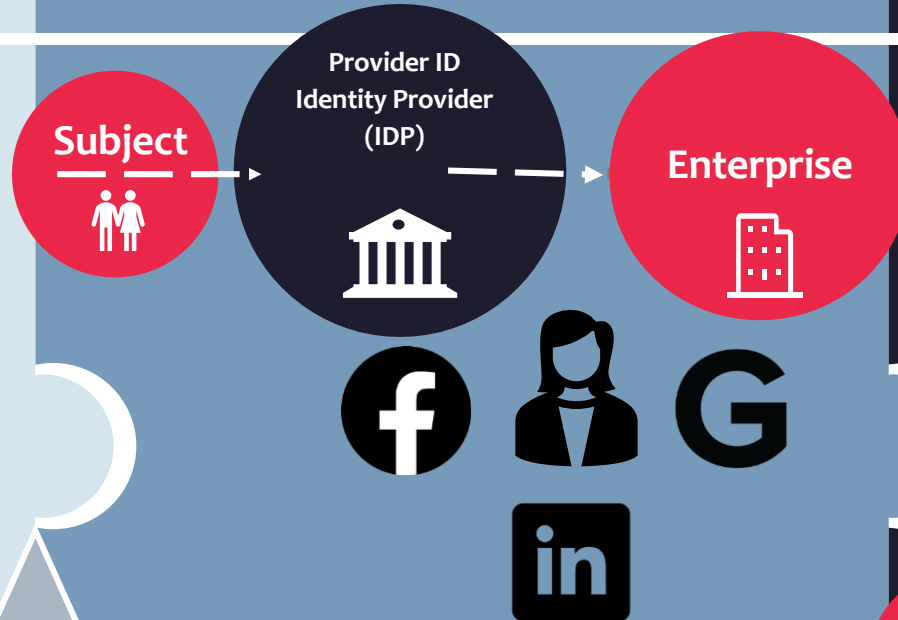
Users' personal data is stored in the organization's "database," and this happens for every organization, application, or website you log into.

As a result, this model requires the creation and management of separate credentials for each relationship.

6



MODEL 2 FEDERATED

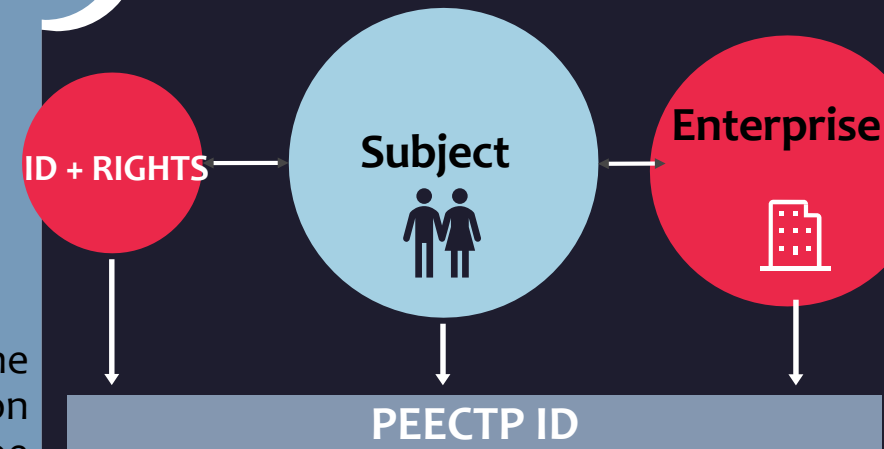


In this model, communication between the identity provider is through common protocols such as SAML or OAuth.

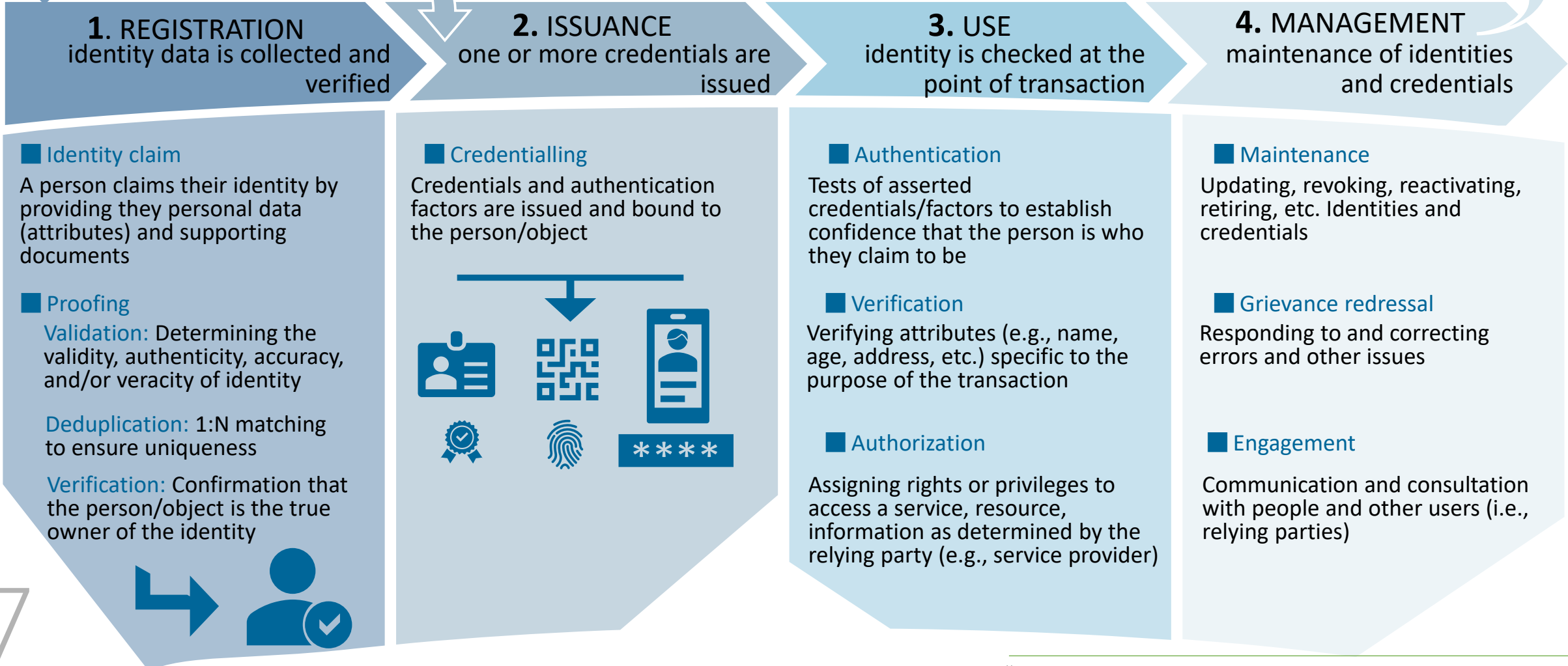
Data is still centralized within the identity provider. A common example of such a model would be Facebook, Connect, or SPID.

MODEL 3 DECENTRALIZED (Self – Sovereign Identity)

This model allows you to create a system in which the participant of the information exchange and the artifacts associated with it are linked together, so that any object becomes an attribute

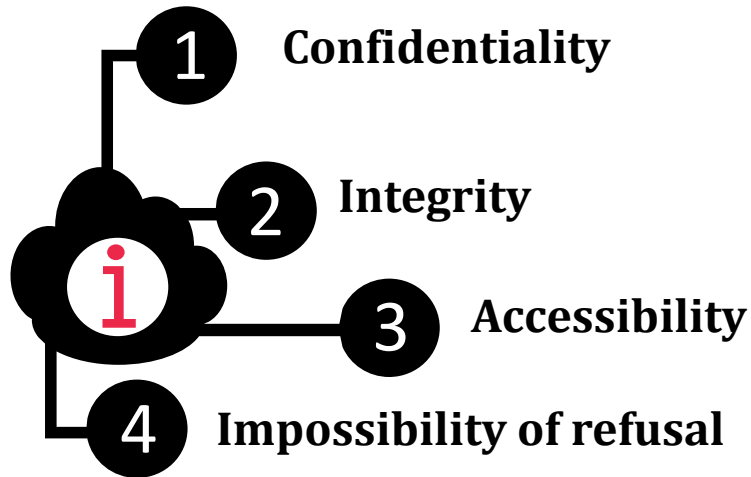


IDENTIFICATION LIFE CYCLE*





Cybersecurity is an activity aimed at protecting systems, networks and programs from digital attacks



Information technologies are aimed at the development and operation of systems that manage data (information).

From the point of view of information, security is a system of organizational and technical measures aimed at ensuring the most important attributes of information are protected (confidentiality, integrity, availability, impossibility of failure)

Traditional security systems were built based on a security perimeter, with the implication of a protected environment inside and an environment of distrust on the outside.

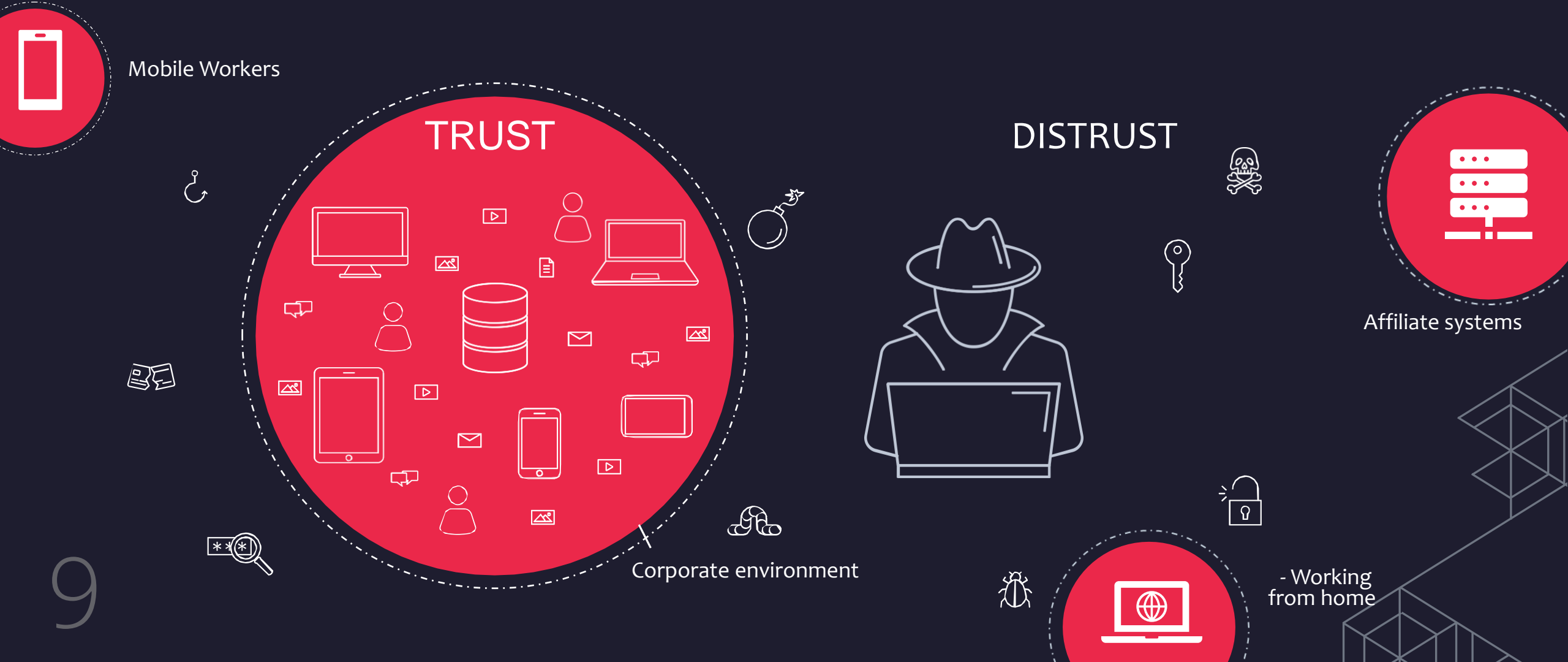
Under these conditions, a threat (attack)-countering system was built, which aimed to reduce, transfer, or eliminate risks

The digital economy has destroyed the concept of the contour due to the rising number of available services, remote work, the complex context (circumstances) of access to information. This gave rise to a new concept of security:

