SnT

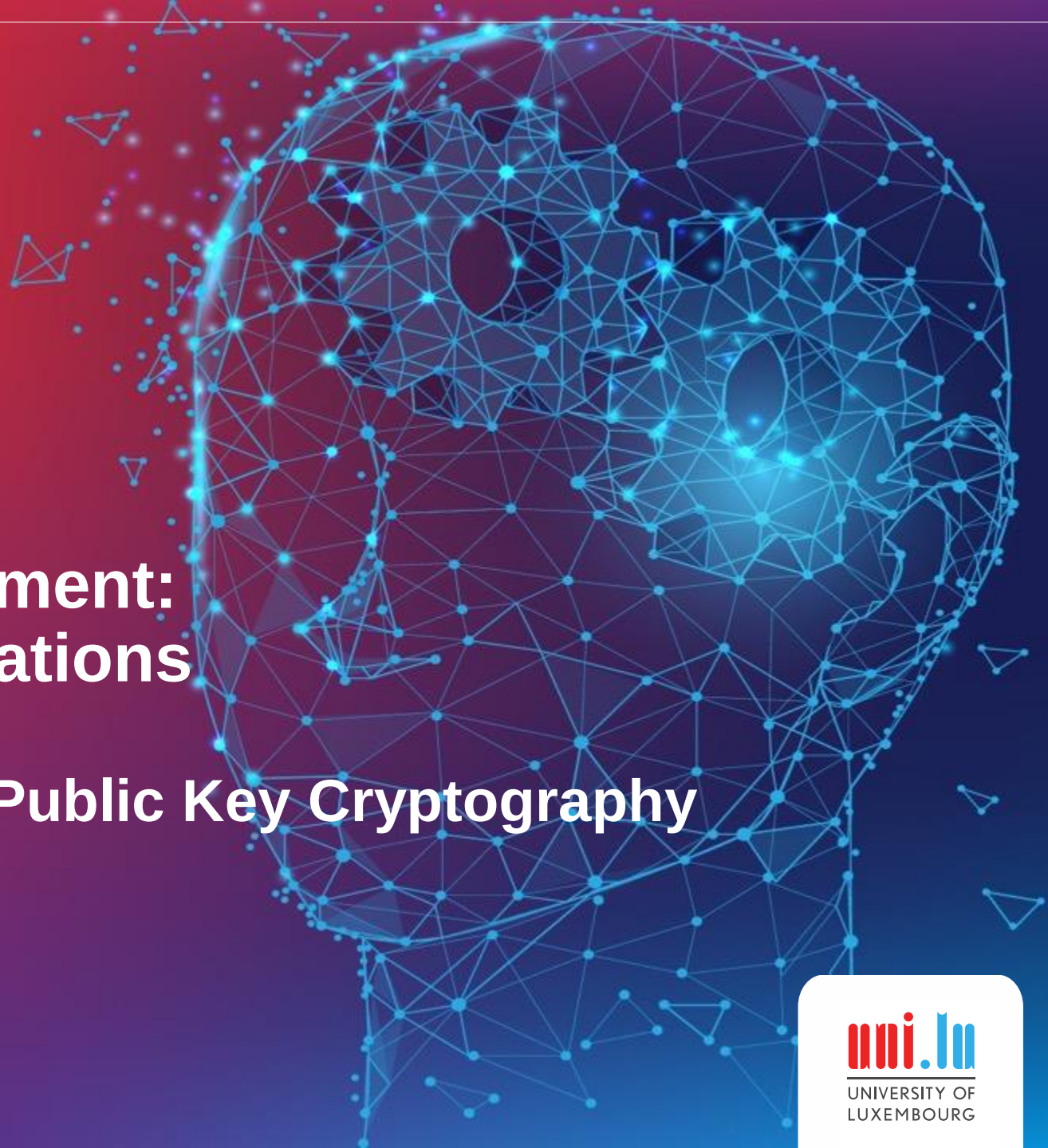
BAN444

Digital Identity Management: Technology and Applications

Technical Background I – Public Key Cryptography and Digital Certificates

Prof. Dr. Alexander Rieger
Dr. Johannes Sedlmeir

2025-01-06



UNIVERSITY OF LUXEMBOURG

SNT

Cryptographic Primitives

UNIVERSITY OF LUXEMBOURG

Definition of Cryptography

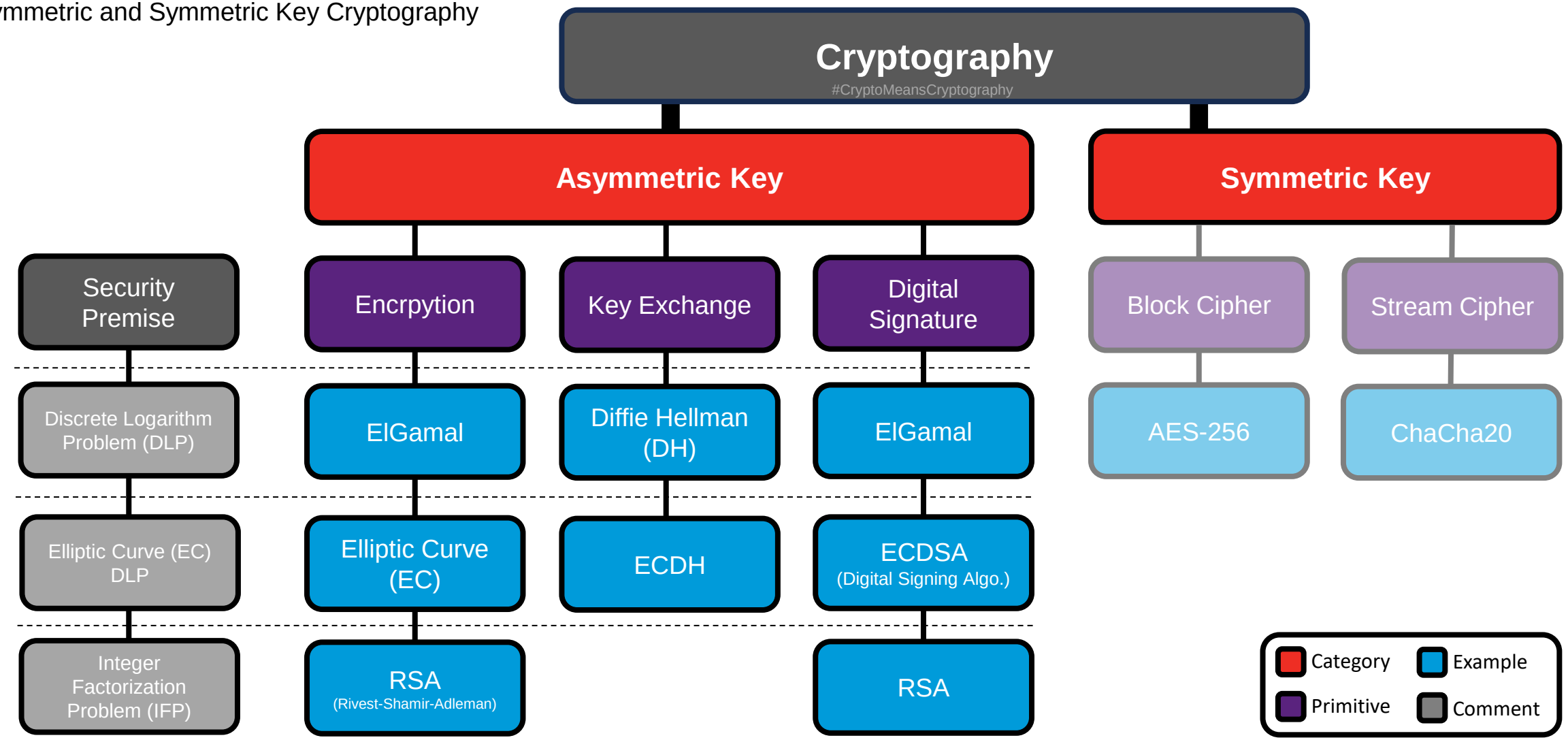
Cryptography, or **cryptology** (from Ancient Greek: κρυπτός, romanized: *kryptós* "hidden, secret"; and γράφειν *graphein*, "to write", or -λογία *-logia*, "study", respectively[1]), is the practice and study of techniques for secure communication in the presence of adversarial behavior.[2] More generally, cryptography is about constructing and analyzing protocols that prevent third parties or the public from reading private messages.[3] Modern cryptography exists at the intersection of the disciplines of mathematics, computer science, information security, electrical engineering, digital signal processing, physics, and others.[4] Core concepts related to information security (data confidentiality, data integrity, authentication, and non-repudiation) are also central to cryptography.[5] Practical applications of cryptography include electronic commerce, chip-based payment cards, digital currencies, computer passwords, and military communications.

or tampering

https://en.wikipedia.org/wiki/Cryptography

Cryptographic Primitives

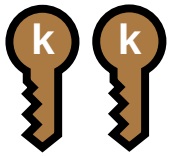
Asymmetric and Symmetric Key Cryptography



Source: Adapted from Kumar, C., Prajapati, S. S., & Verma, R. K. (2022). A Survey of Various Lightweight Cryptography Block ciphers for IoT devices.

Cryptographic Primitives

Asymmetric and Symmetric Keys: The Secret to securing information



Symmetric Keys: A shared secret [*k*]

- Speedy and simple
- Short-lived for secure communications
- Long-lived for secure datastores

- **Application**: Encryption, MAC, AEAD
- **Problem**: Key establishment and key storage

Asymmetric Keys: A public and secret key pair [*pk*, *sk*]

- Slow and generally long-lived
- Public key can be shared with anybody
- Private key [*sk*] has to be stored securely

- **Application**: Key establishment, signature, encryption
- **Problem**: Manage key pairs and trust infrastructure

See also: Technical Guideline – Cryptographic Algorithms and Key Lengths (BSI, 2023)

The Journey Ahead
Keys are like the keys to your home; guard them well and use the right one for the right lock, and if you share your keys, ensure trust to share only with trustworthy parties.

Examples of *secure* key sizes for reference (BSI, 2023)

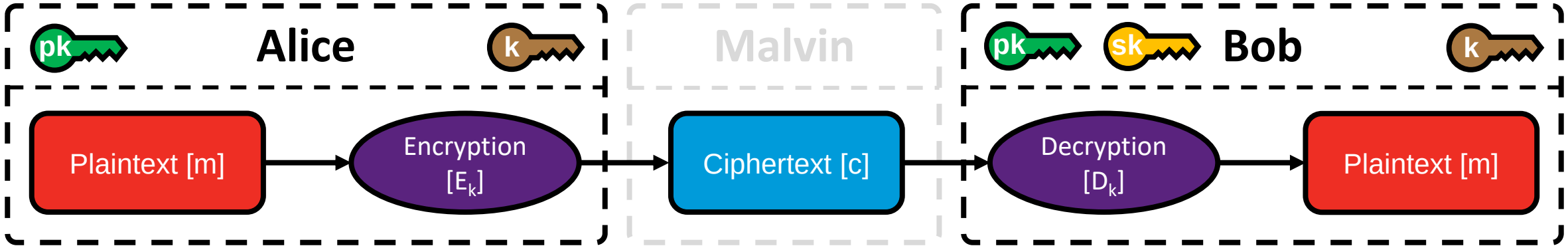
Symmetric:
- Block: AES (256 Bit)
- Stream: ChaCha20 (128 Bit)

Asymmetric:
- IFP: RSA (2048 Bit)
- ECDLP: ECC (256 Bit)
- PQC: Kyber (3168 Bit)

Cryptographic Primitives

Encryption and Decryption Operation: The very naïve approach

- **Encryption** [$E_k(m) = c$]: Converting plaintext message [m] into ciphertext [c] using a key [k]
- **Decryption** [$D_k(c) = m$]: Converting ciphertext [c] into plaintext message [m] using a key [k]



Considerations:

- **Key Management**: Secure generation, distribution, storage, and disposal of keys
- **Applications**: Secure communication, data storage, online transactions, and more
- **Security Goals**: Data confidentiality, integrity, and authenticity

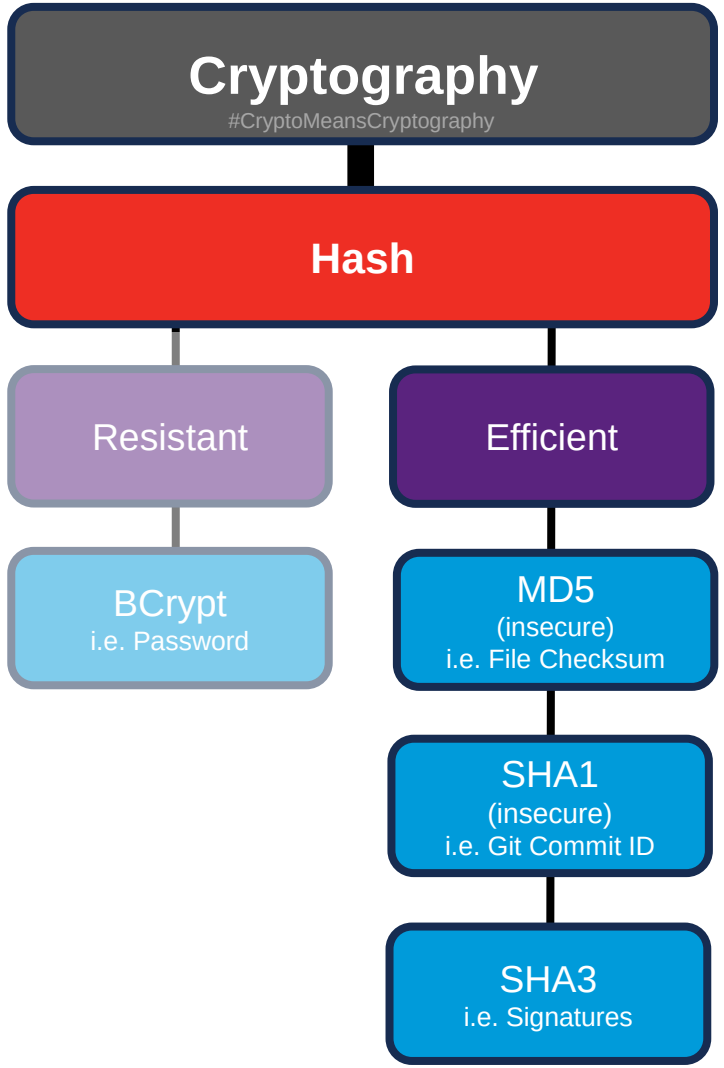
Cryptographic Primitives

Hash Function Characteristics

- **Deterministic**:
  For a given input, the output (hash) is always the same
- **Fixed Output Size**:
  Regardless of input size, the output hash is always of a fixed length
- **Avalanche Effect**:
  A minor change in input produces a major change in the output

Security Assumptions

- **Pre-Image Resistance**:
  Difficult to find the original input given a hash.
  Best via brute-force with $O(n) = 2^{n-1}$ average
- **Second Pre-Image Resistance**:
  Difficult to find a different input with the same hash as a given input
  SHA1 and SHA2 are broken an affected by a hash extension attack
- **Collision Resistance**:
  Difficult to find two different inputs that produce the same hash
  SHA1 and MD5 are broken and collisions have been found



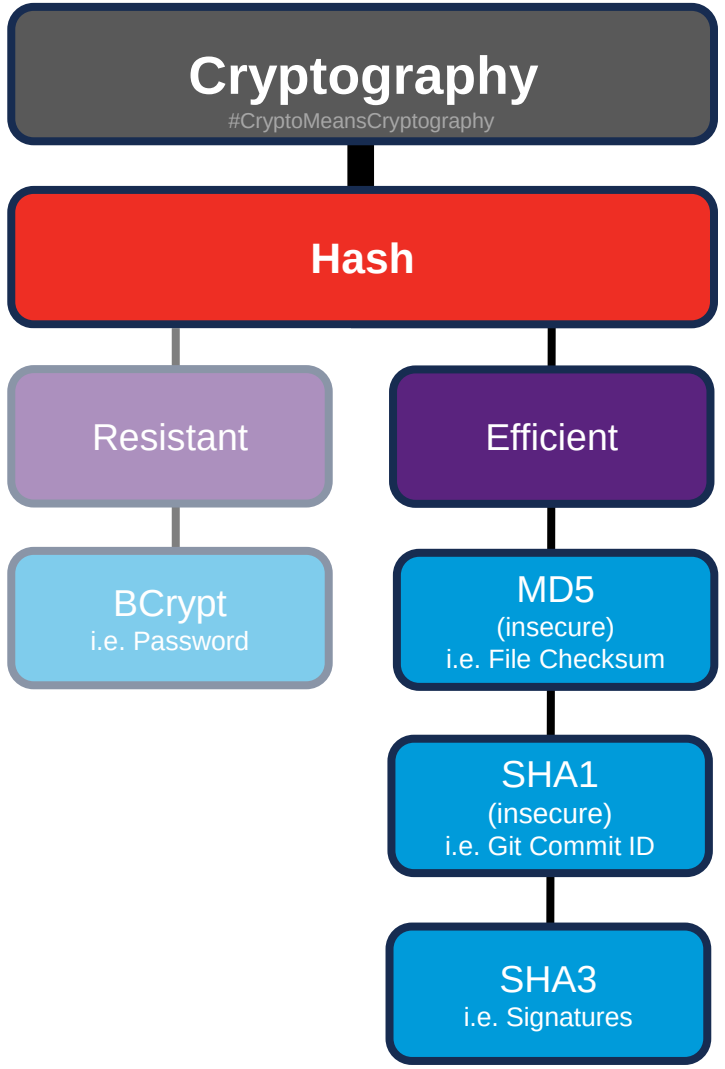
https://sha256algorithm.com/

Source: Adapted from Kumar, C., Prajapati, S. S., & Verma, R. K. (2022). A Survey of Various Lightweight Cryptography Block ciphers for IoT devices.

Cryptographic Primitives

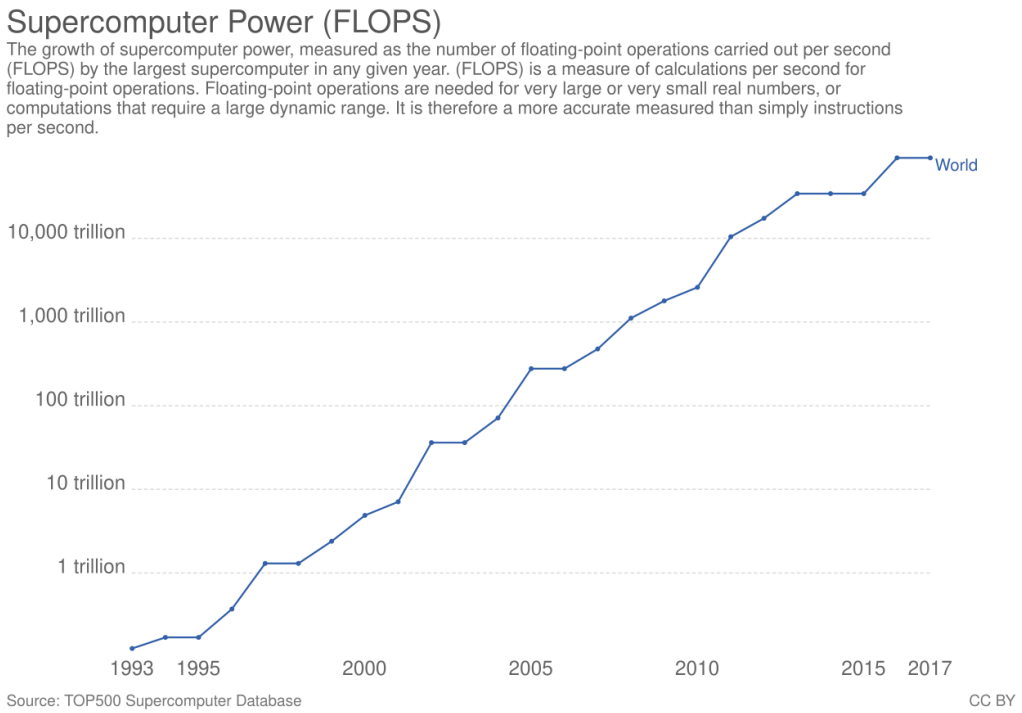
Hash Function Examples

- **md5('DiWa')**
  `12fbd61b3e4b23c23d348ae4b6bbe92c`
- **sha1('DiWa')**
  `65967194ca5d13f80126524bc7bc4a62648aa251`
- **sha1('DiWb')**
  `f84525cc1eafed187c555609151e5b5d45da31d7`
- **sha2('DiWa', 256)**
  `3cc1820fe5410c41153adaa561e2ecc91d50e16b7f3a22bd5fb2e99a74016e84`
- **sha3('DiWa', 256)**
  `54fe156fcb90230c2cbb7dd76bdbf300bed88ea71f2f9165624b45019d76ffee`
- **sha3('DiWa', 512)**
  `fc63f26ff055adecf1c280b65c2202e671c2abbb25363fa26d789119f725e854`
  `c6742c968b0f36432e7258894144d57c01055c4cfd49dace49fdcd8043001650`



See: https://cyberchef.org/#catHashing
Note: All hashes are display using HEX (Base-16) encoding, which is common – consider input and hashes as binary sequences.

All cryptography has a half-life!!



https://en.wikipedia.org/wiki/Supercomputer#/media/File:Supercomputer-power-flops.svg

https://link.springer.com/chapter/10.1007/978-3-642-03356-8_4

!! We keep building better and more computers, and we get better in breaking cryptographic constructions. !!

Many cryptographic primitives are built in a way such that increasing the number of digits (key size, digest (hash output) size) by one or two (in binary representation) doubles the difficulty of breaking the primitive. E.g., moving from SHA3-256 to SHA3-512 mutliplies the effort for a brute-force preimage attack by around a factor of 2^{256}, i.e., 10^{76}. For a brute-force collision attack, the factor would be 2^{128} or 10^{38}.

Discrete log and RSA-based kex exchange

In general, computing the logarithm is easy: $\log_2 50 = ?$ $2^5 = 32; 2^6 = 64; 2^{5.5} \approx 45; 2^{5.6} \approx 48.5; \ldots$

But it gets incredible hard once we work in finite fields (which is also useful because there are no roundings). Examples:

$$7 + 8 \equiv 2 \bmod 13$$

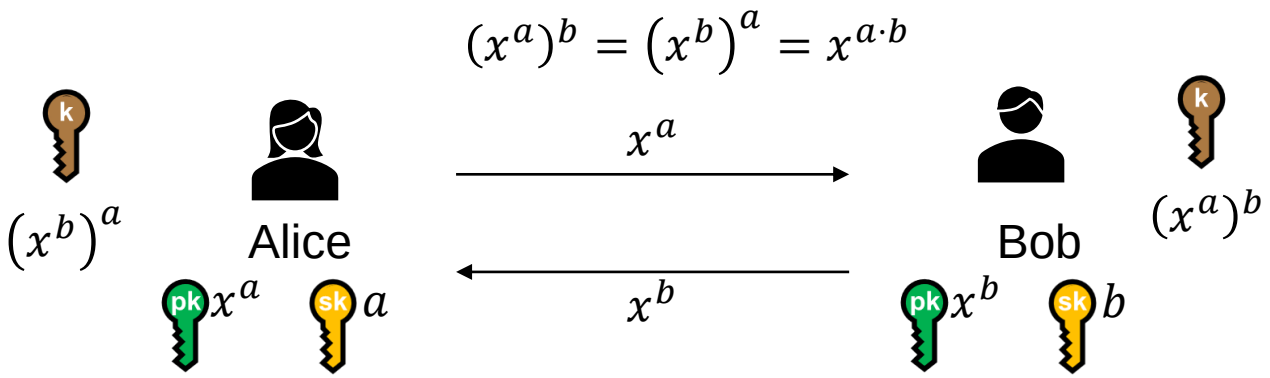
$$4 \cdot 15 = 60 \equiv 8 \bmod 13 = 4 \cdot 2 \bmod 13$$

$$2 \cdot 7 = 14 \equiv 1 \bmod 13 \;\Rightarrow\; \frac{3}{7} = 3 \cdot \frac{1}{7} = 3 \cdot 2 = 6 \bmod 13, \qquad \text{and } \frac{3}{7} \cdot 7 = 3 \equiv 42 \bmod 13.$$

$$3 = 81 \bmod 13 \equiv 3 \cdot 3 \cdot 3 \cdot 3 \bmod 13 = 27 \cdot 3 \bmod 13 = 1 \cdot 3 \bmod 13$$

$$2^5 = 32 \equiv 6 \bmod 13 \;\Rightarrow\; 5 = \log_2 6 \bmod 13$$

We will not dive into digital signatures, but if you believe me that finding a discrete log is hard (trial and error), you will already be convinced about the **Diffie-Hellman key exchange**:



Cryptographic Primitive Methods

Asymmetric Cryptography: Elliptic-Curves (EC)

Definition - Elliptic Curve:

Let E be an algebraic curve defined by the projective solutions,

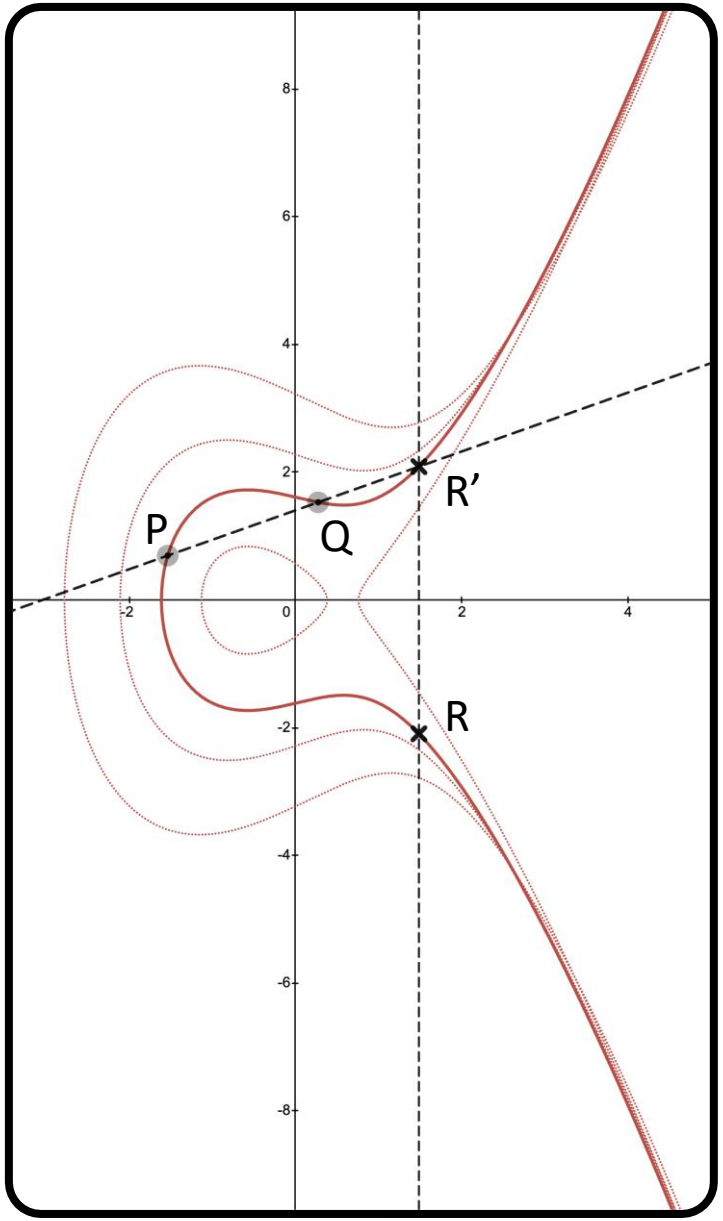
for some point $P = (x, y) \in F_p$ over a finite field $\mathbb{F}$,

of the equation $y^2 = x^3 + ax + b$ (Weierstrass equation)

with the neutral / identity element $I = P + I = P - P$ (point “at infinity” aka. the tip)

Basic Operations:

- Point addition: $R = P + Q$
- Point multiplication: $R = 2P$
- Elliptic curve groups: $R = kP$ (Double-and-Add optimization algorithm)



Try yourself: https://www.desmos.com/calculator/il0chq01kt

Cryptographic Primitive Methods

Asymmetric Cryptography: Elliptic-Curve Cryptography (ECC)

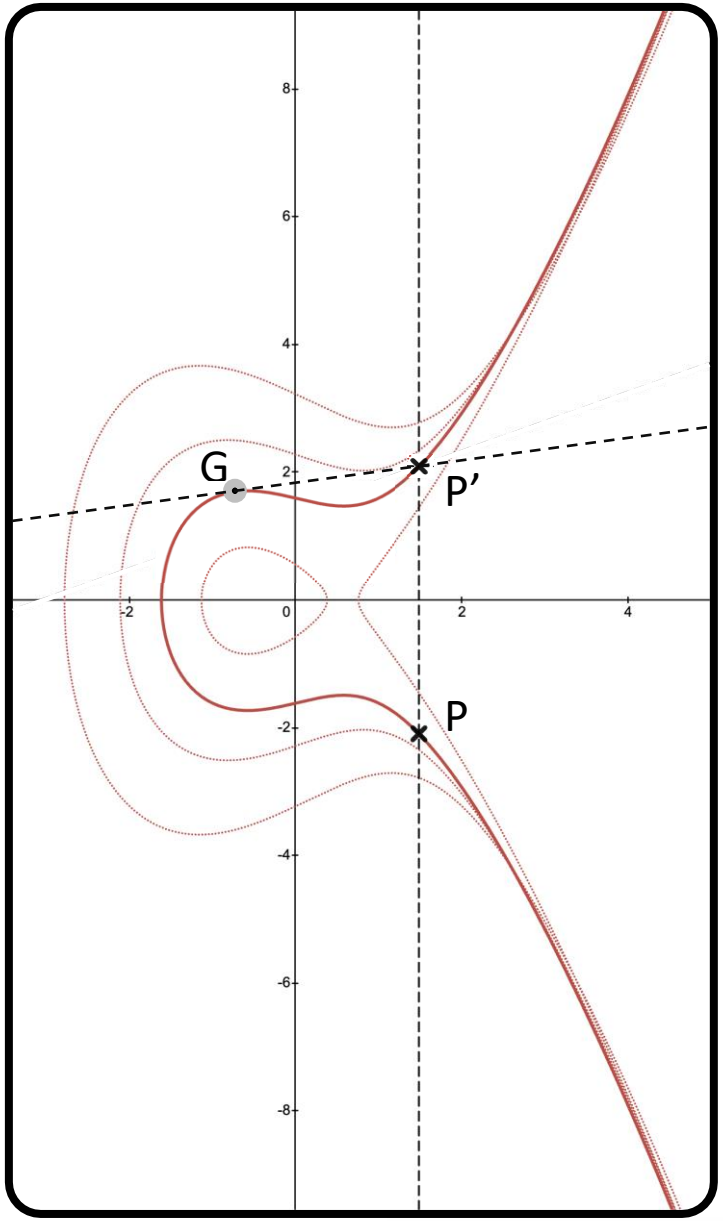
ECCryptography:

- [E]: Elliptic curve over a finite field [$\mathbb{F}_p$]
- [G]: generator point (fixed constant, a base point on the EC)
- [k]: private key (integer)
- [P]: public key (point)

- [c]: {kG, m + kP} (encryption)
- [m]: C − kP = (m + kP) − k(kG) (decryption)

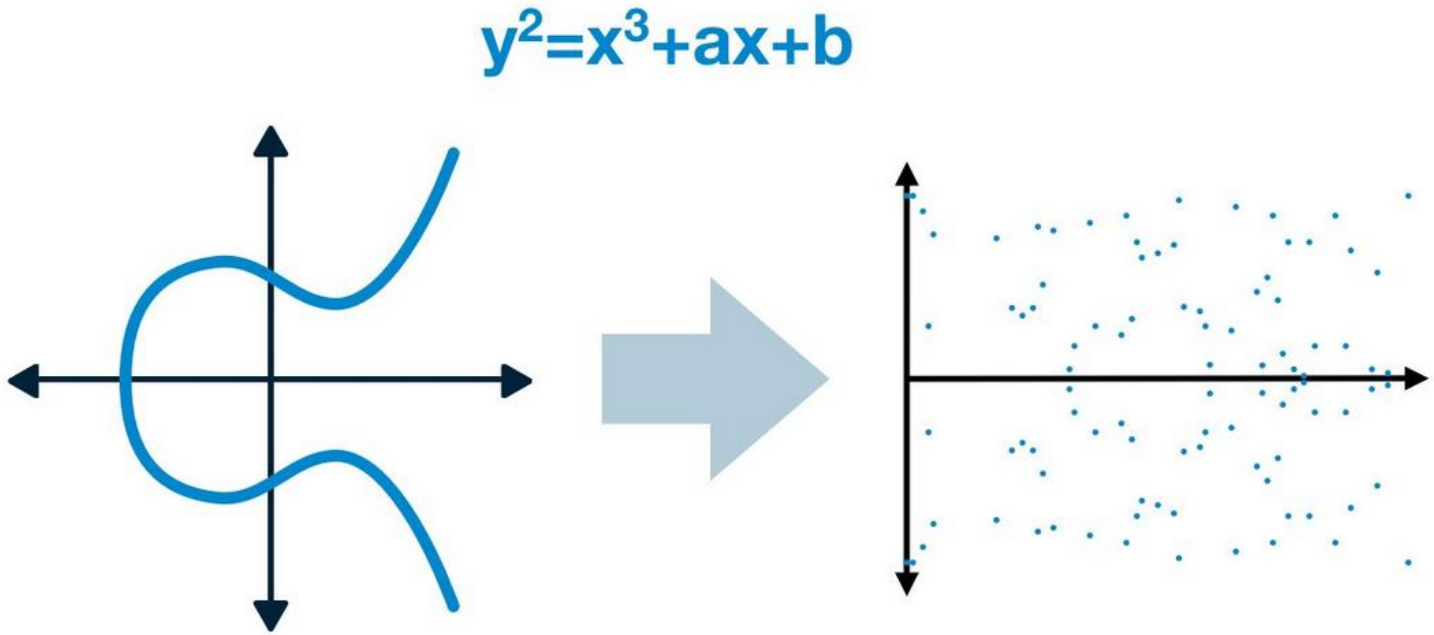
Advantages:

