A.D. 1308
unipg
DIPARTIMENTO
DI MATEMATICA E INFORMATICA

DIGITAL IDENTITIES AND IAM

Cybersecurity with Laboratory

Chiara Luchini

Anno Accademico 2023-2024

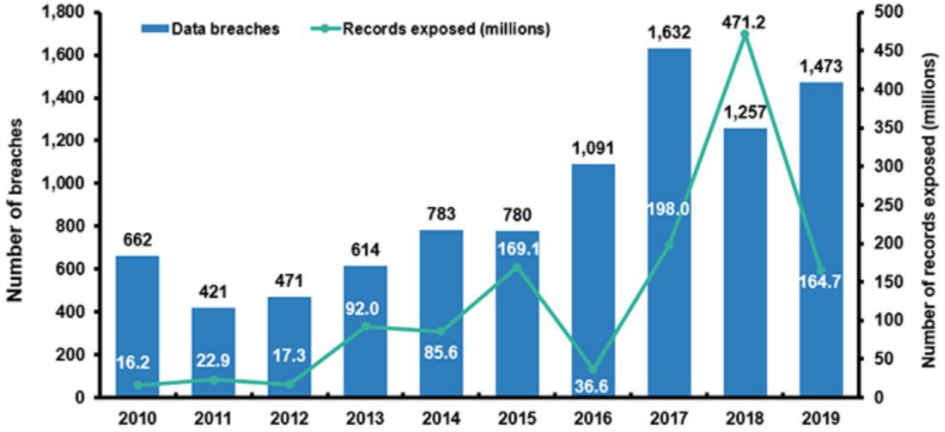
Digital Identities

Identity Theft

Identity theft *occurs when someone uses another person's personal* ***identifying information****, like their name, identifying number, or credit card number,* ***without their permission****, to commit fraud or other crimes.*

Identity Theft and Fraud

In 2019, 14.4 million consumers became victims of identity fraud. Overall, 33 percent of U.S. adults have experienced **identity theft**, which is more than twice the global average. More than one in four older adults, aged 55 and over, have experienced identity theft.



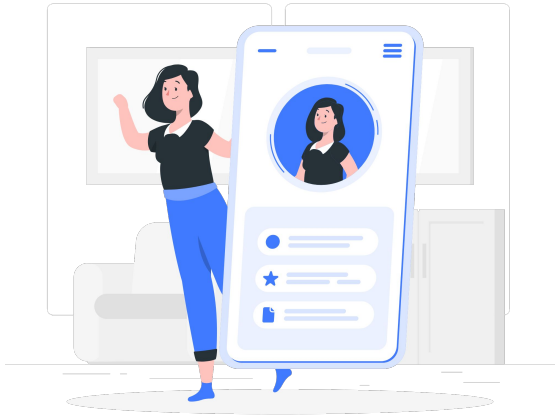
Source: [Identity Theft Resource Center]

Identity Theft and Fraud

Source: [Consumer Sentinel Network Data Book]

Digital Identity

*A digital identity is **information** on an **entity** used by computer systems to **represent** an **external agent**. That agent may be a person, organization, application, or device. ISO/IEC 24760-1 defines identity as **"set of attributes related to an entity"**.*



- They contain the **minimum of attributes** needed in that context
- Useful for **assessment** and **authentication**

Identity, identifier and account

The term “identifier” to refers to a **single attribute** whose purpose is to **uniquely identify** a person or entity, within a **specific context**.

- email addresses,
- passport numbers,
- driver’s license numbers,
- employee numbers.

*Nonhuman entities, such as agents, bots, or devices, may be identified by an **alphanumeric string of characters** assigned at their time of creation or registration within a context where they will act.*

Attributes

- Human identities may include attributes such as **name**, **age**, **address**, **phone number**, **eye color**, and **job title**.
- Nonhuman identities may include attributes such as an owner, **IP address.**
- The attributes which make up an identity may be used for authentication and authorization

*An online identity consists of **at least one identifier** and a **set** of **attributes** for a user or entity in a particular context, such as an application or suite of applications.*

Account

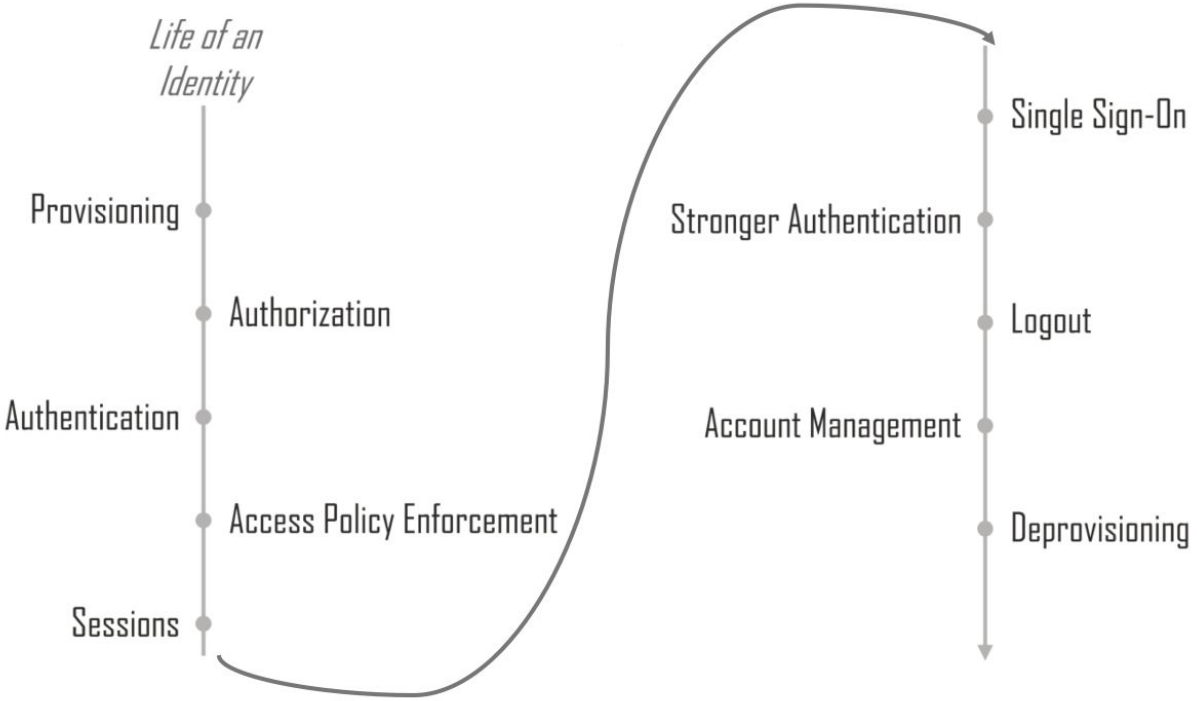
- An identity is associated with an **account** in each such context.
- Identity attributes may be **contained** within an application's account object, or they may be **stored separately** and referenced from the account object.

