DIGITAL IDENTITY

smtp: 550 recipient address rejected

Summary

- Authentication
- Electronic ID
- HTTPS and TLS
- Federated Identity
- Verifiable Credentials
- Decentralized Identifiers

Disclaimer and acknowledgements

- Authentication
- Electronic ID
- HTTPS and TLS
- Federated Identity
- Verifiable Credentials
- Decentralized Identifiers

ipzs.it/

finsec-project.eu/

cherrychain.it/

st.fbk.eu/

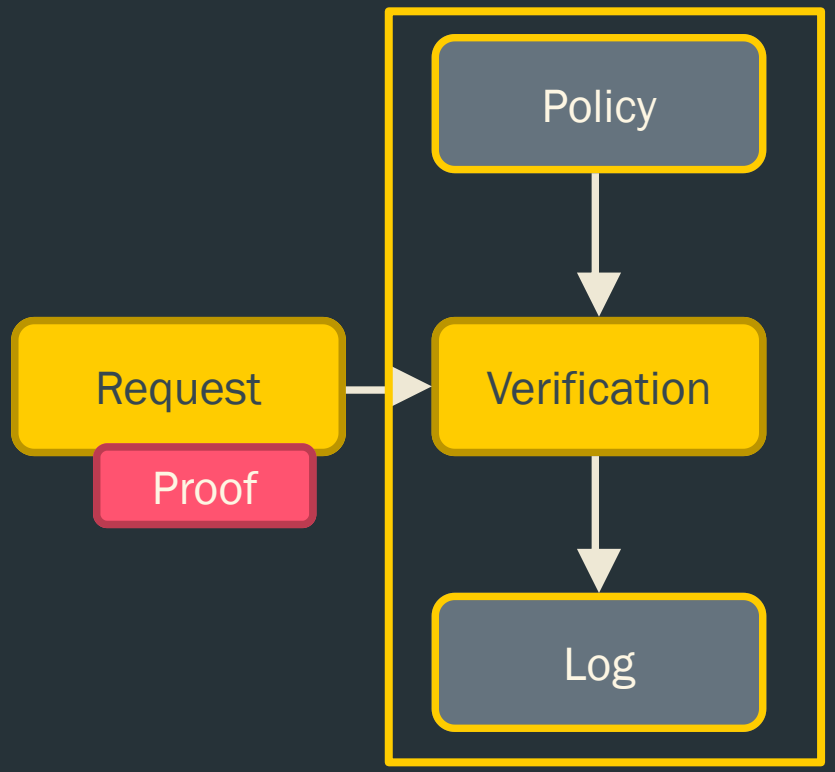
AUTHENTICATION

Authentication

- **Verifying the identity** of a user, process, or object
- Prerequisite to **grant access** to resources

Authentication

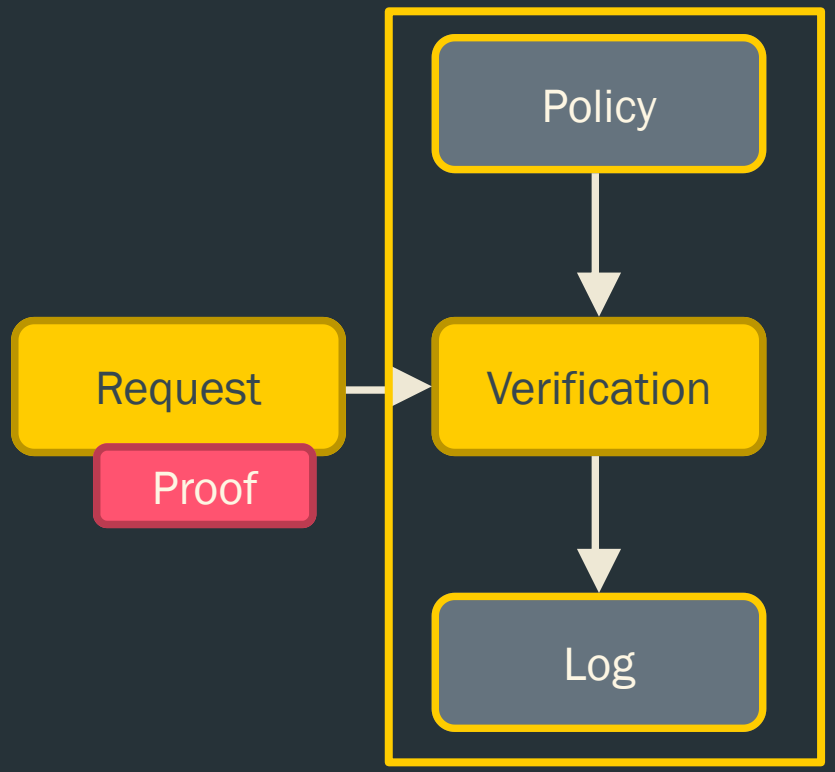
- **Verifying the identity** of a user, process, or object
- Prerequisite to **grant access** to resources



Authentication

- **Proof** generated from authentication factors

Requires enrollment!