- Elliptic Curve Discrete Logarithm Problem (ECDLP) (calculate k = P / G)
- Shorter key lengths and faster computations (perfect for IoT)



See also: https://csrc.nist.gov/pubs/sp/800/56/a/r3/final

Elliptic curves look structured yet are intractable at large scale in finite fields



https://inevitableeth.com/home/concepts/elliptic-curve-cryptography

Asymmetric Cryptography

Use Case: Key Establishment Methods (KEMs)

Purpose: Enable two or more entities to securely generate or exchange cryptographic keys

Types:

- **Key Transport**

  One party selects a key and securely sends it to the other

- **Key Agreement**:

  All parties contribute to generating the key

Designed to Resist:
- Eavesdropping
- Machine-in-the-Middle
- Replay attacks

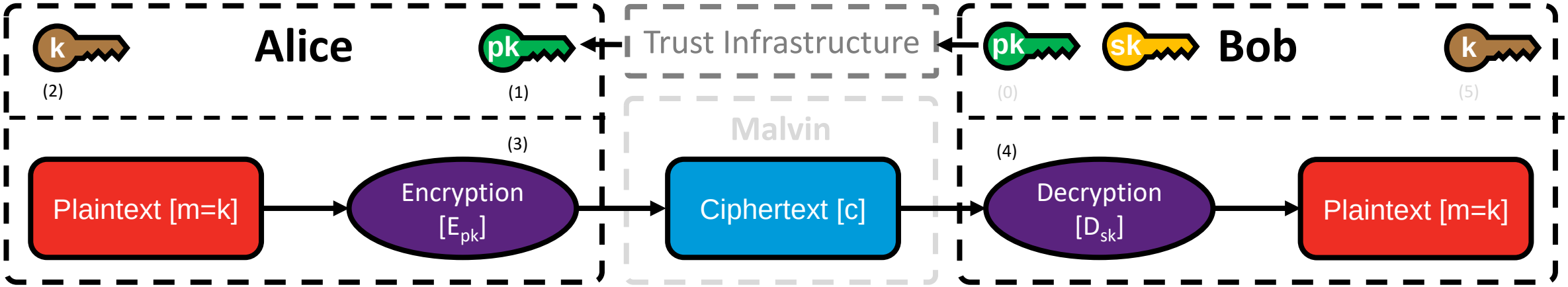
Forward Secrecy:
Ensures that the compromise of one session key won't compromise past sessions, often achieved using **ephemeral keys**

Asymmetric Cryptography

Use Case: Symmetric Key Transport

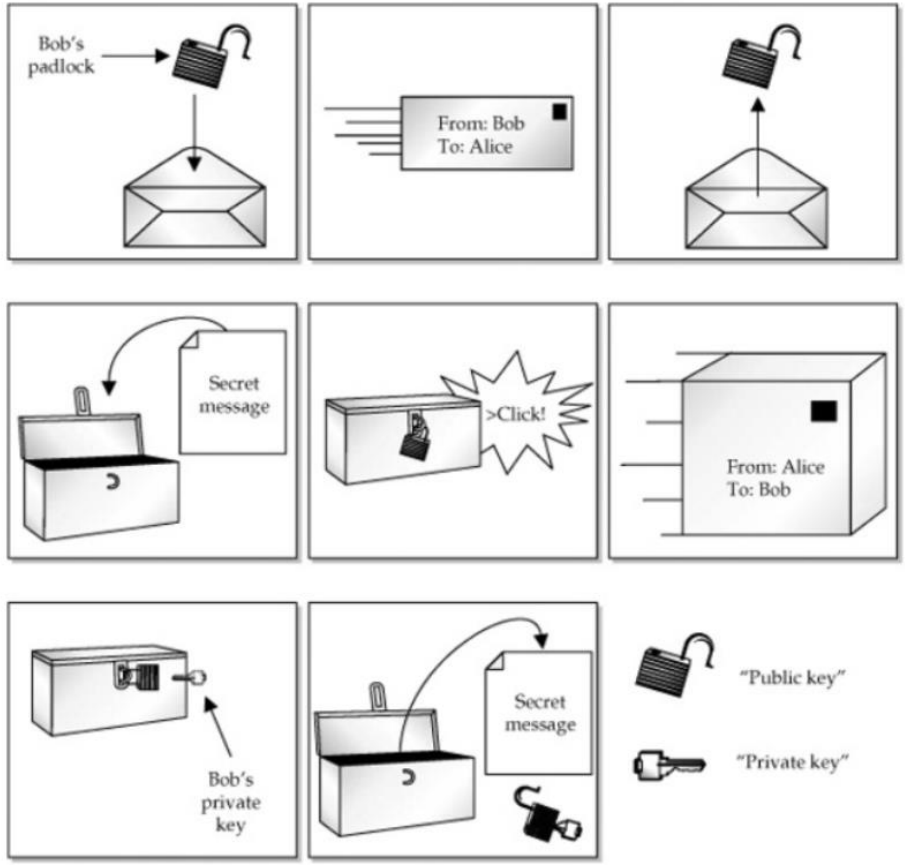
Example Scenario:

0. Bob publishes *pk* to the trust infrastructure
1. Alice gets the public key [*pk*] from Bob via, i.e., Public Key Infrastructure (PKI) or pre-share straight from Bob
2. Alice generates a symmetric key [*k*] *randomly**
3. Alice encrypts [$E_{pk}(k) = c$] and transmits [*c*]
4. Bob decrypts [$D_{sk}(c) = k$]
5. Bob and Alice can now securly exchange information using Symmetric Cryptography [$D_k(E_k(m)) = m$]



Note: Randomness in cryptography is a topic on its own.

Analogy for asymmetric cryptography



https://www.ridingthecrest.com/blog/2020/08/28/public-key-infrastructure.html

Asymmetric Cryptography

Use Case: Key Agreement

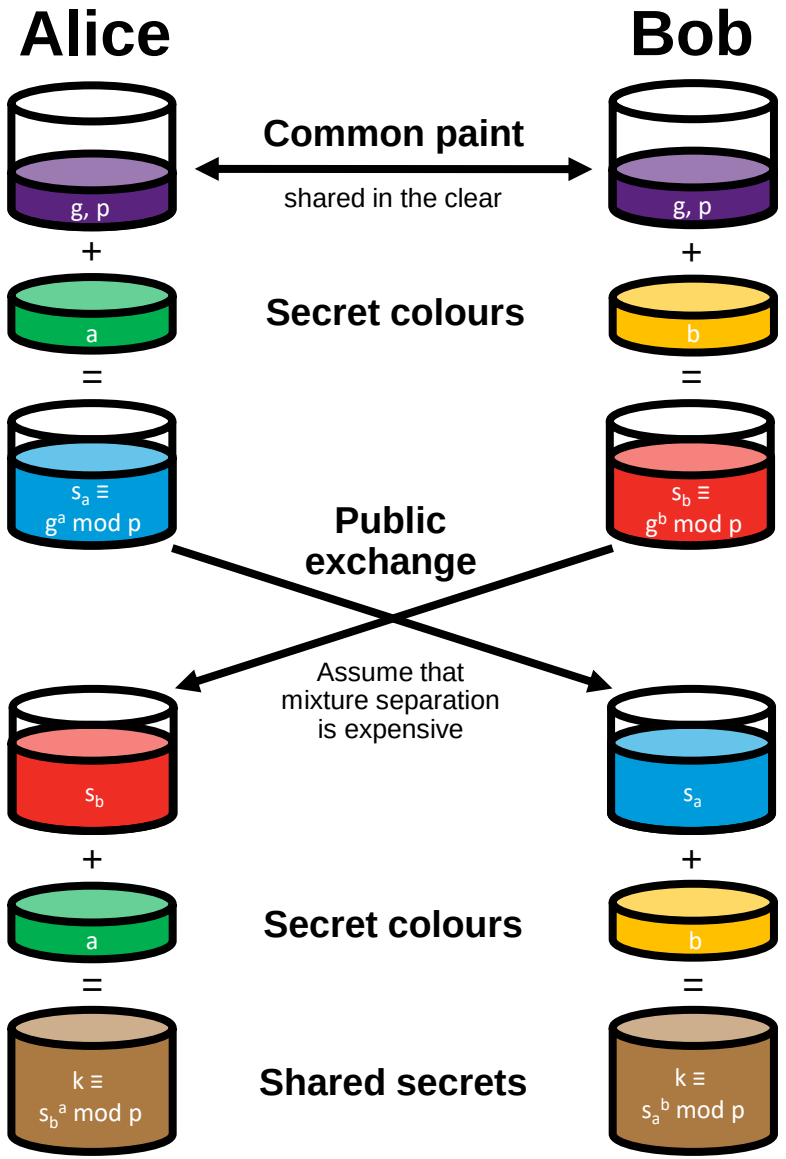
Example (EC) Diffie-Hellman (DH) mathematical method:

- From the *inventors* of Public Key Cryptography (PKC)
- No trust infrastructure (e.g. PKI) needed
- Ephemeral keys and (EC) Discrete Logarithm Problem (DLP)

Psst Kids! Want to buy some Math? It must be discrete, tho!

- Common *paint*: Generator [**g**] and modulus [**p**]
- Alice: Secret *colour* [**a**] and *mix*: [s_a]
- Bob: Secret *colour* [**b**] and *mix*: [s_b]

- **Mixture**: $(g^a \bmod p)^b \bmod p \equiv (g^b \bmod p)^a \bmod p = k$
- **Associative Property**: $(x^a)^b = x^{a*b} = x^{b*a} = (x^b)^a$
- **ECDH**: b*(a*G) ≡ a*(b*G) ≡ k



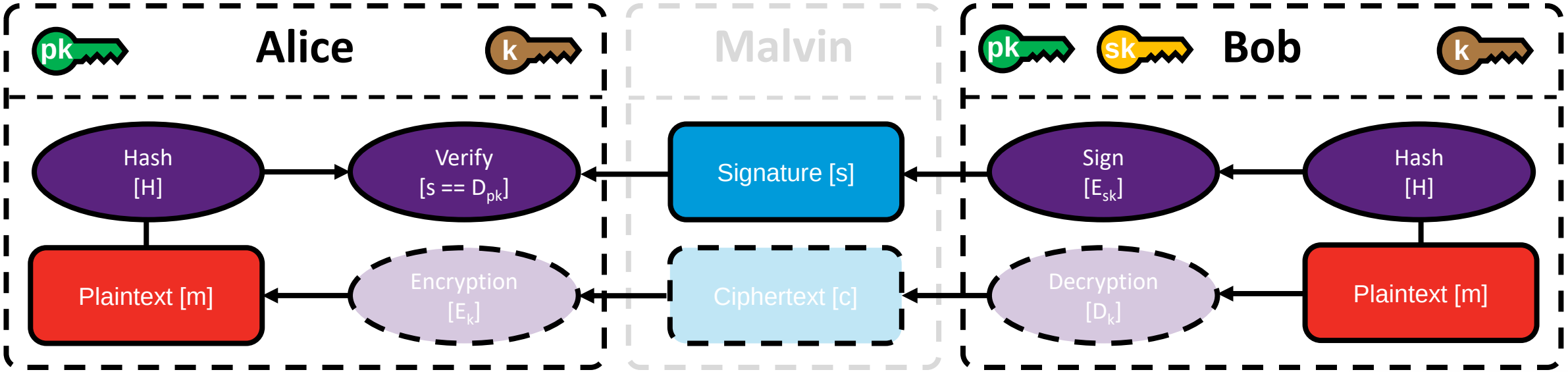
Source: New Directions in Cryptography (Diffie and Hellman, 1976)

Asymmetric Cryptography

Use Case: Encryption-based Digital Signature

Example Scenario (naïve approach):

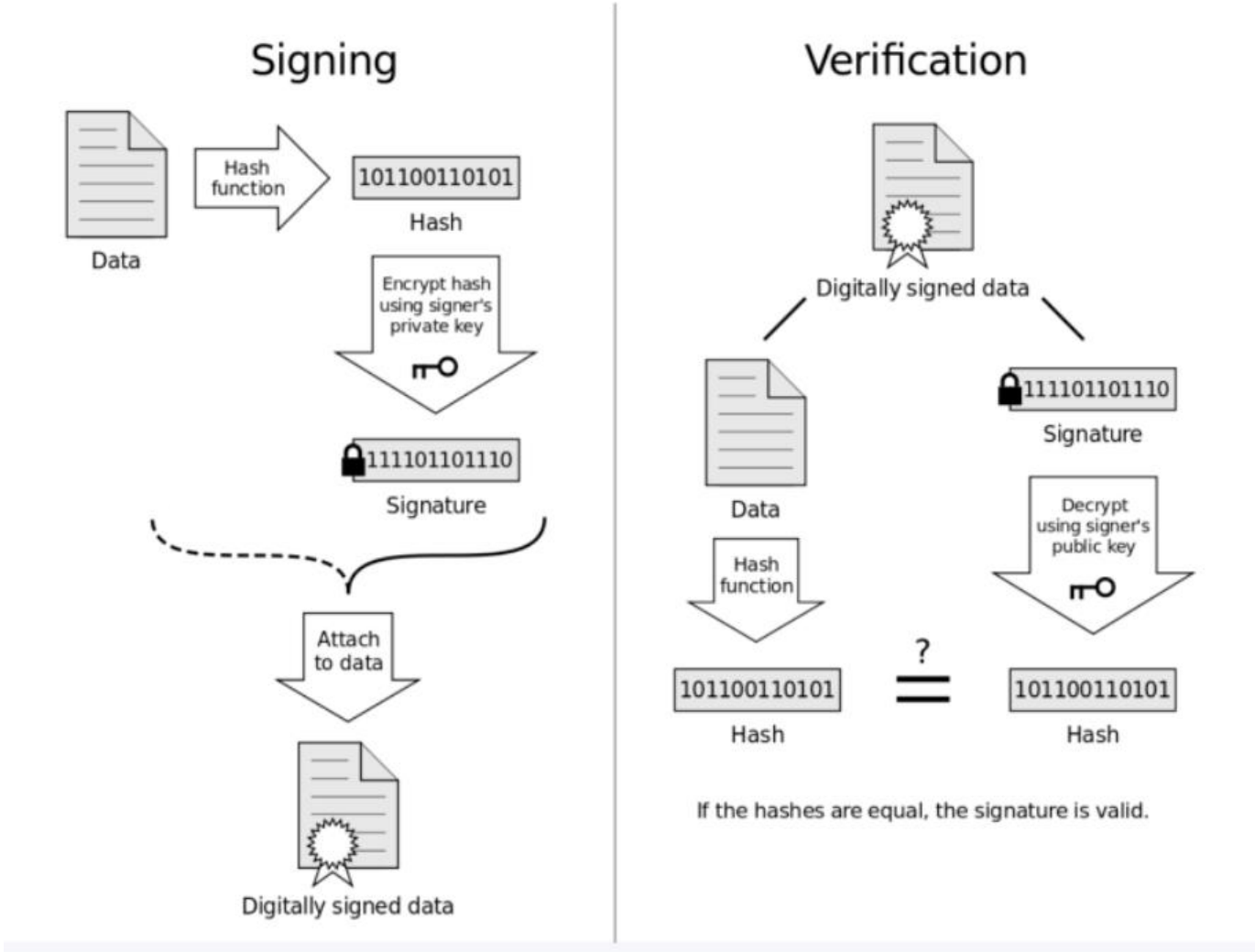
- Alice gets the public key [*pk*] from Bob via, e.g., Public Key Infrastructure (PKI)
- Bob calculates the hash [H(m)=h] and encrypts [E_{sk}] the hash to create a signature [$E_{sk}(H(m))$]
- Message and signature are exchanged
- Alice calculates the hash [H(m')=h'], decrypts the signature [s] and compares if [h==h']



Note: E_k is not necessary for Digital Signatures. One could also sign and send the plaintext message [m]. CIA is fulfilled with E_k.

Illustration: Creating and verifying digital signatures

Hashing the message is necessary to get something short enough to encrypt.



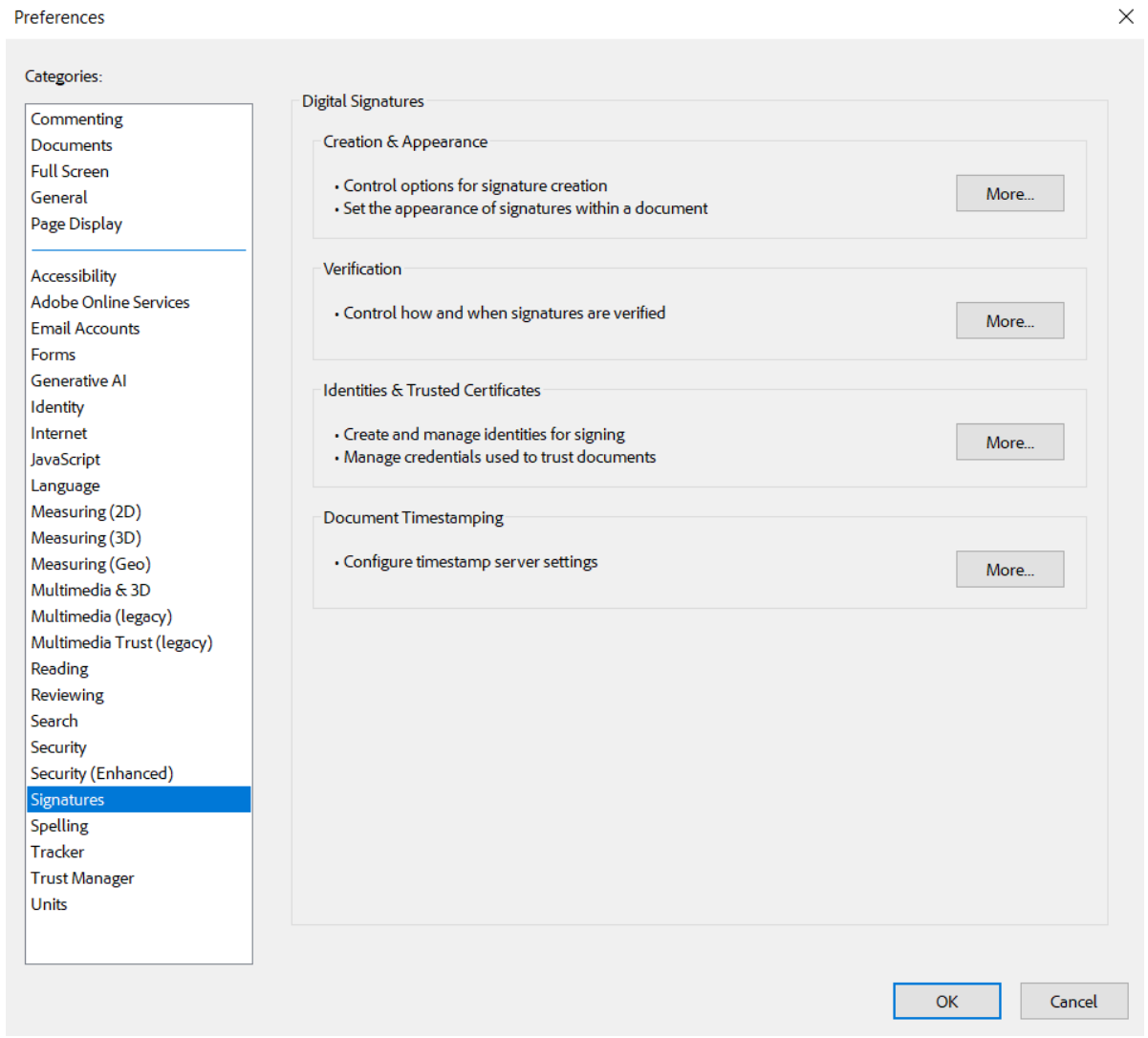
https://proprivacy.com/email/guides/digital-signature

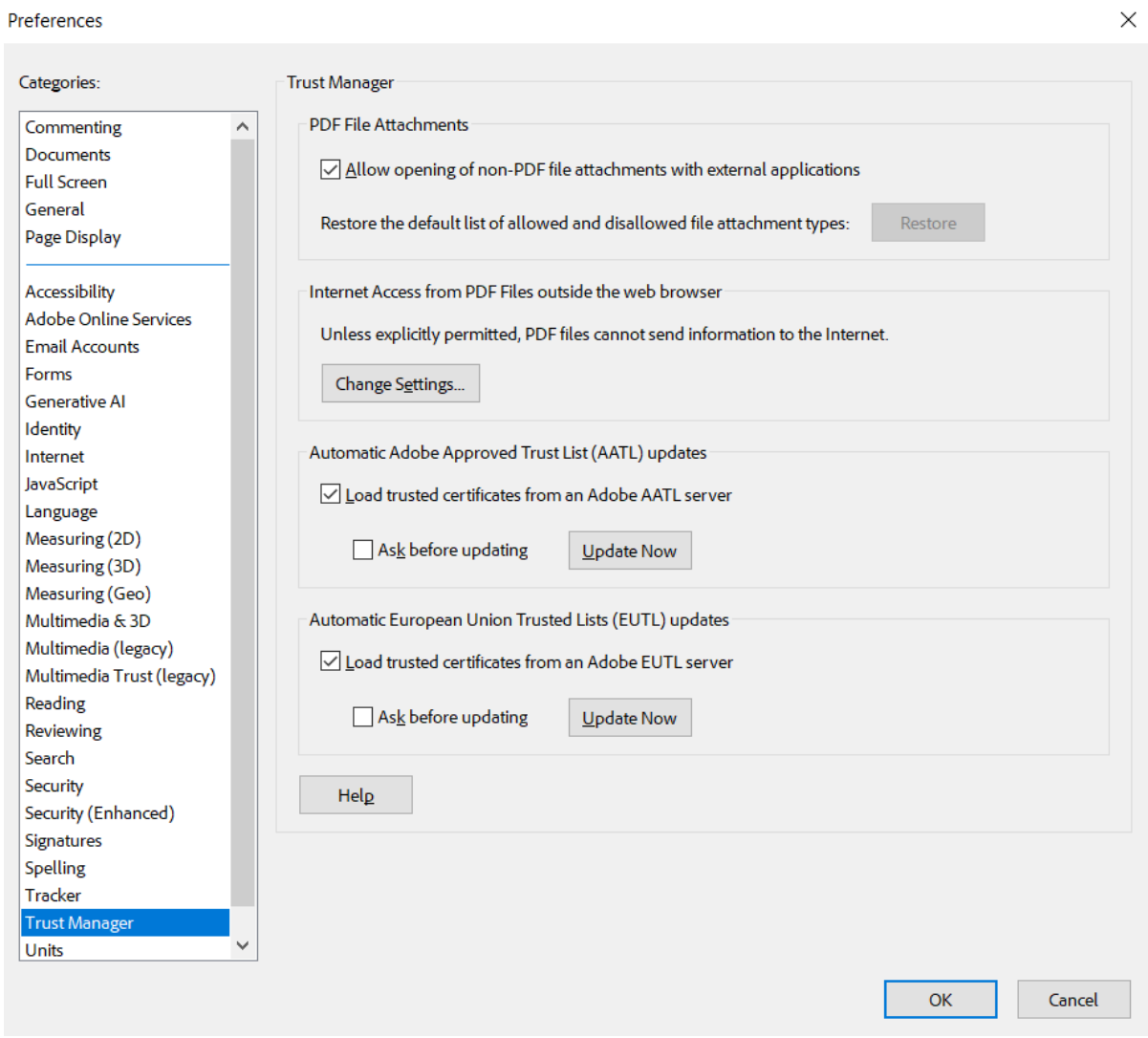
“How would you know my public key?“ Approaches to public key distribution

- Social networks (Pretty Good Privacy, PGP)
- “Contact list“ and consistency checks: iMessage, Signal, (WhatsApp), …
- Hierarchically (WebPKI with Root Stores by Browser vendors, operating systems vendors, providers for identity verification software (e.g., Adobe), ICAO, EU Trusted Lists)

“The phone book‘s editor has its public key included in root stores and uses its private key to sign other companies‘ and individuals‘ keys.“

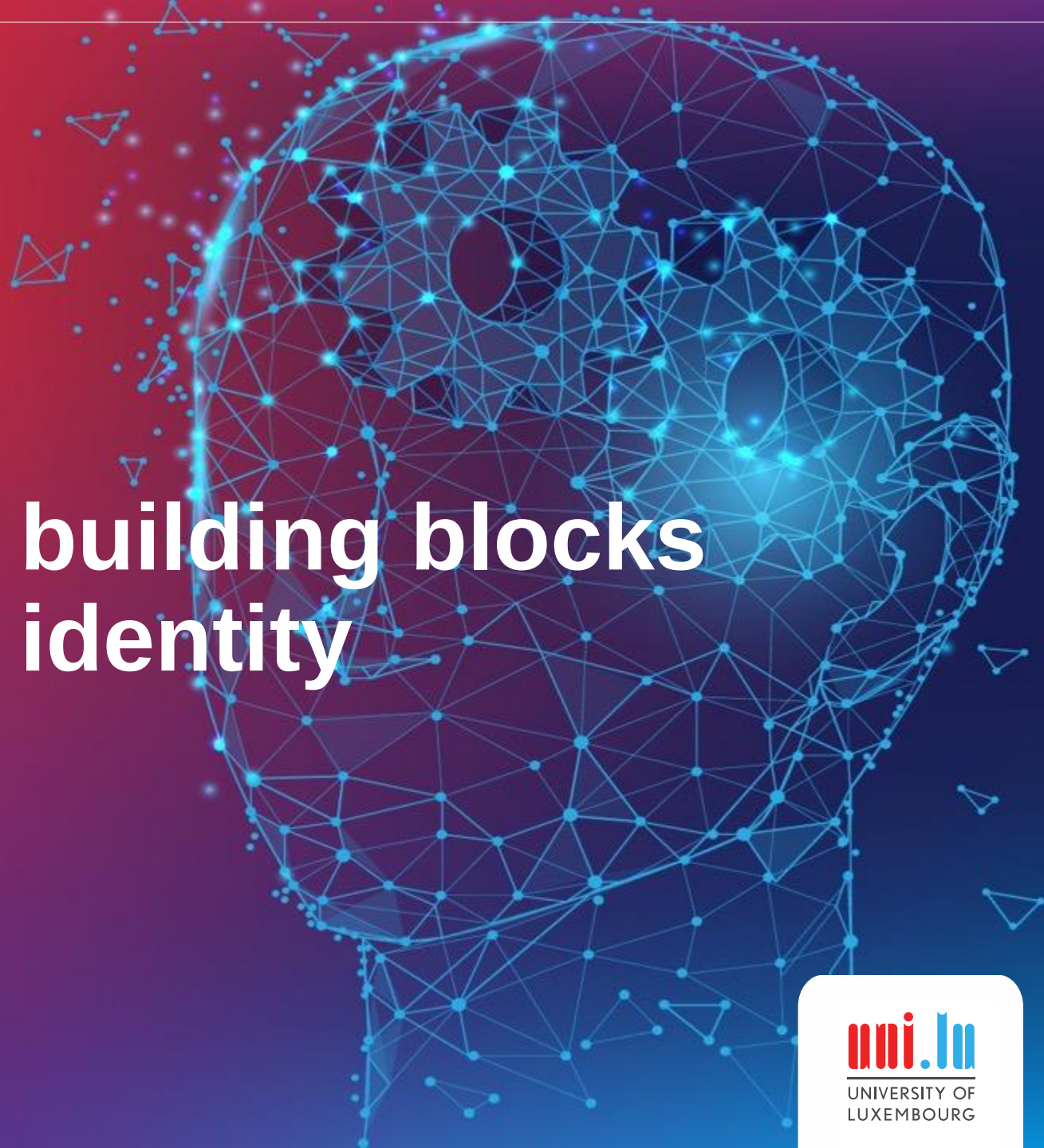
Example: Adobe settings for signature verification





SnT

Further technical building blocks of modern digital identity management



uni.lu
UNIVERSITY OF LUXEMBOURG

Excurs – Trust Registry: Public Key Certificate (X.509)

Skip

The X.509 certificate is a well-known digital credential which binds a subject's identity attributes to a cryptographic public-key.

```
-----BEGIN CERTIFICATE-----
MIIHXDCCBkSgAwIBAgIMEOb8YrdBitUAXkW2MA0GCSqGSIb3DQEBCwUAMGYxCzAJ
BgNVBAYTAkJFMRkwFwYDVQQKExBHbG9iYWxTaWduIG52LXNhMTwwOgYDVQQDEzNH
bG9iYWxTaWduIE9yZ2FuaXphdGlvbiBWYWxpZGF0aW9uIENBIC0gU0hBMjU2IC0g
RzIwHhcNMTYxMTIxMDgwMDAwWhcNMTcxMTIyMDc1OTU5WjB5MQswCQYDVQQGEwJV
[ . . .]
a2luZXdzLm9yZ4IPKi53aWtpcXVvdGUub3JnghAqLndpa2lzb3VyY2Uub3JnghEq
Lndpa2l2ZXJzaXR5Lm9yZ4IQKi53aWtpdm95YWdlLm9yZ4IQKi53aWt0aW9uYXJ5
Lm9yZ4IUKi53bWZ1c2VyY29udGVudC5vcmeCFCouemVyby53aWtpcGVkaWEub3Jn
gg1tZWRpYXdpa2kub3JnggZ3Lndpa2mCDXdpa2lib29rcy5vcmeCDHdpa2lkYXRh
Lm9yZ4INd2lraW1lZGlhLm9yZ4IXd2lraW1lZGlhZm91bmRhdGlvbi5vcmeCDHdp
a2luZXdzLm9yZ4INd2lraXF1b3RlLm9yZ4IOd2lraXNvdXJjZS5vcmeCD3dpa2l2
ZXJzaXR5Lm9yZ4IOd2lraXZveWFnZS5vcmeCDndpa3Rpb25hcnkub3JnghJ3bWZ1
c2VyY29udGVudC5vcmeCDXdpa2lwZWRpYS5vcmcwHQYDVR0lBBYwFAYIKwYBBQUH
AwEGCCsGAQUFBwMCMB0GA1UdDgQWBBQoKiYqV4s7zrTWq1Tv1zghLElcNjAfBgNV
HSMEGDAWgBSW3mHxvRwWKVMcwMx9O4MAQOYafDANBgkqhkiG9w0BAQsFAAOCAQEA
i8Pt0Z05b69Acr0eGF4wVCM1Zl5i1QHiY0dwy20bF7D1TRHkrZRRxV5yA7DVqxjr
tToIqHOV839BGih7RXyDLtMUldjV0V+ZSwz0w5sLT+lJ9Cy1rsMdfSqA9nApTAzm
4MuIiooC7qXRc8KTWCTuQxvjsXuq8BUMYFKPIX2HOhT6gUEAYE+WmmKUWN7LFVw8
9MFNM+P/Of4o+7BBPtKKEdEGASh0fXHUKu8f4yVLLfBm7yb7TPCBhbsamQbJN4fe
jUn3AJGpQjFKuUCgfU9PpurUWAc8AeAaU1Rm4aN+MM07+GlZo0iSSOGeY6sIcJHy
SNKDS5gG+v28mQLanJixow==
-----END CERTIFICATE-----
```

```
Certificate  ::=  SEQUENCE  {
 tbsCertificate       ToBeSigned,
 signatureAlgorithm   AlgorithmIdentifier,
 signature            BIT STRING }

ToBeSigned  ::=  SEQUENCE  {
 version         [0]  Version DEFAULT v1,
 serialNumber         SerialNumber,
 signature            AlgorithmIdentifier,
 issuer               Name,
 validity             Validity,
 subject              Name,
 subjectPublicKeyInfo SubjectPublicKeyInfo,
 issuerUniqueID  [1]  IMPLICIT UID OPTIONAL,
 subjectUniqueID [2]  IMPLICIT UID OPTIONAL,
 extensions      [3]  Extensions OPTIONAL }
```

Excursus – Trust Registry: Public Key Certificate (X.509)

The X.509 certificate is a well-known digital credential which binds identity attributes of a subject with a cryptographic public-key of the subject

Definition:
An X.509 certificate is an ITU standard to verify that a public key belongs to a set of identity information attested within the certificate

Usage: (depends on “Basic Constraints” and “Key Usage”)

- Authentication
- Encryption
- Digital signatures

X.509 Certificate

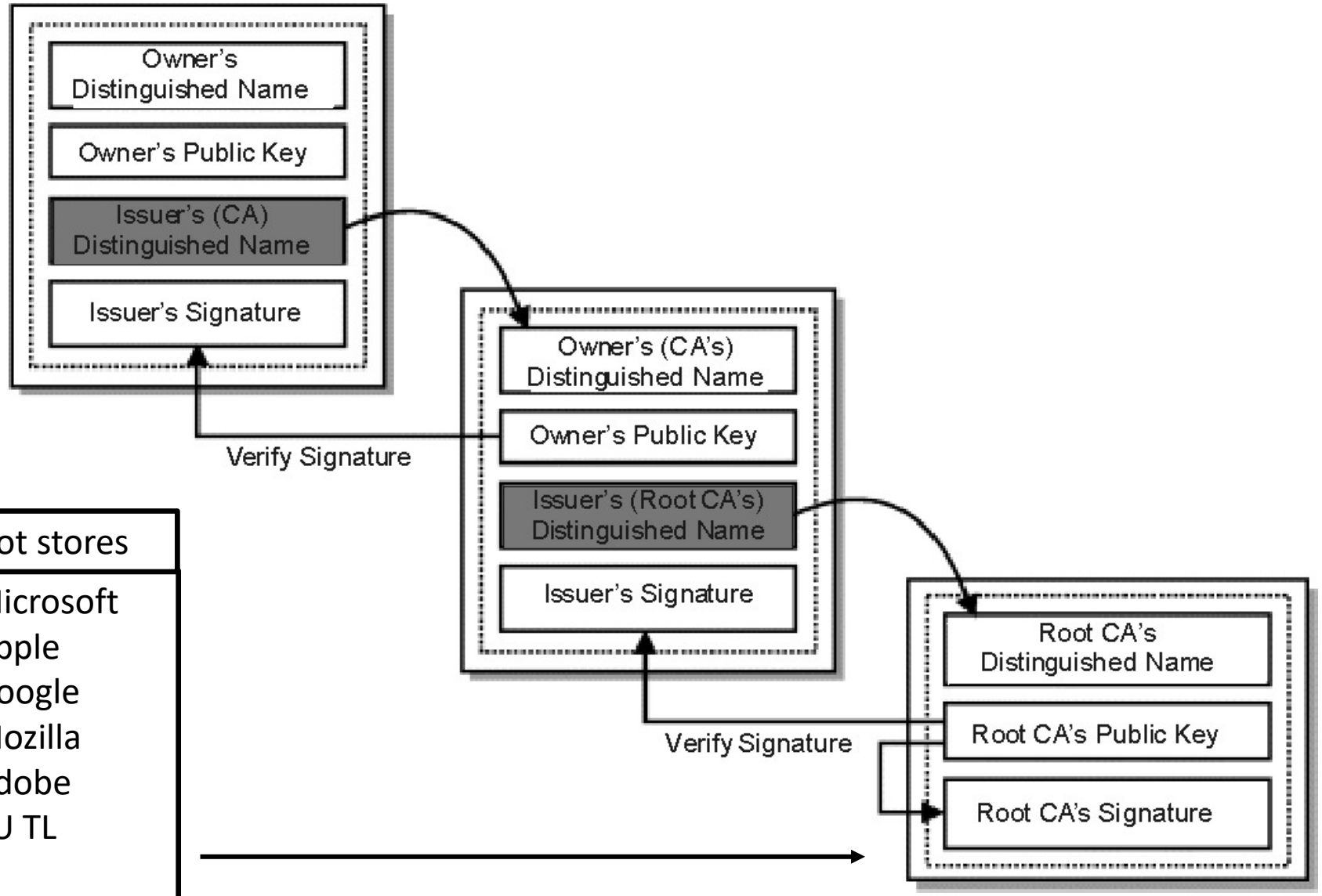
Issuer:
- Version
- Issuer
- Issuer Signature
- Validity (Not before, Not after)
- Revocation information
- Serial number
- Signature algorithm
- Issuer digital signature

Subject:
- Subject public key
- Subject key identifier
- Basic constraints
- (Extended) Key Usage
- Signature Algorithm

Excurs – Trust Registry: Public Key Infrastructure (PKI)

What are other models for trust?