*An account is defined as a **local construct** within a **given application** or **application suite** that is used to perform **actions** within that context.*

Separation between ID/Account

- An account may have its own identifier in addition to that of the identity associated with it.
- Having an account identifier separate from the identity associated with the account provides a **degree of separation**.
- The account identifier can be used in other application records to **make it easier for users to change the username** or other identifier associated with their account.
- It should be noted that **an account can have more than one identity associated with it through account linking**.
- **Nonhuman actors can certainly have identities as well.**

Events in an identity life



Provisioning

- The act of **creating an account** and associated identity information is often referred to as **provisioning**.
- The objective of the provisioning phase is to establish an account with associated identity data.
- It involves **obtaining or assigning a unique identifier for the identity**, optionally a **unique identifier for the account** distinct from that of the identity, **creating an account** and **associating identity profile attributes with the account**.

Authorization

- When an account is created, it is often necessary to specify **what the account can do**, in the form of privileges.
- We use the term **authorization** for the **granting of privileges** that govern what an account is **allowed to do**.
- Authorization for an account is typically done at the time an account is **created** and may be **updated** over time.

Authentication

- A user provides an identifier to indicate the account they wish to use and enters login credentials for the account.
- These are validated against credentials previously registered during the account provisioning phase.

Authentication is a key aspect of trust-based identity attribution, providing a ***codified assurance*** *of the identity of one entity to another. Someone (or something) authenticates to* ***prove that they're the user they claim to be.***

Authentication Factors

Authentication factors are **methods** for proving a user's identity. They commonly fall into these basic types:

- **Knowledge**: "*something you know*" (ex. pin or password).
- **Possession**: "*something you have*" (ex. mobile phone, encryption key device).
- **Inherence**: "*something you are*" (ex. fingerprint, facial recognition, iris scan).

Multi-factor authentication

Multi-factor authentication (MFA) *is a user verification method that requires more than one type of authentication factors.*

MFA factors:

- Push notifications
- SMS notifications
- One-time passwords

Multi-channel protocol: A protocol where messages are sent on two or more at least **independent** channels.

Authentication vs. authorization

Authentication	Authorization
Determines whether users are **who they claim** to be	Determines what **users can and cannot access**
Challenges the user to **validate credentials** (for example, through passwords, answers to security questions, or facial recognition)	Verifies whether **access is allowed** through policies and rules
Usually done before authorization	Usually done after successful authentication
Generally, transmits info through an **ID Token**	Generally, transmits info through an **Access Token**
Generally governed by the **OpenID Connect (OIDC) protocol**	Generally governed by the **OAuth 2.0 framework**

Access Policy Enforcement

*An access control system **validates an identity's access** to a computing **resource**, which can be a service, storage space or online resource.*

- Authorization specifies what a user or entity is allowed to do, and access policy enforcement checks that a **user's requested actions** are **allowed** by the **privileges they've been authorized to use**.
- An **Access Policy** defines the **permissions** and **duration** of **access** to an Asset.

Access Control models

- **Discretionary Access Control (DAC)**: method of limiting access to resources (such as data sets) based on the **identity of users** or **groups** to which the users belong. **End users** have **total control** over their resources.
- **Mandatory Access Control (MAC)**: a method of limiting access to resources based on the **sensitivity of the information** that the resource contains and the **authorization** of the **user** to access information with that level of sensitivity. You define the sensitivity of the resource by means of a **security label** (ex. Top Secret, Secret, Restricted, Confidential, or Internal).
- **Attribute-Based Access Control (ABAC)**: policies consider **user attributes.**
- **Role-based Access control (RBAC)**: an access mechanism defined based on the concepts of **role and permission**.
- **Originator Control (ORCON)**: hybrid between MAC and DAC. Control privileges on an object can only be changed by the "**originator**" of the object.

Sessions

- Some applications, typically traditional web applications and sensitive applications, only allow a user to **remain active for a limited period** of time before requiring the user to authenticate again.
    - A session tracks information
- The **session timeout** settings will typically vary by the sensitivity of the data in the application.

Single Sign-On

Single sign-on (SSO) *is a mechanism that uses a* ***single action of authentication*** *to permit an authorized user to access all related, but independent software systems or applications* ***without being prompted to log in again*** *at each of them during a particular session.*

- After a user accesses one application, they may wish to do something else involving another application.
- SSO is the ability to **log in once** and then access additional protected resources or applications with the **same authentication requirements**, without having to **reenter credentials**.
- Single sign-on is possible when a set of applications has **delegated authentication to the same entity**.

Stronger authentication

- **Step-up authentication** is the act of **elevating** an existing authentication session to a higher level of assurance by **authenticating with a stronger form of authentication**.
- For example, a user might initially log in with a username and password to establish an authentication session.
- Later, upon accessing a more sensitive feature or application with **higher authentication requirements**, the user would be prompted for additional credentials, such as a **one-time password** generated on their mobile phone.

Logout

- At a minimum, the act of logging out should terminate the **user's application session**.
- If they return to the application, they would have to authenticate again before being grantaed access.
- In situations where **single sign-on** is used, there may be **multiple sessions to terminate**.
    - It is a design decision as to which sessions should be terminated when the user logs out of one application.