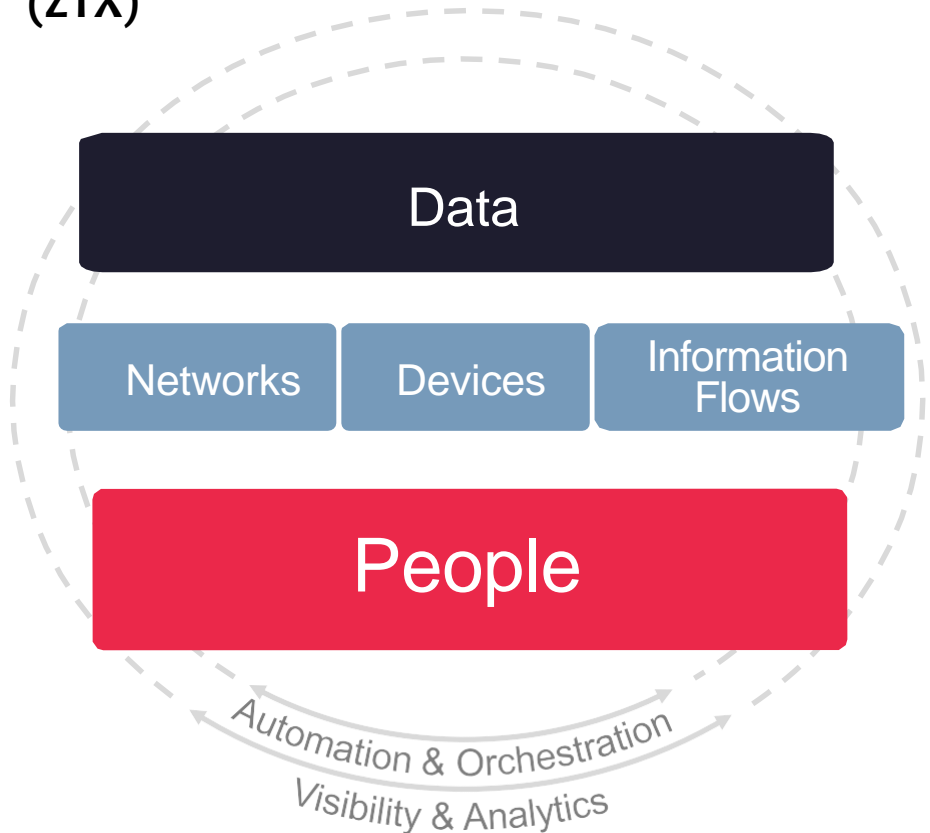
ZERO TRUST



Work in a digital environment means the need to control not one security circuit, but many. In practice, the contour should be built around each participant in the information exchange with an understanding of the context (under what circumstances access to data is requested; what is the sequence of operations; etc.)



Forrester's Zero Trust Extended Ecosystem (ZTX)



Key concepts of ZTX:

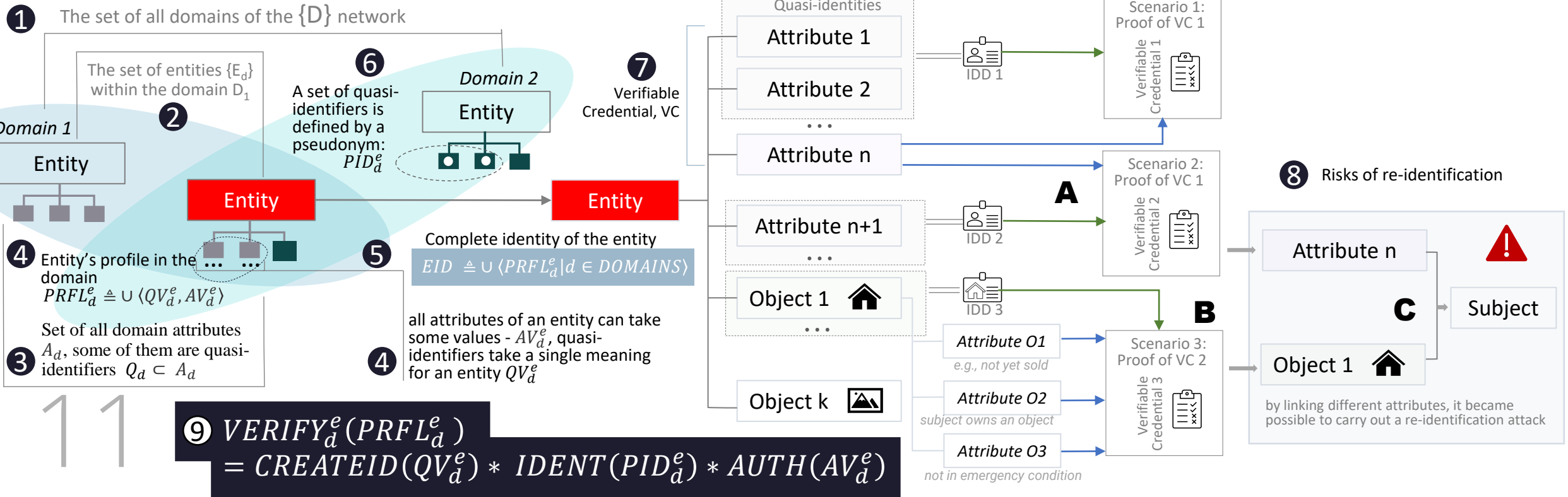
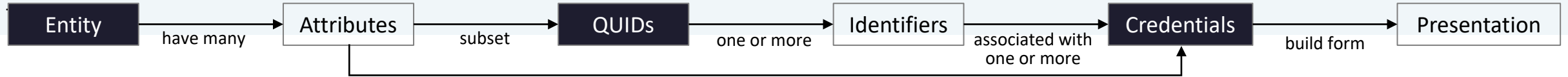
- #1 Secure access**
Access to all resources is provided in a secure way, regardless of your location.
- #2 Access control**
Each subject of information exchange is granted access based on checking access rules
- #3 Inspection + logging**
Fixation of all events on access of information and its transformation

10 When any subject of information exchange is surrounded by its own security perimeter, there is a need for cloud decentralized security, and the problem of identification becomes an integral tool for ZTX strategy.

PARTIAL IDENTITY

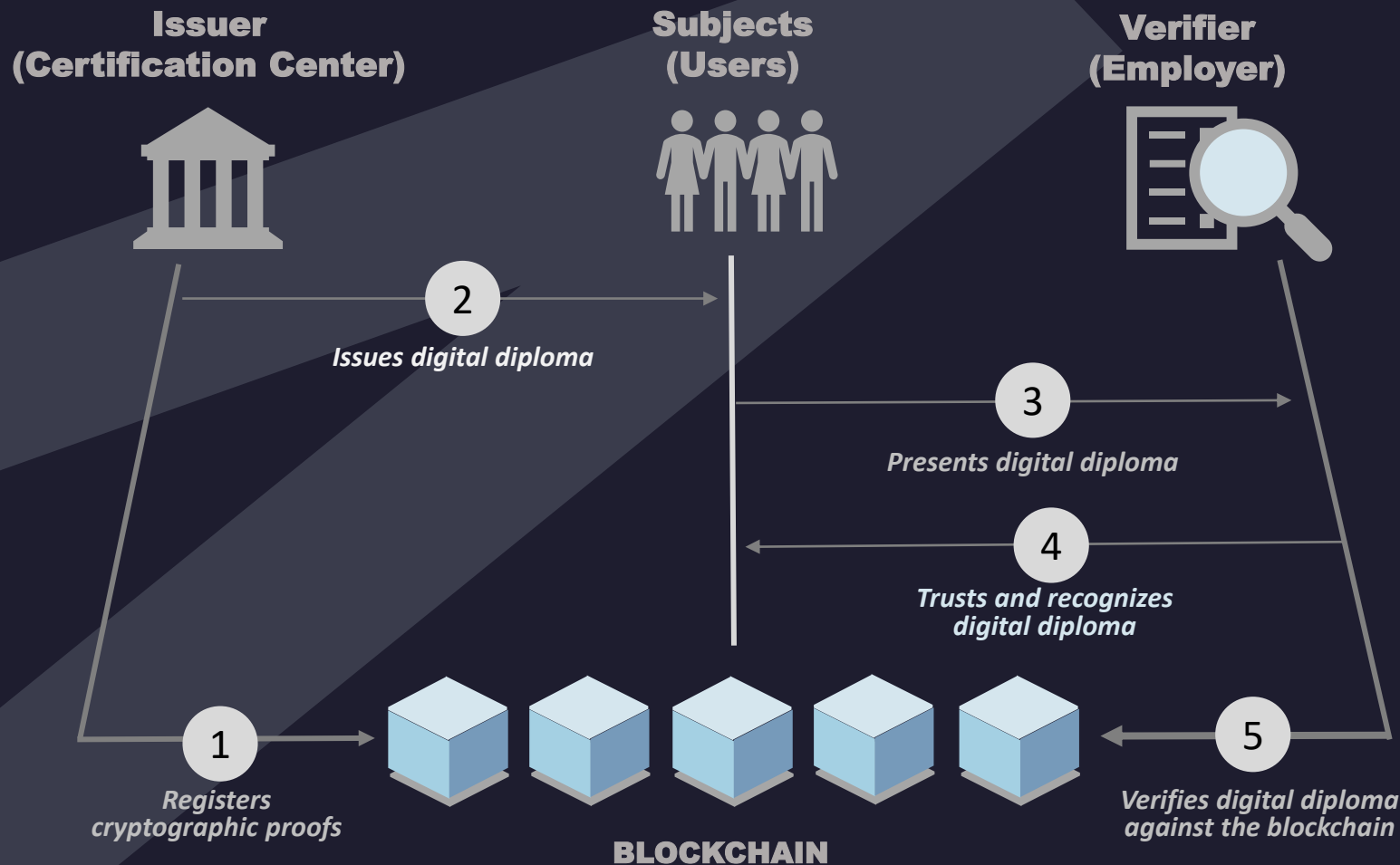
In a selected domain context, each entity is defined by attributes that accept some values. Some of the attributes determine the uniqueness of the entity (the combination of their values allows you to select a single entity). These are called *quasi-identifiers*. Based on a set of quasi-identifiers (in each domain there may be several sets), the entity is assigned an *alias* (ID). An entity can be represented in multiple domains. Each set of quasi-identifiers denoted by an alias forms an entity profile in the domain.