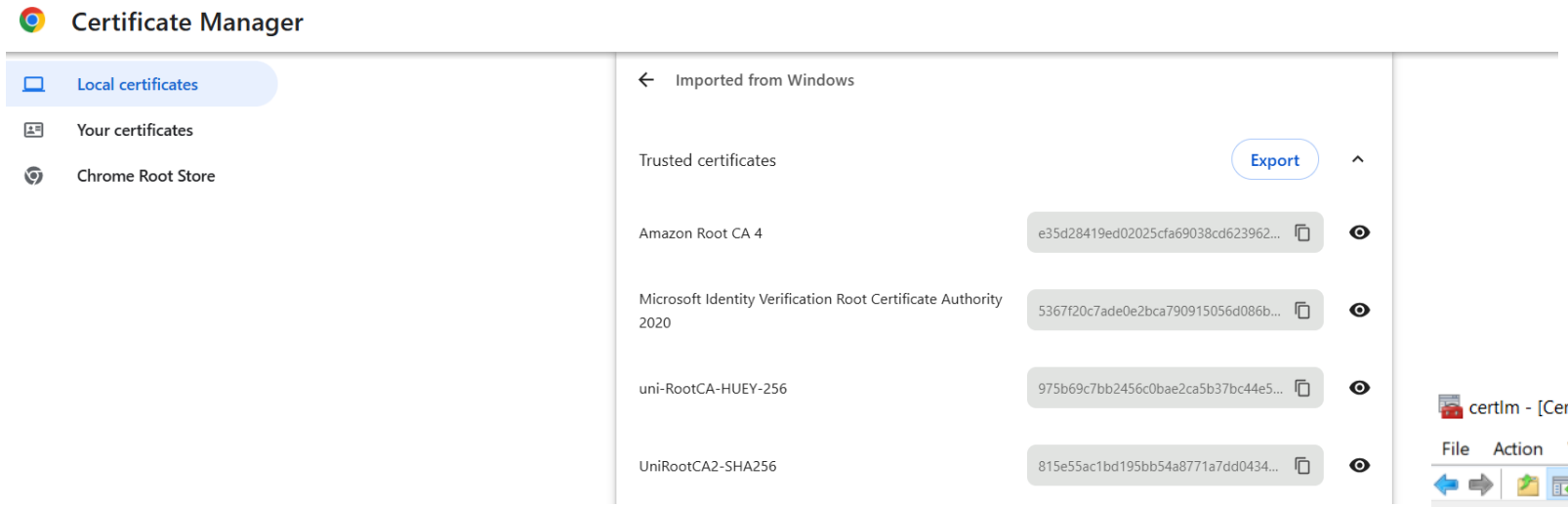
Root of trust: What happens if root leaks key?

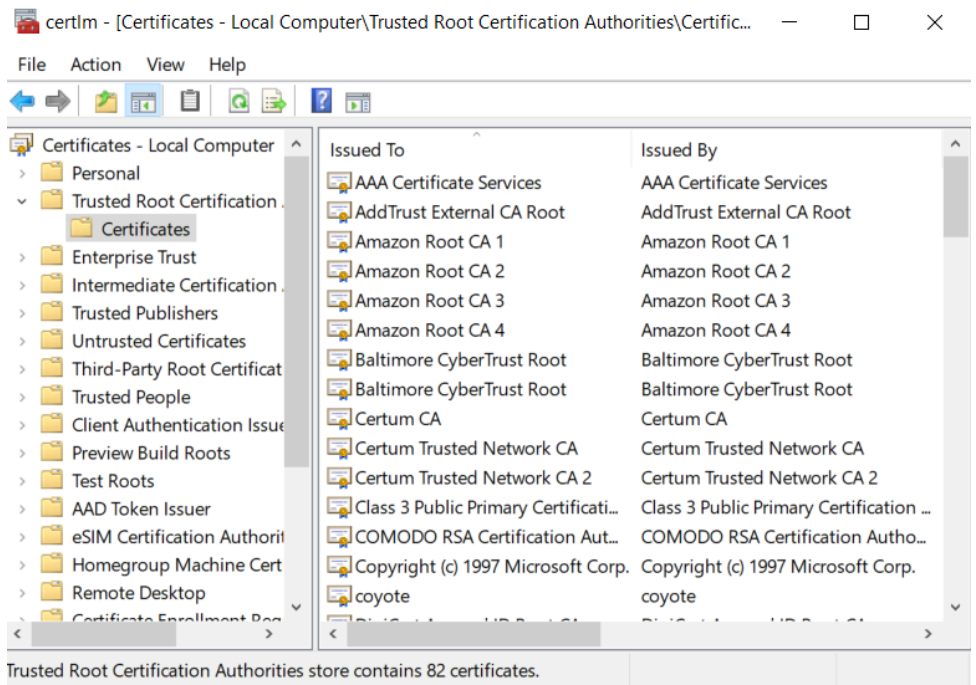


Root stores

- Microsoft
- Apple
- Google
- Mozilla
- Adobe
- EU TL
- …

Example: Trusted public keys in Chrome and Windows



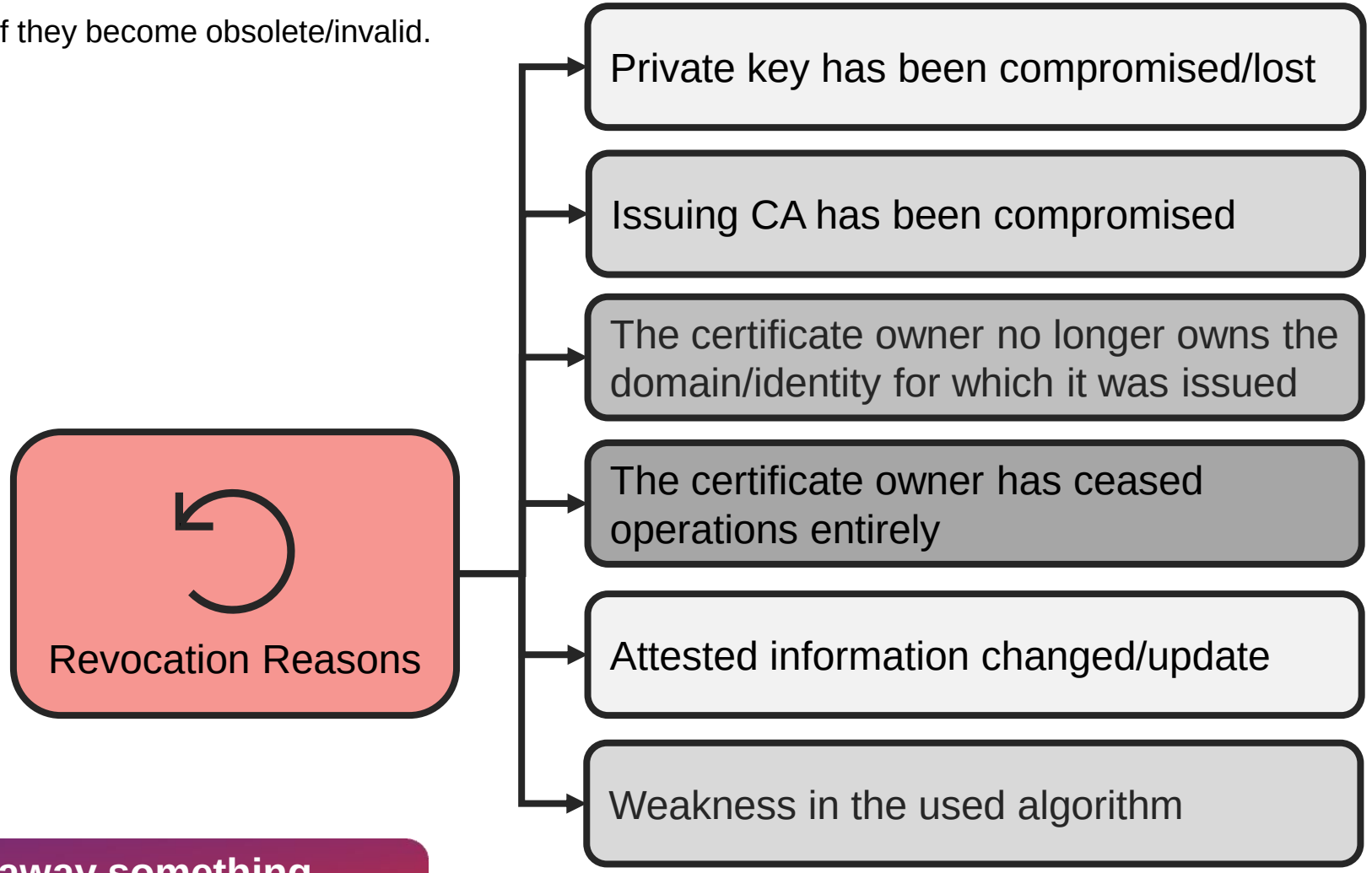


Challenge: Revocation (by example of PKI)

Identities and credentials should be revoked if they become obsolete/invalid.

Revoking digital certificates means invalidating them before they expire. It's a way for CAs (or CRL issuers) to make it is known that one or more of their digital certificates is no longer trustworthy.

E.g., LoA "high" mandates revocation to be in force within 6 hours after notification.

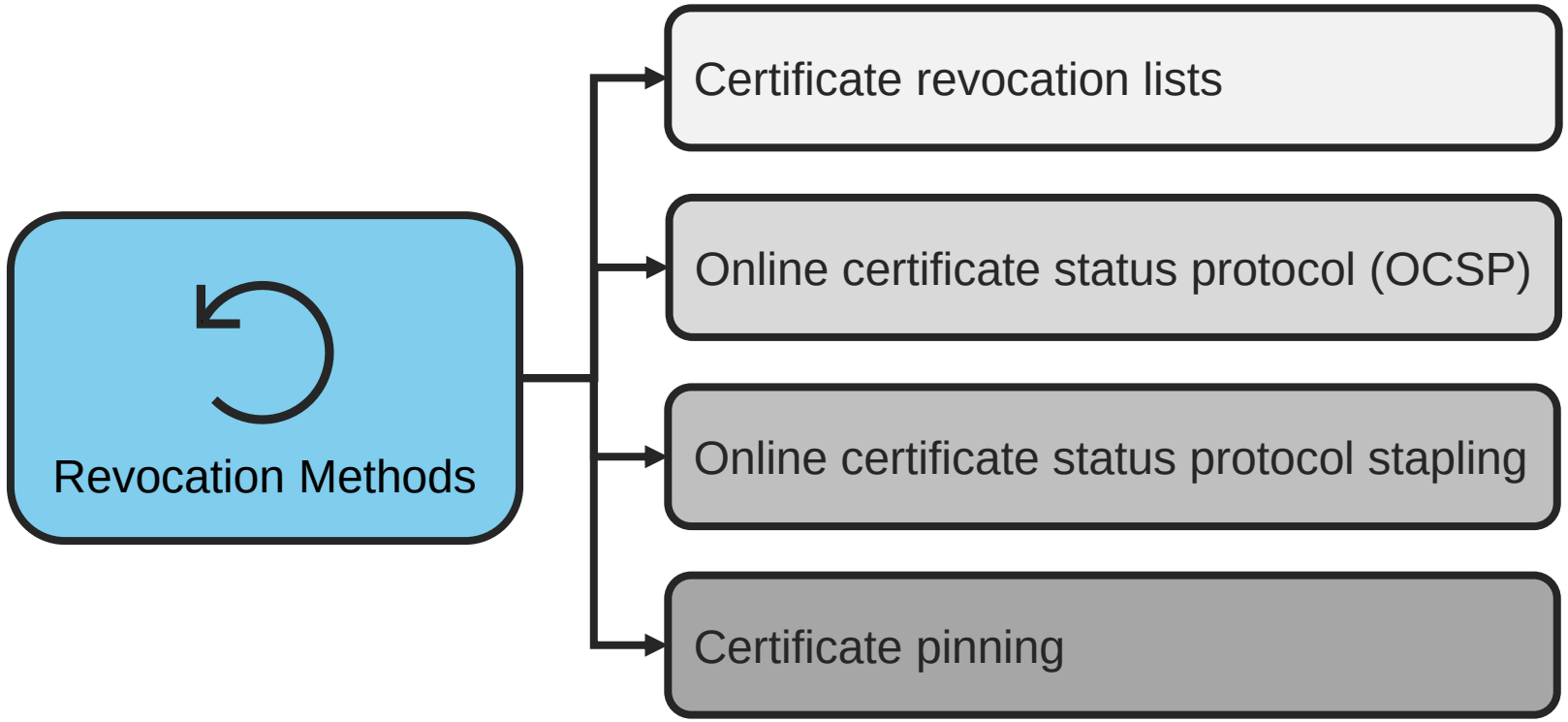


How to reliably and quickly take away something digital from someone, particularly without their cooperation?

Challenge: Revocation (by example of PKI)

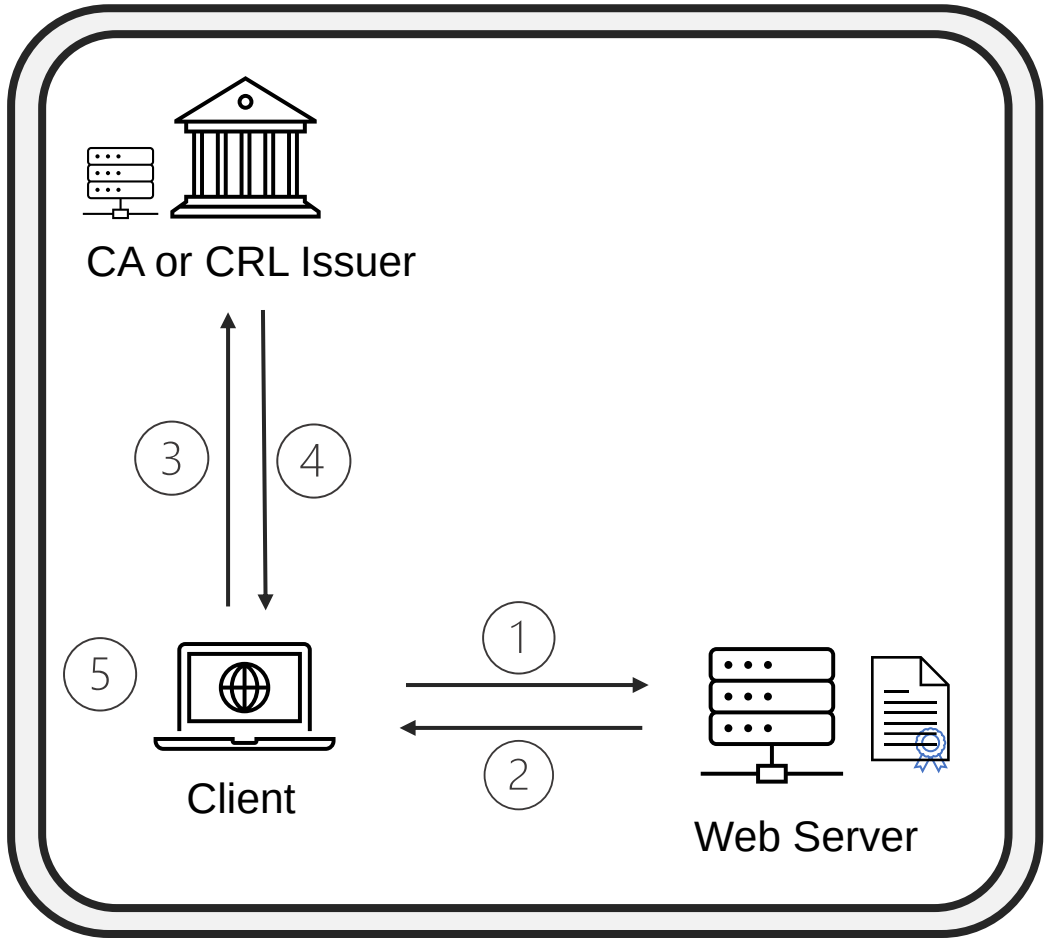
Identities and credentials should be revoked if they become obsolete and/ or invalid.

A crucial aspect of maintaining trust is the capability for revocation, which is pivotal in ensuring that the authentication and authorization processes based on identity data remain valid and reliable over time.



Revocation: Certificate revocation lists (CRLs)

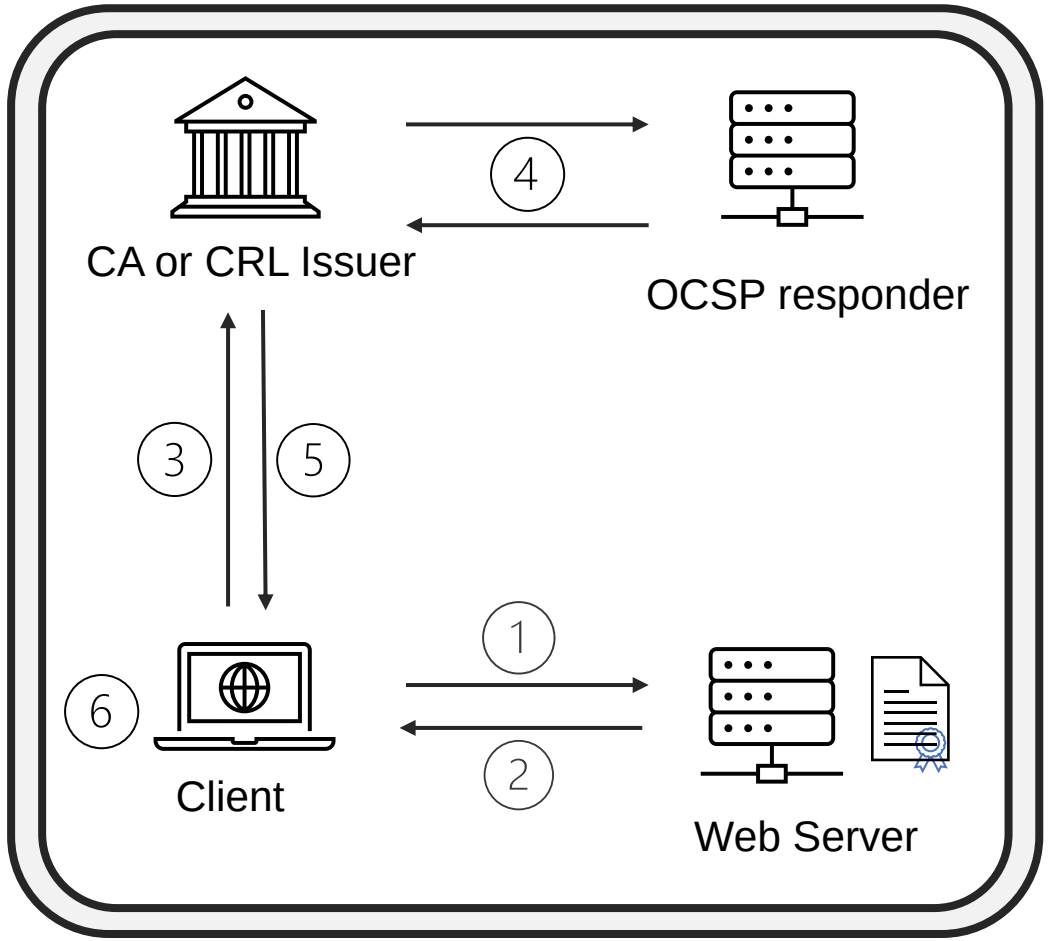
CRLs contain a full list of revoked certificates by a Certification Authority (CA). Users can retrieve them on demand via repositories.



1. Client seeks to connect to a website.
2. Server sends client its SSL/TLS certificate
3. Client contacts the CA's certificate revocation server.
4. Server sends the certificate revocation list.
5. Client checks if the certificate is on the revocation list.

Revocation: Online certificate status protocol (OCSP)

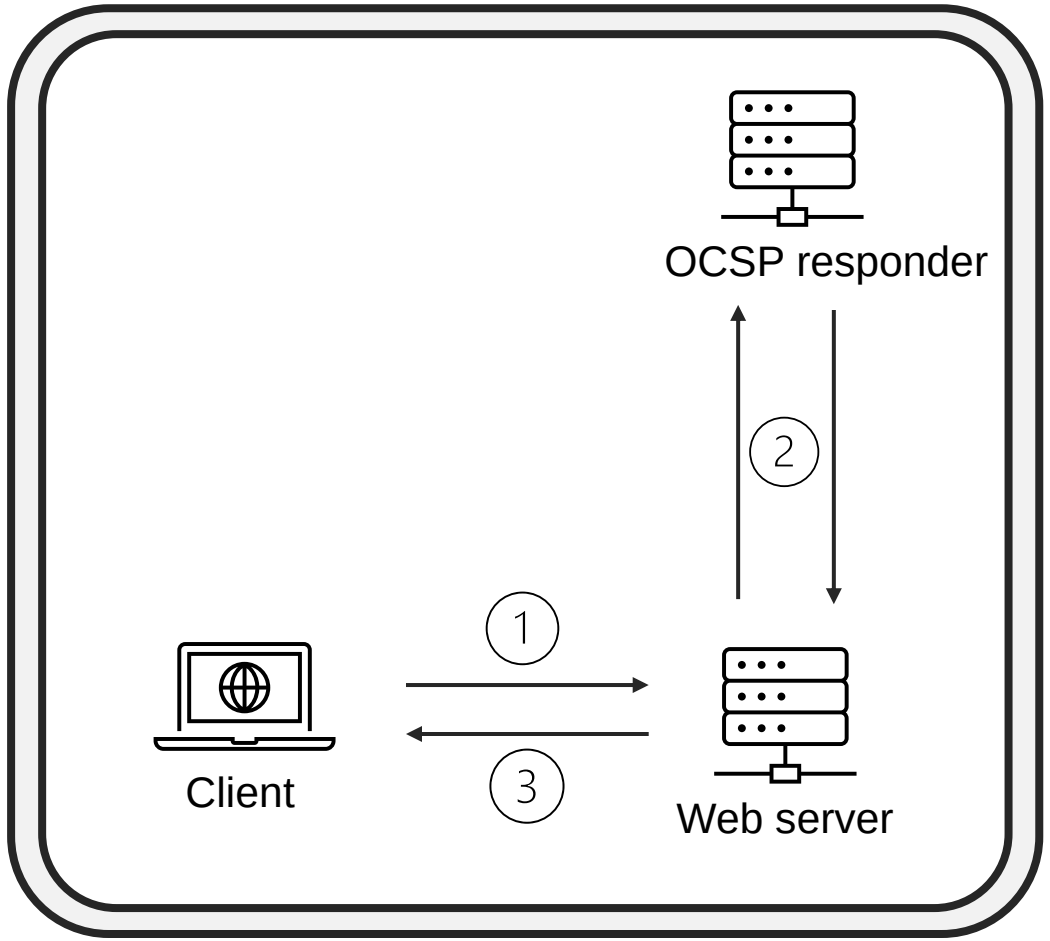
OCSP is a protocol that allows the relying parties to query the revocation status of a specific certificate from a server.



2. The server sends client its SSL/TLS certificate
3. The client sends an OCSP request to the CA (or OCSP responder)
4. The CA's OCSP looks up the revocation status
5. The CA (or OCSP responder) returns a signed OCSP response
6. The Client verifies the signed OCSP response.

Revocation: Online certificate status protocol (OCSP) Stapling

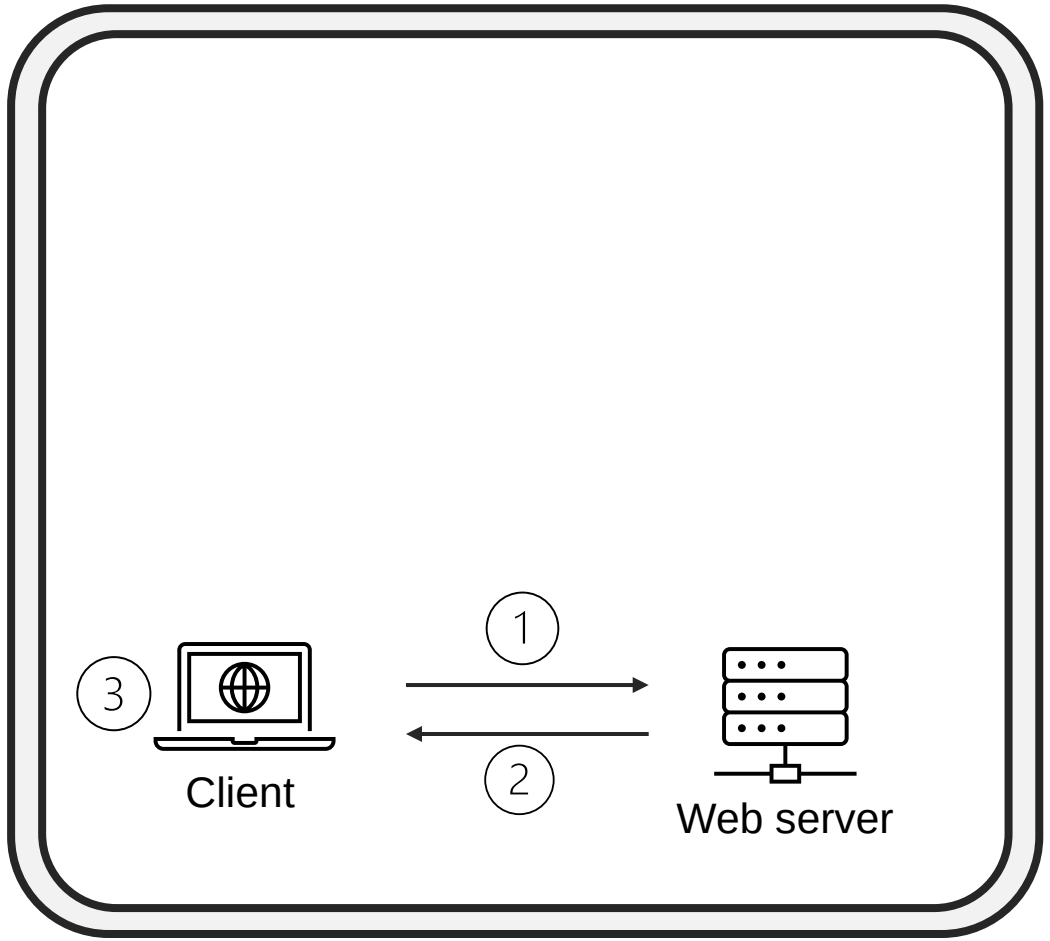
OCSP addresses *privacy* issues, as the OCSP responser may be able to learn about a user's online activities. It also *improves* the connection speed of the SSL handshake by combining two requests into one.



1. Client seeks to connect to a website.
2. Server queries the OCSP responder at regular intervals, obtaining a signed time-stamped OCSP response
3. Server sends client the signed and time-stamped OCSP response

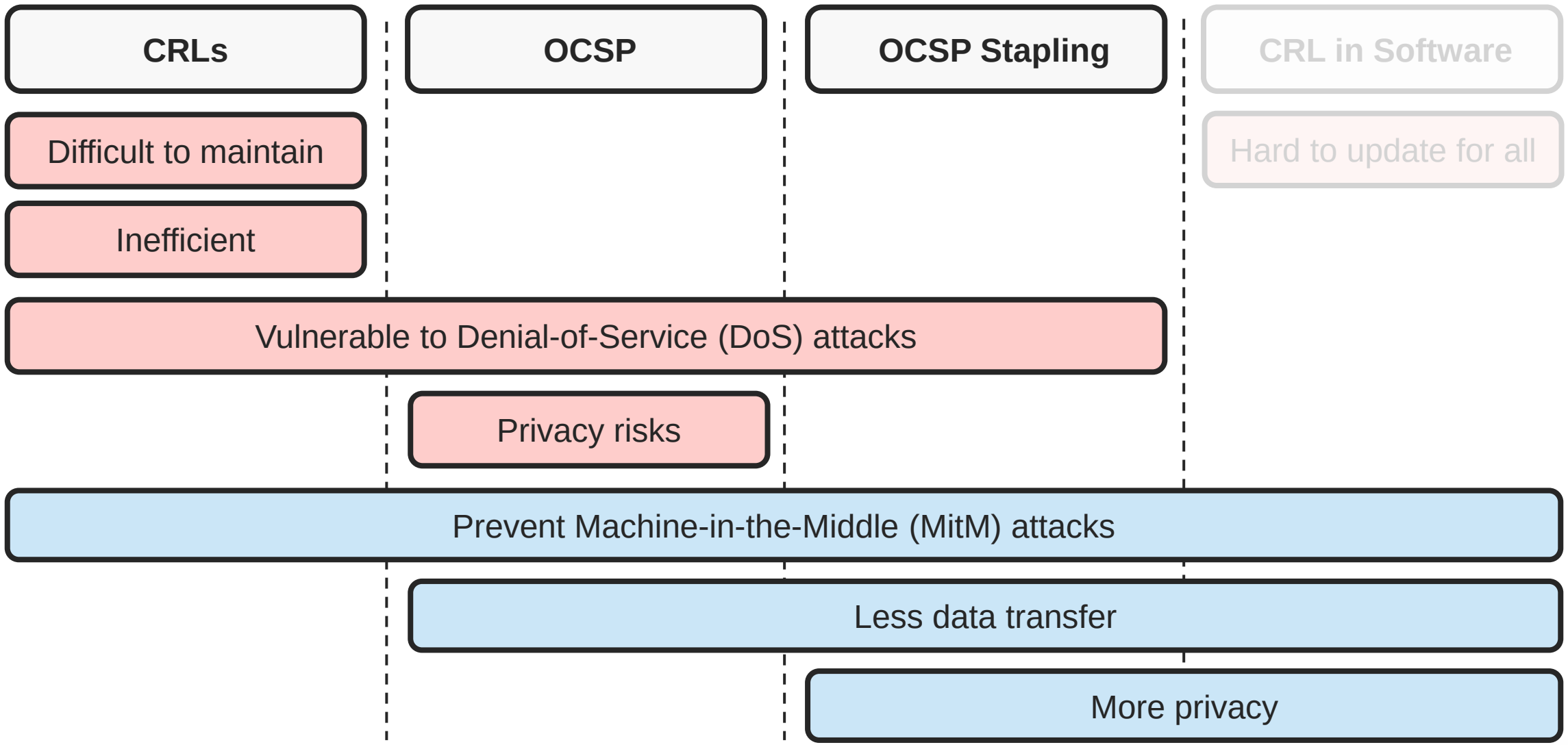
Revocation: Including revocation lists in software distributions (e.g., Browser updates)

The software (Browser) vendor includes certificate revocation lists (CRLs) in their software product.
The frequency of updates is substantially decreased.