Account Management and Recovery

- During the course of an identity's lifetime, it may be necessary to **change various attributes** of the user profile for the identity.
    - For example, a user may need to update their email address or phone number, password, name.
- In a company, a user's profile might be updated to reflect a new position, address, or privileges such as roles
- Account **recovery** is a mechanism to validate a user is the legitimate owner of an account through some **secondary means** and then allow the user to establish **new credentials.**
    - **Lost password reset by email**

Deprovisioning

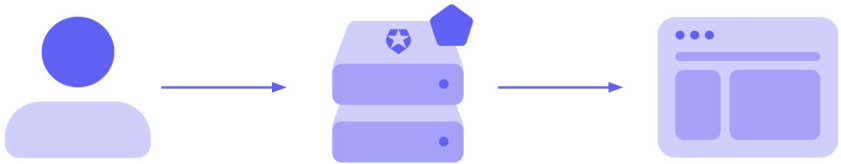
- There may come a time when it is necessary to **close an account**.
- In this case, the user's account and associated identity information must be deprovisioned so that it can no longer be used.
- Deprovisioning may take the form of completely **deleting the account** and associated identity information or simply **disabling** the account, to preserve information for audit purposes

Identity and Access Management

Identity and Access Management (IAM)

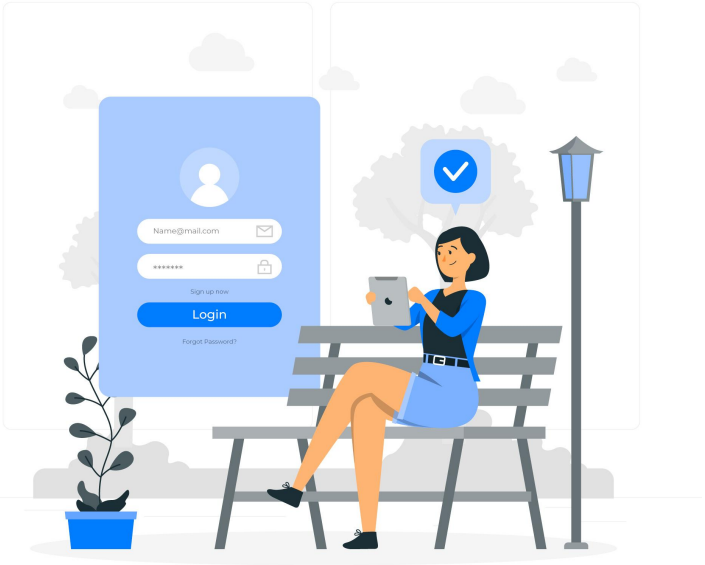
*An **Identity and Access Management (IAM)** is a set of services that support the **creation**, **modification**, and **removal** of identities and associated accounts, as well as the **authentication** and **authorization** required to access resources.*

A **digital resource** is any **combination of applications** and **data** in a computer system. Ex. web applications, APIs, platforms, devices, or databases.



Identity and access management verifies the user and controls their access to the resource.

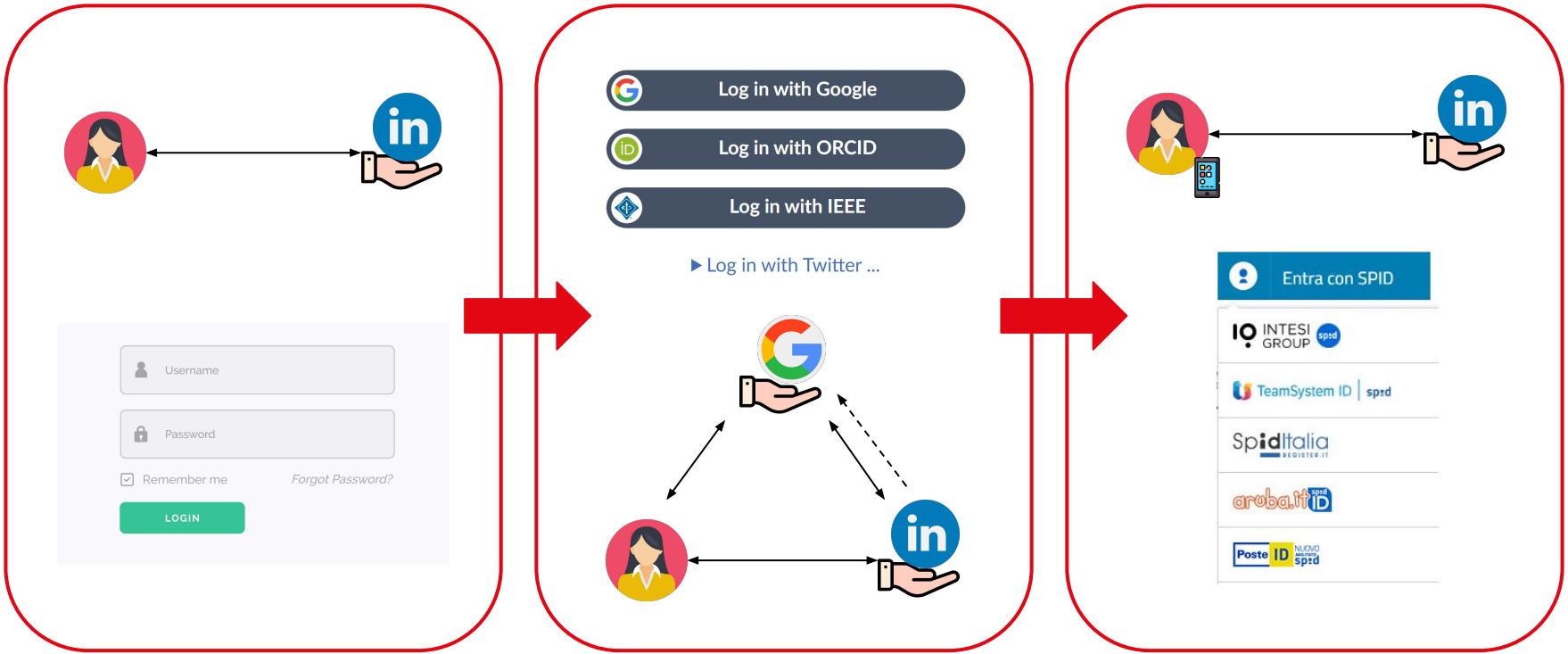
Identity and Access Management (IAM)



Verification of ownership:

- Identification
- Authentication
- Authorization

IAM evolution



Self-Sovereign Identity

Self Sovereign Identity (SSI) *is a sovereign, durable and portable identity for any person, organization or entity that allows its owner to access all digital services using* ***verifiable credentials*** *in a* ***privacy-preserving*** *manner.*

Why SSI?