Thus, the entity is represented by a set of profiles, each of which only partially represents its essence and is thus a *partial identity*. As part of identity management, (1) an alias (**CREATEID** function) is created sequentially based on identifiers. Then (2) a set of attributes (in the aggregate credentials) is mapped to this ID (alias) - represented by the **IDENT** function. Then, when authenticating, (3) the uniqueness of the entity and the existing alias is checked by the **AUTH**



SELF-SOVEREIGN IDENTITY

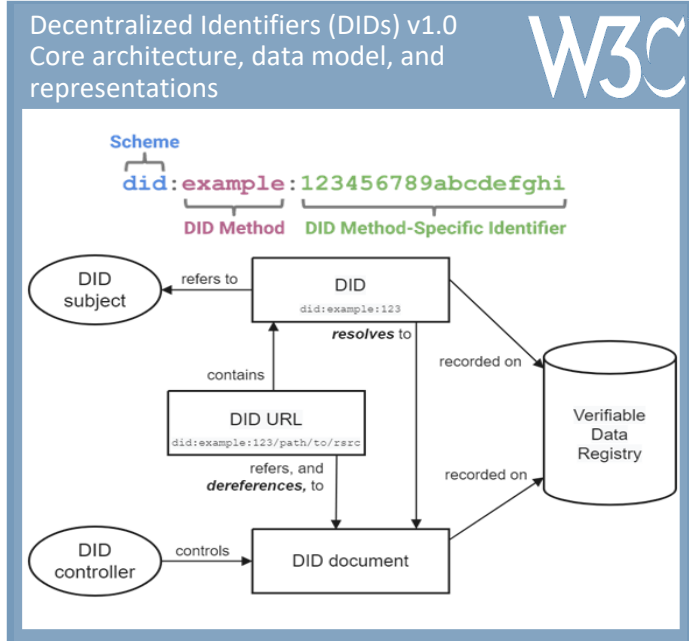
Decentralized identity (Self-Sovereign Identity) is a concept that divides the process of identification, authentication and authorization between different participants in such a way as to exclude the accumulation of personal data by one party, ensure transparency of access to data, as well as store identifiers in blockchain networks



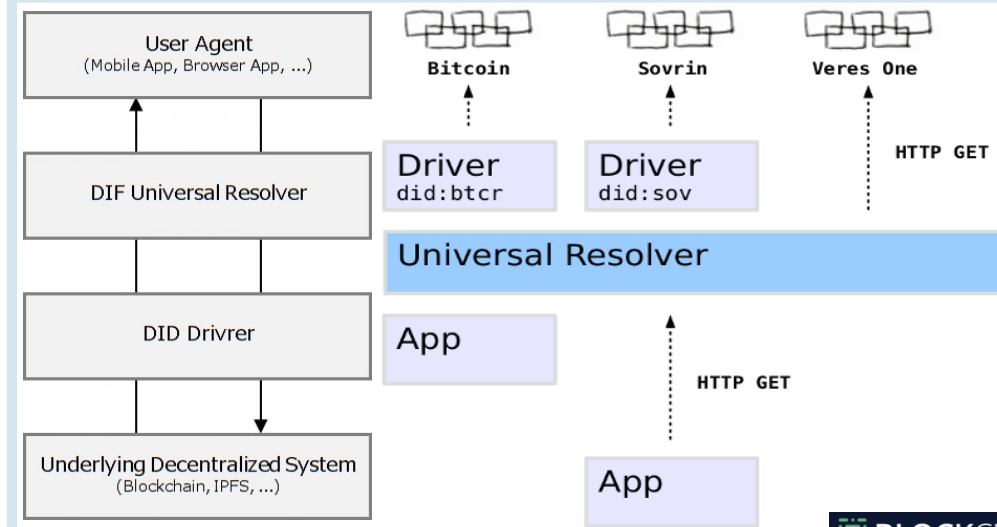


DECENTRALIZED ID STANDARDS AND INITIATIVES

Draft decentralized ID standard defining the format, mechanisms for working with it (JSON-LD), methods, [prepared by the W3C](#)



The [Decentralized Identity Foundation \(DIF\)](#) is the largest organization, including both MS and Hyperledger, which is engaged in the preparation of specifications in various areas, including did Universal Resolver

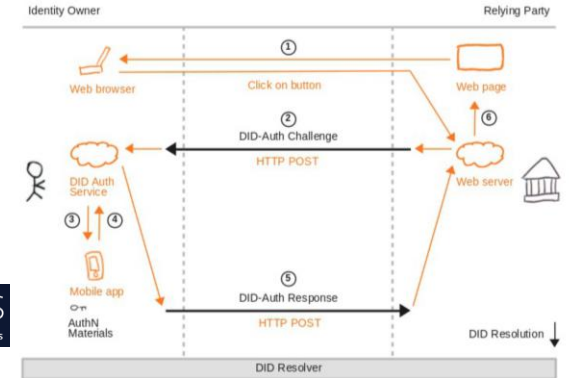


SIDETREE PROTOCOL

DID Decentralized Identity
VERIFIABLE Credentials
JSON-LD

Rebooting Web-of-Trust

[RwOT Development](#): A Set of Specifications and White Papers Defining DID Protocols and Conceptual Architecture

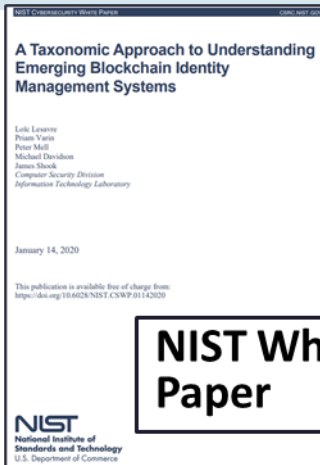
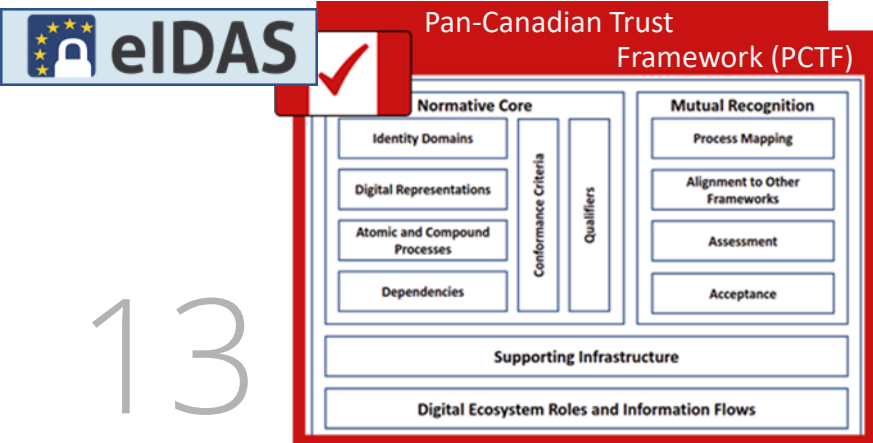


ERC-725 - Ethereum Identity Standard

describes proxy smart contracts that can be managed using multiple keys and other smart contracts. ERC 735 is a related standard for adding and removing requirements for the ERC 725 smart identity contract

ERC-1056 - Lightweight Identity

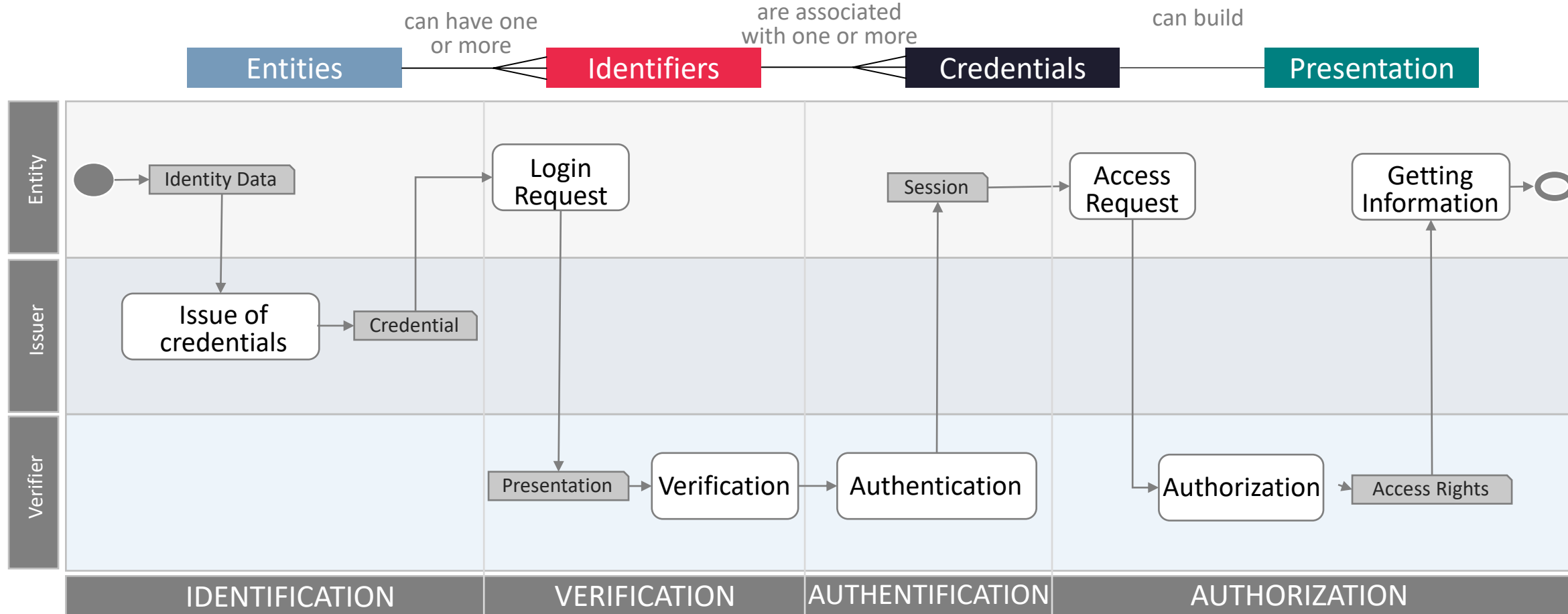
- a standard for creating and updating identities with limited use of blockchain resources. an identity can have an unlimited number of delegates and associated attributes



NIST White Paper



DECENTRALIZED IDENTIFICATION ACTORS

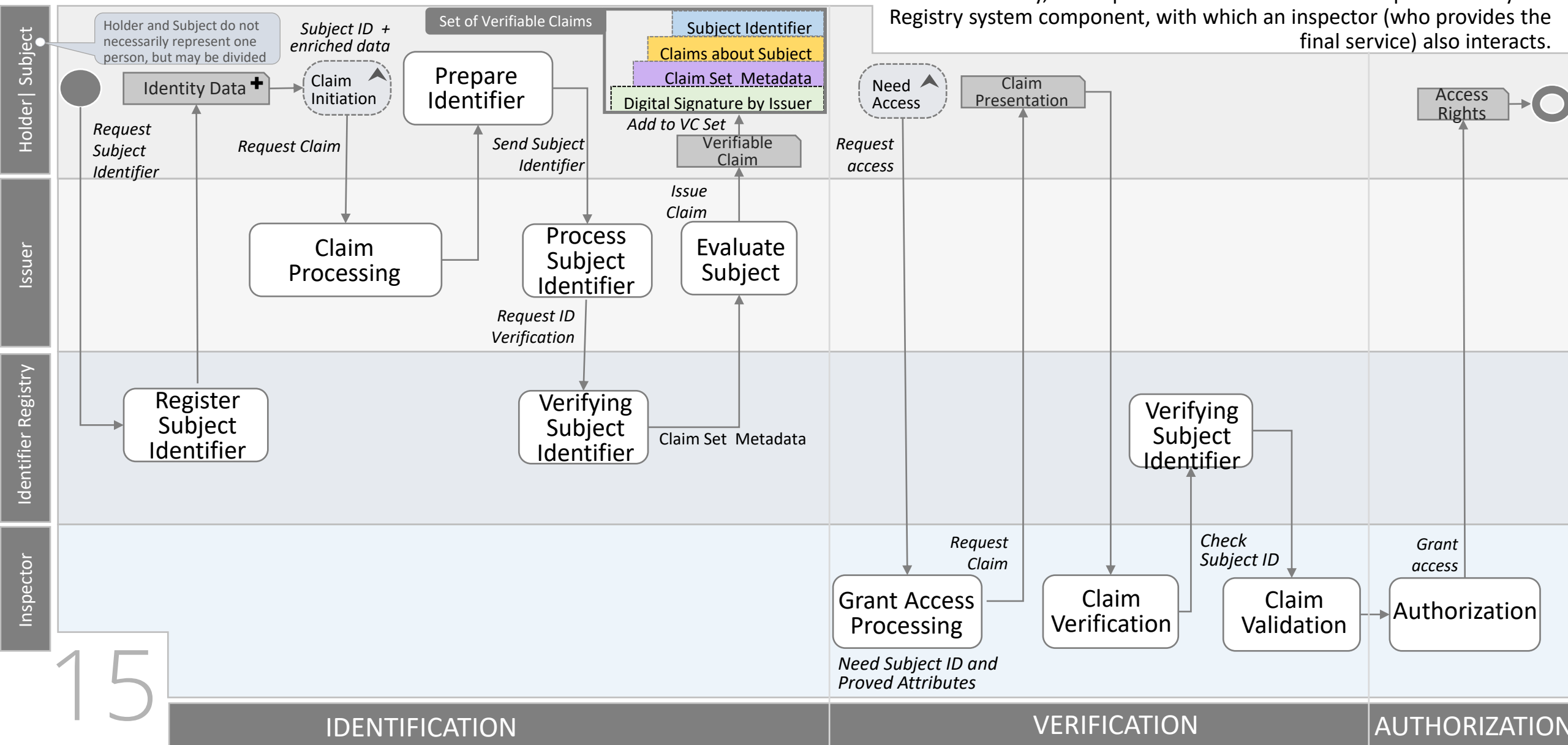


In a decentralized environment, it is natural to divide several roles (parties):

- **Entity** ■ – a subject or object that is identified, can have one or more identifiers ■, each of which can be associated with one or more credentials (verifiable credentials - VC, a set of meta-information) ■, in the course of the work of other parties, transmitted through presentation ■;
- **Issuer** – the party issuing the IDs and the meta-information (VC) associated with them
- **Verifier** – the party that verifies the ID, decrypts and starts the use of the VC, can be separated from the end user of the ID (Service Provider)

IDENTIFIER REGISTRY

Data entry, data update and validation can be represented by the Registry system component, with which an inspector (who provides the final service) also interacts.