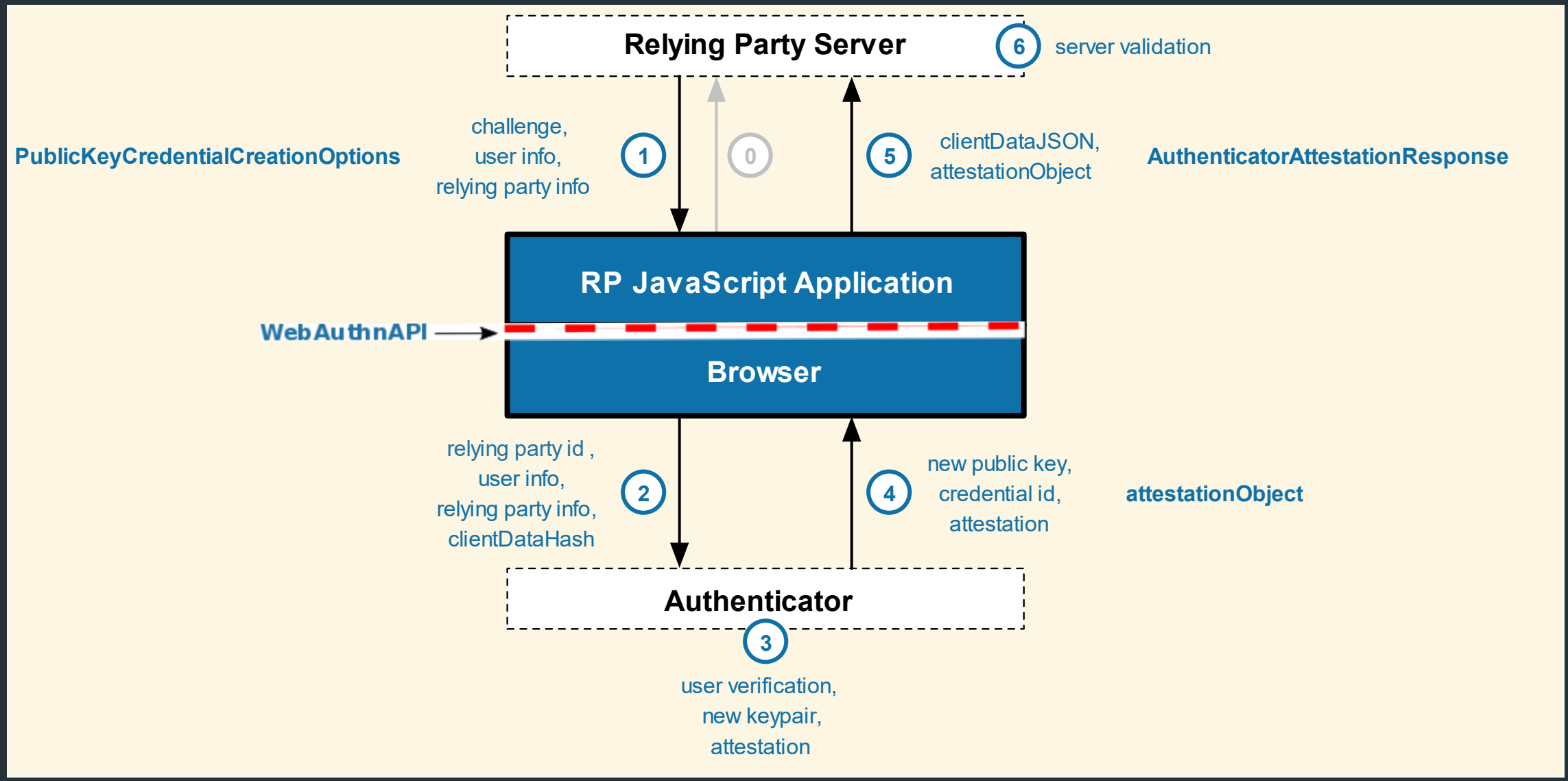
Glossary [SP 800-63-3]

- **Identity**
  - *An attribute or **set of attributes** that uniquely describe a subject within a given context.*
- **Credential**
  - *An object or data structure that **binds an identity** - via an identifier and (optionally) additional attributes - **to** at least one **authenticator** possessed and controlled by a subscriber.*
- **Authenticator**
  - *Something the claimant possesses and controls used to **authenticate** the claimant's identity.*