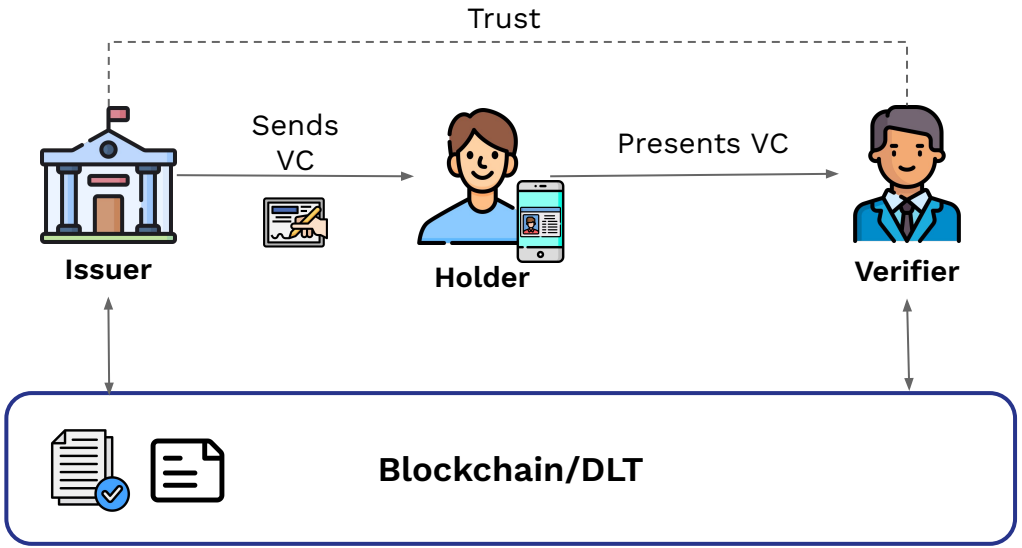
A July 2019 study by MobileIon reported in Security InfoCenter said that when users encounter password troubles:

- 68% feel disrupted.
- 63% feel irritated and frustrated.
- 62% feel they have wasted time.

Moreover

- IT security leaders felt they could reduce their risk of breach by almost half (43%) by eliminating passwords.
- 86% of those security leaders would do away with passwords if they could.

Self-Sovereign Identity



Verifiable Credentials

*A **verifiable credential** is a set of **metadata** and **claims** that cryptographically prove who issued it.*

Decentralized Identifiers

URN-encoded decentralized identifier.

Decentralized Identifiers

*A **Decentralized Identifier (DID)** is a new type of globally unique identifier encoded using a **Uniform Resource Name (URN)**. It provides a verifiable and decentralized means for interacting with a DID Subject controlling the DID.*

An example DID is:

did:sov:WRfXPg8dantKVubE3HX8pw

- **did** indicates that it is a DID;
- **sov** is the DID Method Name for Sovrin DIDs. All DIDs support the same basic functionality, but they differ in how that functionality is **implemented**, e.g. how exactly a DID is created or where and how a DID's associated DID document is stored and retrieved;
- **WRfXPg8dantKVubE3HX8pw** identifies the DID subject.

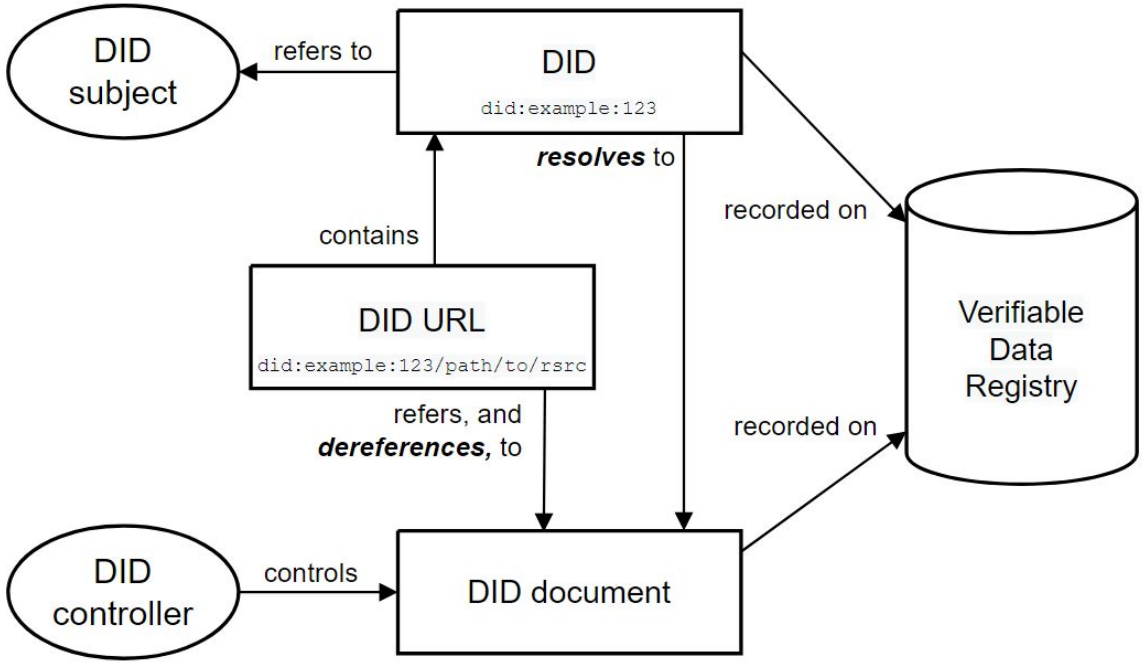
DID Methods

- Different DID “methods”:
  - did:sov:WRfXPg8dantKVubE3HX8pw
  - did:btcr:xz35-jzv2-qqs2-9wjt
  - did:v1:test:nym:3AEJTDMSxDDQpyUftjuoeZ2Bazp4Bswj1ce7FJGybCUu
  - did:uport:2omWsSGspY7zhxaG6uHyoGtcYxoGeeohQXz
  - did:erc725:ropsten:2F2B37C890824242Cb9B0FE5614fA2221B79901E

- DID methods need a method specification.
- Define method-specific syntax.
- Define method-specific **CRUD** operations: **Create, Read (Resolve), Update, Delete (Revoke)**

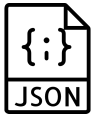
Method	DID Prefix
Sovrin	**did:sov:**
Veres One	**did:v1:**
uPort	**did:uport:**
Bitcoin	**did:btcr:**
Blockstack	**did:stack:**
ERC725	**did:erc725:**
IPFS	**did:ipid:**

DID architecture



DID Resolution

- DID Resolution: DID → DID Document



  - Set of public keys
  - Set of service endpoints
  - Authentication methods
  - Timestamps, proofs
  - Other identifier metadata

- May be dynamically constructed rather than actually stored in this form.
- Can support resolution parameters.
- Can return resolution metadata.