IDENTITY STACK BY IDF

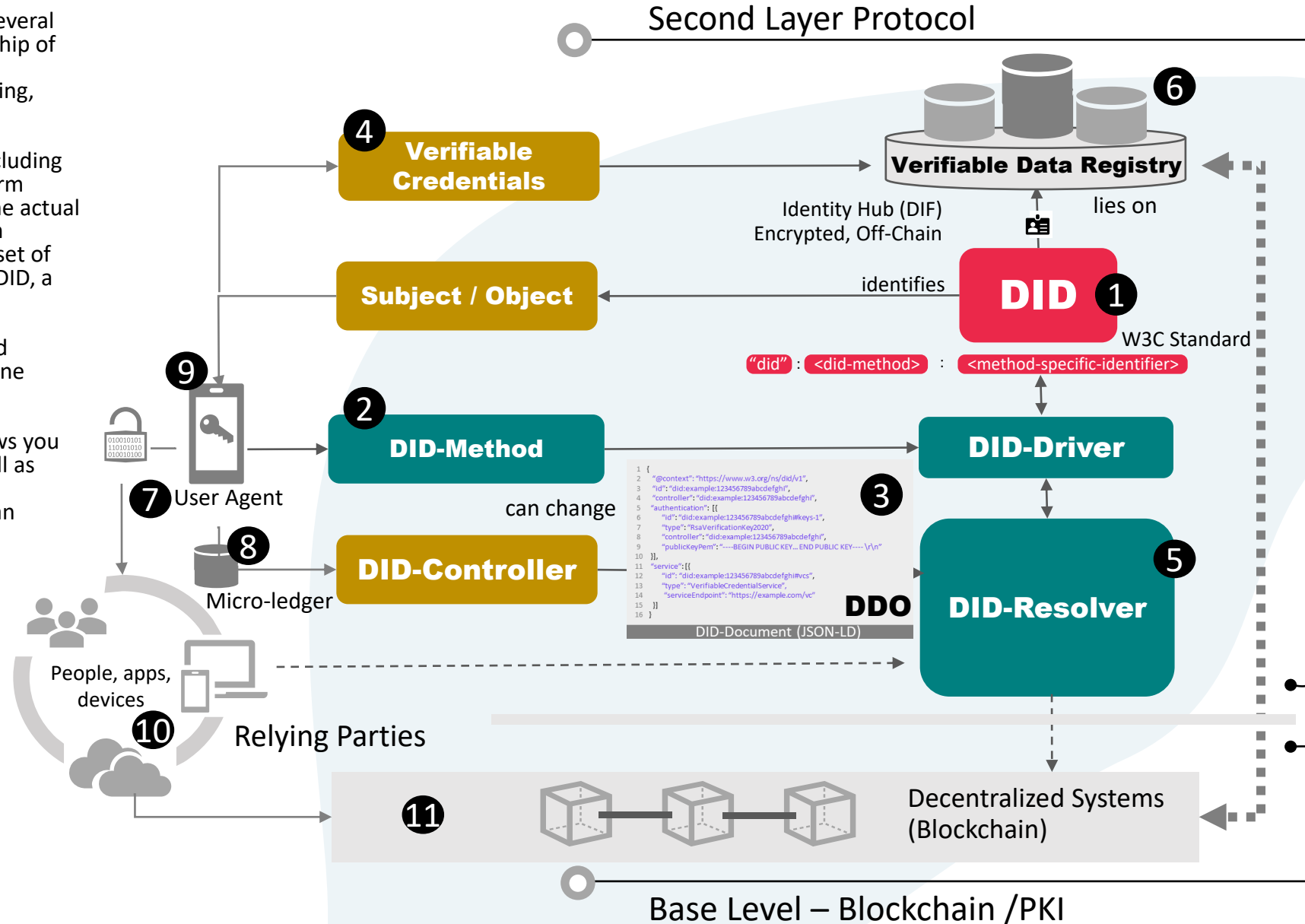
The Decentralized Identity Foundation has published a draft of the protocol stack, shown in the table below as the multi-layered stack to facilitate the emergence of portable and interoperable solutions.

Layer	Description
Application	Applications that interact with a given identity management system through library integrations and API calls
Implementation	Libraries that integrate the system in third-party applications
Payload	Message formats, such as JWT/JSON-LD, used to exchange data between participants
Encoding	Methods for encoding data at both the encryption and payload layers
Encryption	Methods for encrypting messages between participants as well as encrypting the data held by the identifier owner
DID Authentication	Methods to authenticate a participant using their DID
Transport	Transport protocols used for sharing data between participants and devices, such as HTTP or QR code
DID Resolution	DID Resolver used to convert a DID into its corresponding DID document
DID Operation	Create, Read, Update, and Delete operations for a DID document
DID Storage	Methods for storing DID Documents and DIDs
DID Anchor	Networks that serve as medium for DIDs

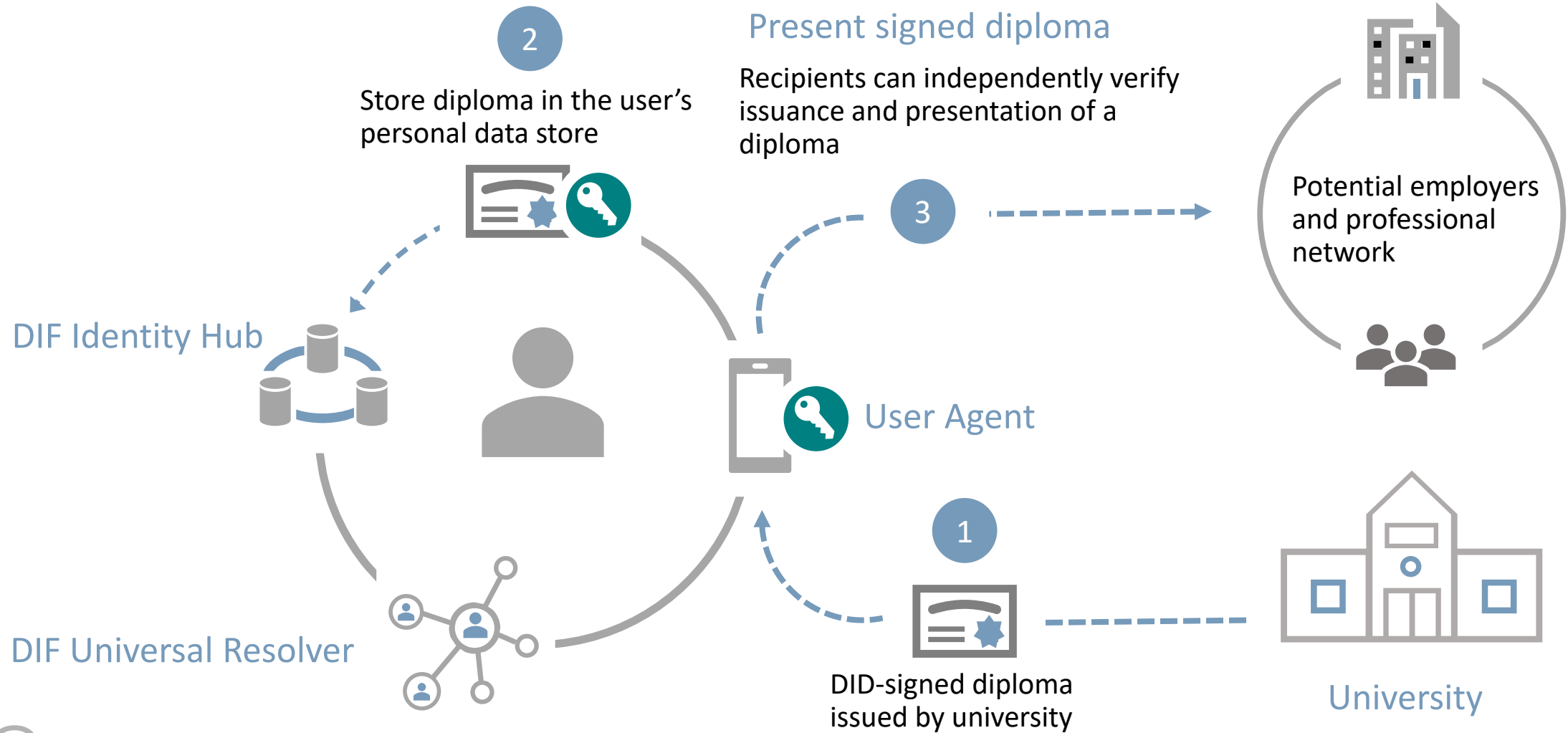


ARCHITECTURE OF DECENTRALIZED IDENTITY

- DIDs** – decentralized identifiers, each object can have several DID, ownership is established by demonstrating ownership of the private key associated with the public key
- The **DID method** is a set of schemas for creating, resolving, updating, and deleting DID, responsible for registering, replacing, rotating, restoring, and expiring DID in IDMS
- DID document (DOD)** – A representation of the DID, including related metadata in JSON-LD notation (includes a Uniform Resource Identifier for accessing the document itself, the actual did reference, a set of public keys, a set of authorization methods that allow delegated organizations to work, a set of services for describing where and how to interact with DID, a time stamp for creating a document and updating it, cryptographic confirmation of integrity)
- Verifiable Credentials** – accounts that can be exchanged between DID include URIs (such as DID), URI to determine credential type, date issued, metadata, cryptographic confirmations, expiration conditions
- DID-Resolver** – provides a universal converter that allows you to write and read data through a single interface, as well as interact with multiple IDMS
- Verifiable Data Registry** – encrypted personal vaults, can be located offline (Identity Hubs), designed to store accounts and DID
- Agent** – a software agent delegated by the entity responsible for data exchange. Edge Agent – located in the user's wallet, cloud agent – works via API
- Micro-ledger**– a replicated store located in each agent, also built on the Merkle tree, contains information about all events
- Identity Wallet** – the user's wallet that allows you to store DID and accounts, edge agent policies and equipped with the necessary cryptographic management
- Relying Party (Service Provider)** - the party that receives information from the relying party and provides the necessary service
- Decentralized System** – the main network, blockchain