1. Client seeks to connect to a website.
2. The server sends client its SSL/TLS certificate.
3. Client checks if the certificate is on the revocation list.

Revocation: Comparison (of PKI-related methods)



Revocation checks in practice

- Revocation checks are often not implemented properly (soft fail)
- They are also handled inconsistently
- Let‘sEncrypt will end support for OCSP Stapling in mid-2025

	Certificate Valid		*Certificate Revoked*		
	OCSP Reachable	*OCSP Blocked*	*OCSP Reachable*	*OCSP Blocked*	
Accept	**Internet Explorer 8.0.6001.18702/WinXP** **Internet Explorer 11.0.9600.17041/Win7** Firefox 28.0/Win7 Firefox 28.0/WinXP Firefox 26.0/Ubuntu 13.04 Safari 5.1.7/WinXP Safari 5.1.7/Win7 **Opera 20.0.1387.91/WinXP** **Opera 20.0.1387.91/Win7** Opera 12.16/Ubuntu 13.04 **Chrome 34.0.1847.116/WinXP** Chrome 34.0.1847.116/Win7 Chrome 34.0.1847.116/Ubuntu 13.04	**Internet Explorer 8.0.6001.18702/WinXP** **Internet Explorer 11.0.9600.17041/Win7** Firefox 28.0/Win7 Firefox 28.0/WinXP Firefox 26.0/Ubuntu 13.04 Safari 5.1.7/WinXP Safari 5.1.7/Win7 **Opera 20.0.1387.91/WinXP** **Opera 20.0.1387.91/Win7** Opera 12.16/Ubuntu 13.04 **Chrome 34.0.1847.116/WinXP** Chrome 34.0.1847.116/Win7 Chrome 34.0.1847.116/Ubuntu 13.04	Safari 5.1.7/WinXP Safari 5.1.7/Win7 Chrome 34.0.1847.116/Win7 Chrome 34.0.1847.116/Ubuntu 13.04	Firefox 28.0/Win7 Firefox 28.0/WinXP Firefox 26.0/Ubuntu 13.04 Safari 5.1.7/WinXP Safari 5.1.7/Win7 Opera 12.16/Ubuntu 13.04 Chrome 34.0.1847.116/Win7 Chrome 34.0.1847.116/Ubuntu 13.04	***Accept***
Blocked			**Internet Explorer 8.0.6001.18702/WinXP** **Internet Explorer 11.0.9600.17041/Win7** Firefox 28.0/Win7 Firefox 28.0/WinXP Firefox 26.0/Ubuntu 13.04 **Opera 20.0.1387.91/WinXP** **Opera 20.0.1387.91/Win7** Opera 12.16/Ubuntu 13.04 **Chrome 34.0.1847.116/WinXP**	**Internet Explorer 8.0.6001.18702/WinXP** **Internet Explorer 11.0.9600.17041/Win7** **Opera 20.0.1387.91/WinXP** **Opera 20.0.1387.91/Win7** **Chrome 34.0.1847.116/WinXP**	***Blocked***
	OCSP Reachable	*OCSP Blocked*	*OCSP Reachable*	*OCSP Blocked*	
	Certificate Valid		*Certificate Revoked*		

SNT

Approaches to electronic identification

UNIVERSITY OF LUXEMBOURG

Today‘s dominant paradigms to digitize identity attributes

FRAGMENTED

Multiple apps and accounts - insecure and/or inconvenient; little control for users over their identity attributes

FEDERATED

Single sign-on enabled by **corporate identity providers / governments**. High convenience but limited control for users over their identity attributes

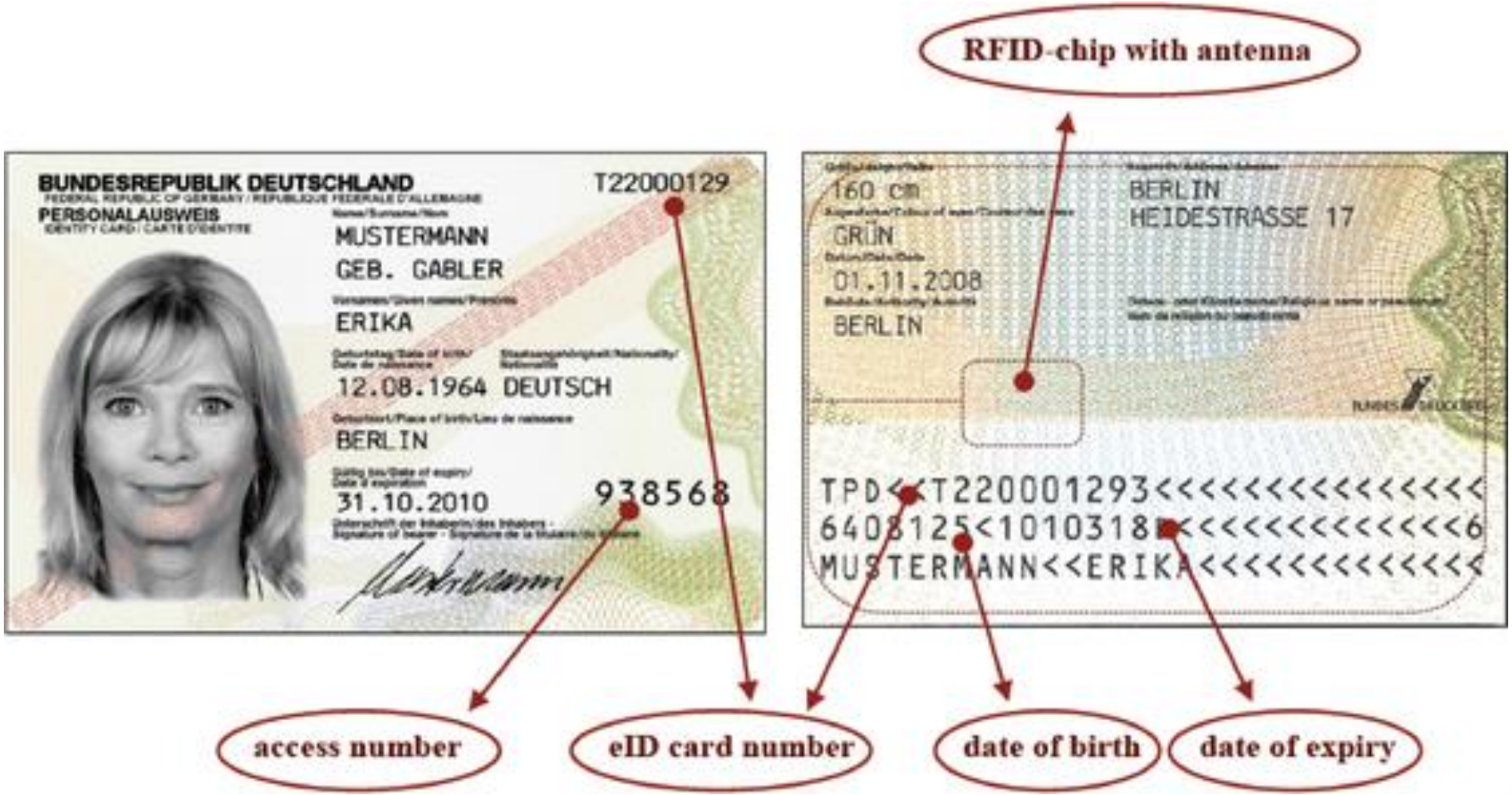
SnT

- **eID**
- Biometric passports
- Video-identification
- Two-factor authentication & federated identity management
- Wallet-based identity management

UNIVERSITY OF LUXEMBOURG

Common capabilities of an eID

"(Contactless) integrated circuit", "RFI chip", "microcontroller", "crypto-chip", "secure element", …



Example: The German eID

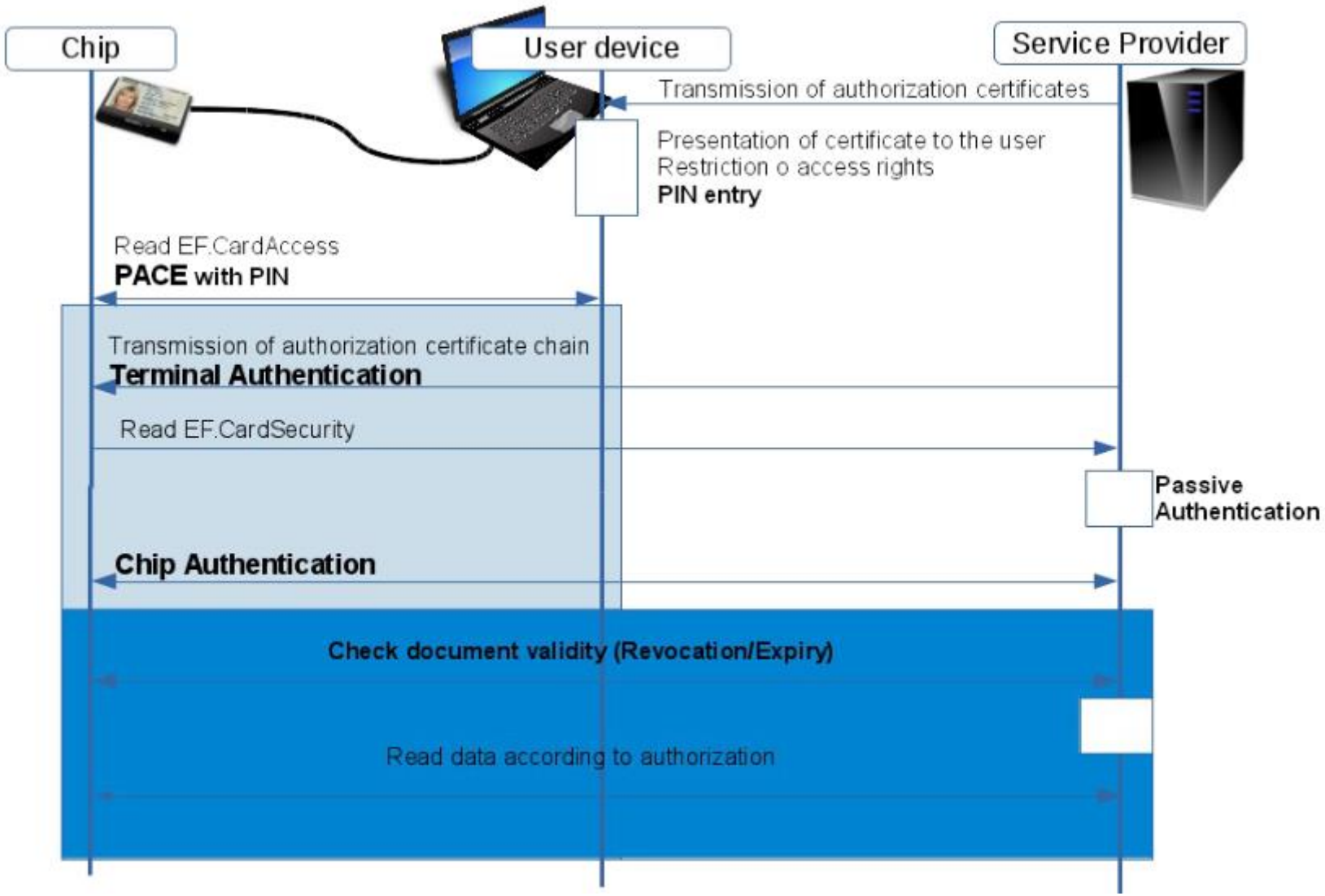
Microcontroller manufactured by Infineon integrated in “plastic card“.

- Password-authenticated connection establishment (PACE): Protection against unauthorized reading; encrypted communication (authorization via user-defined PIN).
- Terminal authentication: Proof of the terminal‘s authorization.
- Passive authentication (PA): Proof of the integrity and authenticity of data groups (certificate for manufacturer and batch of 10.000 microcontrollers with same private key).
- Chip authentication: Proof of the authenticity of the chip (signing challenge with binding key).

More details: https://link.springer.com/chapter/10.1007/978-3-8348-9788-6_35

Most important design paradigm: Security **and** privacy first.
➔ No digitally signed identity attributes are present or communicated.
➔ Selective disclosure, range proofs, different pseudonym for each service.

Example: The German eID



Example: The German eID

Used to require a dedicated reader, but today also a smartphone can be used as communication layer.
Note: The smartphone does **not** get direct read access to eID data, e.g., identity attributes, via NFC.

Very good privacy properties at the highest levels of security.

Opportunity to add further identity attributes (but practically not used).

Need to certify services that use eID for authentication and retrieval of identity attributes.

- Opaque process (formally it is not a monopoly, but practically it is).
- High costs (tens of thousands of €) with not guarantees for certification success.

➔ Scales poorly to broad use.
➔ Scales poorly to heterogeneous needs.
➔ Some level of dependency on a small number certified service providers that *could* log a user‘s activities

Smart eID that uses embedded secure element has faced delays over delays while cooperating only with a single manufacturer (Samsung) for a small set of devices (S20 – S23).

SnT

- eID
- **Biometric passports**
- Video-identification
- Two-factor authentication & federated identity management
- Wallet-based identity management

uni.lu
UNIVERSITY OF LUXEMBOURG

Biometric passports (ICAO travel documents)

Machine-readable zone (MRZ)

https://www.icao.int/publications/documents/9303_p11_cons_en.pdf

Biometric passports (ICAO travel documents)

- Standardized by the International Civil Aviation Organization (ICAO). Includes machine-readable zone (MRZ) and a microcontroller that represents an electronic version of the passport.

- Includes signed digital identity data, including biometric information. Usually no means of active authentication (using a private key that can be invoked using a PIN).

- Protection against unauthorized read access using the card access number (CAN), which is recorded in the document but can also be computed from the data in the MRZ.

- Public key directory including 95 public keys for signing (country signing certificate authority, CSCA) that issue document signer certificates to organizations that manufacture and configure the microcontroller. Similar for reading authorization (country verifying certificate authority, CVCA).

- **In Germany:** According to Section 16a of the Passport Act (PassG), only police enforcement authorities, customs authorities and passport, ID card and registration authorities (PassG Section 16a sentence 1) are authorized to read fingerprints.

- **Applications:**
  - Border control (non-automated and automated)
  - Proving your identity to LinkedIn.

SnT

- eID
- Biometric passports
- **Video-identification**
- Two-factor authentication & federated identity management
- Wallet-based identity management

UNIVERSITY OF LUXEMBOURG

Video identification is facing substantial challenges through AI

Who has gone through a video identification process?
(VideoIdent, AutoIdent, …) are one of the cash-cows for many service providers)

IDnow. WebID Your True Identity Company

Video-based investigation of

- Authenticity of identity document (ID card) → watermarks, holograms etc.
- Authenticity of presenter (eye movement, head movement, responses to questions, …)

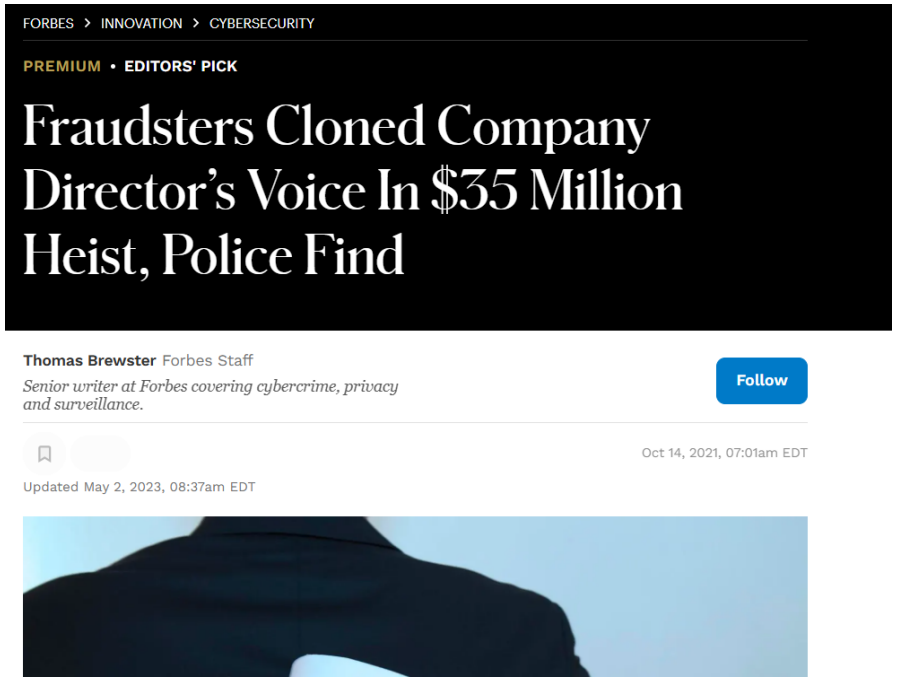
→We know that artificial intelligence can help spoof this!
→A joint publication by ANSSI and BSI attests
"systematic weaknesses"

After a successful demonstration of an attack by the Chaos Computer Club, Video identification was **forbidden** for accessing health data in Germany

→ Fully automated video ident is no more allowed in regulated domains in Germany (it still is in Italy to get high LoA). If accompanied by a person, video identification is still allowed, but the next decision is due in 2025 and also eID must be implemented as alternative.

uni.lu | SnT

Fake calls also pose an increasing risk to businesses



➡ Strong authentication with digital wallets can easily fix this.
For all we know, cryptography is not susceptible to current flavors of machine learning / AI).

The challenges of AI go beyond mere authentication!

opinion

DOI:10.1145/3624721 Wojciech Mazurczyk, Dongwon Lee, and Andreas Vlachos

Opinion
Disinformation 2.0 in the Age of AI: A Cybersecurity Perspective

Why disinformation is a cyber threat.

ACCORDING TO A report from Lloyd's Register Foundation,[a] at present, cybercrime is one of the biggest concerns of Internet users worldwide, with disinformation[b] ranking highest among such risks (57% of Internet users across all parts of the world, socioeconomic groups, and all ages). For years, there has been a discussion in the security community about whether disinformation should

nature machine intelligence

Explore content | About the journal | Publish with us | Subscribe

nature › nature machine intelligence › correspondence › article

Correspondence | Published: 09 October 2023

Battling disinformation with cryptography

Johannes Sedlmeir, Alexander Rieger, Tamara Roth & Gilbert Fridgen

Nature Machine Intelligence **5**, 1056–1057 (2023) | Cite this article

571 Accesses | **98** Altmetric | Metrics

Information and communication technologies have led to an unprecedented surge of disinformation[1], interfering in areas from democratic elections to armed conflicts. With the broad diffusion of generative artificial intelligence (AI) tools such as ChatGPT, Dall-E and Midjourney, the situation may become even worse. These tools make it easy and cheap to fabricate authentic-looking content, such as images, audio files and videos ('deep fakes')[2] that

nature machine intelligence

Explore content | About the journal | Publish with us | Subscribe

nature › nature machine intelligence › correspondence › article

Correspondence | Published: 12 July 2023

Addressing the harms of AI-generated inauthentic content

Filippo Menczer, David Crandall, Yong-Yeol Ahn & Apu Kapadia

Nature Machine Intelligence **5**, 679–680 (2023) | Cite this article

1886 Accesses | **12** Citations | **403** Altmetric | Metrics

Generative AI tools lower the cost of generating false but credible content at scale[1], defeating the already weak moderation defenses of social media platforms. Using inauthentic accounts and other tricks to exploit algorithmic and socio-cognitive vulnerabilities, bad actors can

Analyzing the Strategy of Propaganda using Inverse Reinforcement Learning: Evidence from the 2022 Russian Invasion of Ukraine

DOMINIQUE GEISSLER, LMU Munich & Munich Center for Machine Learning (MCML), Germany
STEFAN FEUERRIEGEL, LMU Munich & Munich Center for Machine Learning (MCML), Germany

The 2022 Russian invasion of Ukraine was accompanied by a large-scale, pro-Russian propaganda campaign on social media. However, the strategy behind the dissemination of propaganda has remained unclear, particularly how the online discourse was strategically shaped by the propagandists' community. Here, we analyze the strategy of the Twitter/X community using an inverse reinforcement learning (IRL) approach. Specifically, IRL allows us to model online behavior as a Markov decision process, where the goal is to infer the underlying

Stay tuned for the lecture on digital identities and misinformation!

SnT

- eID
- Biometric passports
- Video-identification
- **Two-factor authentication & federated identity management**
- Wallet-based identity management

UNIVERSITY OF LUXEMBOURG

FIDO2 as a common second factor scheme

Key feature: Domain-specific (bound) keypairs to address phishing attacks
Typically hardware-bound (Apple‘s secure enclave or Android‘s keystore)

- Microsoft authenticator

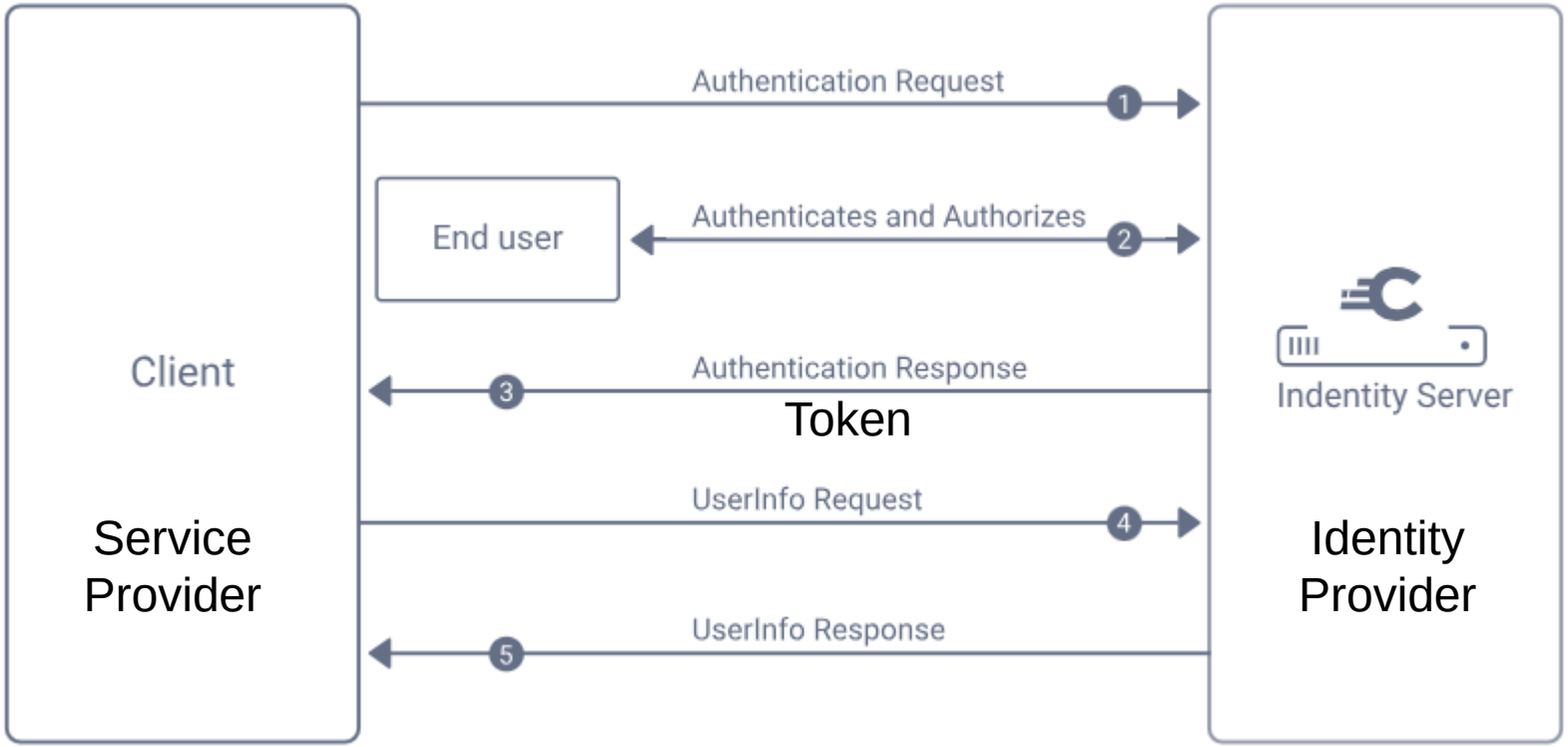
- Yubikey

- Apple/Google passkeys

```
{
  "publicKey": {
    "rp": {
      "name": "Example Service",
      "id": "example.com"
    },
    "user": {
      "id": "YWJjZGVmZw==",
      "name": "user@example.com",
      "displayName": "Example User"
    },
    "challenge": "random_challenge",
    "pubKeyCredParams": [
      {
        "type": "public-key",
        "alg": -7
      }
    ],
    "authenticatorSelection": {
      "residentKey": "required",
      "userVerification": "preferred"
    },
    "timeout": 60000
  }
}
```

The foundations to federated identity management in a nutshell: OAuth and OIDC

OIDC = Open ID Connect



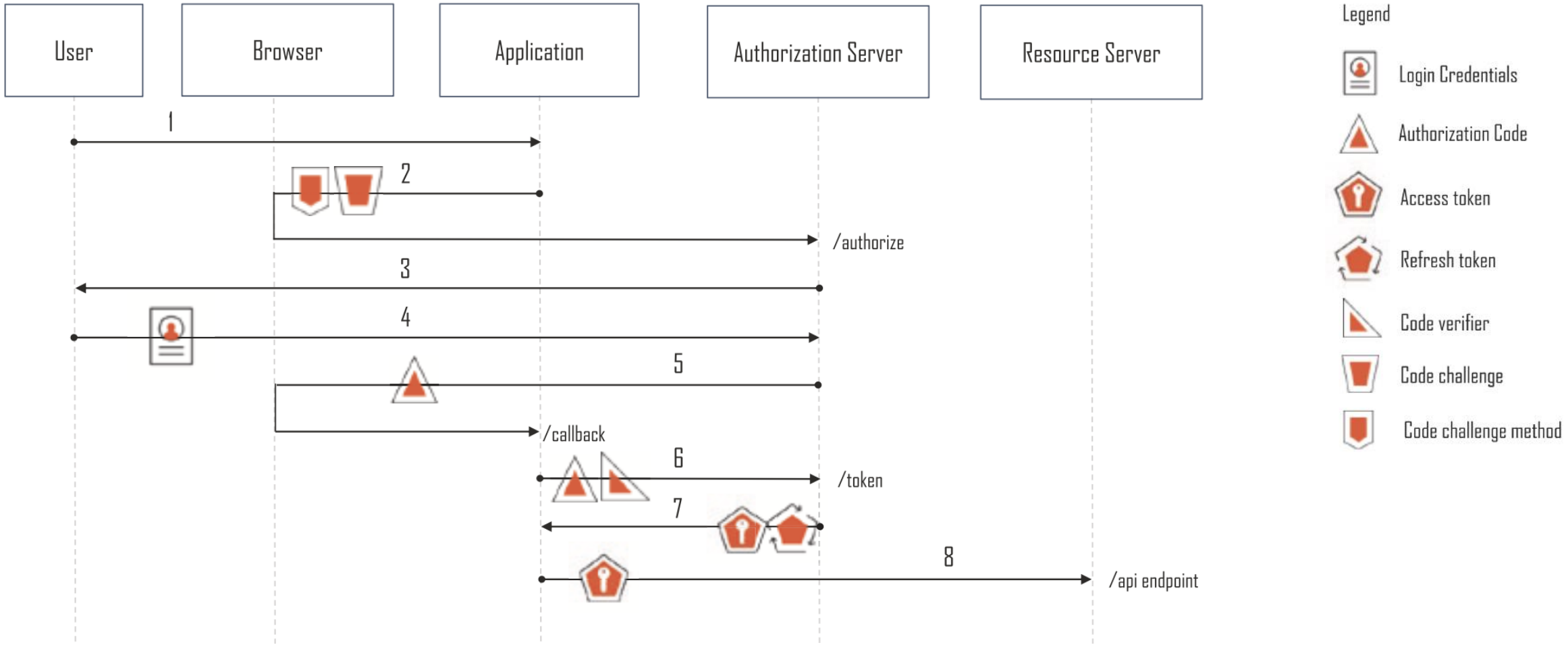
Federated IdM typically uses OAuth for authorization, OIDC for identification

signature

signature

Technologies: OAuth 2.0

OAuth 2.0 provides a framework for authorizing applications to call APIs → not designed for authenticating users to applications

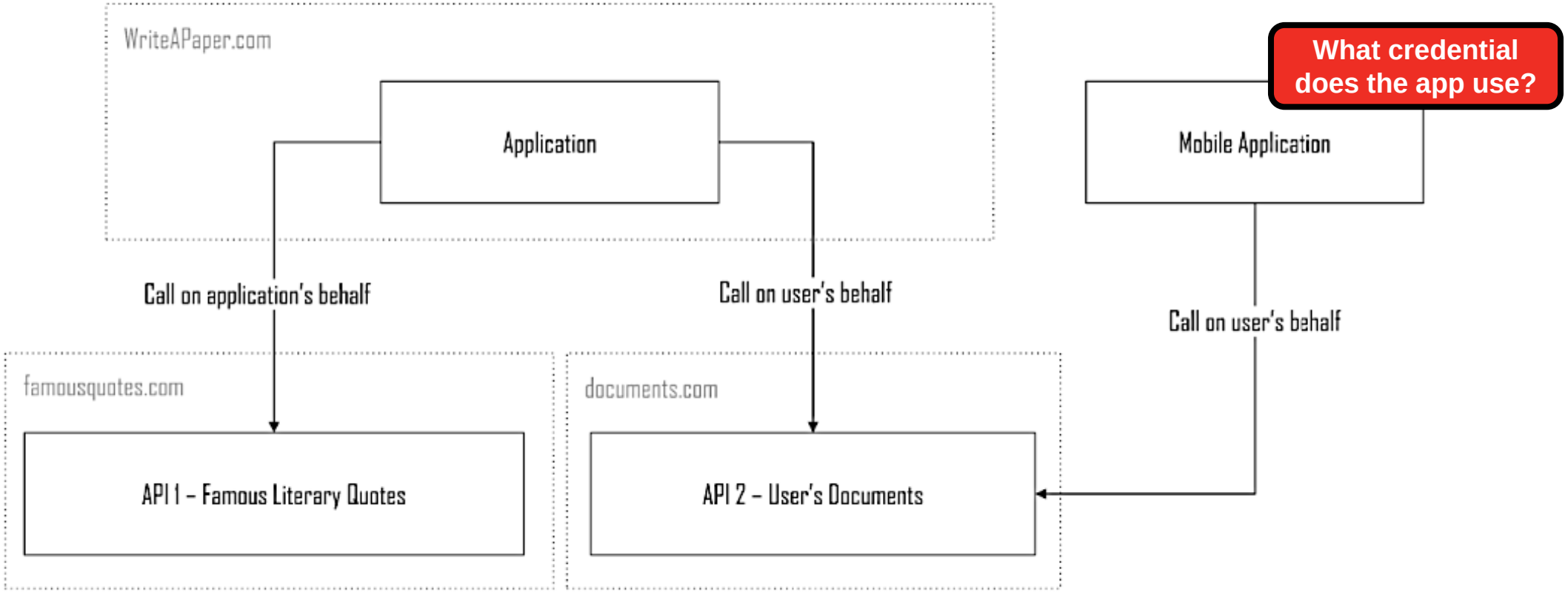


Source: API Authorization: User-based vs. Client-based Flow, (Wilson and Hingnikar, 2019)

Technologies: OAuth 2.0

If an application wants to call an API on a user's behalf to access resources owned by the user, it needs the user's consent

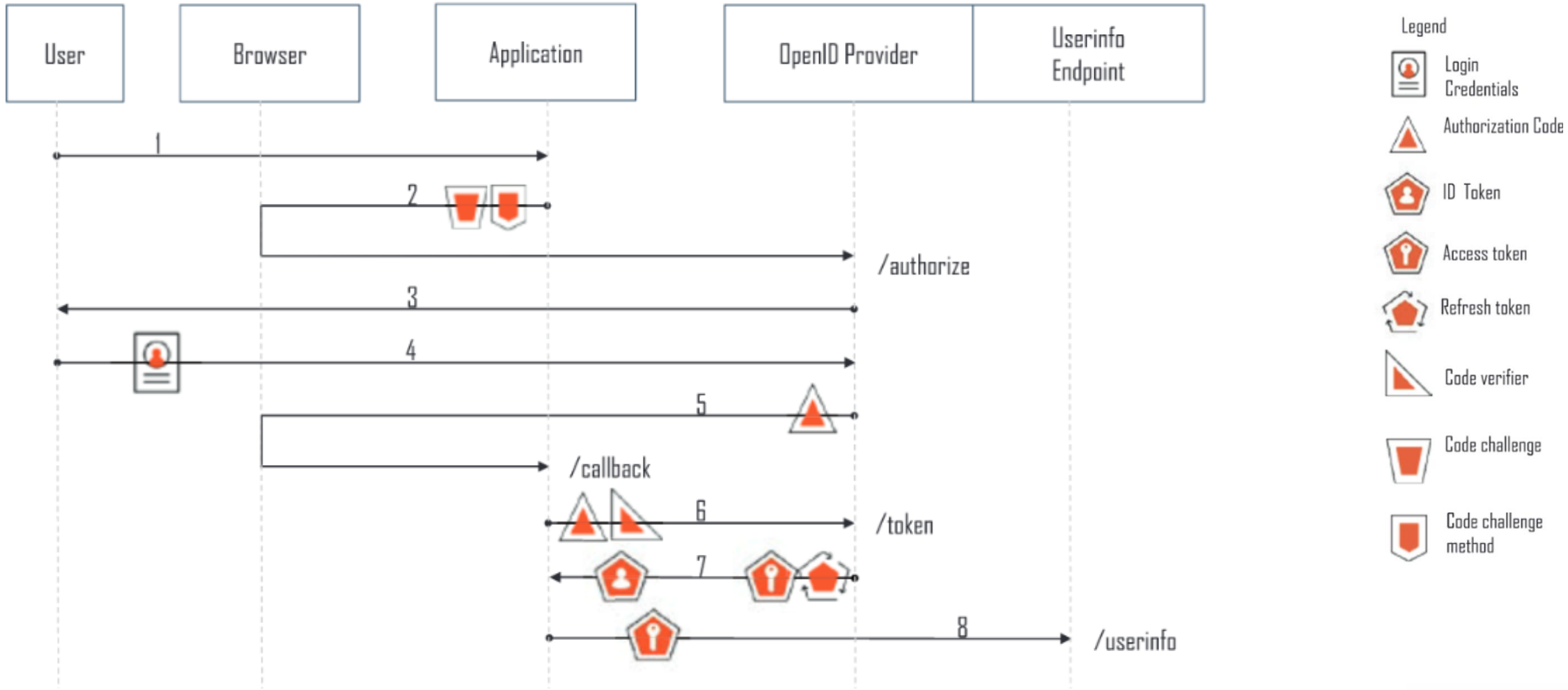
- In the past, a user often had to share their credentials with the application to enable such an API call on their behalf
- Unnecessary amount of access granted to application also the responsibility of safeguarding the credential!