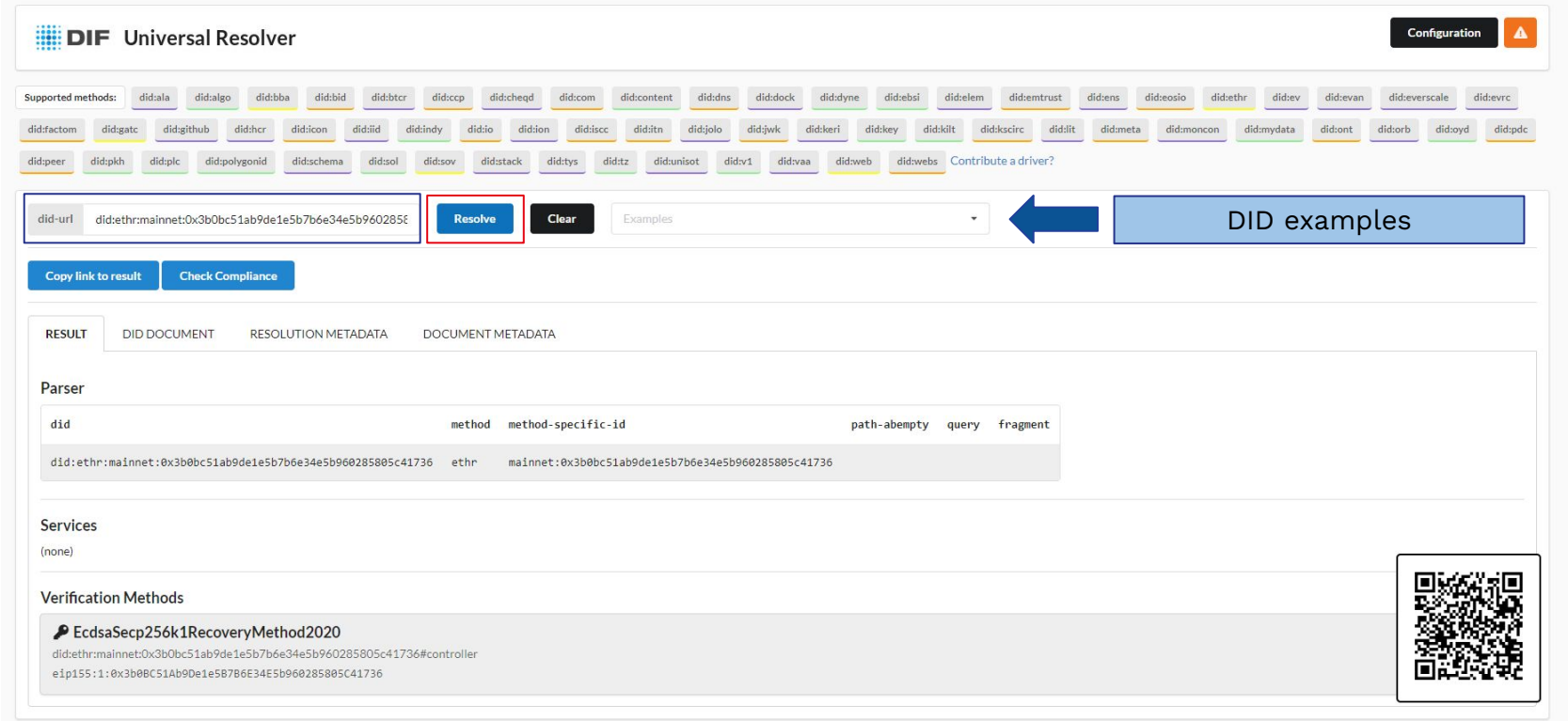
DID Universal Resolver

The **Universal Resolver** resolves **Decentralized Identifiers (DIDs)** across many different **DID methods**, based on the W3C DID Core 1.0 and DID Resolution specifications. It is a work item of the DIF Identifiers&Discovery Working Group.

- Looks up (“resolves”) DID to its DID Document.
- Provides a universal API that works with all DID methods.
- Uses a set of configurable “drivers” that know how to connect to the
- target system.

Link: https://dev.uniresolver.io/
Github link: https://github.com/decentralized-identity/universal-resolver

DID Universal Resolver



Example: https://dev.uniresolver.io/#did:ethr:mainnet:0x3b0bc51ab9de1e5b7b6e34e5b960285805c41736