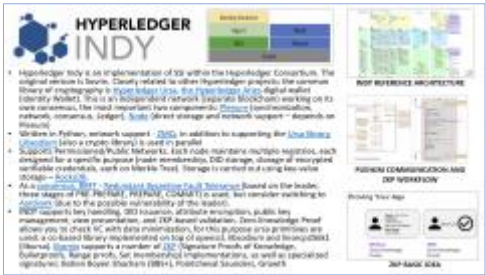
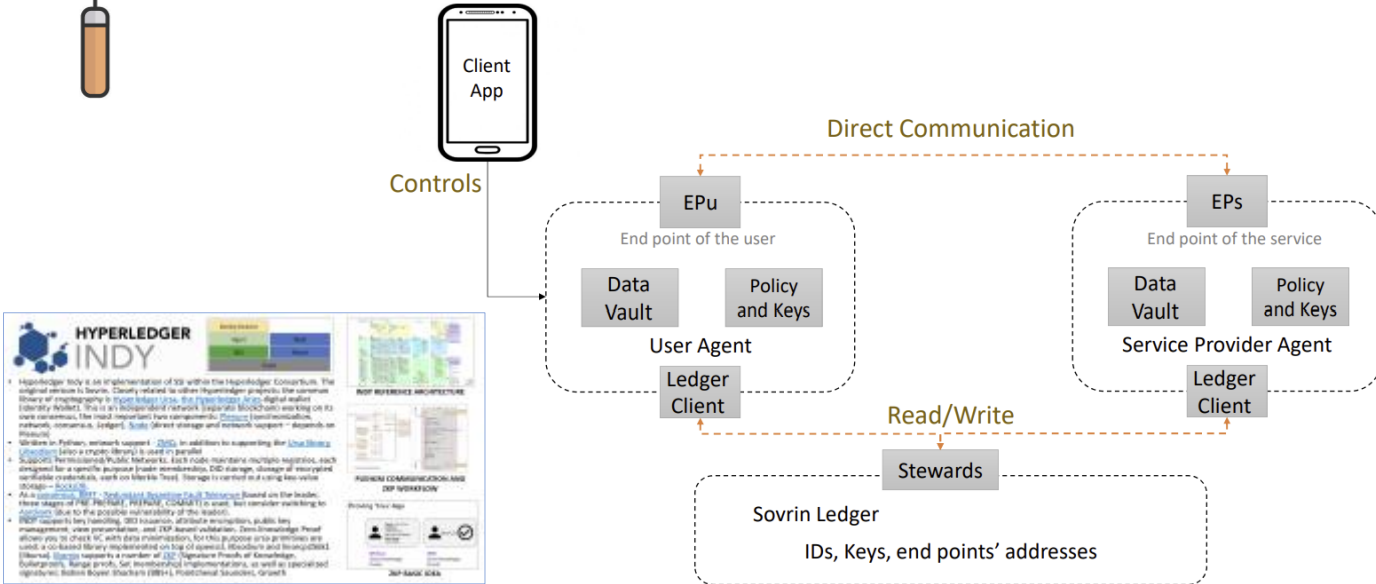
MICROSOFT'S SSI ARCHITECTURE



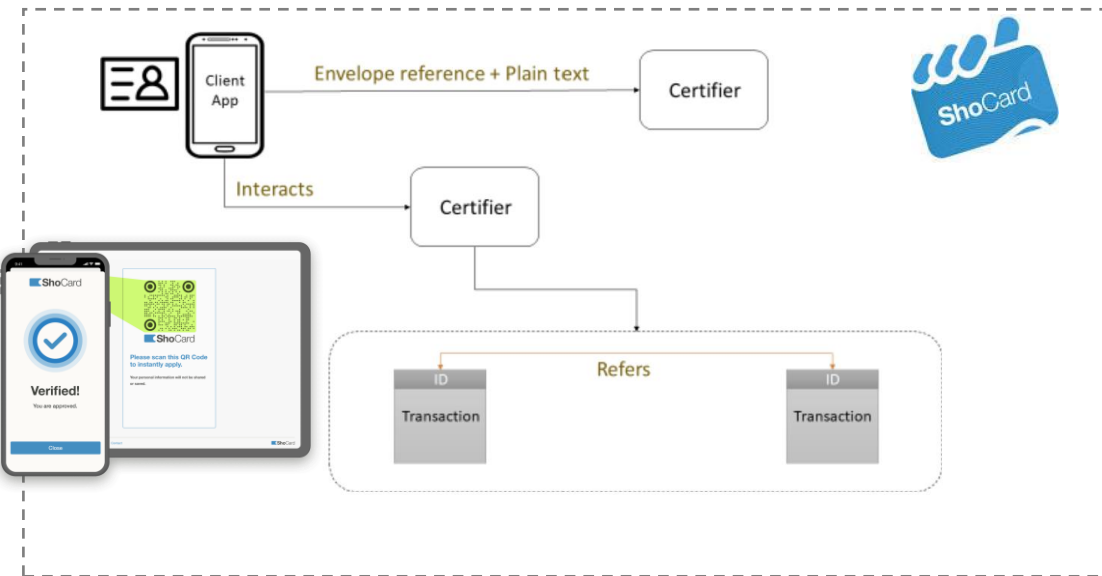
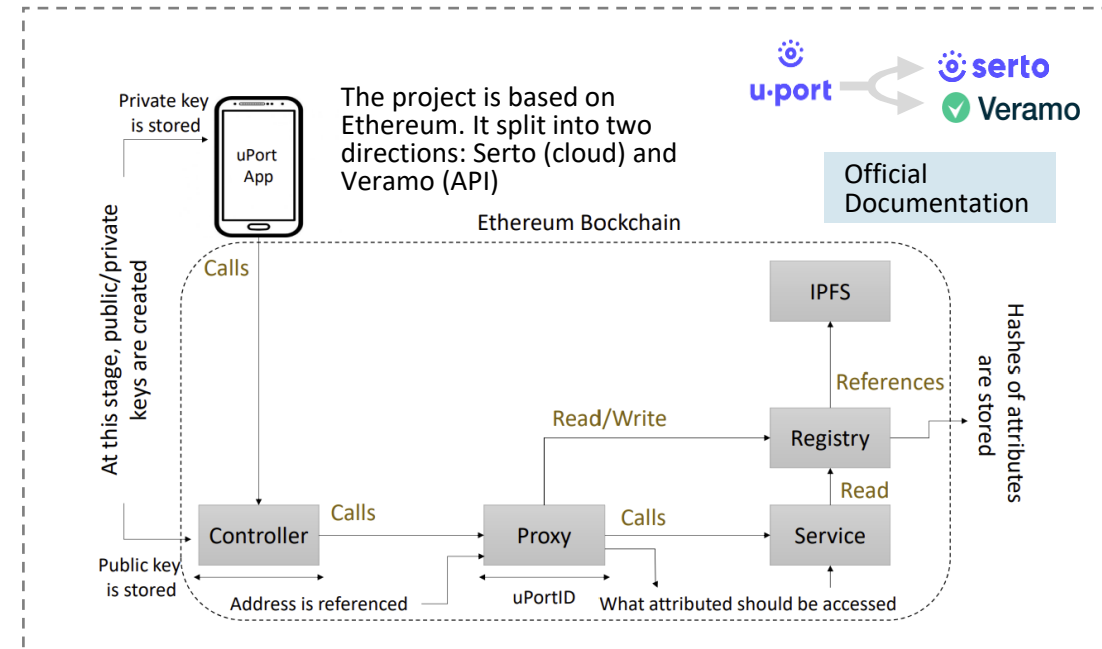


A project originally created by the Sovrin Foundation and developing inside Hyperledger. A separate distributed network, positioned as a decentralized ID system (like Bitcoin = decentralized money, Ethereum = decentralized applications)

- Consists of Indy-Plenum (protocol and Ledger), as well as a node (a distributed server that supports the network)
- Implemented in Python and ZMQ. Depends on Ursa. Storage - RocksDB
- Works with [RBFT](#) consensus support, going to switch to [Aardvark](#)

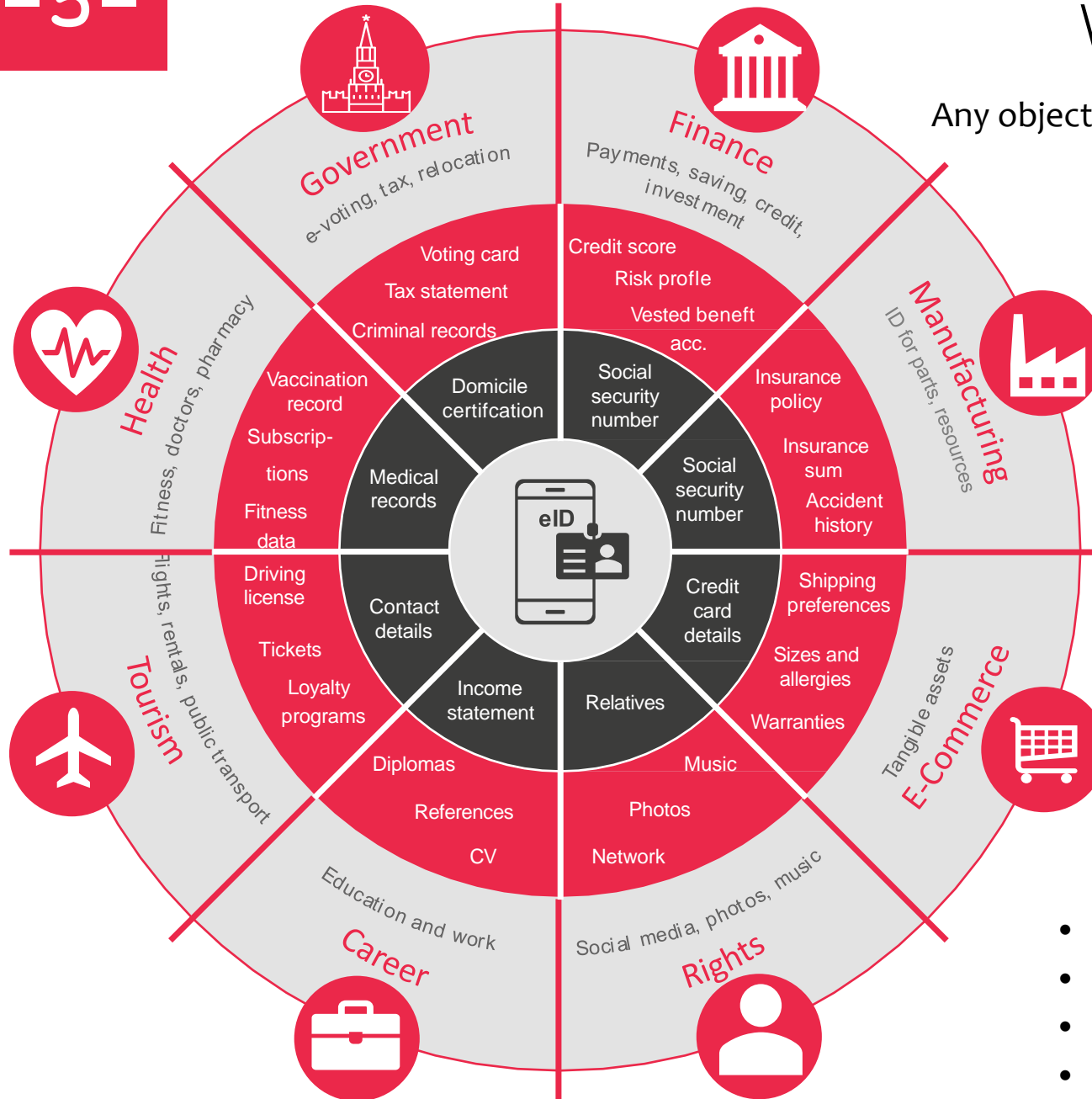


- [Bootcamp Russia Overview](#)
- [Official documentation](#)
- [Indy Architecture From Hyperonomy](#)



WHERE IT IS USED

Any object may be defined by its attributes. Practically any sector may find a use for digital IDs.



- Digital identity papers;
- Educational certificates;
- E-signatures of documents & contracts;
- Medical results, genetics

- ID in manufacturing (Metal ID)
- Tickets and vouchers;
- Digital rights;
- Insurance ID

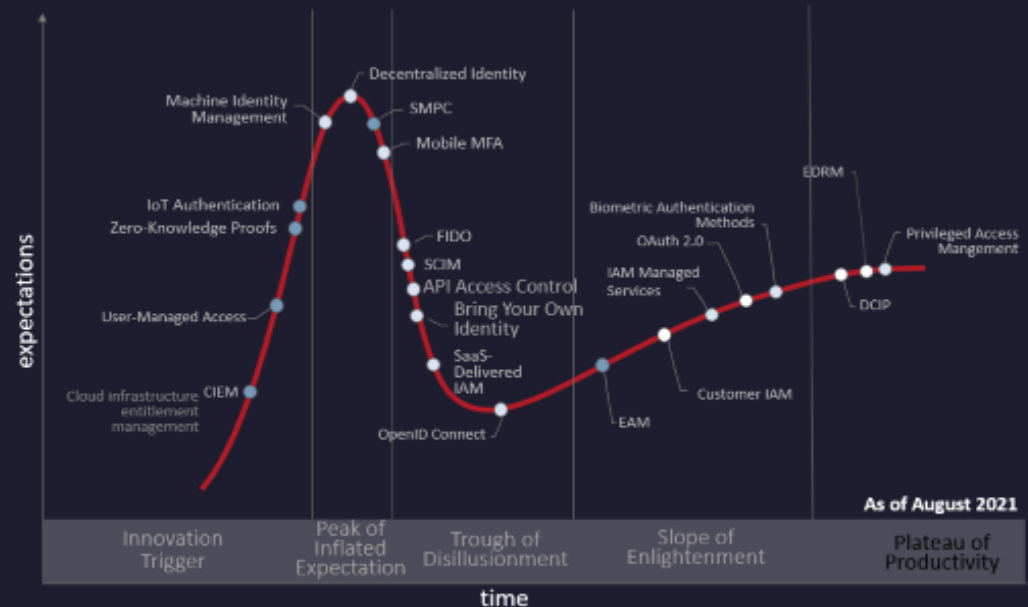
Technology is validated by the market and experts

There is a leader on the western market:



ID as a Service is in demand by many market participants (Metal, logistics, manufacturing) + there is regulatory pressure (i.e., GDPR)

Hype Cycle for Identity and Access Management Technologies, 2021



Plateau will be reached:
 ● less than 2 years ● 2 to 5 years ● 5 to 10 years ▲ more than 10 years ⊗ obsolete before plateau

Source: Gartner Hype Cycle for Identity and Access Management Technologies, 2021

Blockchain and Digital Identity: the path to Self Sovereign Identity

Insight Report

Identity in a Digital World
A new chapter in the social contract

Deloitte.