Source: API Authorization: User-based vs. Client-based Flow (Wilson and Hingnikar, 2019)

Technologies: OpenID Connect (OIDC)

OpenID Connect (OIDC) protocol provides an **identity service layer** on top of OAuth.



Source: Wilson and Hingnikar (2019) Solving Identity Management in Modern Applications: Demystifying OAuth 2.0, OpenID Connect, and SAML 2.0

Technologies: OpenID Connect (OIDC)

Takeaways:

OAuth 2.0 provides a framework for **authorising** applications to call APIs
→ not designed for **authenticating** users to applications

The OpenID Connect (OIDC) protocol provides an **identity service layer** on top of OAuth
→ OIDC is designed to allow authorisation servers to authenticate users for applications and to return the results in a standard way

Overview: OIDC vs. SAML

- Both: Authentication + Authorization
- SSO for consumer apps vs. SSO for enterprise apps
- User-centric vs. Organization-centric
- Native/Mobile/Web apps vs. Web apps
- ID Token/JSON vs. Assertion/XML
- Dynamic client registration vs. Static trust-configuration
- HTTP only vs. Different bindings

Example: JSON Web Token (JTW)

Header (algorithm and type of token)

```
{"alg" : "RS256" ,
 "Typ" : "JWT" }
```

Payload (claims)

```
{
  "iss": "http://openidprovider.com",
  "sub": "1234567890",
  "aud" : "2fb3JsPMrDnQkwLEVNMDzUF",
  "nonce":"47jglw0hmxo2hg0ewhg9582lf",
  "exp": 1516239322,
  "iat": 1516239022,
  "name": "Fred Doe",
  "admin": true,
  "auth_time":1516239021,
  "acr":"1",
  "amr" : "pwd"
}
```

Signature

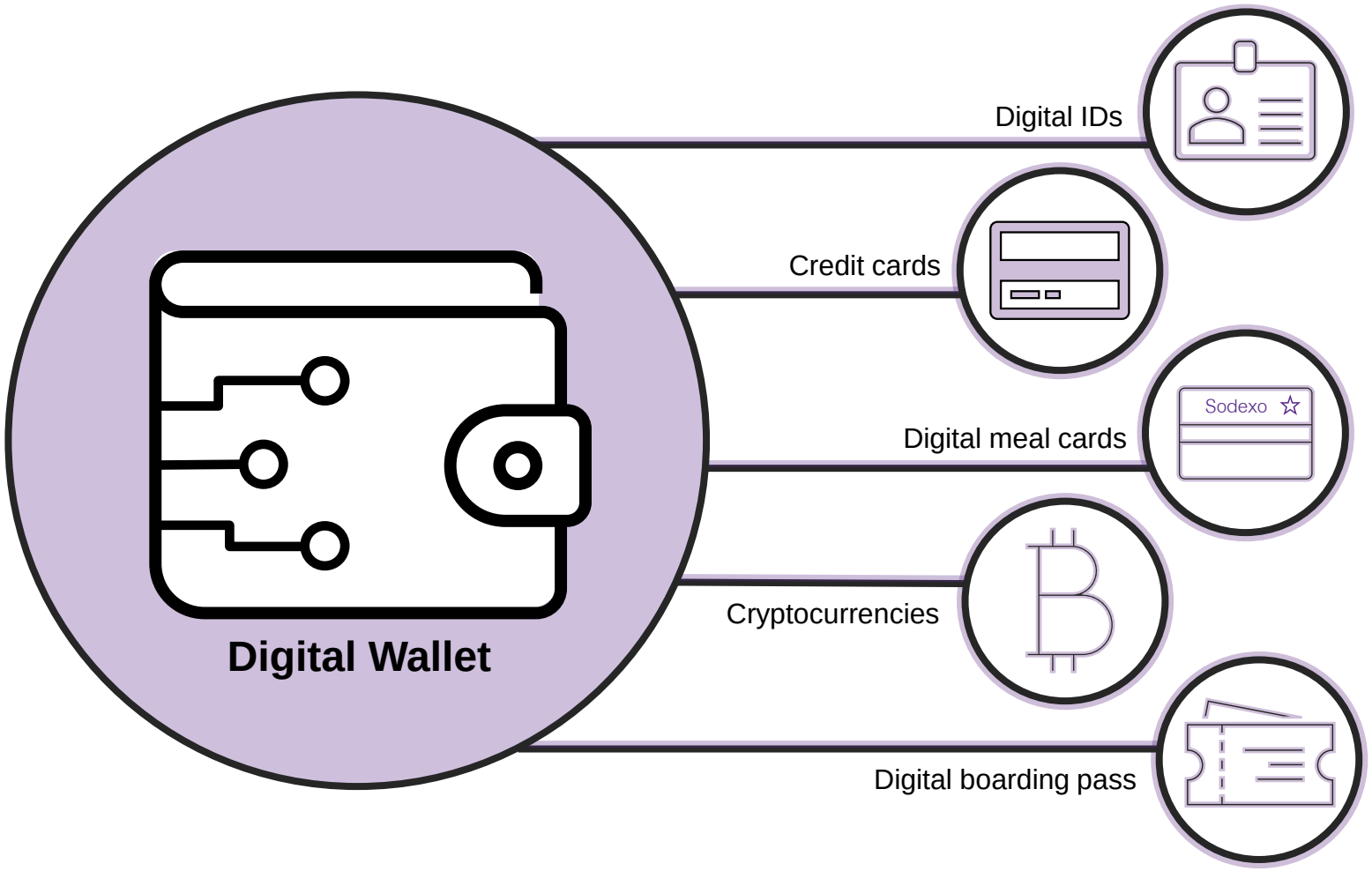
Source: Wilson and Hingnikar (2019) Solving Identity Management in Modern Applications: Demystifying OAuth 2.0, OpenID Connect, and SAML 2.0

SnT

- eID
- Biometric passports
- Video-identification
- Two-factor authentication & federated identity management
- **Wallet-based identity management**

UNIVERSITY OF LUXEMBOURG

Digital certificates (“electronic attestations of attributes“) are at the heart of digital wallets



Digital wallets are **software applications** that store and transmit payment authorization data for one or more credit or deposit accounts.

Digital wallets allows users to store **cryptographic keys** to manage crypto balances.

Digital wallets allow users to digitally identify themselves, store and manage identity data and official documents in an **electronic format**.

Requirements, inspired by today's physical means of representing identity attributes

Must-have: Combination of machine-readable and machine-verifiable attestations to identity attributes in one app

Should-have: Selective disclosure (incl. comparisons)

The verifiable presentation

HOLDER

Sends proof request, including challenge (“nonce”)

VERIFIER

Digital certificate

Attributes:
Last name: Gabler
First name: Erika
…

Binding key: 1234
Expiration: 31.10.2020
Signature: 5678

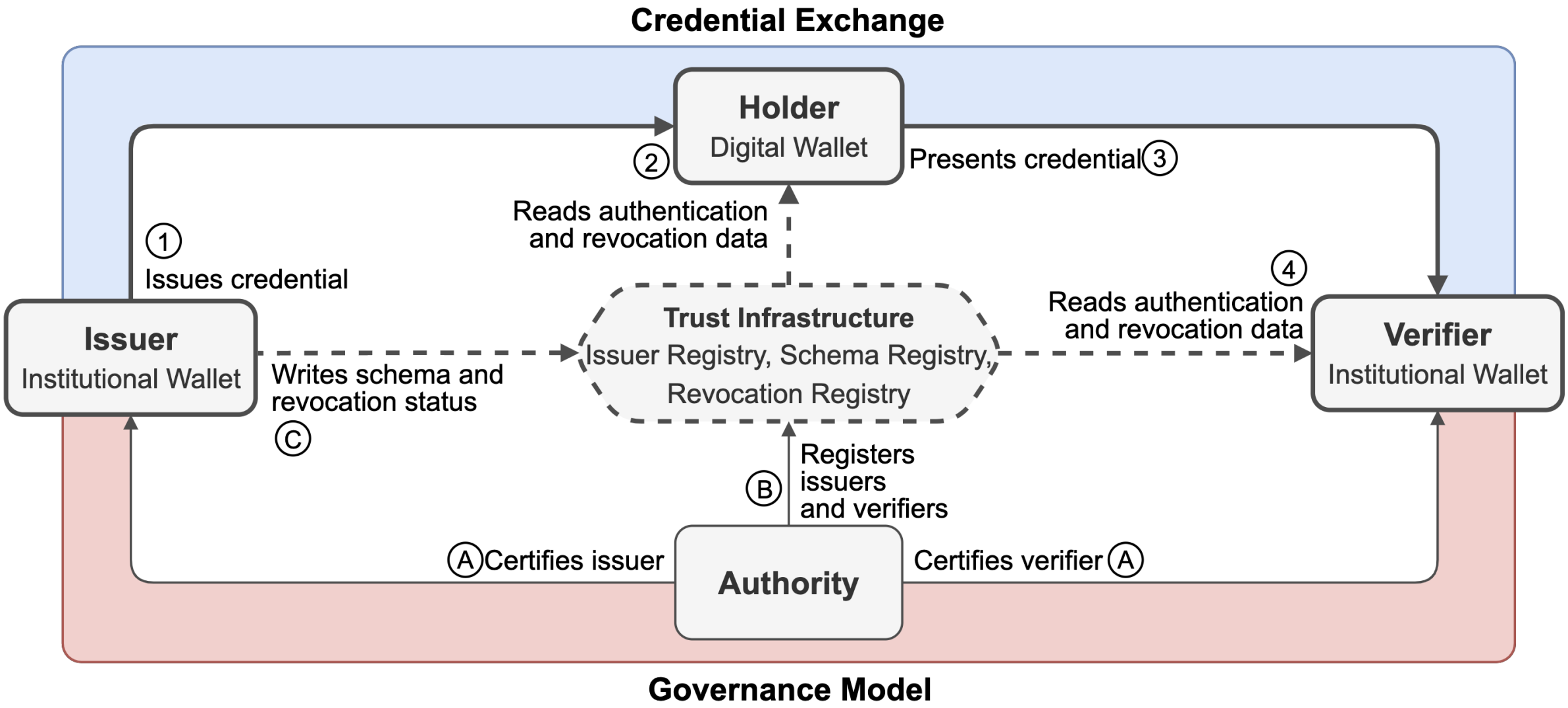
Sends certificate and signed challenge

Digital presentation

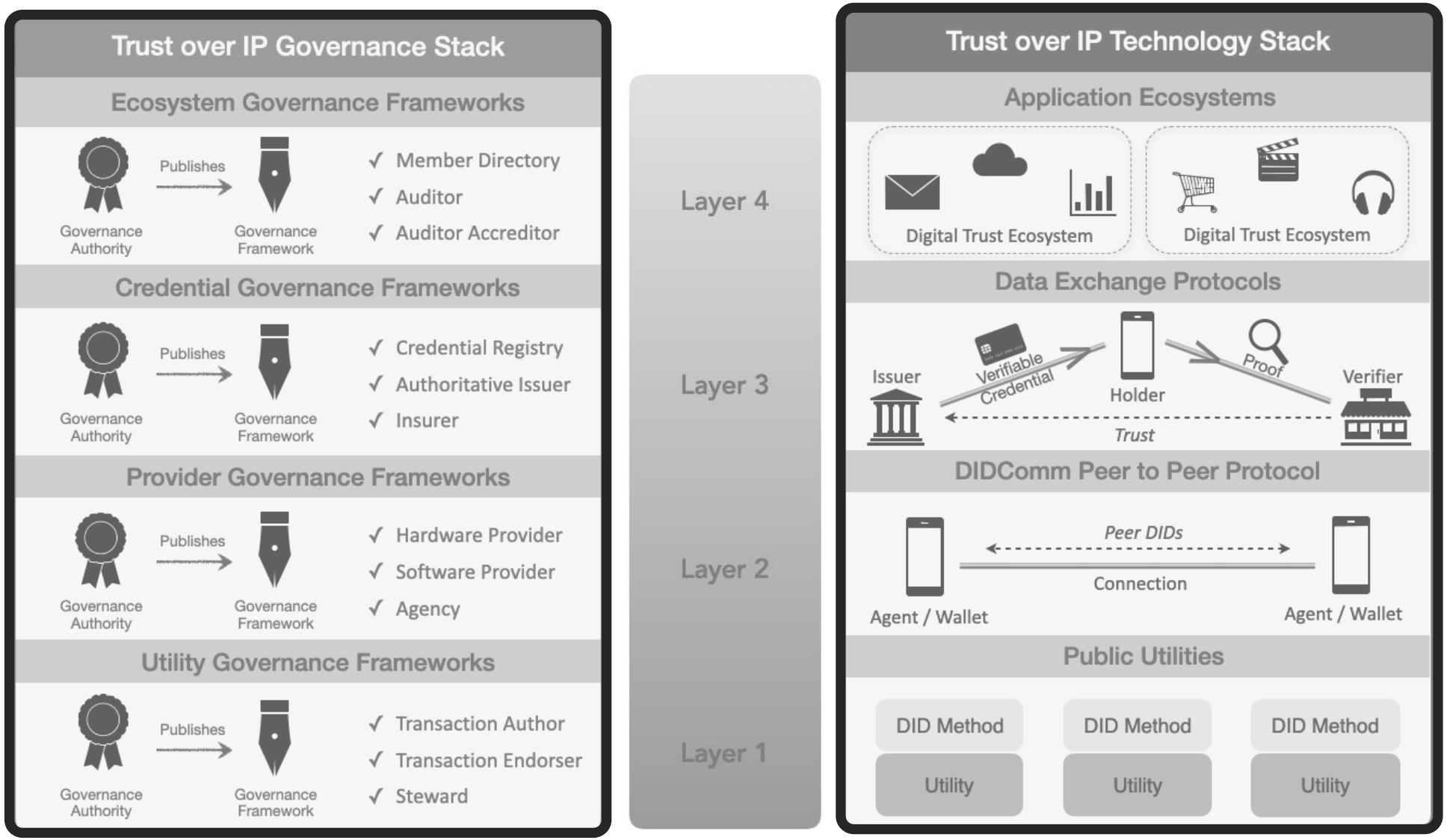
Attributes:
Last name: Gabler
First name: Erika
…

Binding key: 1234
Expiration: 31.10.2020
Signature: 5678

Technologies: Self-Sovereign Identity (SSI) for Decentralized IdM



Trust Over IP (TOIP): Stack



Source: https://github.com/hyperledger/aries-rfcs/blob/main/concepts/0289-toip-stack/README.md

Trust Over IP (TOIP): Layer 1 - Public Utilities for Decentralized Identifiers (DIDs)

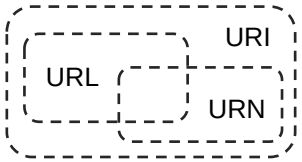
As individuals and organizations, many of us use globally unique identifiers in a wide variety of contexts.

URN (Uniform Resource Name, RFC 8141)

DIDs

URI (Uniform Resource Identifier, RFC 2396)

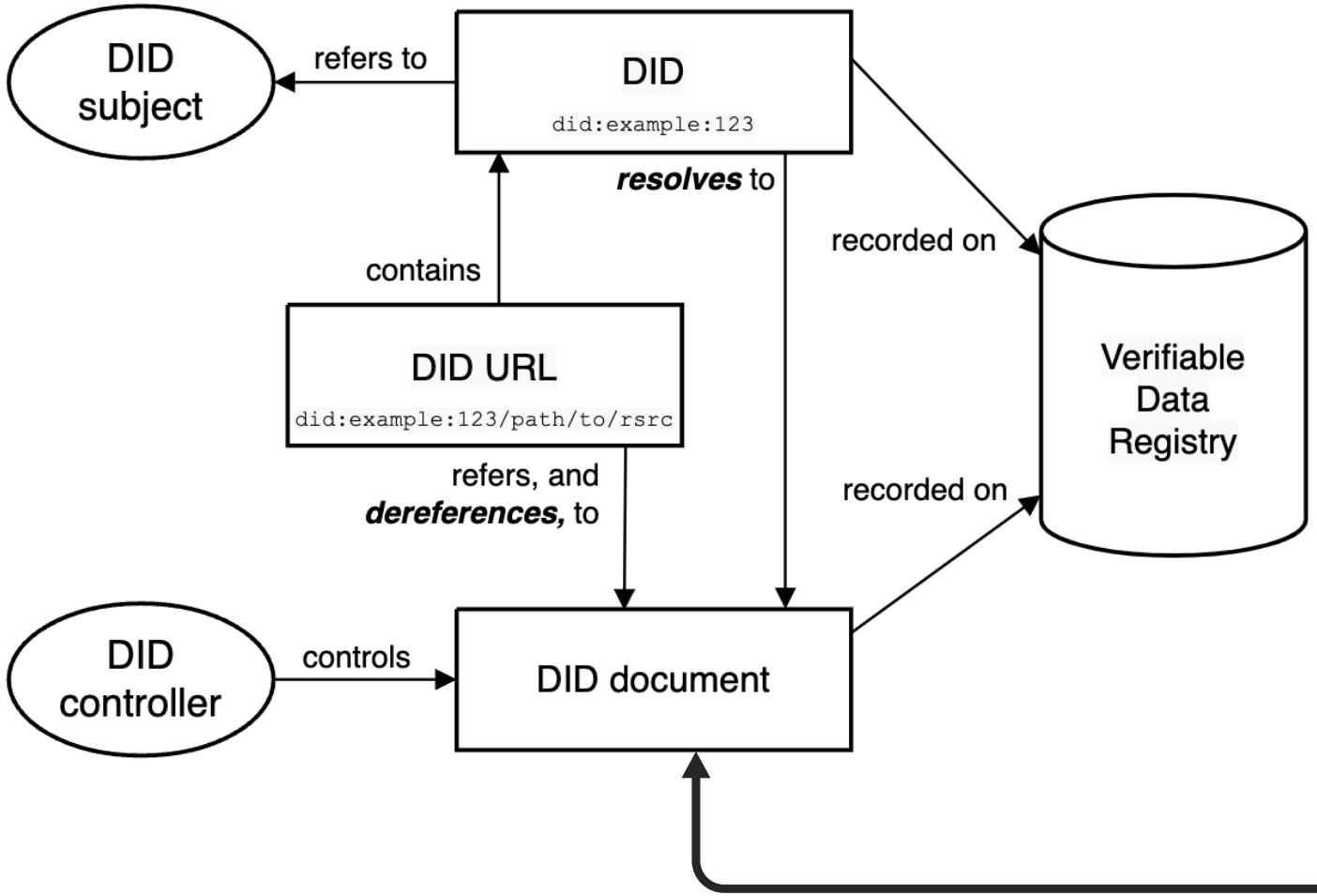
Note:



Source: https://www.w3.org/TR/did-core **and** https://en.wikipedia.org/wiki/Uniform_Resource_Identifier
See also: https://w3c.github.io/did-spec-registries/#did-methods

Trust Over IP (TOIP): Layer 1 - Public Utilities for Decentralized Identifiers (DIDs)

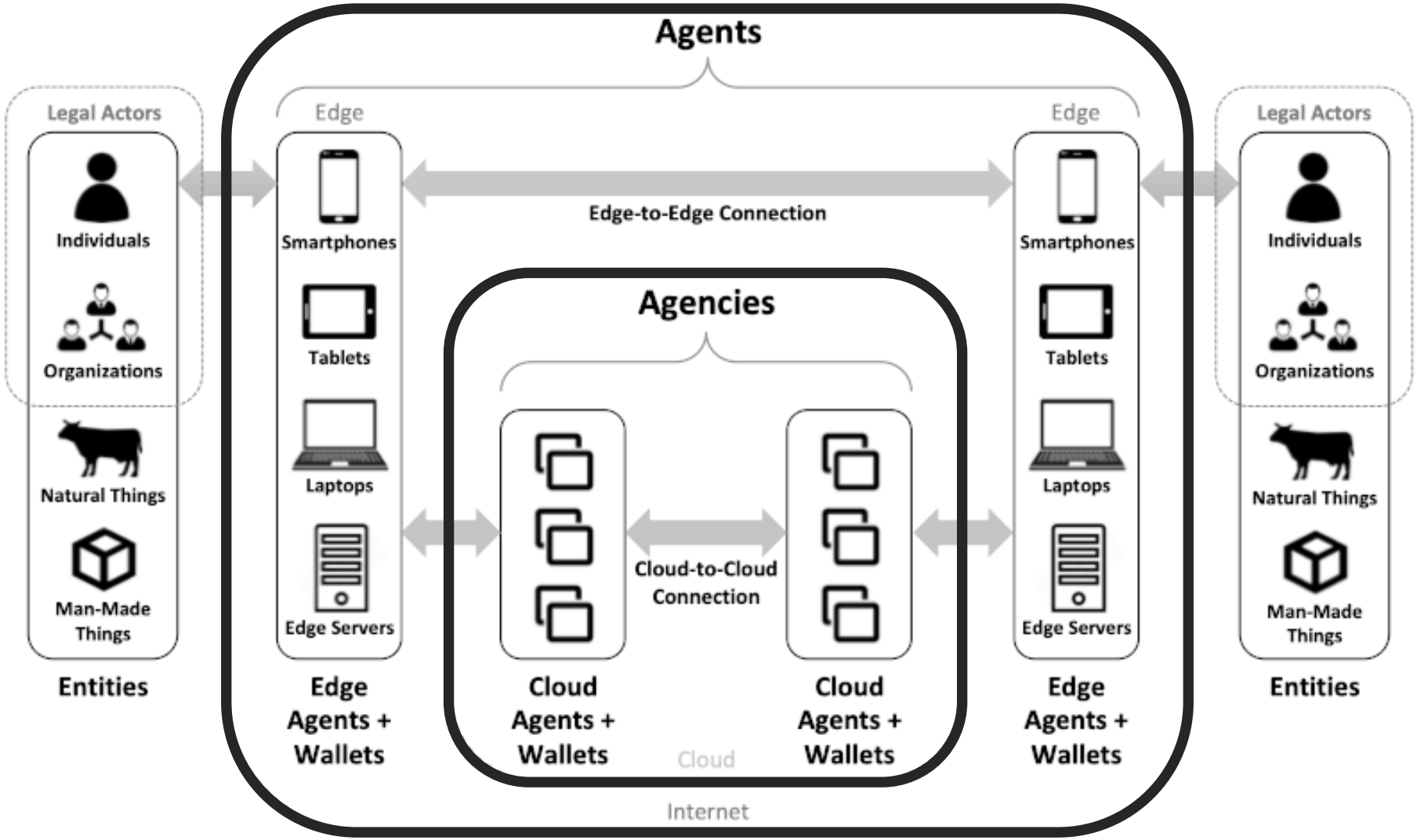
Figure: Overview of DID architecture and the relationship of basic components




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

Source: https://github.com/hyperledger/aries-rfcs/blob/main/concepts/0289-toip-stack/README.md#dids

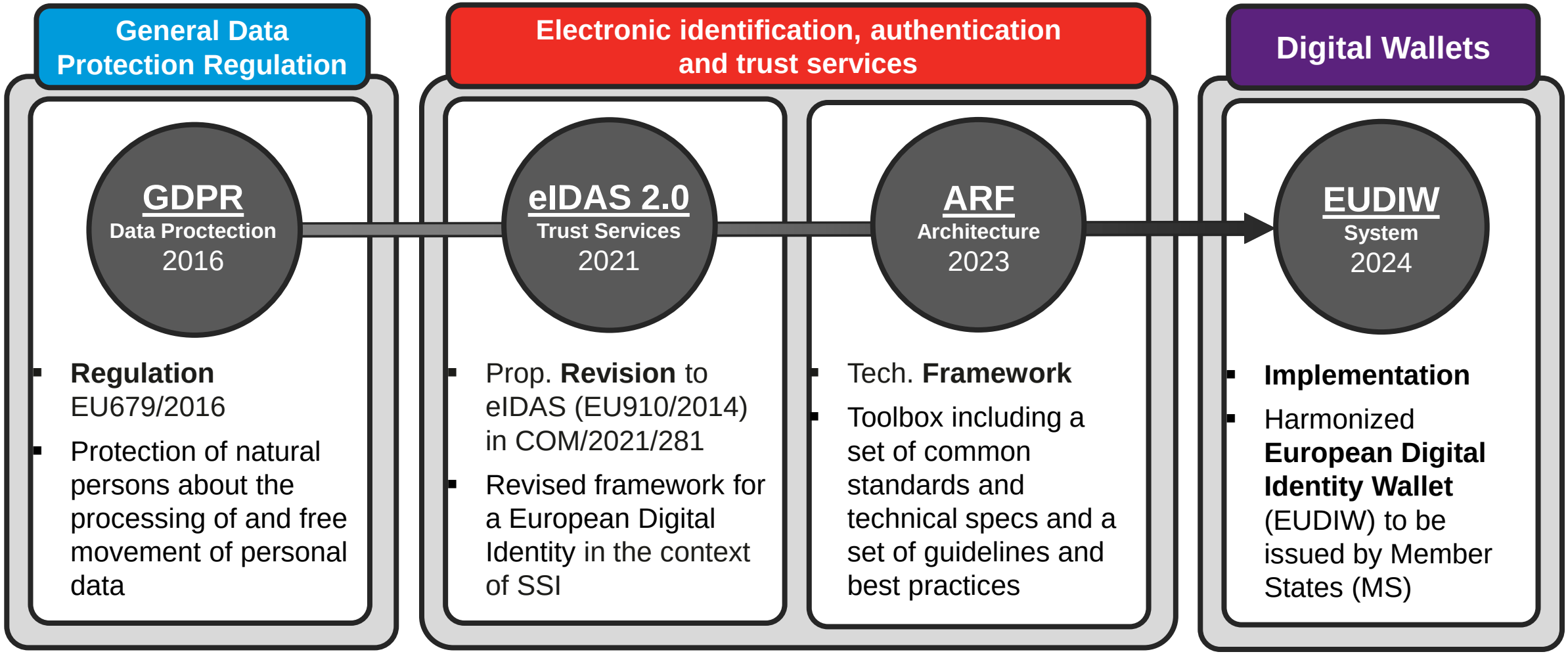
Trust Over IP (TOIP): Layer 2 - The DIDComm Protocol (P2P secure messaging)



Source: https://github.com/hyperledger/aries-rfcs/blob/main/concepts/0289-toip-stack/README.md#layer-two-the-didcomm-protocol

Regulation: Frameworks

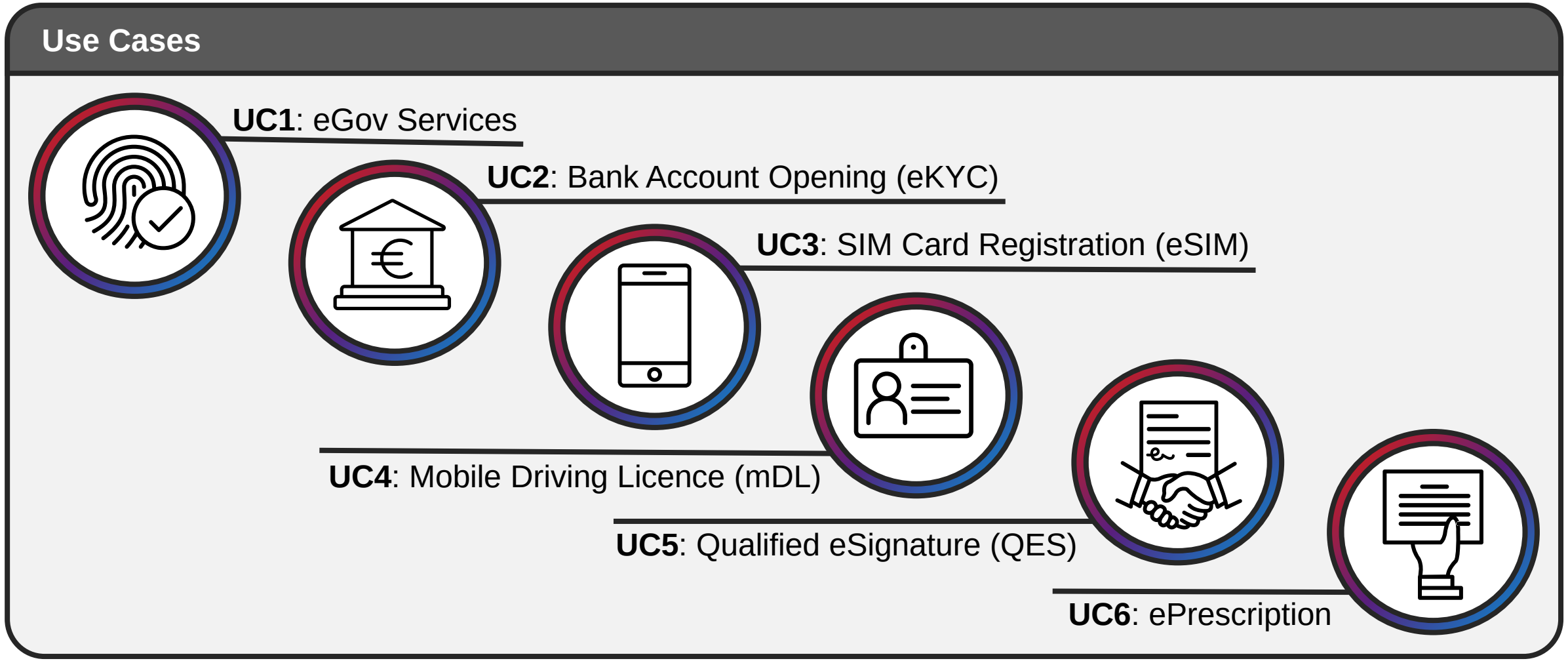
Example: eID and privacy protection in the European Union



See also: GDPR (EU679/2016) **and** eIDAS (EU910/2014) **and** Proposed Revision (COM/2021/281) **and** ARF (outline/toolbox)

Regulation: Architecture and Reference Framework (ARF)

Example: European Digital Identity Wallet (EUDIW)



Source: https://www.digital-identity-wallet.eu/

Technical: Standards and Specification

Standards for SSI technology provide an effective basis for digital identities which protects the privacy of personal data.

World Wide Web Consortium (W3C)

- **Decentralized Identifiers** (DID) v1.0
- Decentralized Identifier (DID) Resolution v0.2
- **Verifiable Credentials** (VC) Data Model 1.0
- … many more specifications

ISO TC 307 and CEN/CLC JTC 19

- ISO TC 307 is concerned with standards for **blockchain and DLT** (distributed ledger technologies)
- European Committee for Standardization (CEN): Standards for Blockchain and Distributed Ledger technologies

Standards and Specifications

Decentralised Identity Foundation (DIF)

- Responsible for developing standards and specifications building on those specifications produced by W3C for SSI
- Working groups: Identifiers and Discovery

ISO/IEC 23220 and 18013-5

- ISO/IEC 2322: Cards and security devices for personal identification – Building blocks for identity management via mobile devices
- ISO/IEC 18013-5: Standard for a mobile driving licence (mDL) application

Source: https://www.enisa.europa.eu/publications/digital-identity-leveraging-the-ssi-concept-to-build-trust

The Wallet-based paradigm: Synthesis or disruption?