DID document

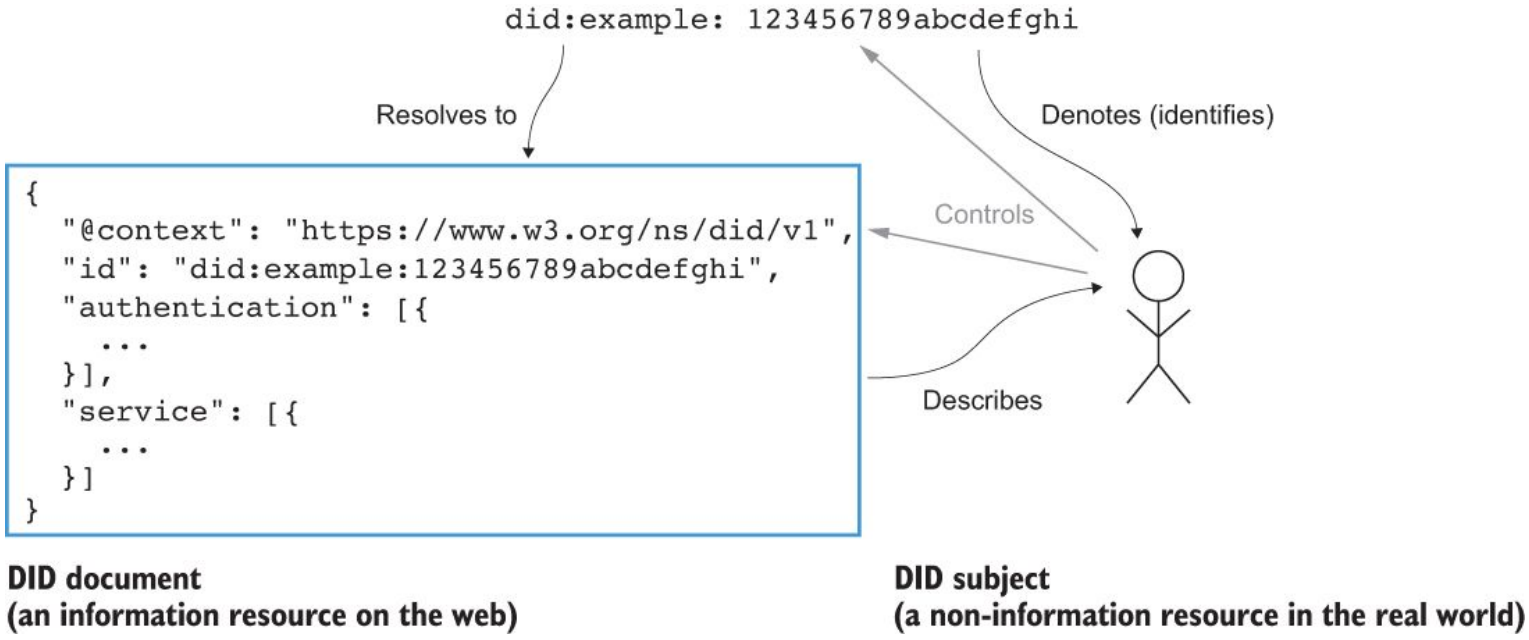
```
{
    "@context": [
        "https://www.w3.org/ns/did/v1",
        "https://w3id.org/security/suites/ed25519-2020/v1"
    ]
    "id": "did:example:123456789abcdefghi",
    "controller": "did:example:123456789abcdefghi",

    "authentication": [{

        "id": "did:example:123456789abcdefghi#keys-1",
        "type": "Ed25519VerificationKey2020",
        "controller": "did:example:123456789abcdefghi",
        "publicKeyMultibase": "zH3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV"
    }]

```

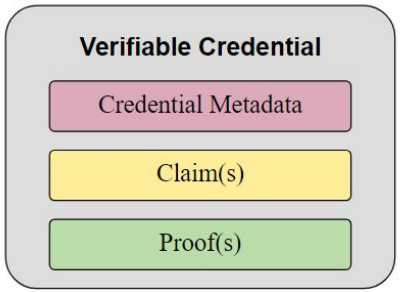
DID example


did:example: 123456789abcdefghi
Resolves to
Denotes (identifies)
{
"@context": "https://www.w3.org/ns/did/v1",
"id": "did:example:123456789abcdefghi",
"authentication": [{
...
}],
"service": [{
...
}]
}
Controls
Describes
DID document
(an information resource on the web)
DID subject
(a non-information resource in the real world)

Verifiable Credentials

*A **credential** is a set of one or more claims made by an issuer.*

*A **verifiable credential** is a set of tamper-evident claims and metadata that cryptographically prove who issued it.*



Credentials might also include an identifier and metadata to describe properties of the credential, such as:

- the issuer;
- the expiry date and time;
- a representative image;
- a public key to use for verification purposes;
- the revocation mechanism and so on.