The future of cyber survey 2019
Cyber everywhere. Succeed anywhere.

Кибербезопасность: больше чем защита?

Международное исследование EY в области информационной безопасности 2018-2019 годы

EY

McKinsey & Company

Digital McKinsey and Global Risk Practice

Cybersecurity in a Digital Era

DGT IDENTITY APPROACH


The need to develop an identity management system (ID_MS) in DGT is substantiated by tokenization (including that of assets). The original blockchain concept allows you to manage operations based on a key pair, where the ownership of the public key remains anonymous. This is acceptable for anonymous financial transfers but does not work in the case of asset management.


Another problem is to store the transaction body in plain text. In order to unambiguously represent digital objects, the attributes of tokenized entities should be hidden from prying eyes and available only to authorized users. This requires storing encrypted attributes outside the blockchain network (off-chain).





The main drivers of the implementation of the identity management system


Current background:

 The introduction of decentralized ID management systems is at the peak of expectations and is accompanied by a large amount of research, standards development and applied projects

 DGT differs from other blockchain implementations in its complex topology and multiple node roles (see hybrid architecture), which provides an advantage when handling horizontal integration tasks, but requires adaptation of the common ID management scheme.

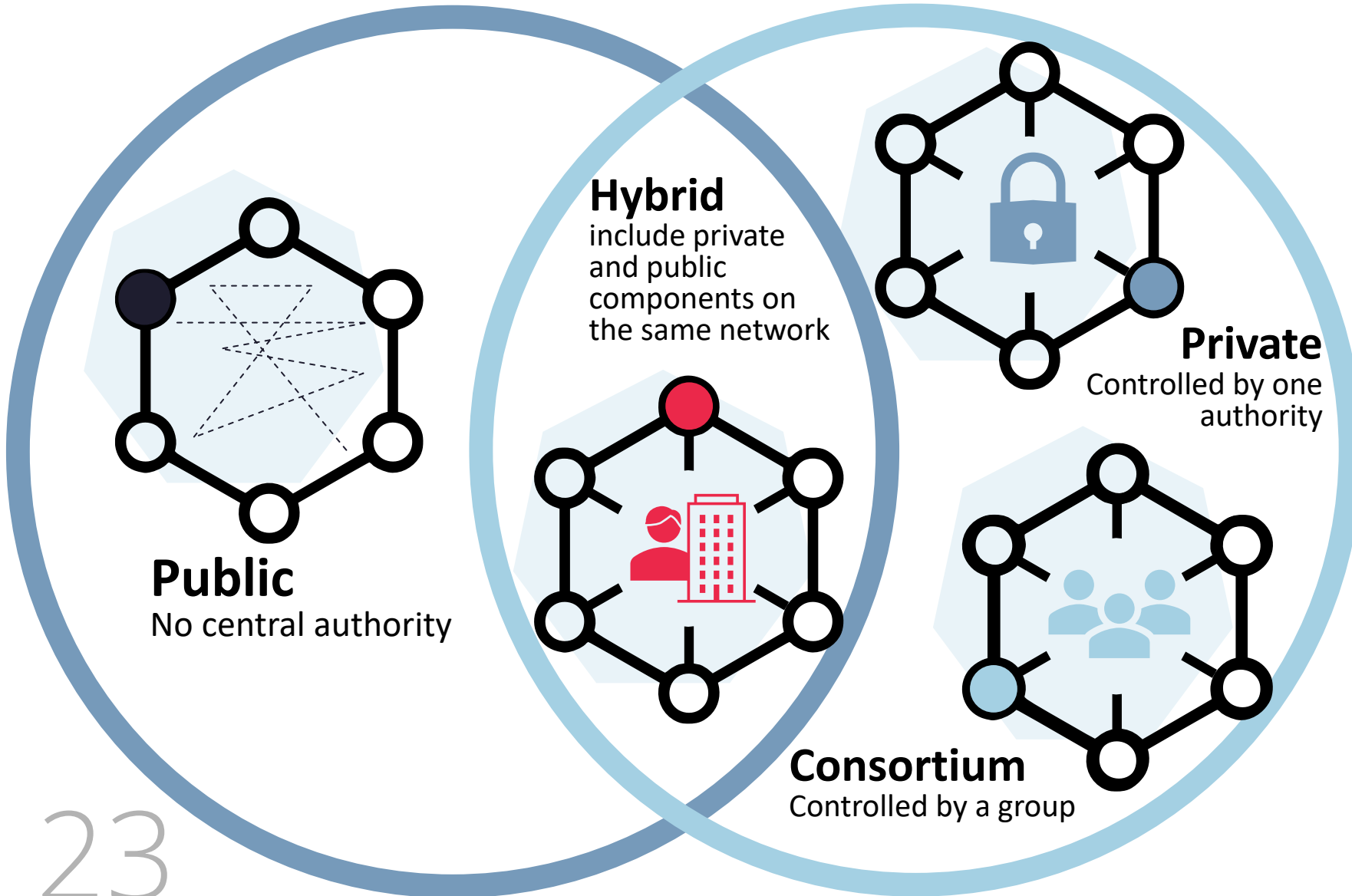
 While the biggest focus of market participants lies in replacing current federated ID management systems, DGT's motivation is to support today's decentralized DeFi business models.

 The main direction of development is support for the W3C DID standard and DIF models, but the transition to the final form through several intermediate architectures

 The development of decentralized ID management systems does not yet have a pronounced leader, except for Hyperledger Indy. The verifiable credentials system will be produced within the DGT (off-chain and on-chain) circuit with a focus on future integration with industry leaders.



DECENTRALIZED HYBRID NETWORKS



Hybrid networks allow for the creation of B2B2C networks that informationally link business partnerships to the end consumers.

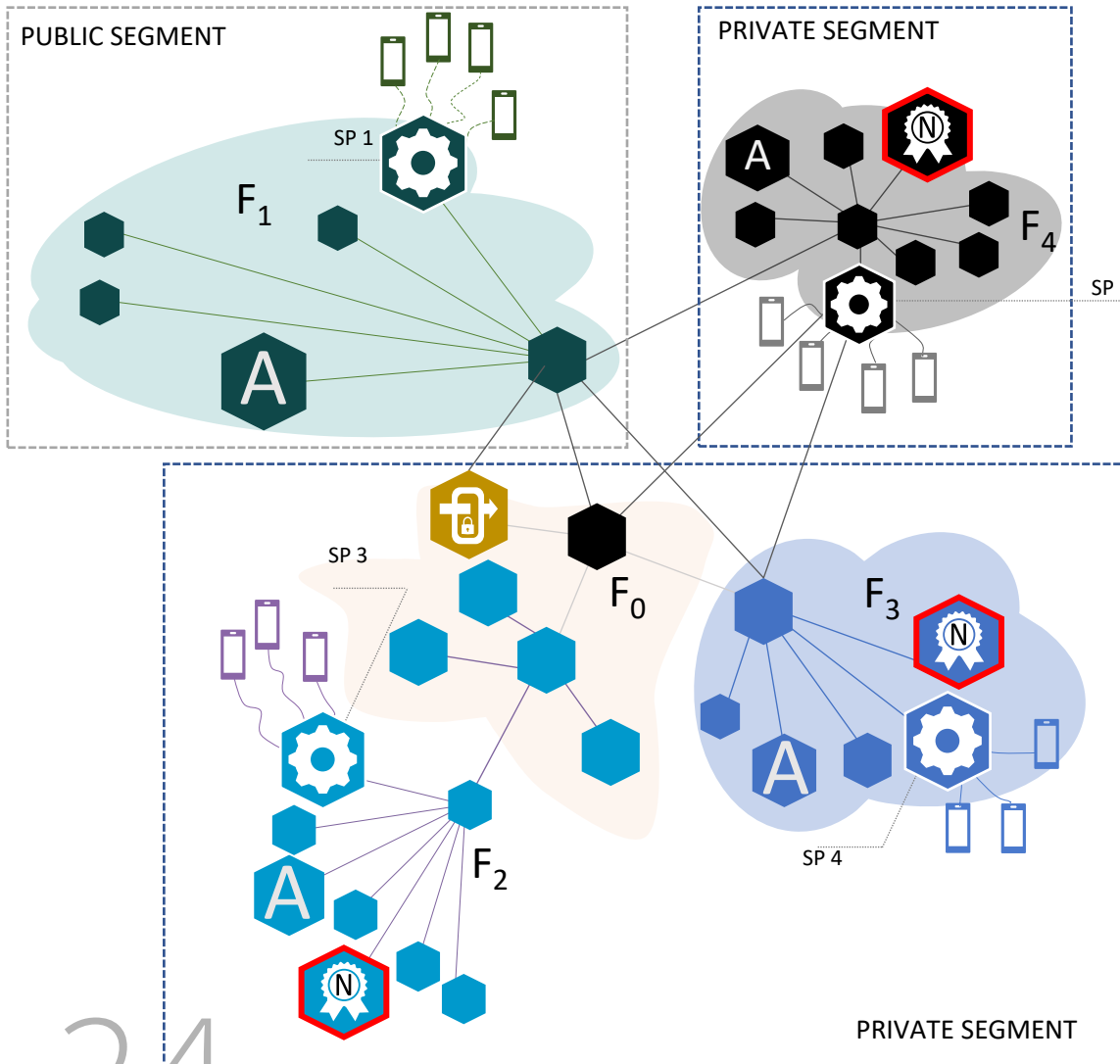
Hybrid networks offer tremendous business benefits through vertical and horizontal integration, combining business interoperability and end-user applications.


At the same time, you must deal with different security loops, which requires complex authentication and authorization operations.