Glossary [SP 800-63-3]

- Credential Service Provider (CSP)
  - *A trusted entity that issues or registers subscriber authenticators and issues electronic credentials to subscribers.*
- Enrollment
  - *Process through which an applicant applies to become a subscriber of a CSP and the CSP validates the applicant's identity.*
- Authentication Protocol
  - *A defined sequence of messages between a claimant and a verifier that demonstrates that the claimant has possession and control of one or more valid authenticators to establish their identity, and, optionally, demonstrates that the claimant is communicating with the intended verifier.*

[WebAuthN]

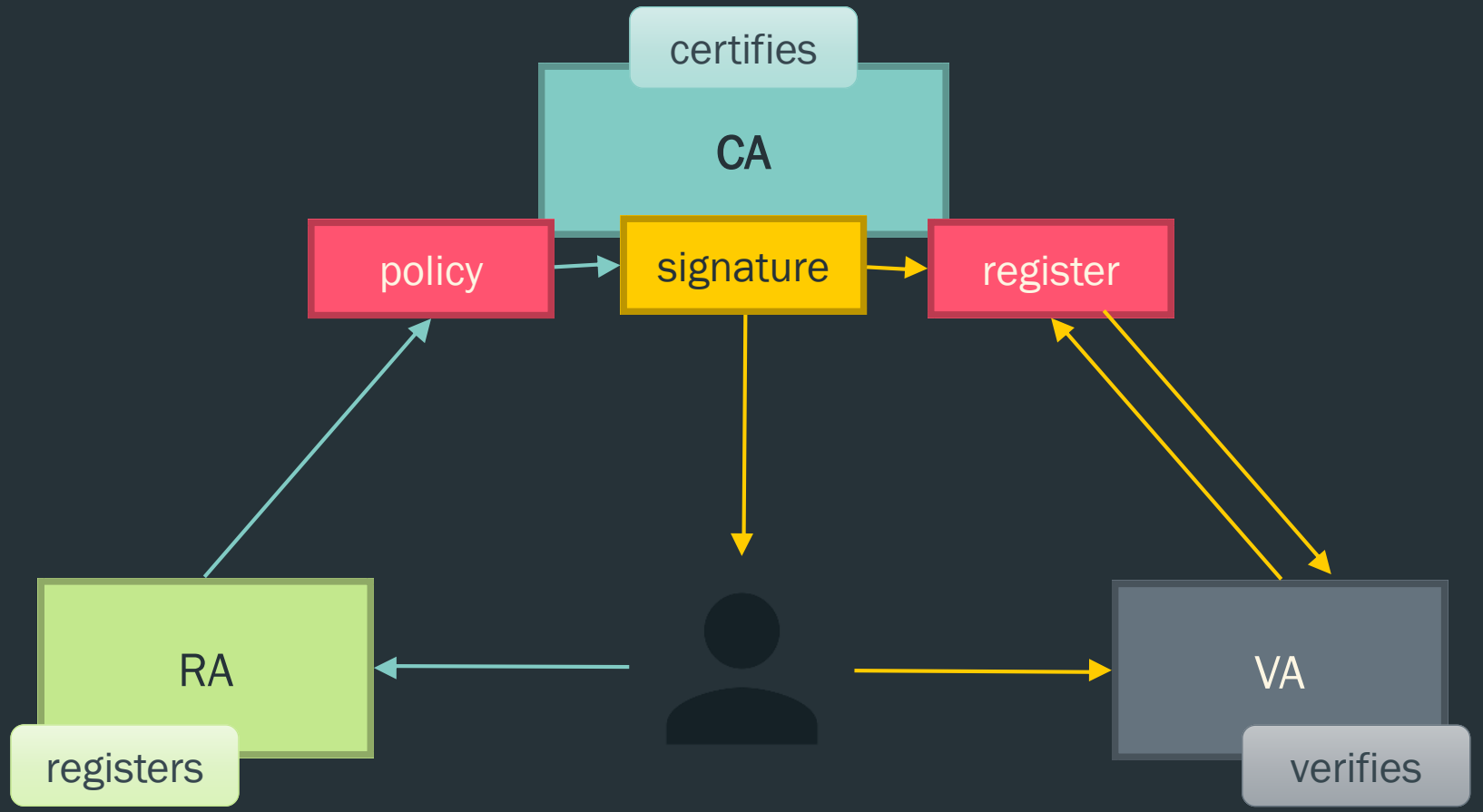

Relying Party Server
6
server validation
PublicKeyCredentialCreationOptions
challenge,
user info,
relying party info
1
0
5
clientDataJSON,
attestationObject
AuthenticatorAttestationResponse
RP JavaScript Application
WebAuthnAPI
Browser
relying party id ,
user info,
relying party info,
clientDataHash
2
4
new public key,
credential id,
attestation
attestationObject
Authenticator
3
user verification,
new keypair,
attestation