- Incorporates learnings from different approaches to digital identity management (the standardized interfaces and mature user experience of federated IdM but without the dependency and privacy issues; the security of ID card without the inconvenience and high entry barriers, etc.). It also extends the scope of verifiable identity data much further.
- Cryptography to achieve higher degrees of security and privacy
- No need to carry analog documents any more
- Consistent user experience across different process flows (public administration and private life; high and low levels of assurance; daily (key usage) and rarely (ID card or passport use).
- Software-based wallets also incorporate more flexibility to customize and automate certain workflows inside the wallet (choice of credentials, renewals, change of device)
- The “funneling“ of identity-related activities can further increase the level of assurance by increasing the barrier to identity sharing and selling

Summary: Electronic identification methods

In-person identification		Online identification				
	Non-electronic	Electronic				
	ID card	(Biometric) passport	eID function of the ID card	Video identification	Other two-factor authentication	Verifiable credentials and digital wallets
Exchanged data	Optical security features, biometric picture, all identity attributes	Digitally signed identity attributes	Cryptographic binding (chip authentication), pseudonym, required identity attributes	Picture from the ID card or biometric passport (including all identity attributes), live video with biometric characteristics	Password in combination with cryptographic binding in the mobile phone or one-time password (OTP) communicated via SMS/TAN	Cryptographic binding in the mobile phone, required digitally signed identity attributes
Users require access to	ID card	Passport (including CAN) NFC reader application	ID card PIN Card reader with eID functionality or mobile phone with NFC interface and app (e.g., Ausweis-App 2)	ID card or passport	Password, private key stored in the mobile phone or corresponding SIM card in the mobile phone	Digitally signed identity attributes, private key stored in mobile phone
Relying parties require access to	–	–	Relying party access certificates	–	–	Potentially relying party access certificates or inclusion in national registries
Attack vectors / disadvantages	Manual Not applicable to online identification	Only one factor (possession of the passport) for online identification	High entry barriers (certification) for relying parties	Long-term security questionable because of improvements of AI-generated pictures and videos	Bad password management by users, sim-swap fraud, potentially required TAN generator	Tradeoff between entry barriers for relying parties and security
Level of assurance	High	In person: high Remote: low	High	Substantial to high (long-term: probably low)	Substantial – high	Low – high

Source: Omlor, Sedlmeir & Urbach (2024). Identifizierungsniveaus bei notariellen (Online-) Verfahren: technische Grundlagen. *ZIP – Zeitschrift für Wirtschaftsrecht*

SNT

Identity Management: More Terminology

UNIVERSITY OF LUXEMBOURG

Identity: Definitions

It is challenging to define ”identity" as its meaning varies significantly based on context and interpretation

- **Legal**: (Thales, 2021)
  *“A legal identity is the **registration** and documentation of a person that enables **access** rights, benefits and responsibilities in their country. This can include **documentation** of name, personal data, date of birth…”*

- **Technical**: (ITU-T Y.2720, 2009)
  Digital identity is *“information about an entity that is sufficient to identify that entity in a **particular context**.”*

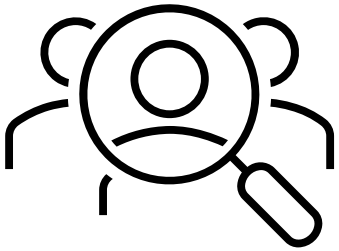
- **Management**: (Pfitzmann and Hansen, 2009)
  *“An identity of an individual person may comprise many **partial identities** of which each represents the person in a specific context or role. A partial identity is a subset of **attribute values** of a complete identity, where a complete identity is the union of all attribute values of all identities of this person”*

- **Philosophical**: (Noonan, 2019)
  TL;DR: It’s complicated …

Digital Identity: Composition

Identity (defined by) **Attributes** (verified by) **Credentials**

Identity is a concept applicable to all entities, encompassing individuals, systems, and organizations.



Identity consists of

Identifier != identity

Identifier
A series of digits, characters, and symbols or any other form of data used to identify a subject.

Example: username, pseudonyms, public key, SSN, IP, MAC, URI, ...

Weak vs. strong

Credentials
A set of data providing evidence for claims about parts of our entire identities.

Example: passwords, SAML assertions, bearer token, Kerberos ticket

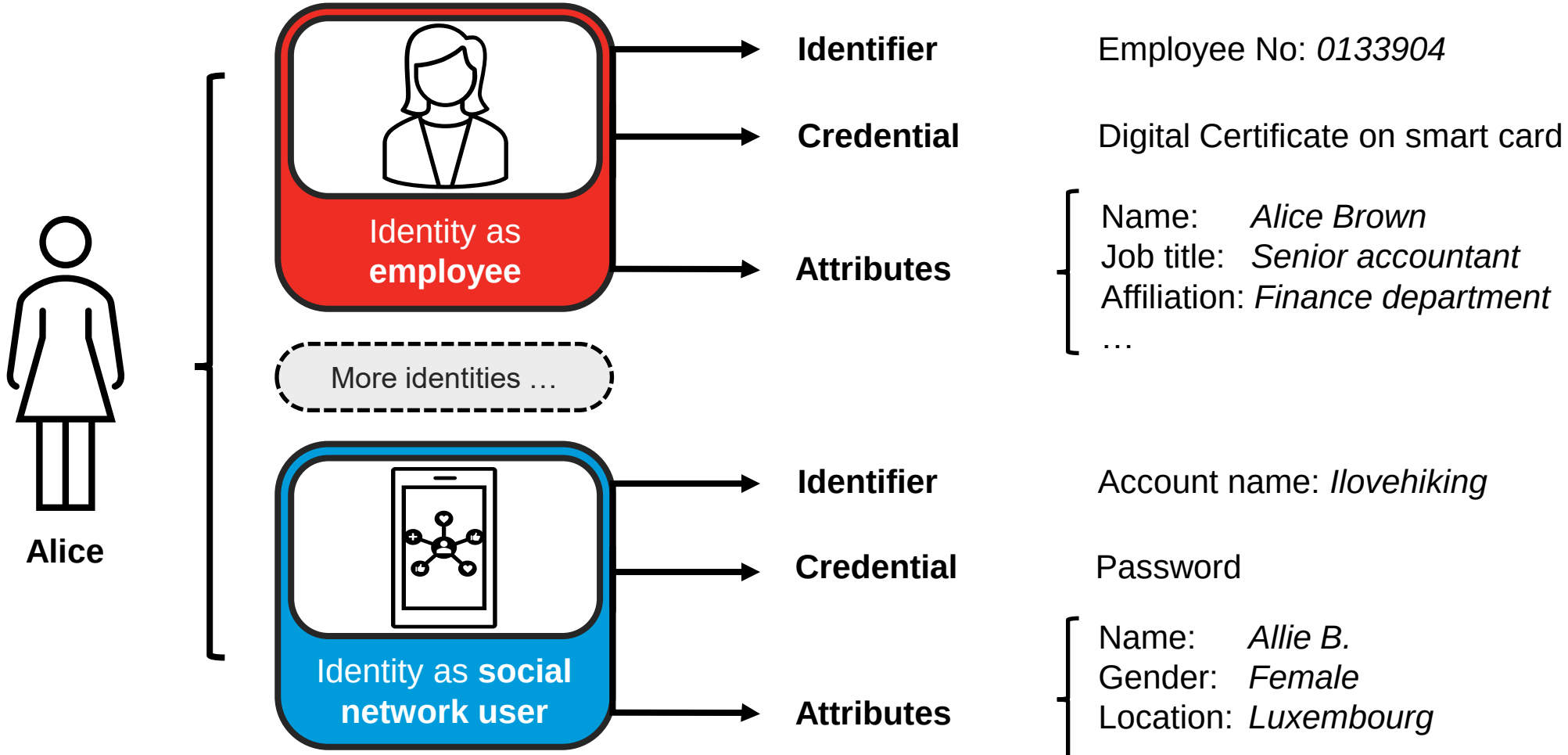
Attributes
A set of data that describes the characteristics of a subject.

Example: full name, date of birth, roles, genders, titles, affiliations, activity records, reputations

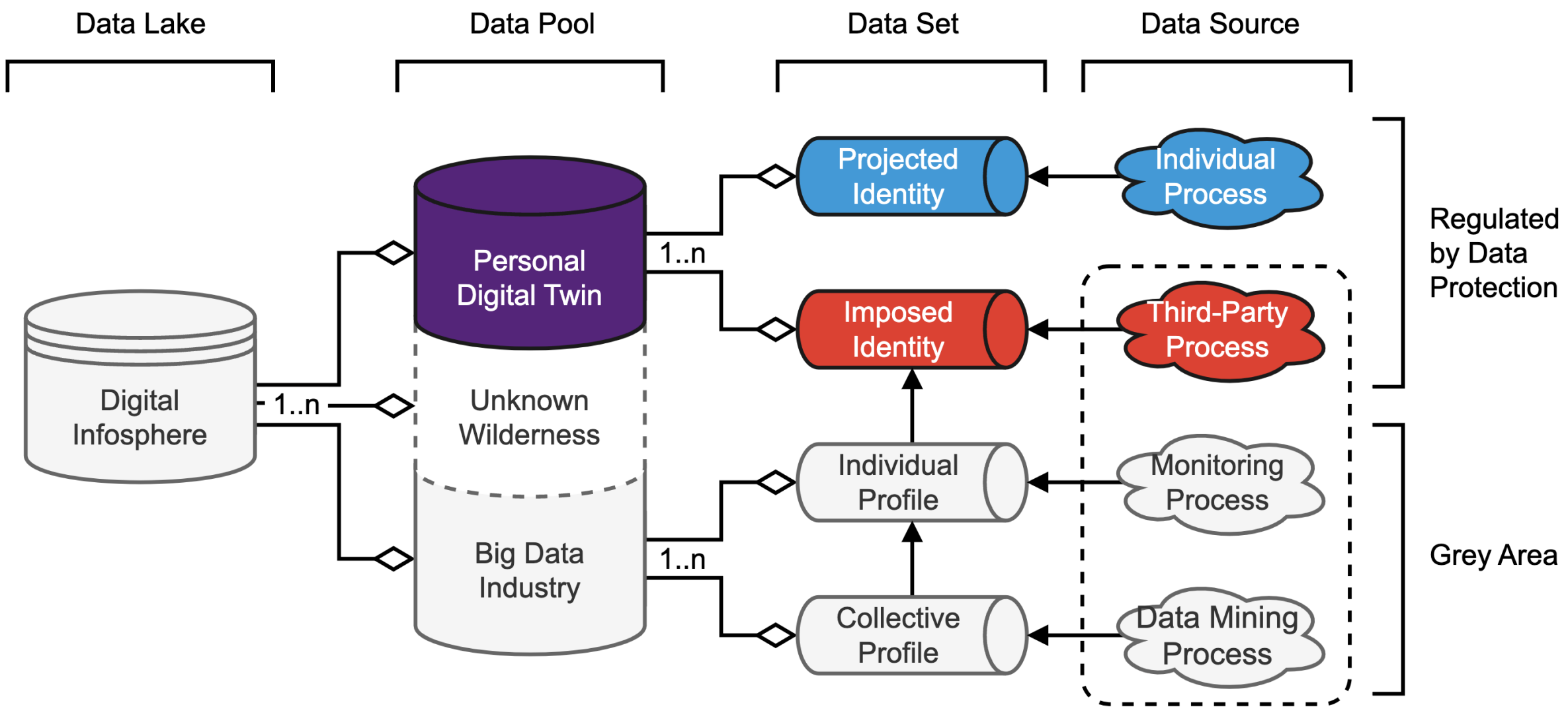
Source: Y.2720:NGN Identity Management Framework, 2009

Fragmented identity and digital fingerprints

A subject can have more than one identity



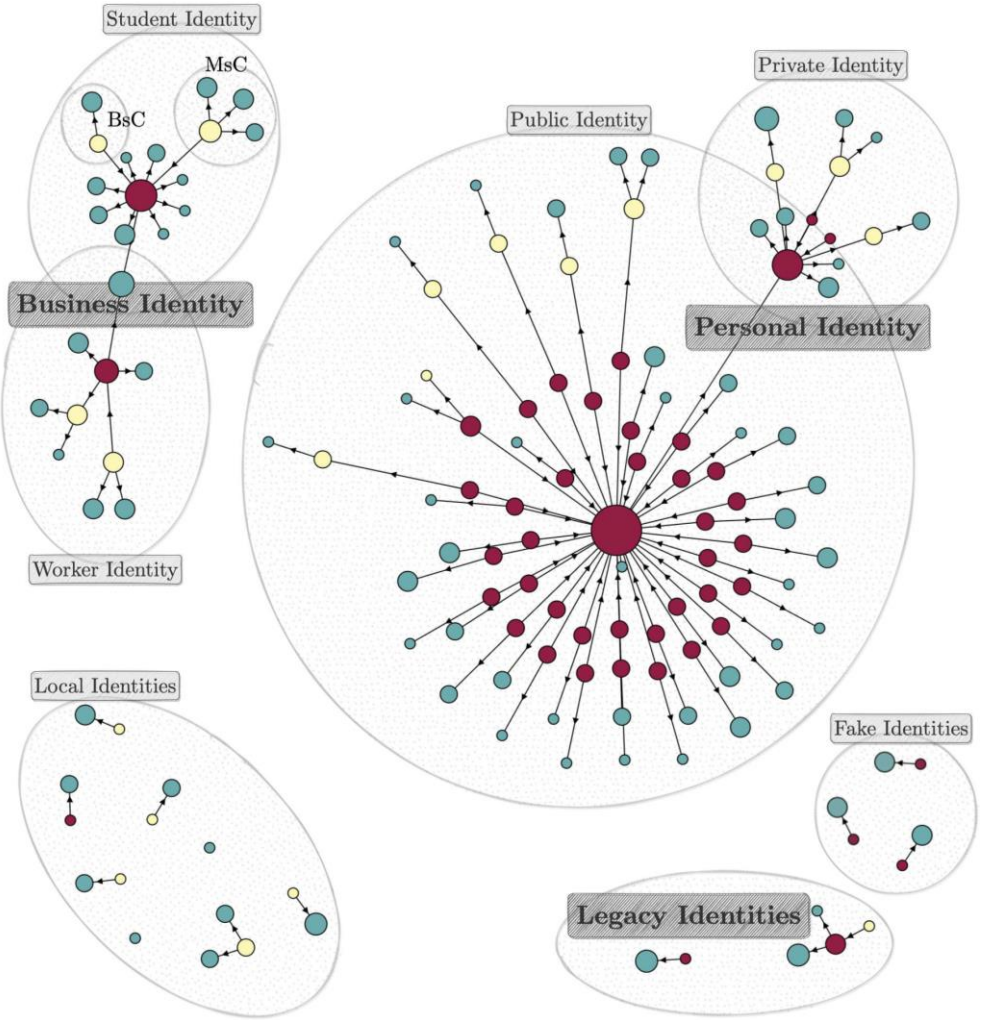
Digital infosphere: Digital identity vs. digital profile

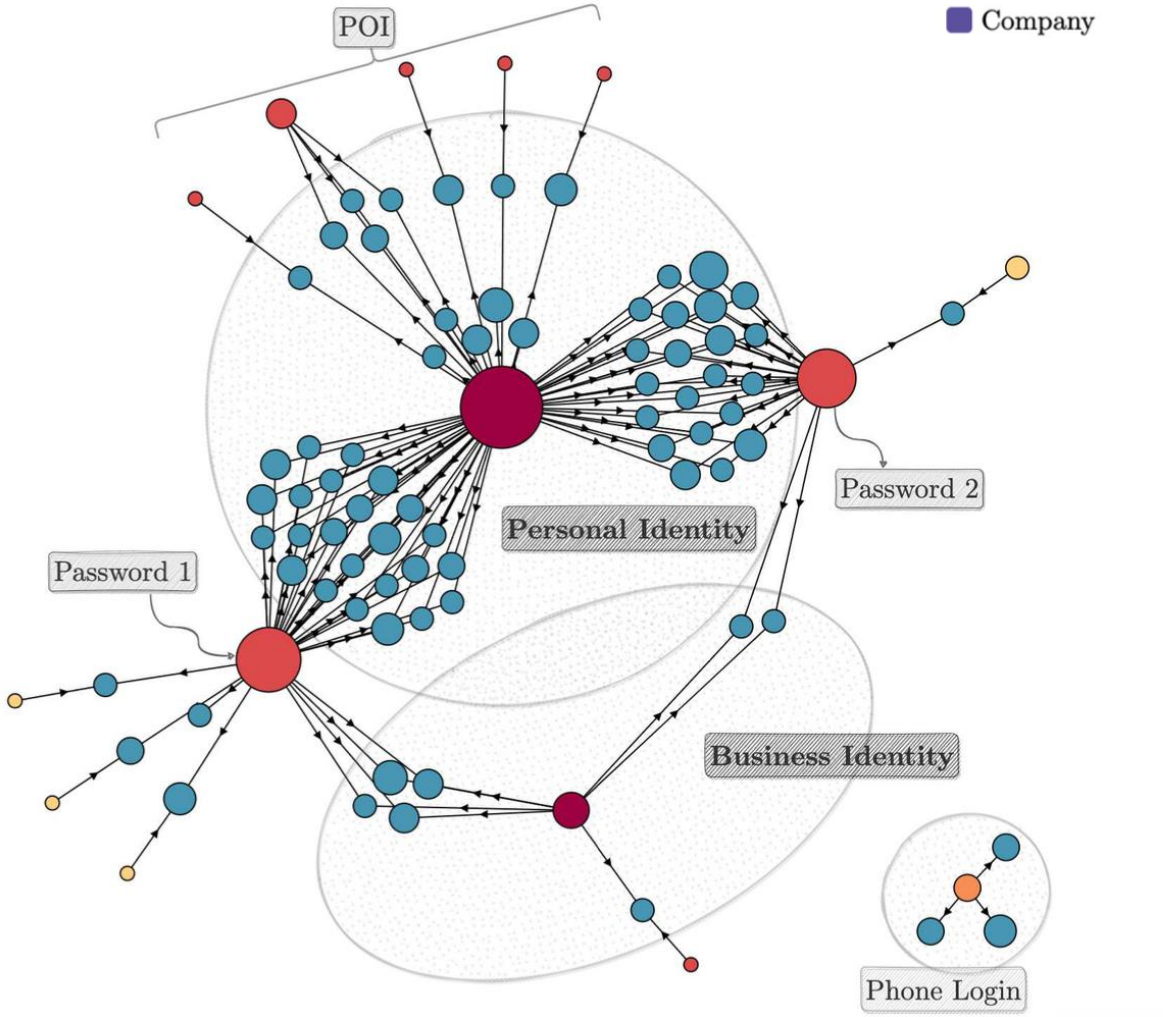


Source: Hölzmer (2022) Evaluation of Personal Digital Twins for Improving Privacy Awareness and Protection in the Digital Era – Figure adapted from Roosendaal (2010)

Digital personae aka. Personal Digital Twin

Case Study: Visualizing an individual's digital counterpart, composed of several partial identities





Try yourself: https://gitlab.com/personal-digital-twin/pdt (Check tested setups in the wiki)

Identity Management: Definitions

Mind the distinctions and nuances between each concept

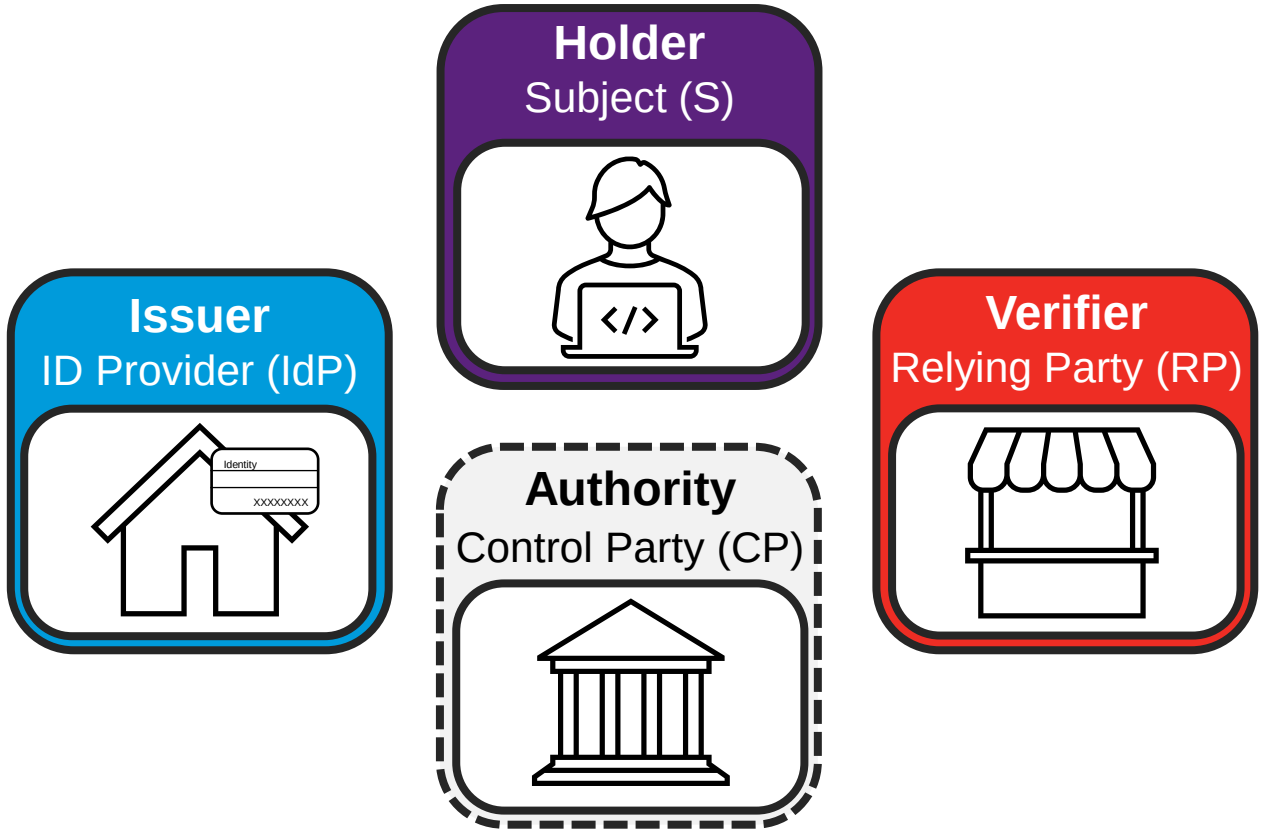
Processes:

- **Identification**:
  The process of presenting or verifying an entity's characteristics that (uniquely) identifies them in a set of entities.
- **Authentication** (AuthN):
  The process of verifying the identity of an entity by validating credentials against a trusted source.
- **Authorization** (AuthZ):
  The process of determining what access permissions an authenticated entity has.
- **Accountability**
  The process of keeping track of a user's activity while accessing the system resources

Frameworks: Processes and Technologies

- **Identity Management** (IdM):
  Involves the management of identifiers and credentials, their authentication, authorization, and privileges within or across information systems.
- **Identity and Access Management** (IAM):
  A broader term that covers identity management, as well as provisioning access to system resources by including technologies like Single Sign-On (SSO) and Multi-Factor Authentication (MFA), or processes such as Role-Based Access Control (RBAC) and Attribute-Based Access Control.
- **Authentication, Authorization and Accountability** (AAA):
  A framework of processes for IdM and IAA as the core of identity and network security policies.

Repetition: Identity (and Access) **Management**: Stakeholders



Note: IP, CP, and RP could be the same entity

Credentials: Definition

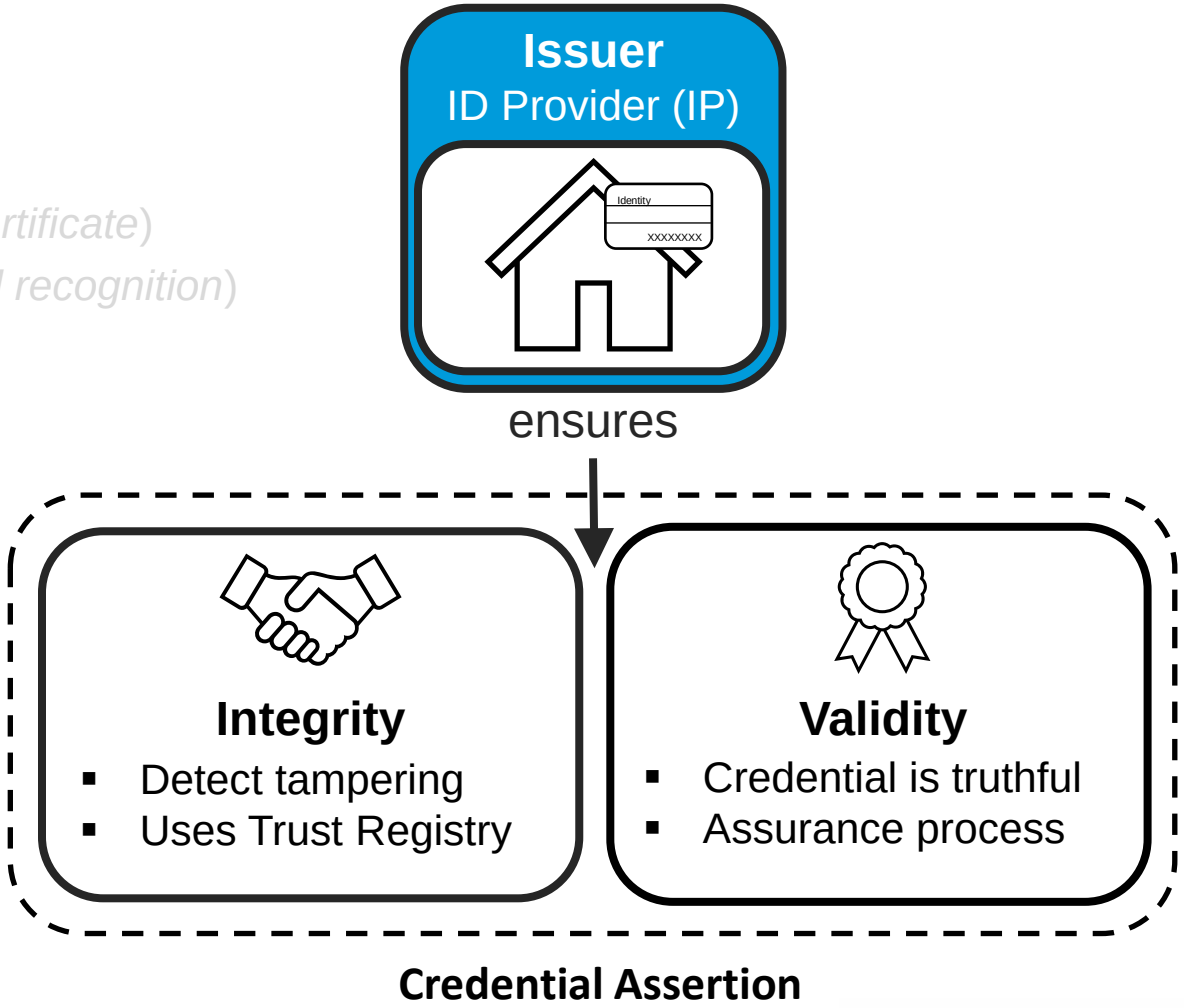
A credential is a set of attributes and assertions by an IdP that verifies the claim of an identity.

Types of Credentials:

- Something you know (*e.g., passwords, PINs, memnonic*)
- Something you have (*e.g., smart cards, security tokens, certificate*)
- Something you are (*e.g., biometrics like fingerprint or facial recognition*)

Security Requirements:

- Secure storage (*e.g., encrypted databases*)
- Secure transmission (*e.g., SSL/TLS encryption*)
- Secure handling (*e.g., multi-factor authentication*)
- Assurance Level Compliance (*e.g., eIDAS*)



Credentials: Classification

This classification has been devised for physical credentials, such as passports and driving licenses.

Primary identity credentials	**Secondary** identity credentials	**Tertiary** identity credentials
▪ Core identity elements usually issued by authoritative entities	▪ Derived from primary credentials or independently issued with lesser authoritative rigor	▪ Commonly self-asserted or less strictly vetted, often used for lower assurance scenarios
▪ **Issued**: by-products of significant life event (mostly issued once)	▪ **Issued**: response to AuthZ request to perform an action	▪ **Issued**: by an authority or organization for a limited purpose
▪ **Examples**: Birth certificate, passport	▪ **Examples**: Driver's licenses, employee ID	▪ **Examples**: Library cards, club memberships
▪ **Digital Analog**: Biometric Passports	▪ **Digital Analog**: mDL, smart cards	▪ **Digital Analog**: UN/PW, social media profiles

Source: https://pages.nist.gov/800-63-3/

Credentials: Digital Analog

Physical vs. digital credentials:

- **Physical credentials**: Tangible items proving identity, usually carried by the individual and having tamper-proof characteristics like watermarks, holograms, …
- **Digital credentials**: Electronic counterparts providing (legally) the same proof digitally
- **Transition**: Various technologies, like NFC, RFID, and biometrics, are bridging the gap, transitioning physical credentials to digital platforms, and enhancing verification processes.

Digital certificate

Attributes:

Name:	Specimen
Given name:	Jeanne
Sex:	F
Nationality:	LUX
Date of birth:	19.08.1983
…	
Binding Key:	123456789
Signature:	56c8…fa34
Expiration:	31.10.2020

Credentials: Digital Transition

NFC Chip Integration:
- ID cards embedded with NFC chips
- The chip contains a digital certificate
- Provides a secure method of identity verification

Machine Readable Text:
- A strip of machine-readable text on the back of the card
- Contains machine-readable essential information
- Facilitates quick data capture and verification chips

Advantages: Enhanced security; speedy verification; digital transition
Disadvantage: Slow adoption; bad realization

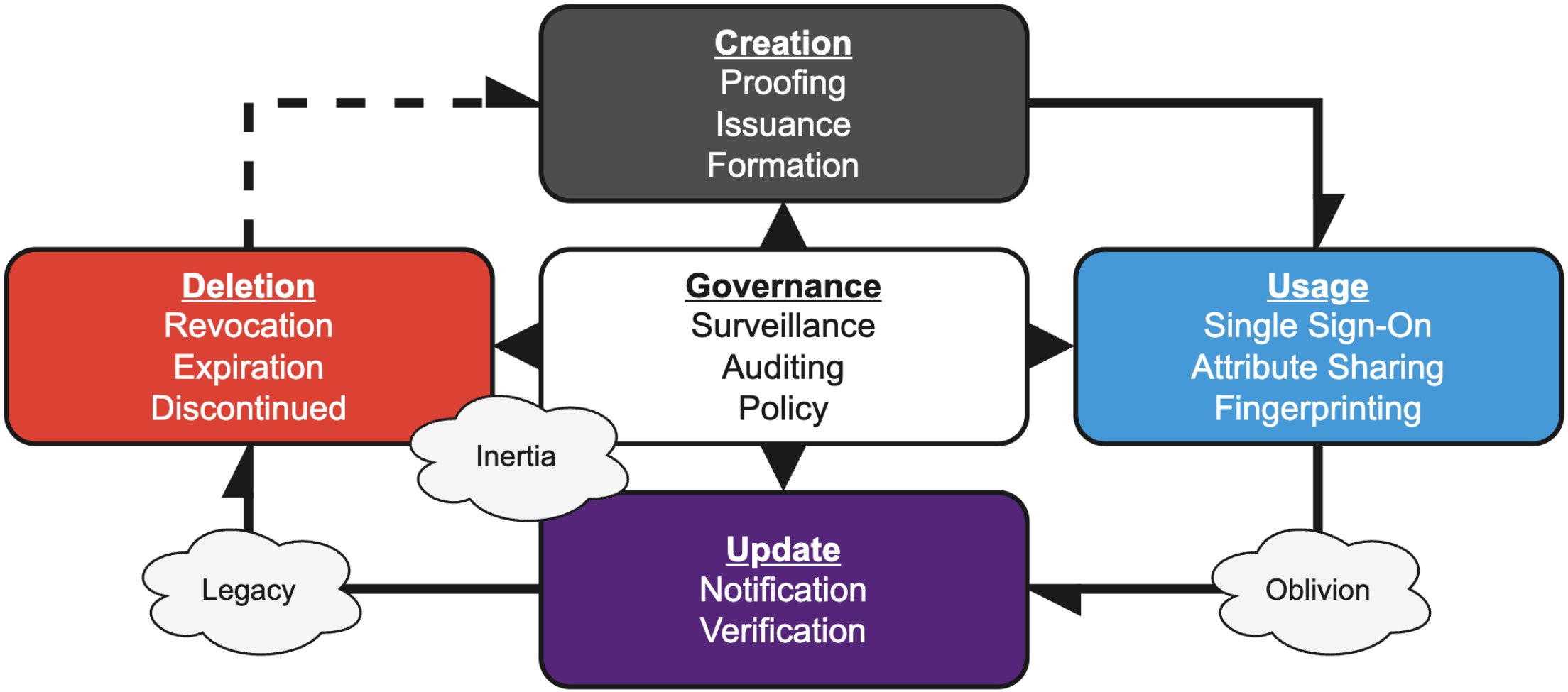
Digital certificate

Attributes:

Name:	Specimen
Given name:	Jeanne
Sex:	F
Nationality:	LUX
Date of birth:	19.08.1983
…	
Binding Key:	123456789
Signature:	56c8…fa34
Expiration:	31.10.2020

Identity Lifecycle: Overview

Conceptualized stages a digital identity may reside in with example processes and special cases.



Identity Lifecycle: Creation

1. Attribute proofing

- The process by which an Identity Provider (IdP) collects, validates and verifies information about a person for the purpose of issuing credentials.

2. Credential issuance

- Post-verification, credentials are issued by IdP (e.g., digital certificates) or by the subject whom credentials are about (e.g., chosen password).

3. Identity formation

- Identities are comprised of proofed attributes, issued credentials, and identifiers that are assigned by third parties or by the subject.

Example Scenario: Employee Onboarding

- HR collects personal information, including name, date of birth, and proof of identification (e.g., passport).
- Educational and professional credentials are verified by contacting the respective institutions and prior employers.

- Post-verification, a corporate email address and an initial password is issued by the IT department.
- A digital ID card that grants access to secure areas within the corporate premises is also provided.

- A unique employee ID is generated, forming the corporate identity within the organisation.
- The identity is associated with the role, department, email address, and digital ID card within the corporate directory.

Identity Lifecycle: Usage

Three functions are commonly used by identity-enabled services: trusted communication, single sign-on, and attribute sharing.