H-NET ARCHITECTURE


H-NET architecture – the architecture of a blockchain network built in a federated way:


- Network nodes are grouped into Fi clusters, allowing BFT consensus to be reached in the fastest way;
- The network simultaneously presents private and public structures (hybrid network) with delineated segments
- Each node supports several types (families) of transactions at the same time (the network is multifunctional)
- The distributed ledger is built on DAG technology and is a combination of related graph structures, each of which can have its own visibility for different segments (the so-called private branches);
- The nodes of the system have various roles, including:




 Validators that approve transactions within a cluster

 Nodes that provide data validation at the network level (on-chain Trust Providers). Any transaction within a trust group (node clusters) requires approval outside that group (Circle of Trust, CoT)

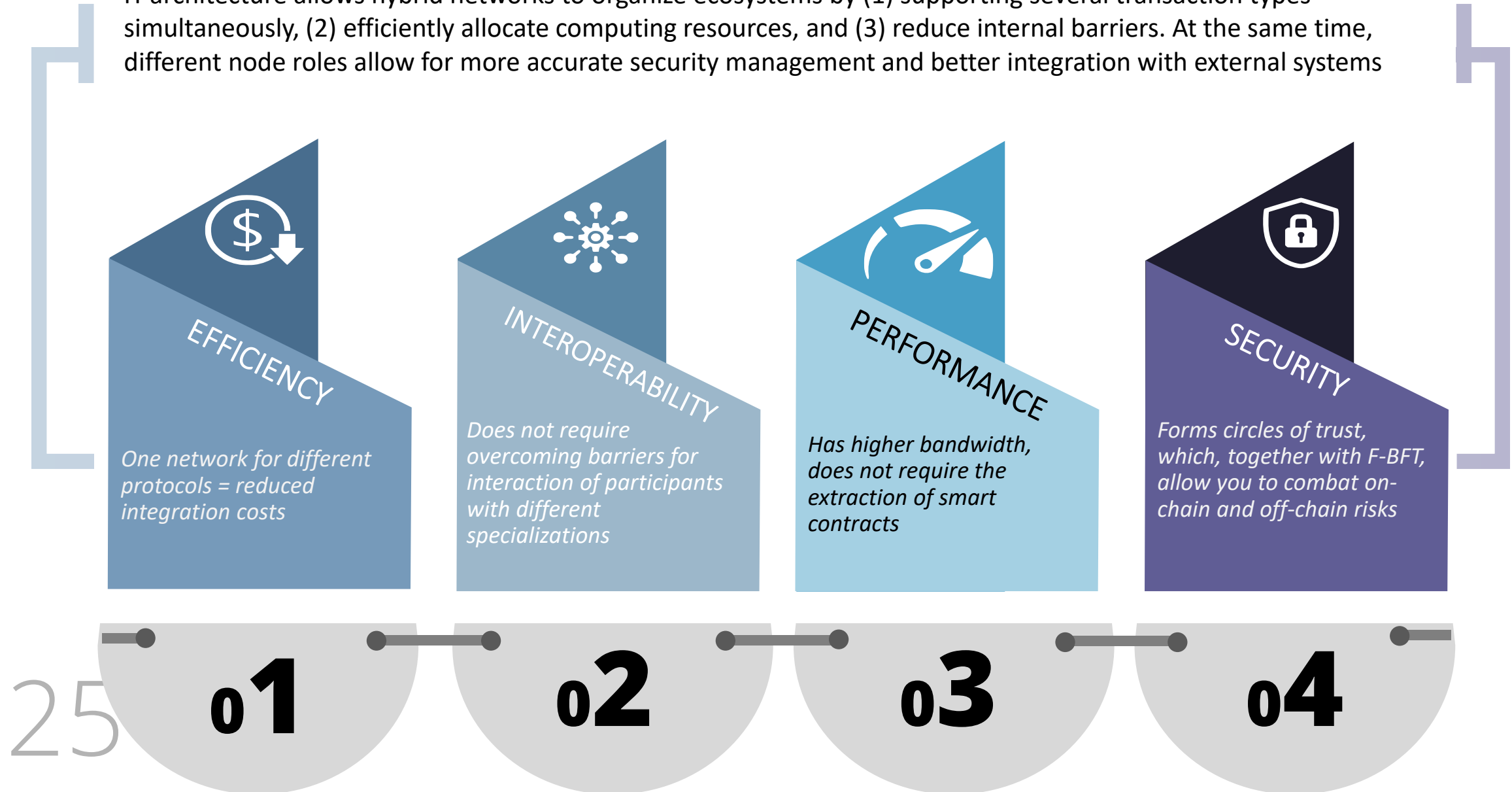
 Service Provider (SP) – a node that provides service for its clients connected via API

 Notary: an off-chain node commonly called an “oracle” but with off-network validation

 Relay (Gateway): a node that provides data synchronization with external blockchain networks and payment systems

H-NET ARCHITECTURE

H-architecture allows hybrid networks to organize ecosystems by (1) supporting several transaction types simultaneously, (2) efficiently allocate computing resources, and (3) reduce internal barriers. At the same time, different node roles allow for more accurate security management and better integration with external systems

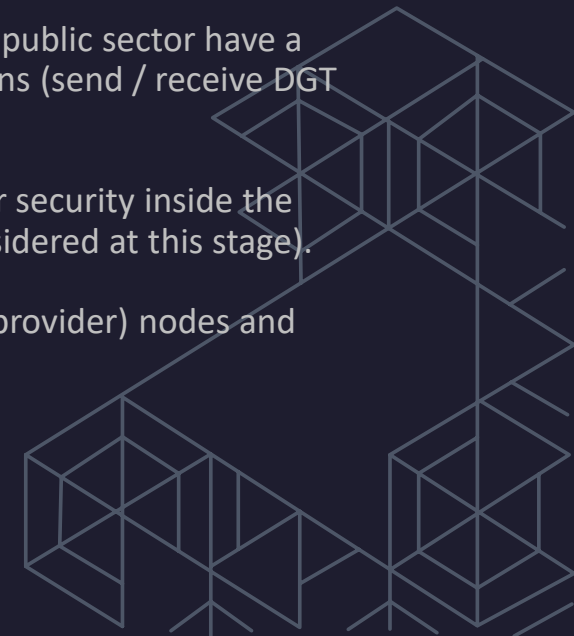


DGT IDMS is an identification management system. Its core functions include:

- Supporting the process of generating public and private keys. Keys are created on the client side, but require cryptographic unification, including the selection of algorithm and coding method (keys can be created). In the current version, one cryptographic system for the entire network is proposed. Keys can be created for a user, node, digital object, tokenization event, etc.
- Creating a decentralized DID identifier and its body (DID – DOD document); assigning a public key to DID and verifying whether a DID corresponds to an identified object. Also, assigning additional public keys, deleting DID, etc.
- Creating and verifying a set of properties for an identified object (Verifiable Credentials, VC)
- Storage of Verifiable Credentials (VC, informative object attributes, including quasi-identifiers); including their decryption and generating representations

Core solutions:

- While most SSI systems aim towards ensuring interoperability among different blockchains, DGT primarily solves an internal problem – a unified ID. Interoperability will be phased in during the next DGT version (Athabasca).
- The identification process will be conducted both for user-subjects and for different object types. Users of the network's public sector have a limited selection of verifiable attributes, but can also receive decentralized DID identifiers for allowable public transactions (send / receive DGT currency)
- There is a division between OFF-CHAIN and ON-CHAIN operations. As per H-NET architecture, arbiters are responsible for security inside the network (ON-CHAIN), while OFF-CHAIN relies on notary (oracle) nodes to work with external systems (Gates are not considered at this stage).
- The network does not store long-term encrypted information that could be used for user-aimed attacks, but SP (service provider) nodes and arbiters are proxies for working with notaries, which in turn have access to encrypted storage.
- Development is carried out through an intermediate architecture approach, which contains simplifications.



Client, Agent (Identity Wallet)



Contains ID, including DID + VC

Request for creating DID (identification), then to store VC (verification), and if needed – request to verify VC (authorization)

Service Provider, SP

Plays a Proxy role for ID-based operations



Transactions for creating DID, adding and verifying VC

Validator Cluster



VC check – YES / NO

Replicative VC storage; encrypted

Notaries verify VC (off-chain), and lead the VC registry



Notary Ring

OFF-CHAIN

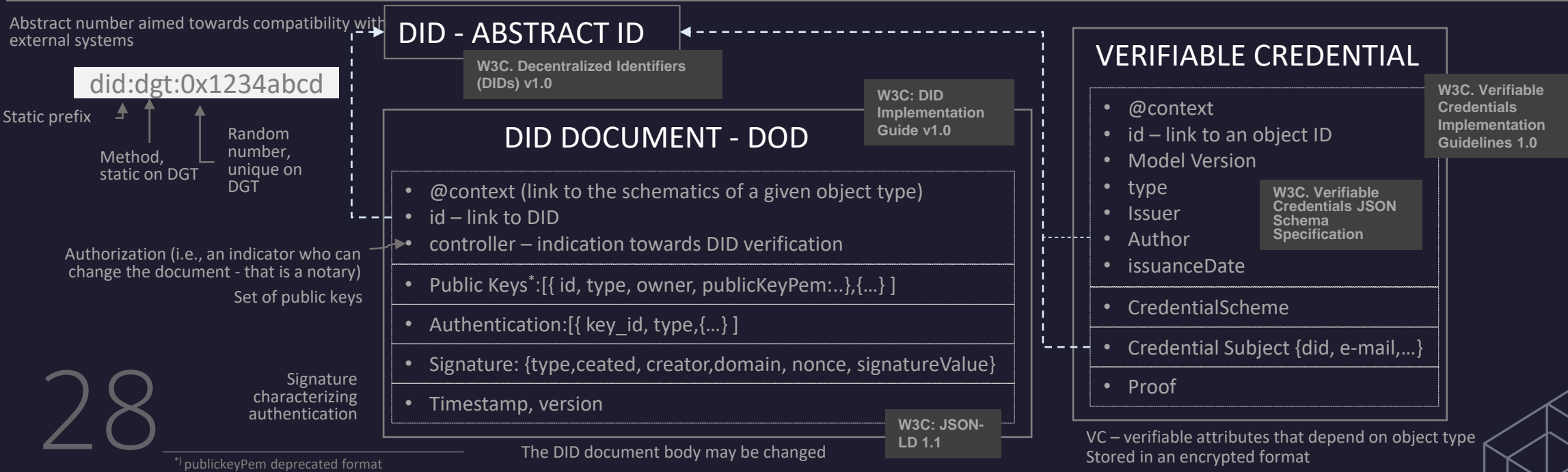
ON-CHAIN

Arbiter Ring



- When implementing DID, we take W3C specification into account.
- Each object (person, node, digital object, or even event – such as an emission) may receive a decentralized ID in the W3C ID format. In approximation, this DID (its body – see DOD) is a collection of information in the form of unique ID, object type, and one or more public keys. ID that belong to an object are managed by a relevant admin (the simplest case does not involve multi-signature).
- DID format: «did:dgt:0x1234abcd», where the URI (unique ID) is highlighted. The full identifier make only be received through the DOD format (JWT or JSON-LD¹), which will include one or more public keys, a set of authorization methods, a DID issuance time stamp, DID document update date, and cryptographic proof of integrity. The content of the DID-registry is located on-chain and does not include other information about the identified object.

1. **DID** – a unique identifier (just a number, but in a format that allows for later connection with other networks)
2. **DID (DOD) Document** – an expanded DID representation (practically its body); considering accepted standards, it is processed in JWT or JSON-LD standards. It is reminiscent of the X.509 certificate but can store several public keys and does not contain meaningful attributes.
3. **Verifiable Credentials (VC)** – any attribute for an identified object that allow for the object or its properties to be identified (including those that require verification). VC are stored in encrypted form outside the network (off-chain) and can only be accessed through oracles – notaries.



DID DOCUMENT (DOD)

DID-DOCUMENT defines the parameters of working with DID. It does not contain direct attributes and is stored in DAG (a separate branch). This is an updateable set that allows for the addition of public keys and methods in the form of incremental transactions, each of which adds to the DID status (if the operation is allowed – only for arbiters). DOD does not contain any encrypted elements. The format of working with DOD is JSON-LD¹.

```

{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://dgt.world/security/did/211123-people-v1"
  ],
  "id": "did:dgt:0x1234abcd",
  "controller": "did:example:123456789abcdefghi",
  "verificationMethod": [{
    "id": "did:dgt:0x1234abcd#key-1"
    "type": "Ed25519VerificationKey2020",
    "controller": "did:dgt:notary89abcdefghi",
    "publicKeyMultibase": "zAKJP3f7BD6W4iWEQ9jwndVTCBq8ua2Utt8EEj6Vxsf«
  },
  {
    "id": "did:example:123#_Qq0UL2Fq651Q0Fjd6TvnYE-faHiOpRIPVQcY_-tA4A",
    "type": "JsonWebKey2020",
    "controller": "did:example:123",
    "publicKeyJwk": {
      "crv": "Ed25519",
      "x": "VCpo2LMLhn6iWku8MKvSLg2ZAoC-nlOyPVQaO3FxFxVeQ",
      "kty": "OKP",
      "kid": "_Qq0UL2Fq651Q0Fjd6TvnYE-faHiOpRIPVQcY_-tA4A"
    }
  }
  "proof": {
    "type": "LinkedDataSignature2015",
    "created": "2016-02-08T16:02:20Z",
    "creator": "did:example:8uQhQMgZWxR8vw5P3UWH1ja#keys-1",
    "signatureValue": "QNB13Y7Q9...1tzn4w=="
  }
}

```

Context represents references to specifications that define verifiable data. First, there is a link to w3 (did-core), then to the specification of a given object type. Such a specification defines the existence of verification methods, possibilities of delegation, and verifiable objects (**to be defined later**)

`did:dgt:0x1234abcd` DID#

A link to the controller (its DID), who can change the document. There may be several controllers (indicated in accordance with the JSON standard).