ELECTRONIC ID

CIE 3.0 smart authentication processes

	Physical identity	Digital identity
Mobile	IDEA	CieID
Desktop		Middleware Windows 10, mac, linux
CIE 3.0	ICAO MRTD Identifying data Biometrics – fingerprint, photo	ECC-IAS Identifying data Unique identifier Authentication key Authentication certificate

Public Key Infrastructure



X.509 [RFC 5280] - CIE 3.0

Subject

Common Name : <CF>/<DocID>
Serial Number : IDCIT-<SubjectID>
Given Name : <>
Surname : <>
Country : IT

Issuer

Common Name :
Issuing sub CA for the Italian Electronic Identity Card - SUBCA1
Organizational Unit :
Direz. Centr. per i Servizi Demografici - CNSD
Organization :
Ministero dell'Interno
Country : IT

Issued Certificate

Version : 3
Serial Number : [8 bytes]
Not Valid Before : yyyy-mm-dd
Not Valid After : yyyy-mm-dd
Certificate Fingerprints : SHA1 : [20 bytes] MD5 : [16 bytes]

Public Key Info

Key Algorithm : RSA
Key Parameters : 05 00
Key Size : 2048
Key SHA1 Fingerprint : [20 bytes]
Public Key : [256 bytes]

Extensions

Signature

Signature Algorithm :
1.2.840.113549.1.1.11 [sha256WithRSAEncryption]
Signature Parameters : 05 00
Signature : [512 bytes]

Extensions

[Authority Information Access] Identifier : 1.3.6.1.5.5.7.1.1
Access Method=On-line Certificate Status Protocol (1.3.6.1.5.5.7.48.1)
Alternative Name : URL=https://ocsp.cie.interno.gov.it/

[CRL Distribution Point] Identifier : 2.5.29.31
[1] CRL Distribution Point
Distribution Point Name :
Full Name : URL=http://ldap.cie.interno.gov.it/ciesubca1.crl

Extensions

[Certificate Policies] Identifier : 2.5.29.32
[1] Certificate Policy :
Policy Identifier=1.3.76.47.4
[1,1] Policy Qualifier Info :
Policy Qualifier Id=User Notice
Qualifier : Notice Text=X.509 authentication certificate issued by the Italian Ministry of Interior for the Electronic Identity Card
[1,2] Policy Qualifier Info :
Policy Qualifier Id=CPS
Qualifier : http://www.cartaidentita.interno.gov.it/policy/cittadini_cps.pdf

Certificate Policy and Certification Practice Statement
Public Key Infrastructure for the Italian Electronic Identity Card "CIE"

Key Usage
Usages : Digital signature
Extended Key Usage
Allowed Purposes : Client Authentication

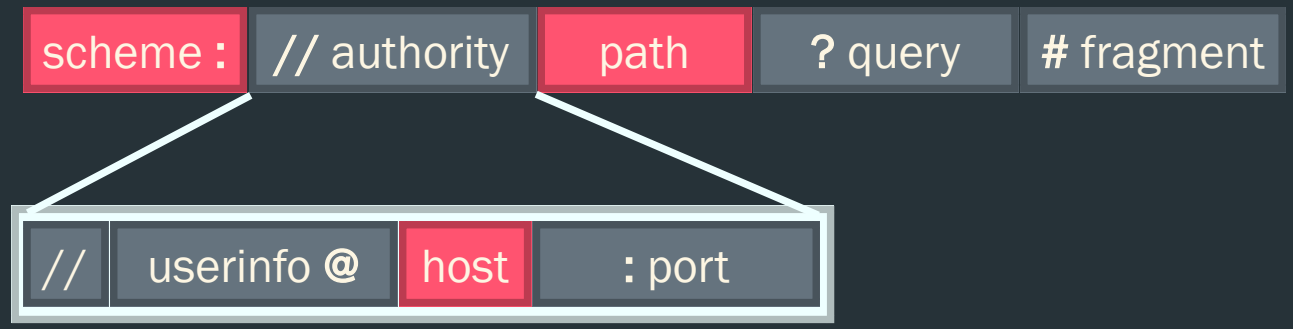
```
KeyUsage ::= BIT STRING {
        digitalSignature        (0),
        nonRepudiation          (1) / contentCommitment
        keyEncipherment         (2),
        dataEncipherment        (3),
        keyAgreement            (4),
        keyCertSign             (5),
        cRLSign                 (6),
        encipherOnly            (7),
        decipherOnly            (8) }
```