Trusted communication

- Ensures that RP can accurately discover, differentiate, and authenticate identities, fostering a secure and reliable exchange of messages, e.g., simple Login with UN/PW
- Necessary basis for trusted and secure **identity transactions**

Single sign-on

- A method that enables a subject to use an access credential issued by the IdP to authenticate against multiple RPs
- **Technical Standards**:
  SAML, OIDC, OAuth, Kerberos, LDAP, JWT

Attribute Sharing

- A method that allows identity providers to share the attributes of subjects with relying parties (upon authorization by the subject).
- **Technical Standards**:
  SAML, OIDC, OAuth, LDAP, SCIM, FIM

Identity Lifecycle: Update

Identity data are continuously updated through their life cycles.

Reasons

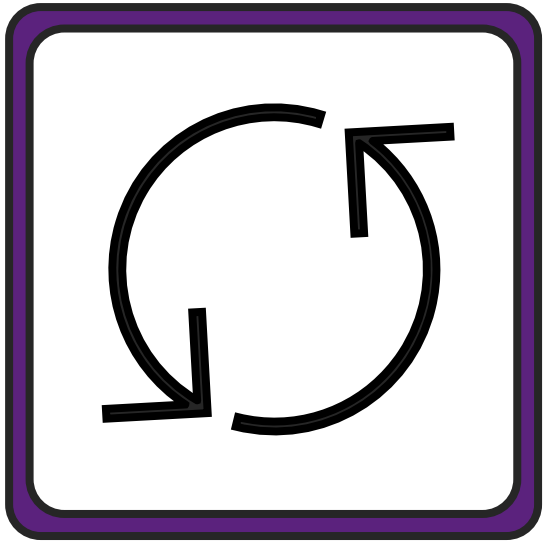
- Credentials revoked and re-issued
- Attributes change over time
- Credentials expires

Requirements

- User notification
- Attribute verification

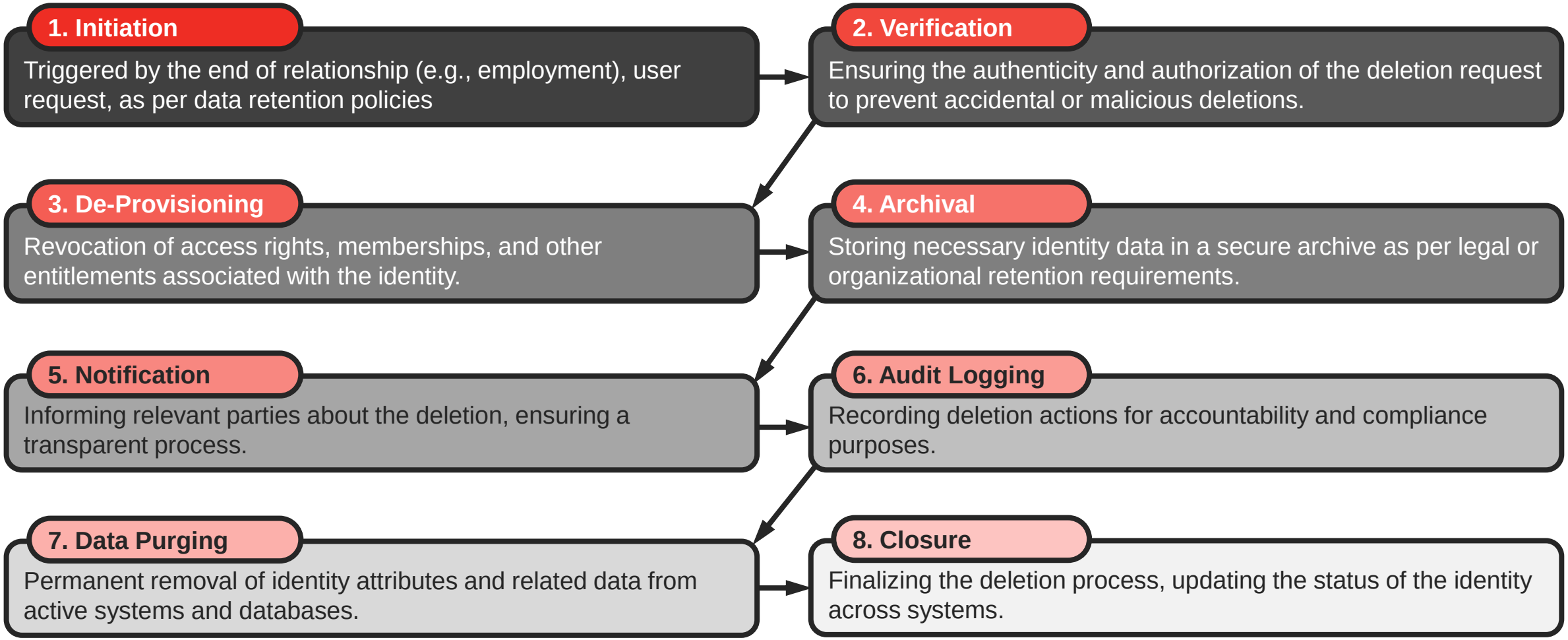
Types

- Implicit
- Explicit



Identity Lifecycle: Deletion

Ensures a structured and secure approach towards the removal of identities while adhering to legal and organizational standards.

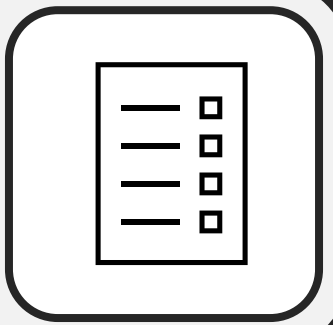


Identity Lifecycle: Governance

Identity governance is the key framework for the compliance to internal control and governance-related regulations.

Audit Trails

- Include detailed information of each transaction that involves identities in a trusted and provable manner so that any repudiation can be mitigated.

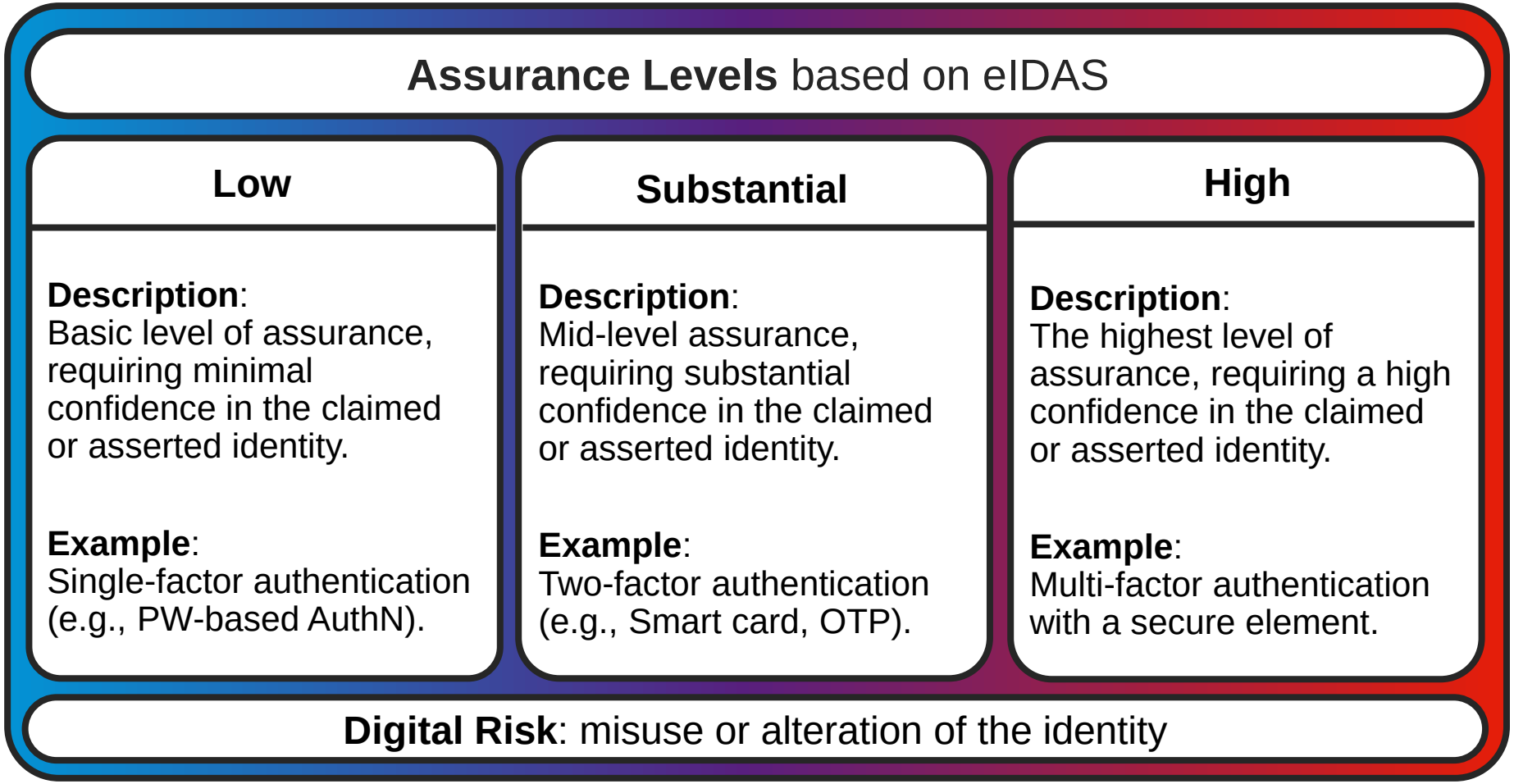


Identity Policies

- For authentication and authorization
- Authentication policies define the required **level of identity assurance** for a given transaction.

Governance: Assurance Level

Some identity transactions require specific assurance levels to comply with legal requirements and best practices to mitigate digital identity risk.



Source: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32014R0910

Excerpt from the implementing act specifying the LoAs

COMMISSION IMPLEMENTING REGULATION (EU) 2015/1502

of 8 September 2015

on setting out minimum technical specifications and procedures for assurance levels for electronic identification means pursuant to Article 8(3) of Regulation (EU) No 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market

2.2.1. Electronic identification means characteristics and design

Assurance level	Elements needed
Low	1. The electronic identification means utilises at least one authentication factor. 2. The electronic identification means is designed so that the issuer takes reasonable steps to check that it is used only under the control or possession of the person to whom it belongs.
Substantial	1. The electronic identification means utilises at least two authentication factors from different categories. 2. The electronic identification means is designed so that it can be assumed to be used only if under the control or possession of the person to whom it belongs.
High	Level substantial, plus: 1. The electronic identification means protects against duplication and tampering as well as against attackers with high attack potential 2. The electronic identification means is designed so that it can be reliably protected by the person to whom it belongs against use by others.

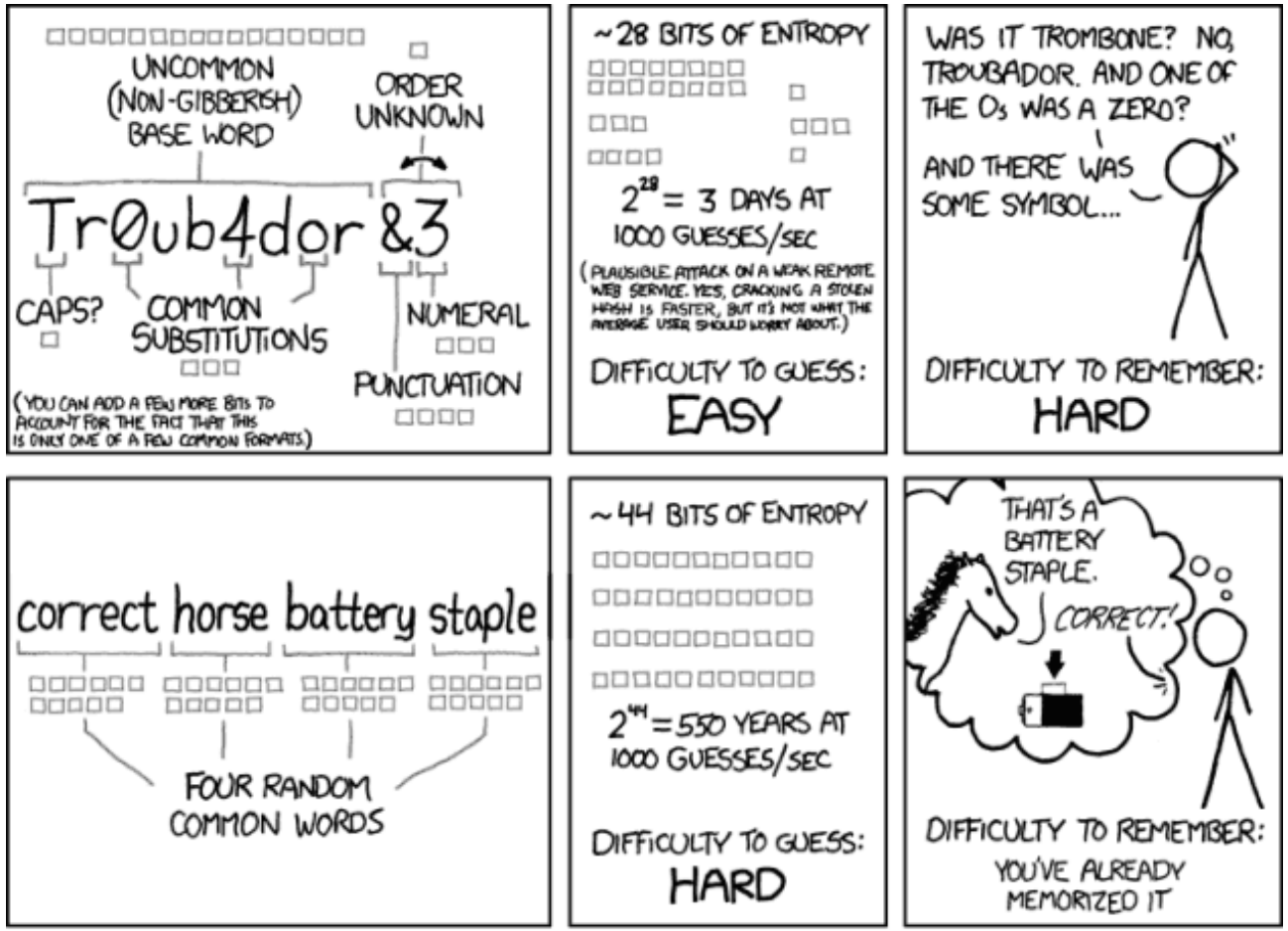
SNT

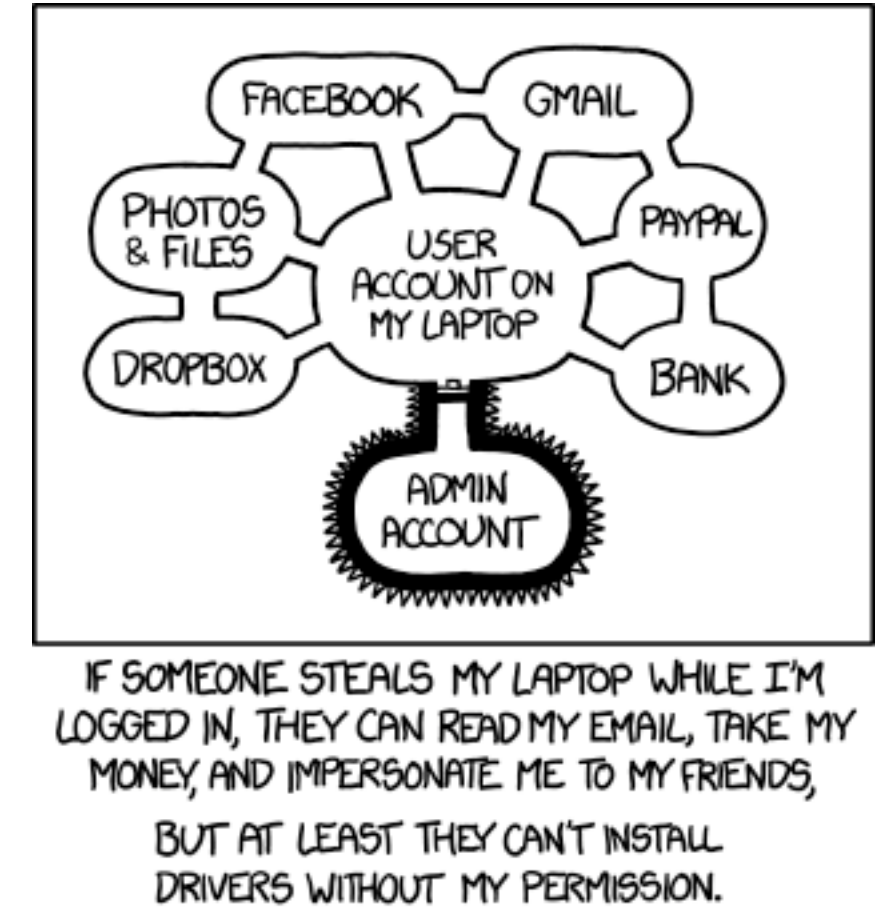
Identity Management: Common Challenges

UNIVERSITY OF LUXEMBOURG

Challenge: Security

Identity Hijacking: Authentication and access control

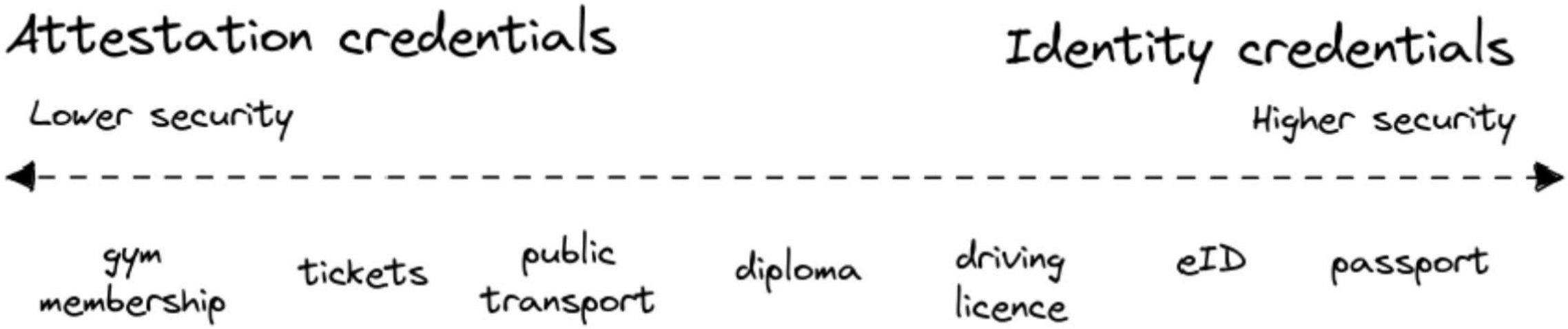




Source: https://xkcd.com/936 (Password Strength) **and** https://xkcd.com/1200 (Authorization)

Challenge: Security

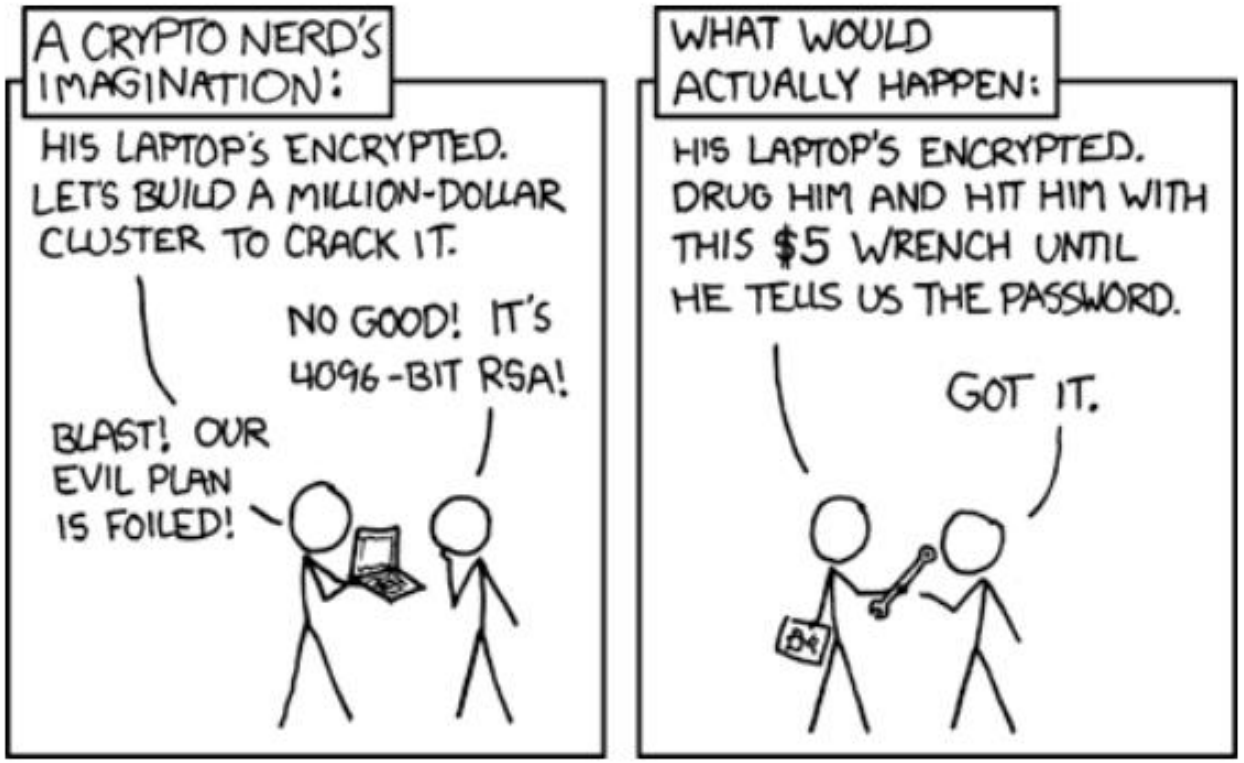
Attestation vs. identity credentials: Differentiation of credentials based on security requirements



Source: Bastian et al. (2023) Concepts for Secure Wallets in Decentralized Identity Ecosystems

Challenge: Security

Ultimately, cryptography is only a small part of security.



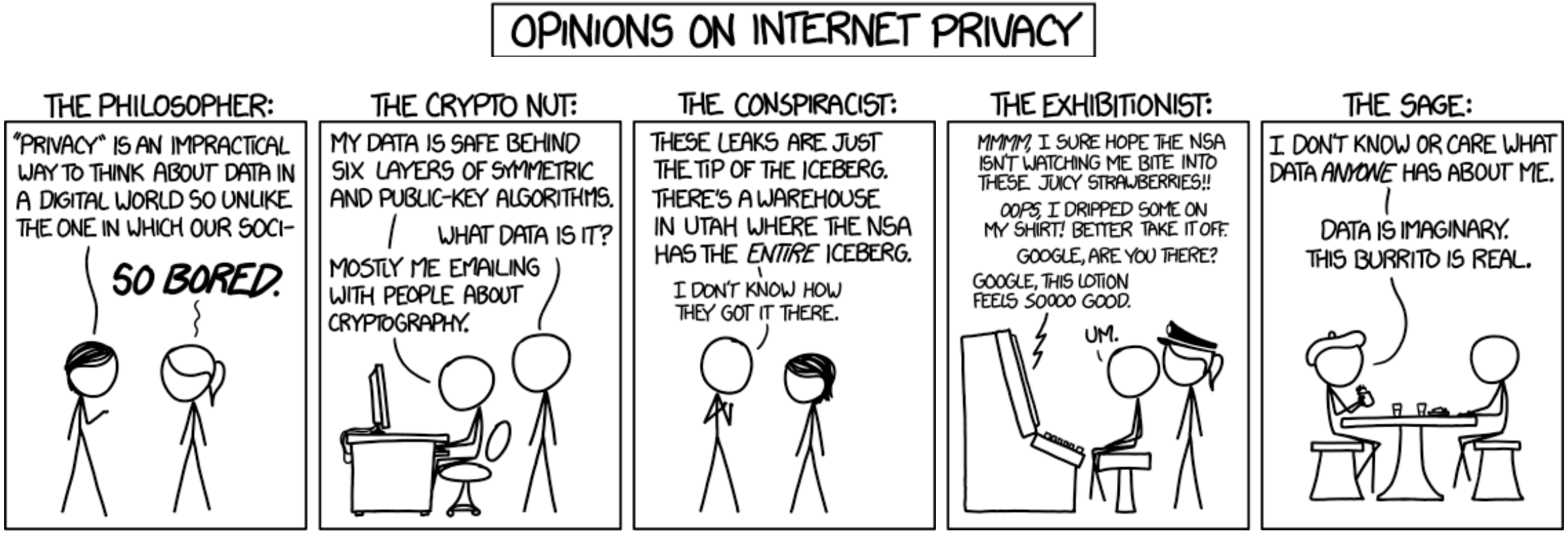
https://xkcd.com/538/

Challenge: Privacy

Important: We require adequate security measures to ensure privacy.

Challenge: Privacy ⊂ Security

Privacy Risk: The right to be let alone



Source: https://xkcd.com/1269 (Privacy Opinions)

Challenge: Privacy – OECD Recommendations

Recommendations for the protection of privacy and trans-border flows of personal data

Principles:

1) Collection limitation
2) Data quality
3) Purpose specification
4) Use limitation
5) Security safeguards
6) Openness
7) Individual participation (Right for information/confirmation)
8) Accountability

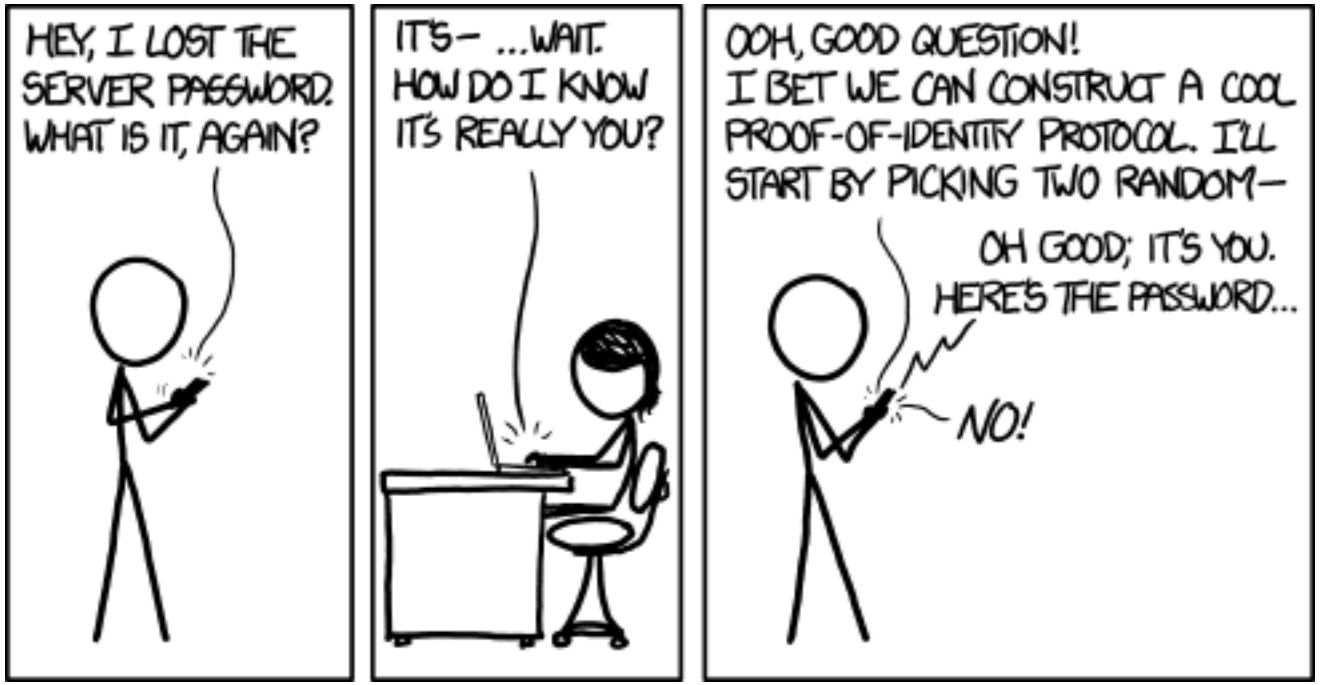
Protection by Design:

1) Data minimization
2) Pseudonymization (encryption, hashing, aggregation, masking)

Note: Are these challenges or solutions?

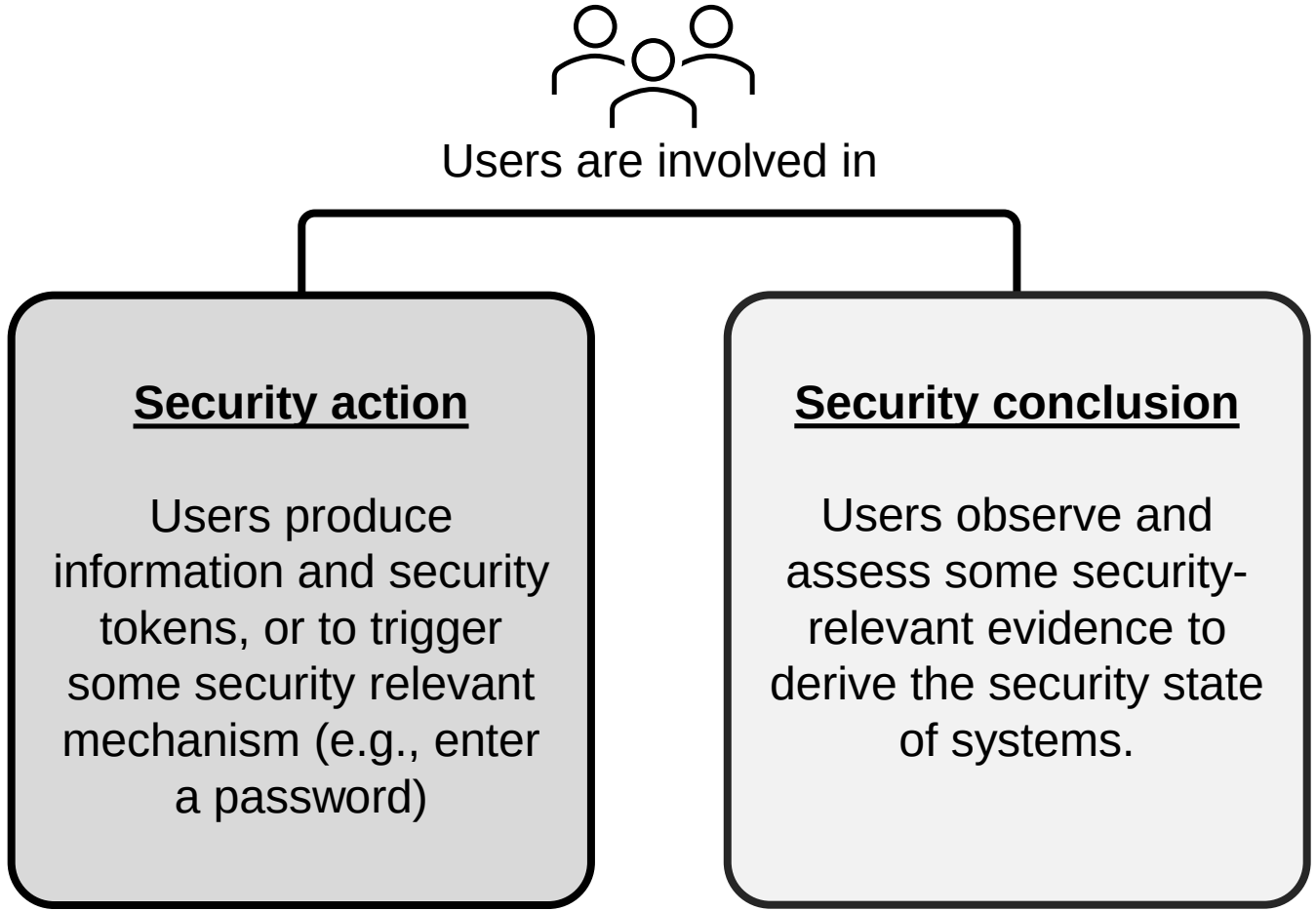
Challenge: Usability and Accessibility

The user should not have drawbacks from a weakly designed system.