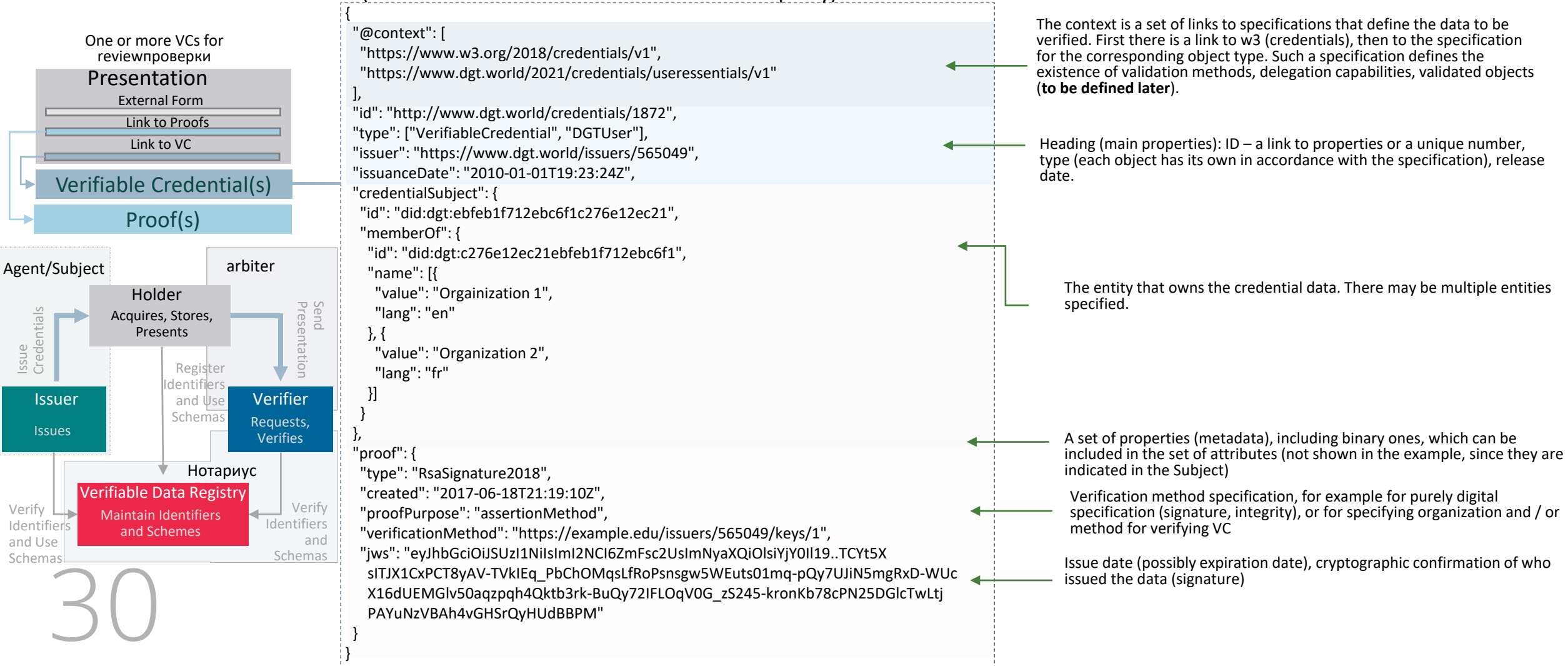
Validation method (proof of the ability to act with the given DID). There may be different methods for various situations, including delegating (then the appropriate inputs are used). In this given example, there are two keys one in the [Multibase](#) format (encoding), the other in [publicKeyJwk](#)

Each method may have its own controller, i.e., a node that validates the signature in DGT terms

The “basement” of the DID-document, which contains information about the document’s creation, updates, as well as a signature with the changes

¹) For example, see the library for working with JSON-LD in Python- <https://github.com/digitalbazaar/pyld>

The VC-DOCUMENT defines the representation of VC: a set of meta-information associated with a DID that is subject to verification. The [draft WC standard](#) (see also [here](#) and [here](#)) distinguishes between the VC itself and its presentation. VC sets already arise when registering a DID in the process of receiving checks (see list of checks), which are then attached to the existing DID. Even though in a fully decentralized world (network of networks) the lifecycle of a VC may be complex, in the context of DGT the VC objects are a set of verifiable data (key-value), which are not stored in the network for long, but retrieved and transferred for amendments to the notaries (the network itself works as a decentralized proxy).



GETTING DID

There are several operations that include creating a DID, changing the associated public keys, creating and linking accounts (VC) to a DID



The initializing event for a node, user, or object to receive a DID is to request it from its service-provider (that is, through the standard API). Depending on the type of the object identified, the applicant forms a Claim (application), which is a completed form in which the applicant includes the necessary data, some of which is encrypted for reading by the notaries.

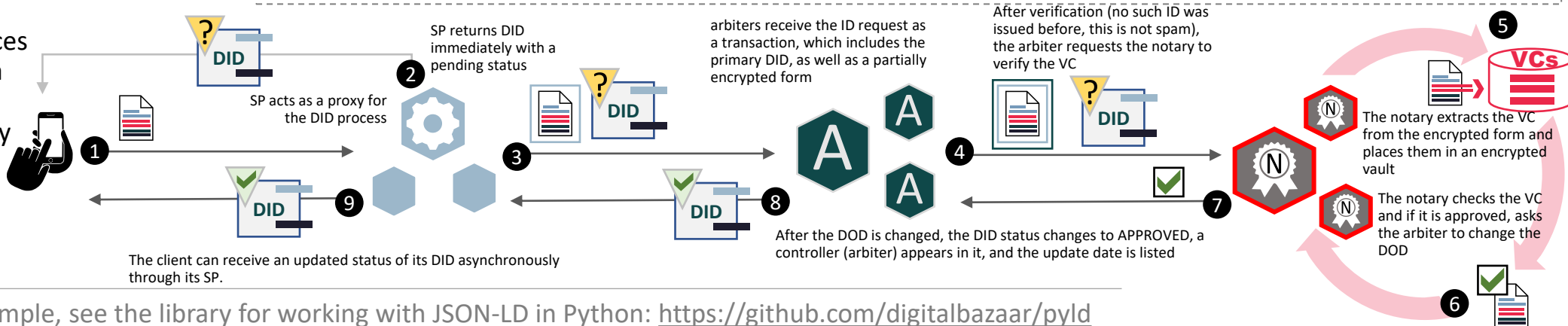
In return, the user immediately receives a “pending” DID (if the SP can guarantee the first communication request). SP does not have access to encrypted data and inserts it into a transaction, which is then approved by a cluster (for cryptographic integrity). Then there is a check on the arbiter level (guaranteeing no concurrent requests on other clusters).

To initiate receiving a DID, the user fills out a form that depends on the type of object. The closed information is a quasi-identifier, which is encrypted and then placed in the storage of VCs.

	USER	NODE	REALESTATE
Open information	Username E-Mail	Server name Admin E-Mail	Property Name Admin E-Mail
Closed information	[Phone number] Last Name/First Name Nationality (Citizenship) Date of Birth Gender	IP [Segment] [Organization] Expected End Users [...]	Property Identification Number Country, Region, City/Town Address Property Owner Description
Cryptographic additives to ensure integrity	[Current Location (Country, Region) [Organization]	Country, City [OS Version] [Node Role]	
	User Public Key User Signature	Node Public Key Node Signature	Property Public Key Owner Signature

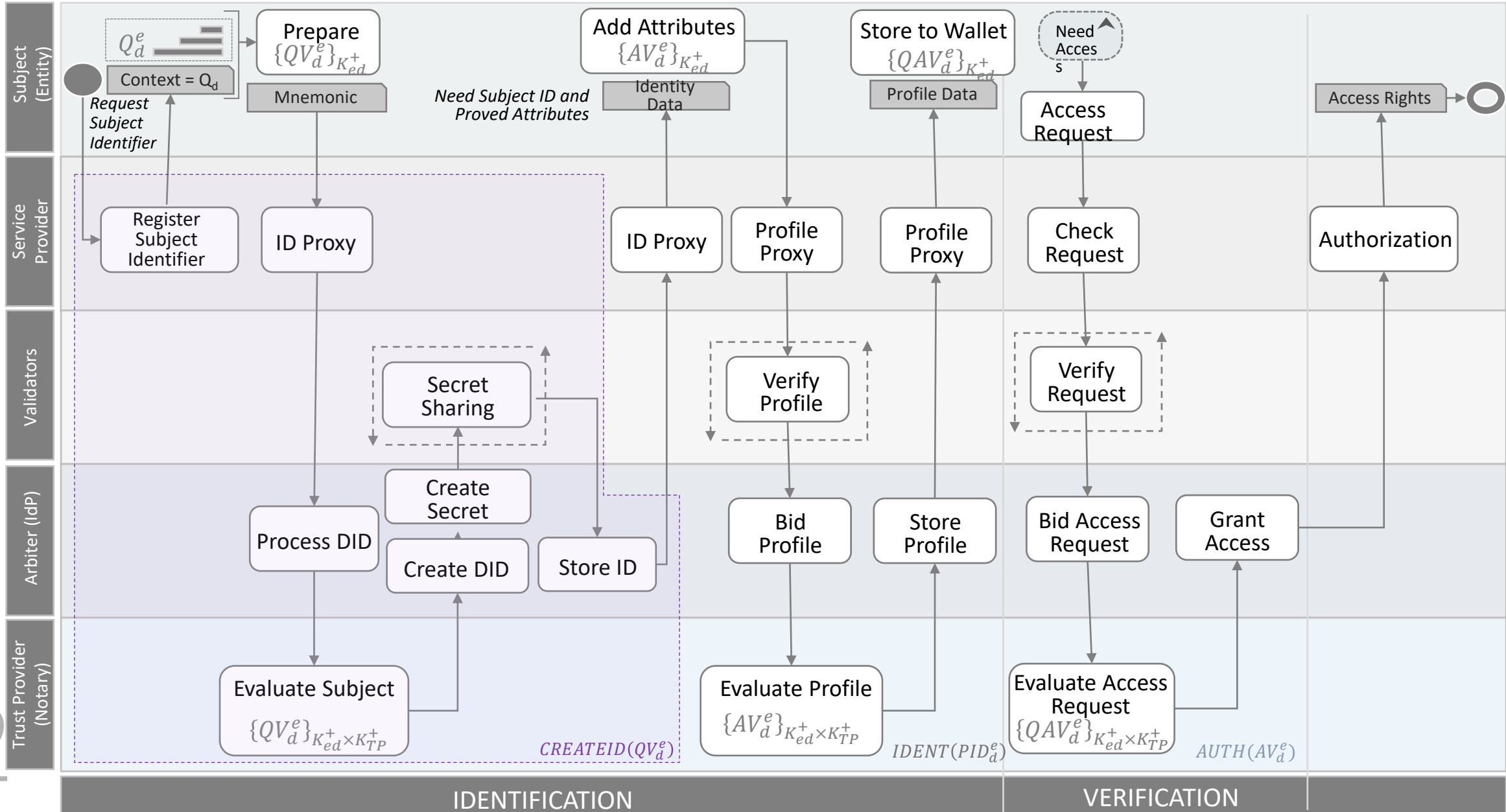
DID Claim

The arbiter requests a notary (this is a long process), who directly verifies the VCs, places them in storage, and sends a transaction to update the DOD. If successful, the notary changes the status.



1) As an example, see the library for working with JSON-LD in Python: <https://github.com/digitalbazaar/pyld>

DGT H-NET ID WORKFLOW



REACH OUT TO DGT

info@dgt.world



medium.com/@dgtworld



www.dgt.world

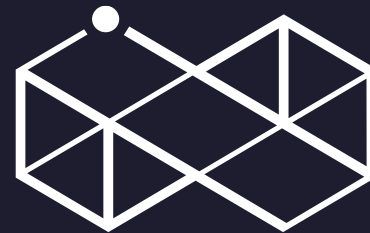
WWW



twitter.com/dgtnetwork



CONNECT TO



DGT