Recap

- Digital identity consists of claims
- Claims are entered in signed certificates, with a public key
- Authentication is based on proof of possession of the corresponding private key
- Trust is based on the PKI behind the signature, starting with root CAs

HTTPS AND TRANSPORT LAYER SECURITY

https: Uniform Resource Identifier



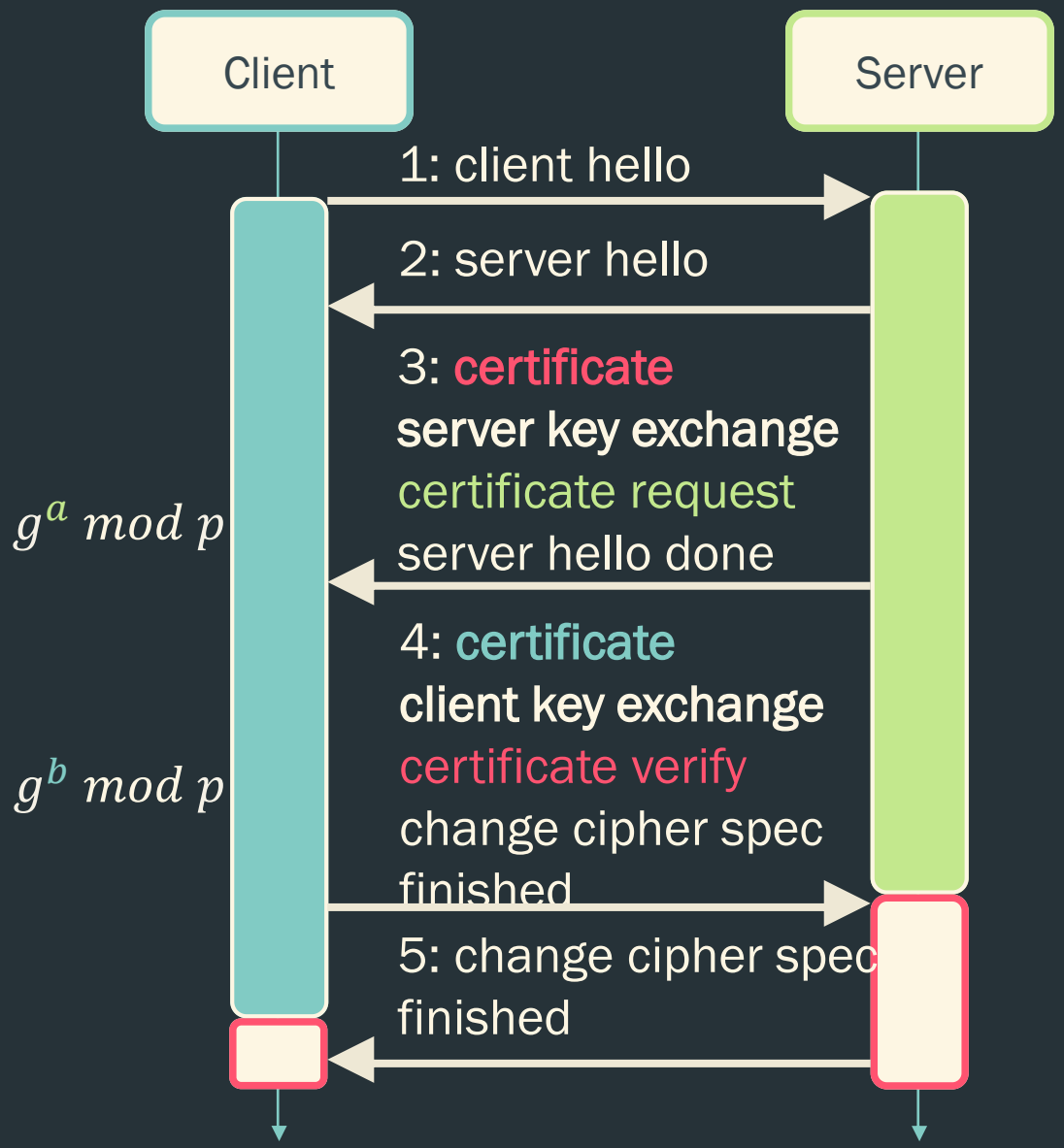
- Examples of URI:
  - *https://en.wikipedia.org:443/wiki/Uniform_Resource_Identifier#Examples*
  - *mailto:st@fbk.eu*
  - *file:///usr/local/bin/*
  - *tel:+1-816-555-1212*
  - *doi:10.1000/1*