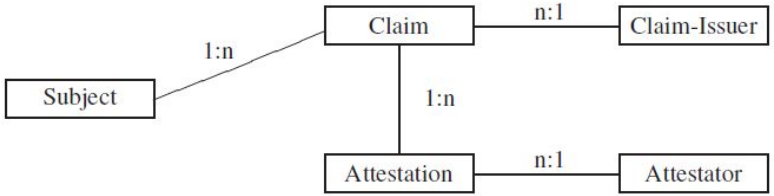
VC example

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "id": "http://example.edu/credentials/3732",
  "type": ["VerifiableCredential", "UniversityDegreeCredential"],
  "issuer": "https://example.edu/issuers/565049",
  "issuanceDate": "2010-01-01T00:00:00Z",
  "credentialSubject": {
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    "degree": {
      "type": "BachelorDegree",
      "name": "Bachelor of Science and Arts"
    }
  }
}
```

The example uses two types of identifiers. The first identifier is for the verifiable credential and uses an **HTTP-based URL**. The second identifier is for the subject of the verifiable credential (the thing the claims are about) and uses a **decentralized identifier**, also known as a DID.

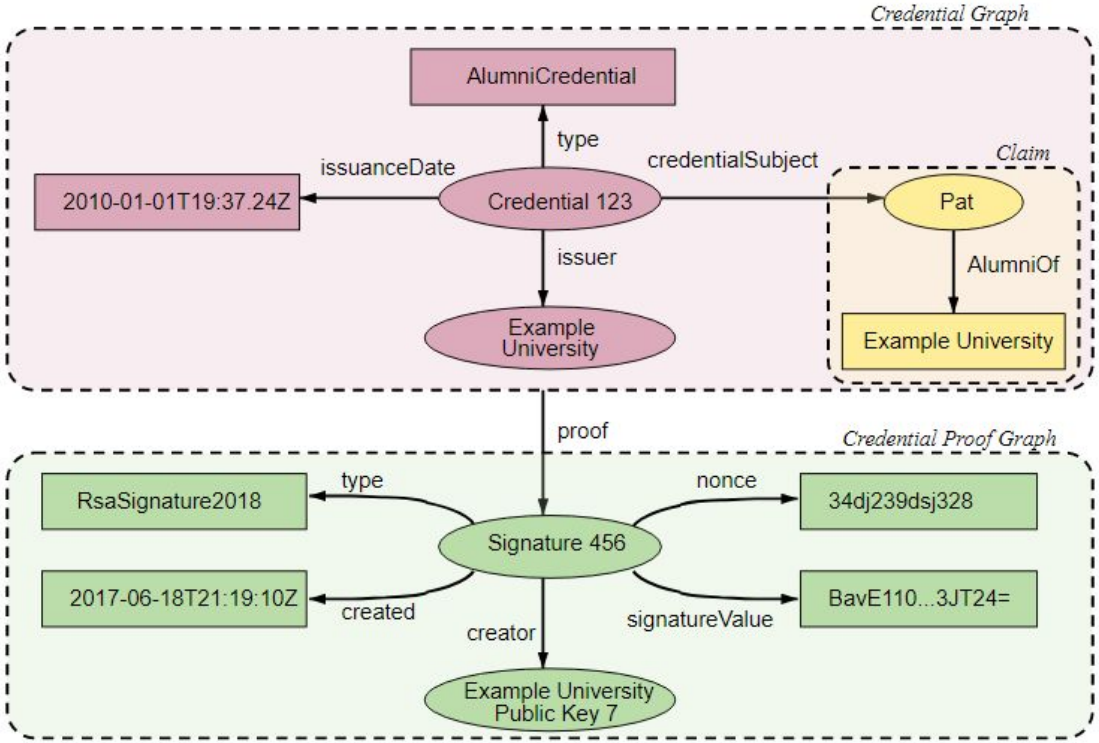
Attestation

Verifiable credentials are verifiable through a **signature** of an **attestation issuer** that has either issued the claim himself or can attest the correctness of it .



*An **attestation** can be seen as a **proof** in form of a **signature** attesting to a certain claim and meta data needed for verification .*

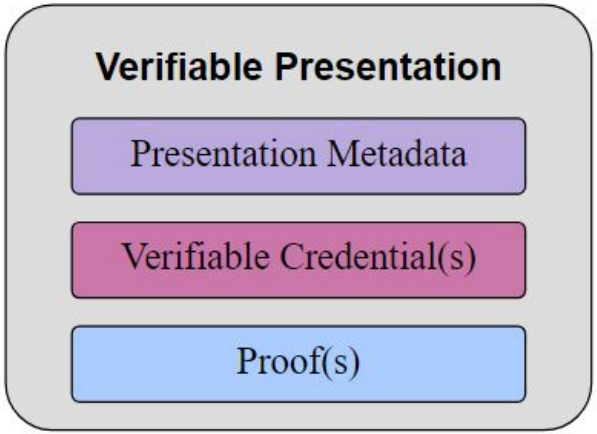
Verifiable Credential



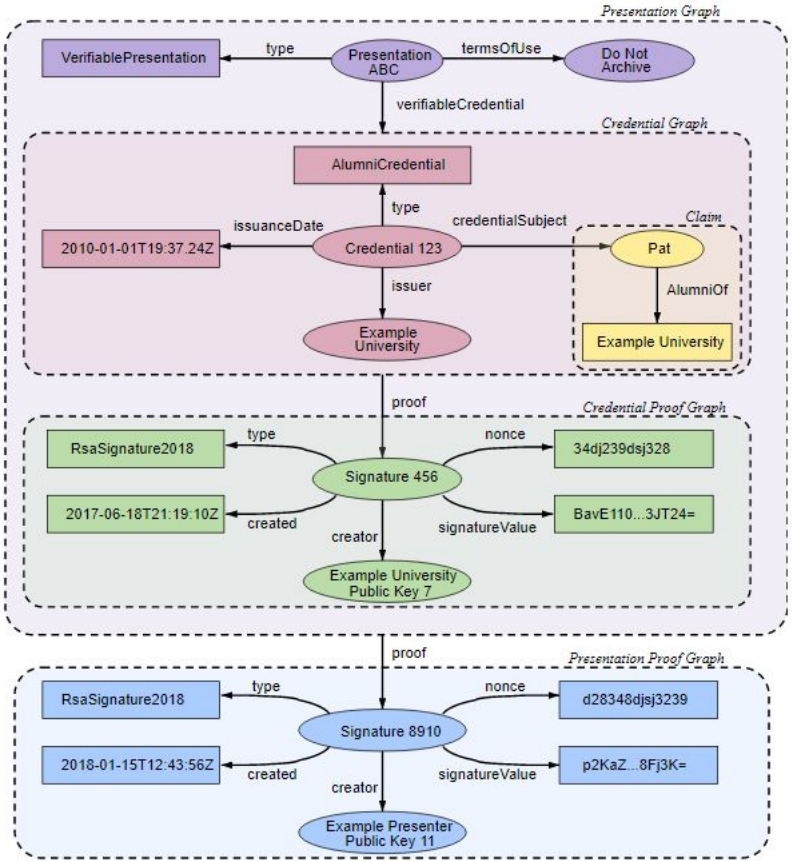
Verifiable Presentation

*A **verifiable presentation (VP)** expresses data from **one or more verifiable credentials** and is packaged in such a way that the authorship of the data is verifiable.*

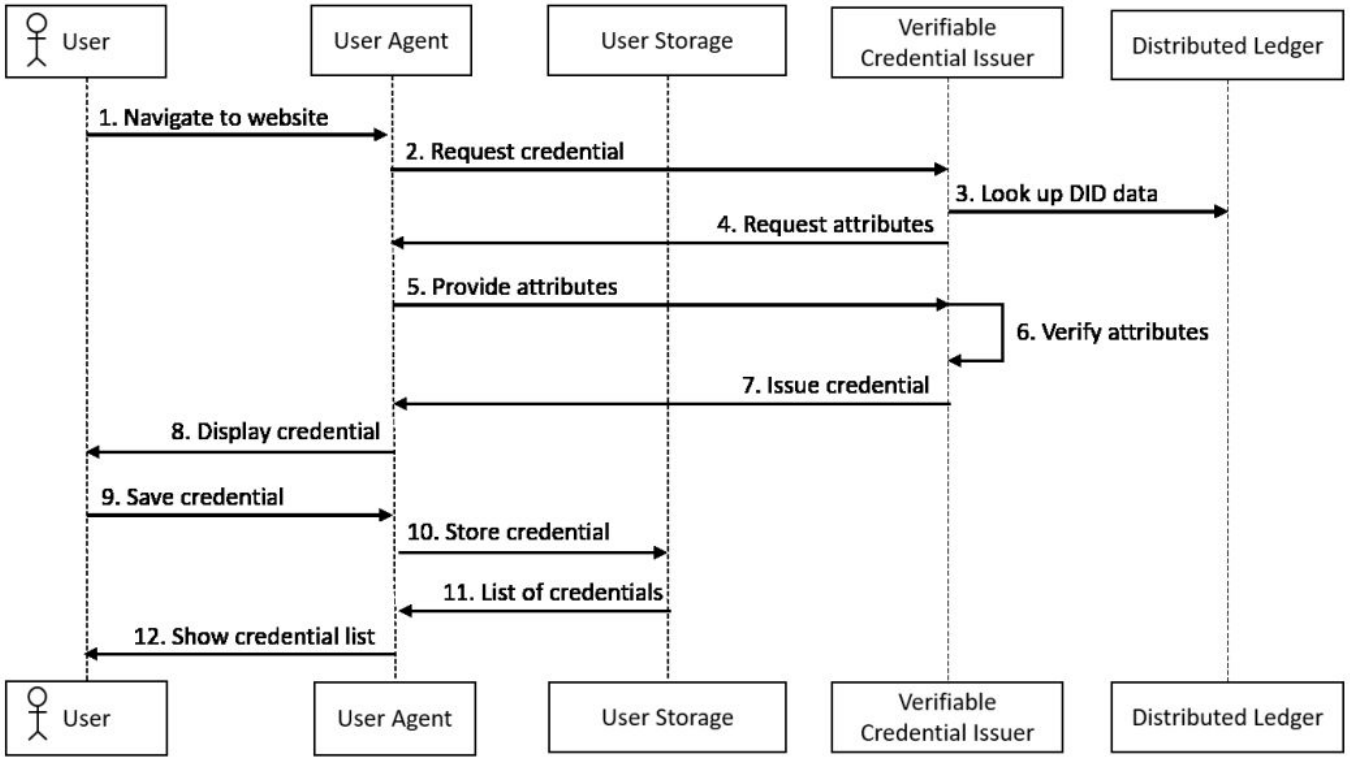
Presentations may be used to combine and present credentials. The data in a presentation is often about the same subject, but might have been issued by multiple issuers.



VP example

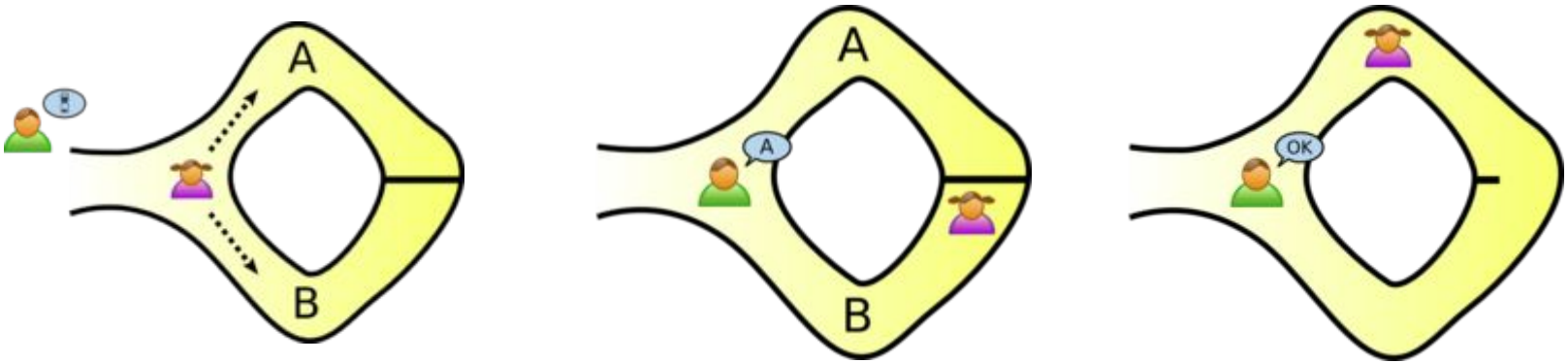

Presentation Graph
VerifiablePresentation
type
Presentation ABC
termsOfUse
Do Not Archive
verifiableCredential
Credential Graph
AlumniCredential
type
issuanceDate
credentialSubject
Claim
2010-01-01T19:37:24Z
Credential 123
Pat
issuer
AlumniOf
Example University
Example University
proof
Credential Proof Graph
RsaSignature2018
type
nonce
34dj239dsj328
Signature 456
2017-06-18T21:19:10Z
created
signatureValue
BavE110...3JT24=
creator
Example University Public Key 7
proof
Presentation Proof Graph
RsaSignature2018
type
nonce
d28348djsj3239
Signature 8910
2018-01-15T12:43:56Z