Source: https://xkcd.com/1121 (Identity)

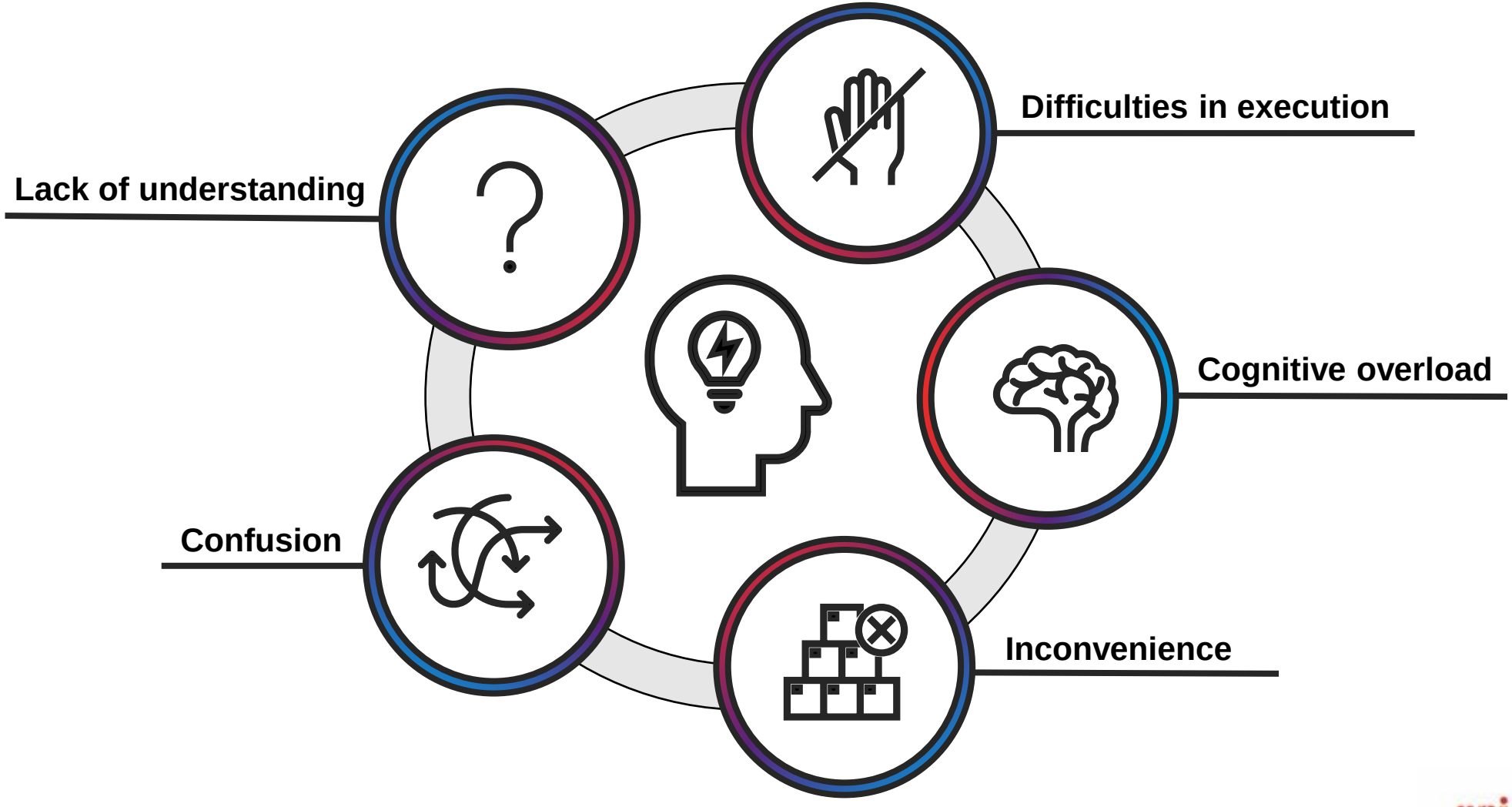
Challenge: Usability and Accessibility



Source: Jøsang et al. (2007) Usability and Privacy in Identity Management Architectures

Challenge: Usability and Accessibility

Warning: Users are struggling due to many reasons.



Challenge: Usability and Accessibility

Warning: Users are struggling due to many reasons

Usability principles

Principles for security actions

- User must understand required security actions
- User must have sufficient knowledge and the practical ability needed
- “The mental and physical load of a security action must be tolerable”
- “The mental and physical load of making repeated security actions for any practical number of transactions must be tolerable”

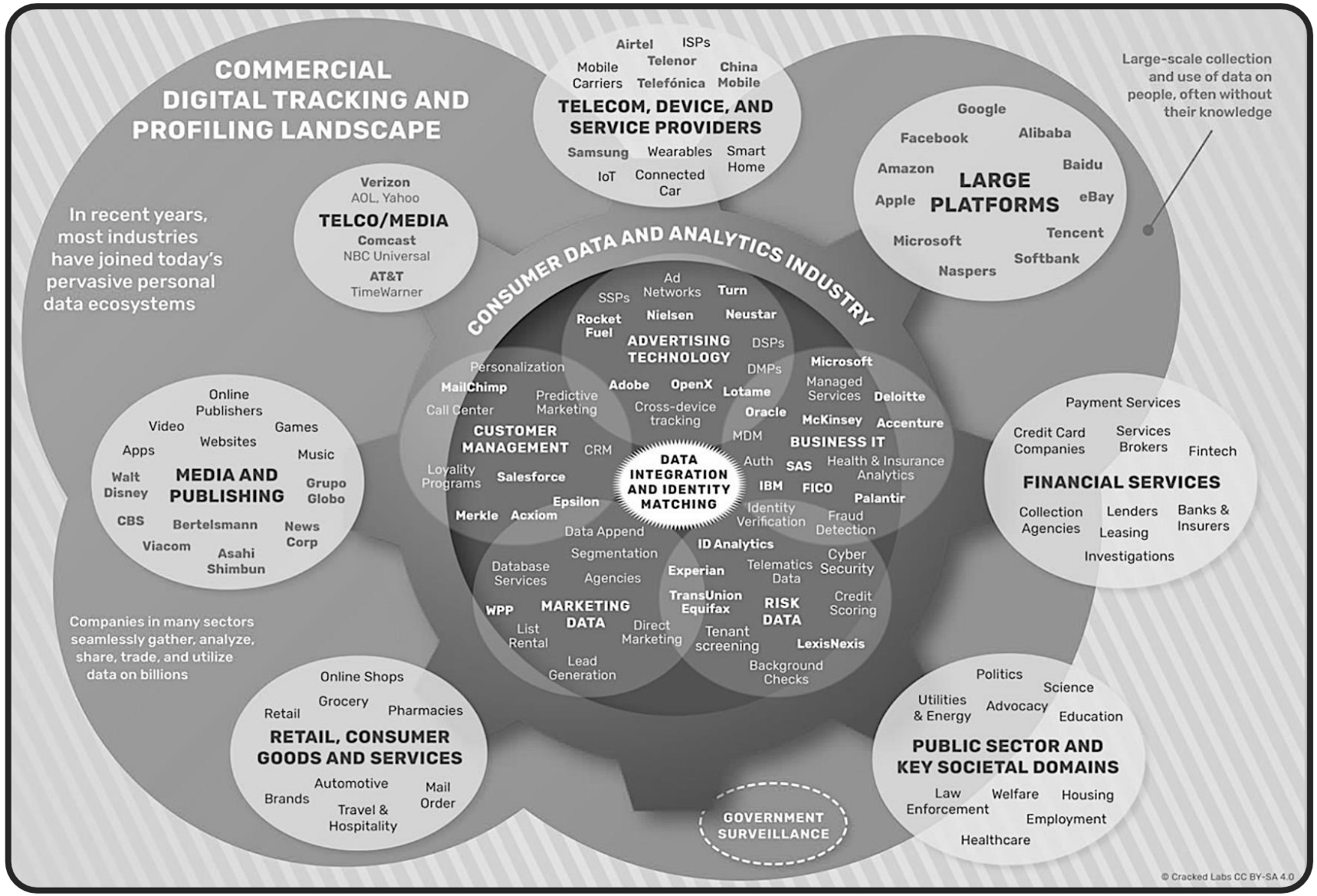
Principles for security conclusion

- User must understand the security conclusion.
- The system must provide the user with sufficient information for deriving the security conclusion.
- “The mental load of deriving the security conclusion must be tolerable”
- “The mental load of deriving security conclusions for any practical number of service access instances must be tolerable”

Source: Jøsang et al. (2007) Usability and Privacy in Identity Management Architectures

Challenge: Economics

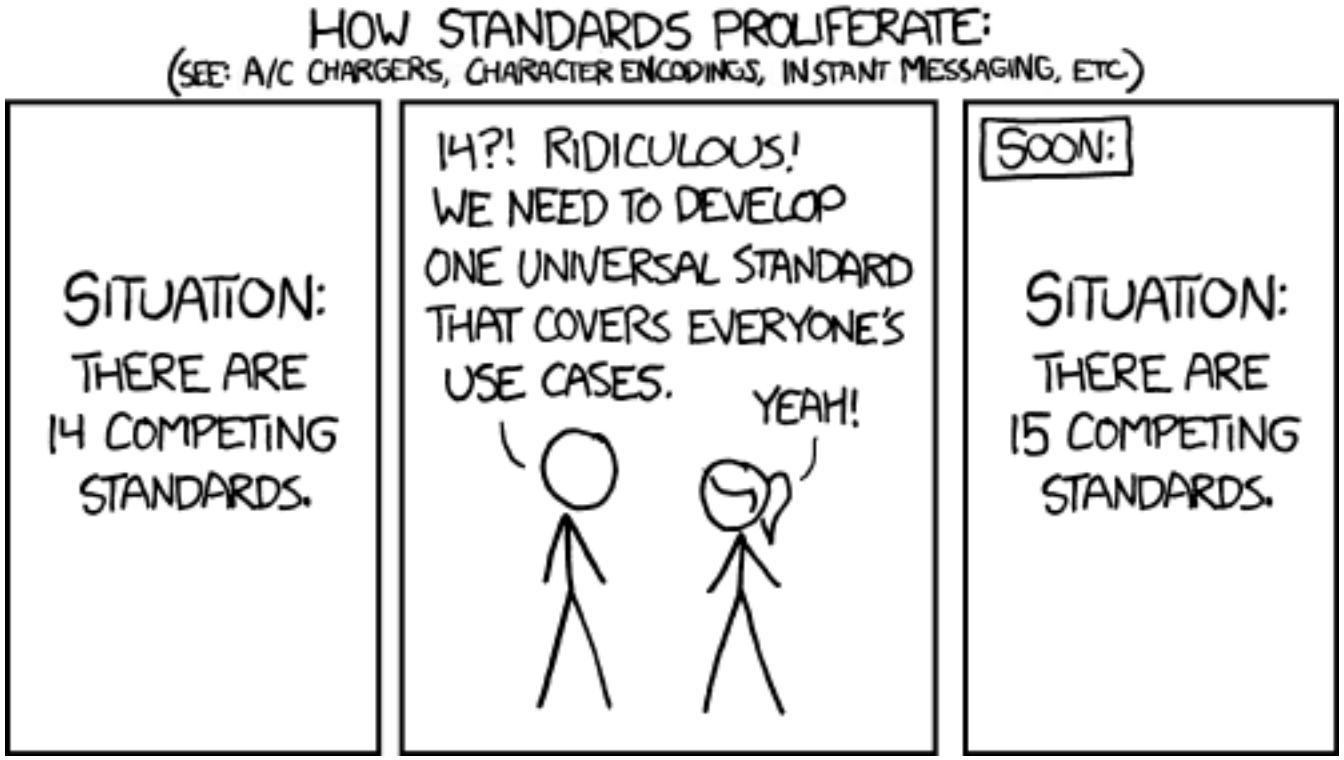
Big Data Market:
Mapping the commercial digital tracking and profiling landscape



Source: Christl et al. (2017) CORPORATE SURVEILLANCE IN EVERYDAY LIFE – How Companies Collect, Combine, Analyze, Trade, and Use Personal Data on Billions

Challenges: Regulation and Standards

Are solutions just new challenges?



Source: https://xkcd.com/927 (Standards)

Further reading

- Elisa Bertino and Kenji Takahashi, 2011
  Identity Management: Concepts, Technologies, and Systems
- Maryline Laurent and Samia Bouzefrane, 2015
  Digital Identity Management
- Yvonne Wilson and Abhishek Hingnikar, 2019
  Solving Identity Management in Modern Applications: Demystifying OAuth 2.0, OpenID Connect, and SAML
- Johannes Sedlmeir, Reilly Smethurst, Alexander Rieger, and Gilbert Fridgen. 2021
  Digital identities and verifiable credentials. *Business & Information Systems Engineering* 63, no. 5, pp. 603-613.


SNT
Questions?
UNIVERSITY OF LUXEMBOURG

Digital Identity: Identifiers

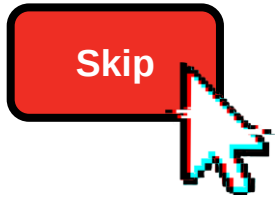
A somewhat extensive list of examples for strong and weak identifiers for digital entities*

- **User**:
  Social security; Username;
  User handle; Account ID;
  UID (*User Identifier*)
- **Device**:
  Serial numbers; Product numbers
- **Browser**:
  Cookie; Fingerprint; Tracking pixel; ETag (*Entity Tag*)
- **Network access**:
  IPv4; IPv6;
  MAC (*Media Access Control*)
- **Access point**:
  SSID (*Service Set IDentifier*);
  BSSID (*Basic Service Set IDentifier*);
  ESSID (*Extended Service Set IDentifier*)
- **Phone number**:
  IMEI (*International Mobile Equipment Identity*);
  IMSI (*International Mobile Subscriber Identity*);
  ICCID (Integrated Circuit Card Identification Number)
- **Advertisement**:
  IDFA (*Apple's Identifier For Advertising*);
  GAID (*Google Advertising ID*);
  MAID (Microsoft Advertising ID)
- **Remote endpoint**:
  URI (*Unified Resource Identifier*);
  URL (*Unified Resource Locator*)
- **Arbitrary entity**:
  UUID (*Universally Unique IDentifier*);
  GUID (*Globally Unique IDentifier aka. UUID*);
  UDID (*Unique Device IDentifier*); Barcode

***Note**: Strong identifiers are difficult or impossible to change and uniquely identify an entity of an extended period of time

Cryptographic Primitive Methods

TL;DR: No discrete math…



Hash Functions

Requirements	
Compression	**Input** of **arbitrary length** to **output** of **fixed length**
Preimage Resistance	Difficult to define x if y = h(x) is known
2nd Preimage Resistance	Difficult to define x' such that h(x') = h(x), if x is known
Collision Resistance	Difficult to find two x such that h(x') = h(x)
Speed	Based on **logical/boolean** functions

Components	
Compression Function	Function with two concatenated **fix inputs** [m] to **fix output** [n] with [m > n] ▸ Custom Function (e.g. MD5, SHA1/2/3) (**OR**) ▸ Block-Cipher e.g.: Davis-Meyer: $h(x,y)=E_y(x)\oplus y$; Miyaguchi-Preneel: $h(x,y)=E_x(y)\oplus x\oplus y$
Domain Extender	Modify function with **fix length input to** one with **arbitrary length input** e.g. **Merkle-Damgård Construction:** ▸ Input divided into [n] blocks [x_i] of same length as input for compression function ▸ $h_i = f(h_{i-1}, x_i)$; $h(x) = h_{n+1}$; h_i = internal state of f (!!) ▸ Append length of input to x_n -> Same ending. Different hash. ▸ **Length Extension Attack:** If H(key||m) state given, then new MAC of extended m easy.

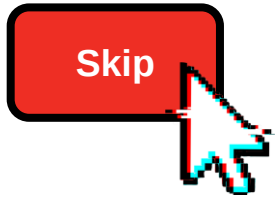
Attack Difficulty	n = Output length; Effort of 2^{80} is considered difficult
Preimage Resistance ①	Break **One-Way function:** $O(2^n)$ = Horrible Complexity
2nd Preimage Resistance ②	Weak **Collision Resistance:** $O(2^n)$ = Horrible Complexity
Collision Resistance ③	Strong **Collision Resistance:** $1{,}2*2^{n/2}$ = Much better than the others (**birthday paradox**)

https://preshing.com/20110504/hash-collision-probabilities/

Algorithms	MD5 (1991)	SHA-1 (1995)	SHA-2 (2001)	SHA-3 (2015)
Output	**128** Bit	**160** Bit	224/**256**/384/512 Bit	224/256/**384**/512 Bit
Function	64 rounds **512 Bit blocks** 128 Bit state 4 round functions	80 rounds **512 Bit blocks** 160 Bit state 4 round functions		3-Dimensional state absorbing phase squeezing phase 224-1600 Bit output
Structure	Merkle-Damgård	Merkle-Damgård	Merkle-Damgård	**Sponge**
Best Attack	2^{18} (2013) on ③	2^{61} (2011) on ③	Theoretical	
Security	**Broken**	**Broken**		
Other	Ron Rivest			NIST (Keccak)

Cryptographic Primitive Methods

TL;DR: Discrete math…



Asymmetric Cryptography: Rivest-Shamir-Adleman (RSA)

Text-Book Version		
Prime Factors	$\mathbf{p \neq q}$	random; large; similar length;
Modulus	$\mathbf{n} = p \cdot q$	
Euler's Phi	$\mathbf{\Phi} = (p-1)(q-1)$	$\Phi(n)$
Enc. Key	$\mathbf{e} \equiv d^{-1} \pmod{\Phi}$	$\gcd(\Phi, e) = 1;\ 1 < e < \Phi$; --> Usually: **3** or **65537** = $2^{16}+1$
Dec. Key	$\mathbf{d} \equiv e^{-1} \pmod{\Phi}$	$e^{-1} \pmod{\Phi} = e^{\Phi-1} \pmod{n}$
Secret	$\mathbf{x} = \{d, n\}$	$p, q, \Phi(n)$
Public	$\mathbf{y} = \{e, n\}$	
Encryption	$\mathbf{c} \equiv m^e \pmod{n}$	$m \in]1; n[$
Decryption	$\mathbf{m} \equiv c^d \pmod{n}$	

Theoretical Attacks		
Factorization Problem	If $n = \mathbf{p \cdot q}$ feasible $\mapsto e = d^{-1} \pmod{\Phi}$	$\lvert n\rvert < 512$ in 1h for 20$ on AWS
Calculate d from (e, n)	If $1 = \mathbf{d} \cdot e \pmod{\mathbf{\Phi}}$ feasible	Calc Φ and e equivalent to factoring n
Small encryption exponent	If $e = 3$, and $c_i \equiv m^3 \pmod{n_i}$ $(i = 1,2,3)$ $\mapsto x \equiv c_i \pmod{n_i};\ m = \sqrt[3]{x}$	Using Gauss Algo. & Chinese remainder **Fix:** m||salt and e = $2^{16}+1$
	If $\lvert m\rvert < n^{1/e} \mapsto m = c^{1/e}$	**Fix:** m||salt and e = $2^{16}+1$
Small decryption exponent	If d "small", Φ also small and $\lvert d\rvert \approx \lvert n\rvert \cdot \frac{1}{4}$ $\mapsto 1 \equiv \mathbf{d} \cdot e \pmod{\mathbf{\Phi}}$ efficient (see above)	**Fix:** $\lvert d\rvert \approx \lvert n\rvert$
Forward search attack	If $\lvert M\rvert$ small, and $\mathbf{C'} \equiv M^e \pmod{n}$ feasible $\mapsto c' = c$ (like a rainbow table)	**Fix:** m||salt
Adaptive chosen-c attack	If A decrypts all $c'^d = m'$ for M except $c = m^e$, $\mapsto c' \equiv cx^e \pmod{n}$ $\mapsto m \equiv m'x^{-1} \pmod{n}$	Homomorph: $m^e = m_1^e m_2^e = c_1 c_2 = c \pmod{n}$ **Fix:** Include well-known structure in the message to detect tampering
Common modulus attack	If $n_i = n$ $(i > 1)$, and Fac. n feasible $\mapsto$ Leak all d_i	**Alternative:** If $n_1 = n_2$, $m_1 = m_2$, and $e_1 \neq e_2$, then m can be derived.
Cycling attack	If $c^{e\hat{}k} \equiv c \pmod{n}$ $\mapsto c^{e\hat{}(k-1)} \equiv m \pmod{n}$	Difficult because most cycles are long Difficulty equivalent to factoring n
Message concealing	If $m^e \equiv m \pmod{n}$ $\mapsto$ m is unconcealed, e.g. $m \in \{0, 1, n-1\}$	$9 = [1+\gcd(e-1,p-1)][\ 1+\gcd(e-1,q-1)]$ 9 possible unconcealed m

Security in Practice	
Large modulus	Min. 1024; **More than 2048** Bit for long-term security
Strong prime	**Similar size:** $\lvert p\rvert \approx \lvert q\rvert$; **Diff not too small:** $p-q \neq 0 \wedge p \neq \sqrt{n}-q$; **Large prime factor** for: $p \pm 1 \wedge p-2$
Small encryption exponent	With k = |e| in Bits, square-and-multiply requires k squares and avg. k/2 multiplies If |e| small, and few 1's, then better performance Therefore: e = $\{3_d = 11_b,\ 65537_d = 1000000000000001_d\}$

Asymmetric Cryptography

TL;DR: More discrete math…

Skip

Use Case: DLP-based Digital Signature

General	**ElGamal** (1985)	**Schnorr**© (1989/91)	**DSA** (1991/94)	**ECDSA** (1991/94)
Integrity Authenticity	Minimum 1024 Bit Prime	Patented and Licensed	Hybrid of the ElGamal and Schnorr Sig. Schema Efficient Version of ElGamal Schema ▸ Verification uses 2x Exp. instead of 3x ▸ Exponents of max length 160Bit ▸ Calculates in Sub-Group q instead of p ▸ (P, Q){(1024, 160), (2048, 256), (3072, 256)}	
Key Material				
Group	**p, g** ▸ q = p−1 (**OR**) ▸ q = Large Prime Factor ▸ $g^q = 1 \% p$ ▸ g = Primitive Root	**p, q, g** ▸ $g^q \equiv 1 \% p$ ▸ g = Primitive Root	**p, q, g** ▸ $q \in]2^{Q-1}; 2^Q[$ ▸ $p \in]2^{P-1}; 2^P[$ ▸ $p-1 \equiv 0 \% q$ ▸ $a \in (\mathbb{Z}/p\mathbb{Z})^*; g \neq 1$ ▸ $g \equiv a^{(p-1)/q} \bmod p$ ▸ g = Primitive Root	**P, n** ▸ $nP \equiv \theta$
Private	$x \in]1; q[$	x	$x \in [1; q[$	x
Public	$y \equiv g^x \% p$	$y \equiv g^{-x} \% p$	$y \equiv g^x \% p$	$Y = xP$
Random	$k \in]1; q[$ ▸ Random per Sig.! ▸ gcd(k, q) = 1		$k \in [1; p-1[$ ▸ Random per Sig.! ▸ gcd(k, q) = 1	k ▸ Random per Sig.! ▸ gcd(k, q) = 1
Exchange				
Public Key	(p, g, y)	(p, q, g, y)	(p, q, g, y)	(P, n, Y)
Signature	(m, r, s)	(m, r, s)	(m, r, s)	(m, r, s)
Signature				
Hash	h = H(m)	h = H(m||r)	h = H(m)	h = H(m)
	$r \equiv g^k \% p$	$r \equiv g^k \% p$	$r \equiv g^k \% p \% q$	$r \equiv (kP)_x$
	$s \equiv k^{-1}(e-xr) \% p-1$	$s \equiv k+xh \% q$	$s \equiv k^{-1}(h+xr) \% q$	$s \equiv k^{-1}(h+xr) \% q$
Verification				
Input	$0 < r < p-1$		$0 < r < q-1$	
Preparation	$a = H(m) \% p$	h = H(m||r) $H(G^s y^h \| \| m) \stackrel{?}{=} h \% p$	$w \equiv s^{-1} \% q$ $b \equiv w \cdot H(m) \% q$ $c \equiv wr \% q$	$w \equiv s^{-1} \% n$
Signature	$g^a \stackrel{?}{=} y^r r^s \% p$	$r \stackrel{?}{=} g^s y^e \% p$	$r \stackrel{?}{=} g^b y^c \% p \% q$	$(ewP+rwY)_x \stackrel{?}{=} r \% n$

Technologies: Security Assertion Markup Language (SAML)

Security assertion markup language is a set of **technical specifications** for implementing **federated IdM**

- **Identity federation** establishes a logical link between two different identities of a subject, each of which is managed by a different service provider
- It enables subjects to **create, update, and delete links** and allows only limited entities, on a "**need to know**" basis, to access the information about the links.
- SAML 2.0 specifies a **mechanism** for identity management in two different aspects:
  - **data format** (or "assertions")
    for expressing facts about identities, and
  - **procedures** for transmitting the assertions between identity providers and relying parties.
- One of the most well-known applications of SAML is a browser-based **single sign-on**.

SSO Advantages

- Solved the problem of authenticating users
- Convenient for users
  (only one password to remember)

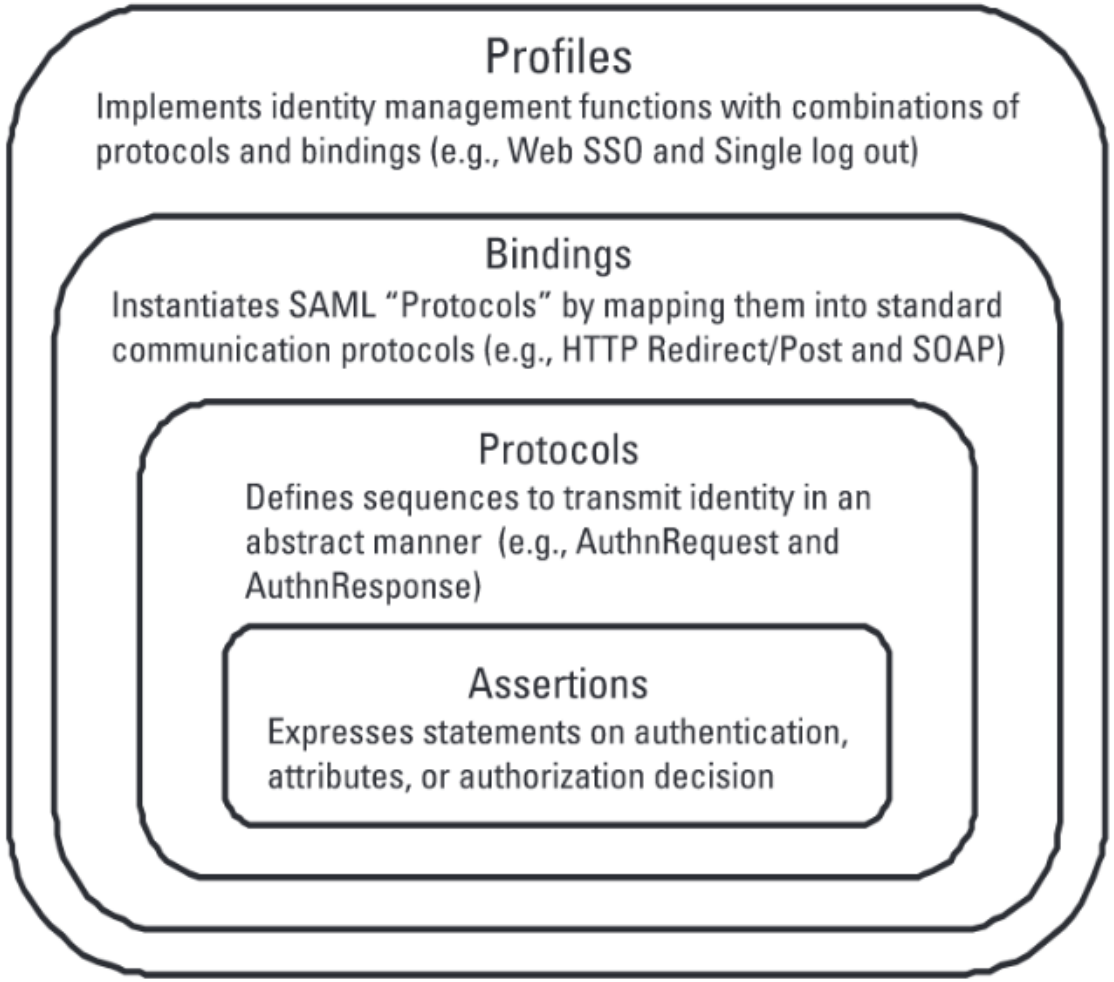
SSO Disadvantages

- Complex configuration
- No viable business model

Source: http://saml.xml.org/saml-specifications

Technologies: Security Assertion Markup Language (SAML)

SAML specification structure and example

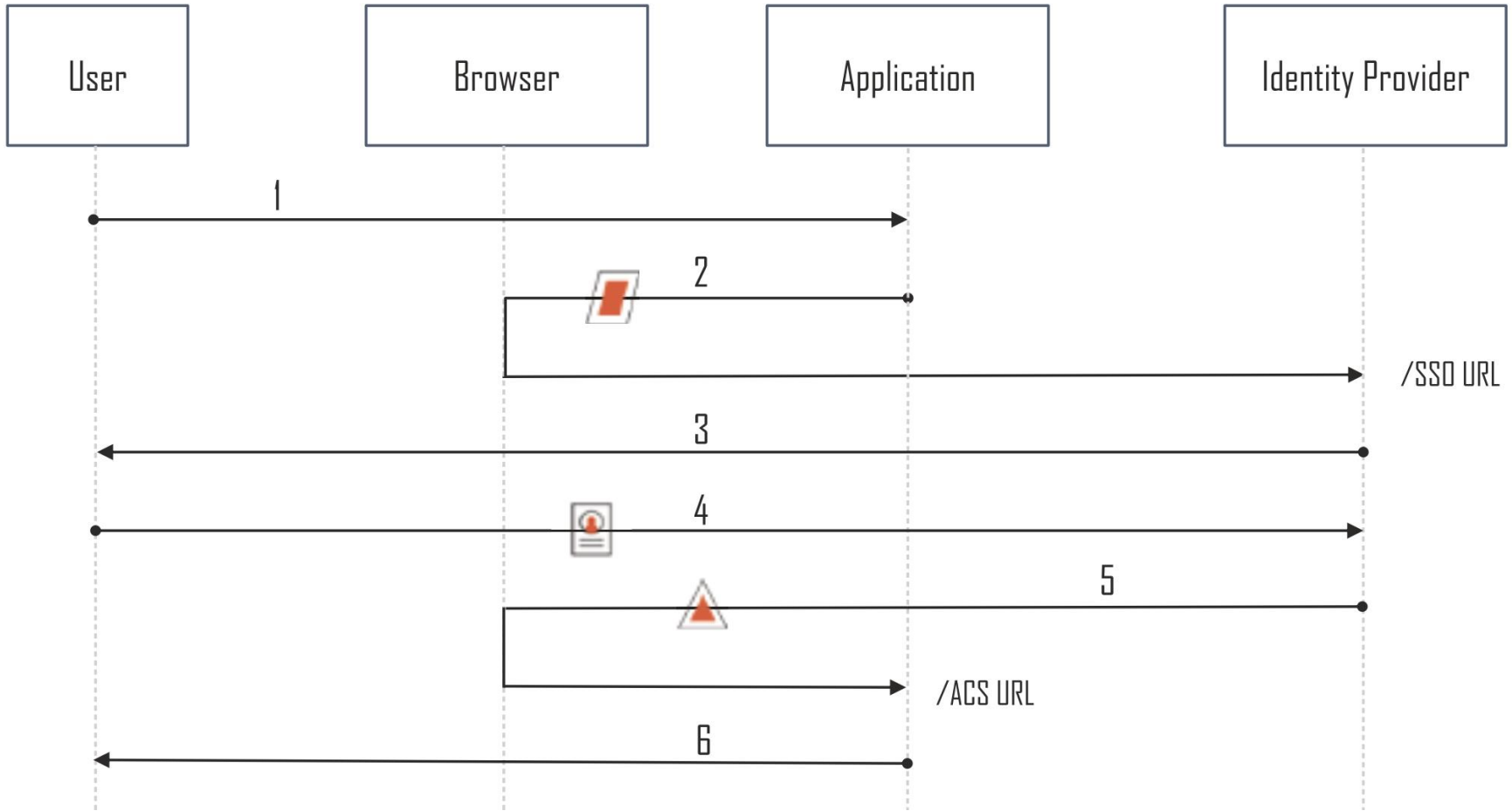


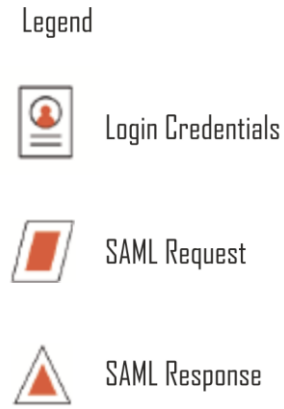

```
<saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    Version="2.0" IssueInstant="2021-06-10T12:00:00Z">
 <saml:Issuer Format="urn:oasis:names:SAML:2.0:nameid-format:entity">
  https://example.idp.com
 </saml:Issuer>
 <saml:Subject>
  <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:
      transient">
   Ax9GO0f08FFakjQwQgiTI3x7e811HpL
  </saml:NameID>
 </saml:Subject>
 <saml:Conditions NotBefore="2021-06-10T12:00:00Z" NotOnOrAfter="
     2021-06-10T12:03:00Z">
  <saml2:AudienceRestriction>
   <saml2:Audience>https://example.sp.com</saml2:Audience>
  </saml2:AudienceRestriction>
 </saml:Conditions>
   <saml:AuthnStatement AuthnInstant="2021-06-10T12:00:00Z" SessionIndex
       ="35452006787179">
  <saml:AuthnContext>
   <ac:AuthnContextClassRef>
    urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
   </ac:AuthnContextClassRef>
   <ac:AuthnContextDeclaration>
    <ac:Identification>
     <ac:PhysicalVerification/>
    <ac:Identification/>
    <ac:TechnicalProtection>
     <ac:PrivateKeyProtection>
      <ac:KeyStrage "medium"="smartcard"/>
     <ac:PrivateKeyProtection/>
    <ac:TechnicalProtection/>
   </ac:AuthnContextDeclaration>
  </saml:AuthnContext>
 </saml:AuthnStatement>
</saml:Assertion>
```

Source: Bertino and Takahashi (2011) Identity Management: Concepts, Technologies, and Systems

Technologies: Single Sign-On

One of the most well-known applications of SAML is a browser-based **single sign-on.**





Source: Wilson and Hingnikar (2019) Solving Identity Management in Modern Applications: Demystifying OAuth 2.0, OpenID Connect, and SAML 2.0