Transport Layer Security

1. Authentication
2. **Key** exchange
3. Symmetric cryptography

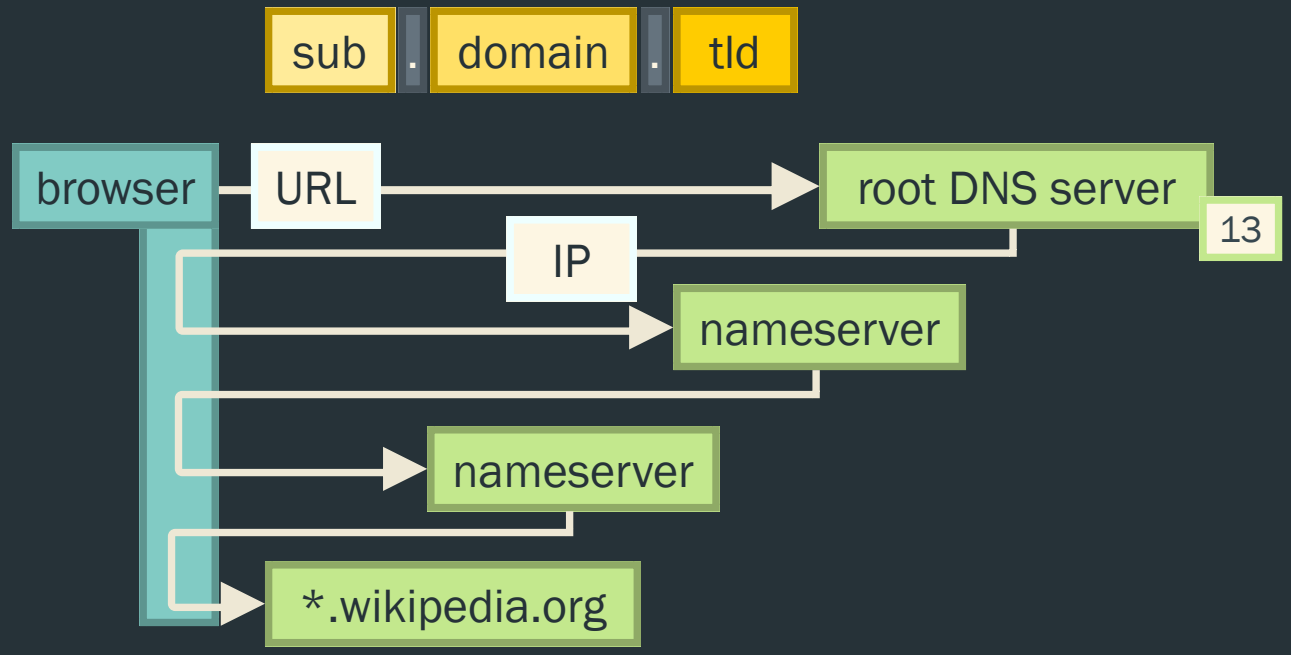
TLS handshake

1. version, cipher suite
2. protocol, suite, session ID
3. server cert.,
   premaster secret,
   client cert. request
4. client cert.,
   premaster secret,
   client cert. verification
5. OK, h(handshake)



DNS resolution

- But wait – how did we find “en.wikipedia.org” in the first place?
- We used a Domain Name Service resolver.



Subject

CN = *.wikipedia.org
O = Wikimedia Foundation, Inc.
L = San Francisco
S = California
C = US