created
signatureValue
p2KaZ...8Fj3K=
creator
Example Presenter Public Key 11

VC workflow


User
User Agent
User Storage
Verifiable Credential Issuer
Distributed Ledger
1. Navigate to website
2. Request credential
3. Look up DID data
4. Request attributes
5. Provide attributes
6. Verify attributes
7. Issue credential
8. Display credential
9. Save credential
10. Store credential
11. List of credentials
12. Show credential list
User
User Agent
User Storage
Verifiable Credential Issuer
Distributed Ledger

Zero-Knowledge Proof

Zero Knowledge Proof (ZKP) *is a digital method whereby one party proves to another party the* ***possession of information without revealing it****.*



References

- *Identity fundamentals,* Auth0. Available at: https://auth0.com/docs/get-started/identity-fundamentals
- Mühle, Alexander, et al. "A survey on essential components of a self-sovereign identity." Computer Science Review 30 (2018): 80-86.
- Preukschat, Alex, and Drummond Reed. Self-sovereign identity. Manning Publications, 2021.
- Epping, M. & Morowczynski, M., (2021) “Authentication and Authorization (v2)”, IDPro Body of Knowledge 1(10). doi: https://doi.org/10.55621/idpro.78
- *About SSO*, Google Workspace Admin Help. Available at: https://support.google.com/a/answer/60224?hl=en
- Radha, V., and D. Hitha Reddy. "A survey on single sign-on techniques." Procedia Technology 4 (2012): 134-139.
- Preukschat, A. (2018) “DID resolution: Given a did how do I retrieve its document?”, SSI Meetup. Available at: https://ssimeetup.org/did-resolution-given-did-how-do-retrieve-document-markus-sabadello-webinar-13/.

References

- P. Grassi, M. Garcia, e J. Fenton, «Strong Authentication», giu. 2017, doi: https://doi.org/10.6028/NIST.SP.800-63-3.
- Hultgren, Andrew J. "A Holistic View of Identity Theft Tax Refund Fraud." (2019).
- *Verifiable credentials data model V1.1*, W3C. Available at: https://www.w3.org/TR/vc-data-model/
- *Mobileiron research reveals 8 in 10 it leaders want to eliminate passwords and expect mobile devices to become primary authentication to the Enterprise* (2019) Business Wire. Available at: https://www.businesswire.com/news/home/20190627005221/en
- Commision, FEDERAL TRADE. "Consumer Sentinel Network Data Book 2021." (2022).
- Images designed by Freepik
- Icons designed by Flaticon