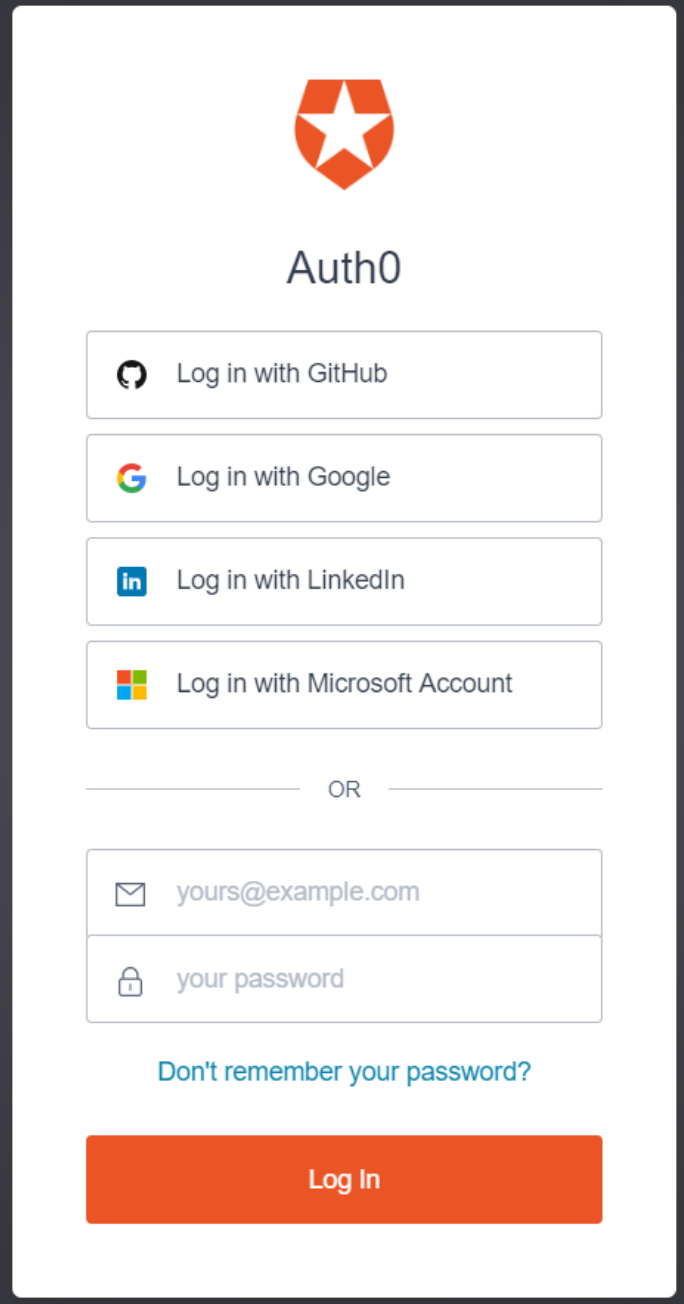
Recap

- Resources are identified / located by a URL / URI
- URL is resolved through a service
- The DNS resolution service is nominally free for users, but secondary servers are run by ISPs. Registering a (sub)domain also incurs fees
  - *OpenDNS: 140billion daily DNS requests. dns.google.com/: 400bn in 2014*
- Root CA certificates are embedded in the browser

FEDERATED IDENTITY

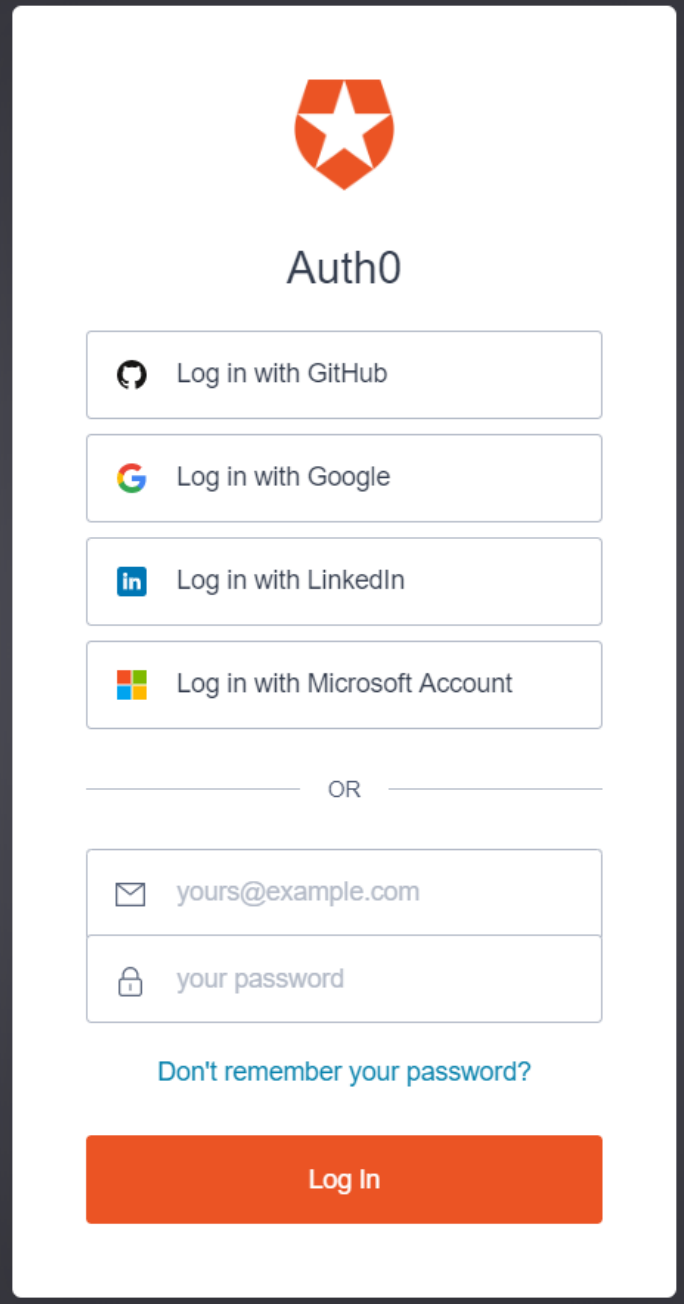
Federated sign-on

- Third-party apps verify user identity and get claims
- Trust between authentication provider and service provider
- Outsources much of the burden of enrollment and authentication to ID providers
- Main authentication protocols:
  - *OIDC – OpenID Connect [JWT]*
  - *SAML – Security Assertion Markup Language [XML]*



OIDC summary

- **Subject** starts log in to App with **Provider**
- App sends Authorization Request to **Provider**
- **Provider** authenticates **Subject**
- **Provider** lists all the permissions that App wants, e.g. email, and asks you if Subject authorizes
- **Provider** sends Access Token, and (if requested) ID Token, to App
- App can retrieve user information from the ID Token or use the Access Token to invoke a **Provider** API.



JWT [RFC 7519, jwt.io]

- JSON Web Token (JWT):
  - *Representation of claims to be transferred between two parties*

Issuer	Subject	Audience	Not before	Issued at	JWT ID

  - *payload of a JSON Web Signature (JWS)*
  - *plaintext of a JSON Web Encryption (JWE)*
- JWS: header.payload.signature [RFC 7515], [RFC 7797]
- Note: Audience is app-specific, Subject and ID are issuer-specific

OIDC ID JWT standard claims

Subject	Profile	Zoneinfo
Name	picture	Locale
Given_name	Website	Phone_number
Family_name	Email	Phone_number_verified
Middle_name	Email_verified	Address
nickname	Gender	Updated_at
Preferred_username	Birthdate	

https://openid.net/specs/openid-connect-core-1_0.html#StandardClaims

Recap

- A few dominant services act as Credential Service Providers
  - *Recall: credentials bind identity (attributes) to authenticators*
- Requires federation to ensure consent and disclosure to intended service
- Claim subjects are tied to an identifier that is CSP-specific
- Claim issuing and verification is based on the CSP

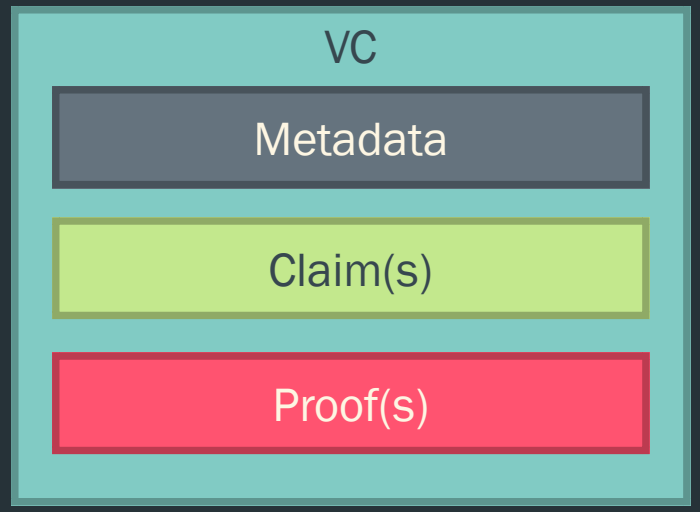
Motivation for alternatives

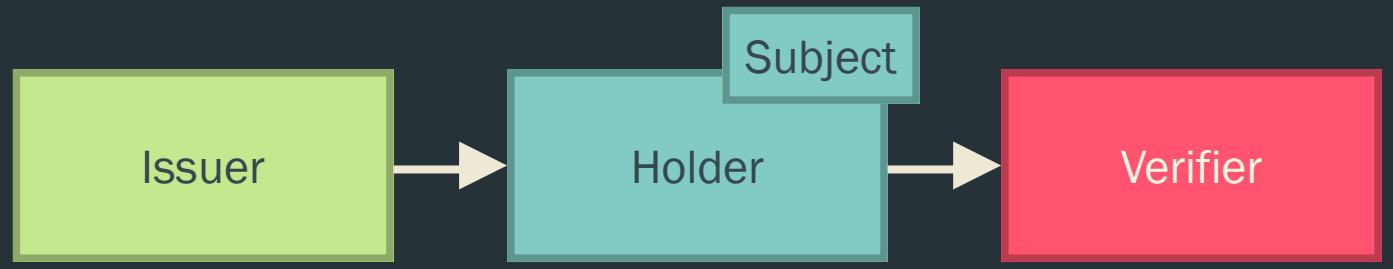
- [CSP] can issue claims about [identity] and [app] can verify them
- How does [app] make claims about [identity] and have them verified by other [app]s?
- Is there a single way to refer to [identity] without relying on many [CSP]?
- Contrived use case: education qualifications.
  - *Suppose universities could offer claims of alumni qualifications instantly verifiable online by any authorized employer. What if claims were tied to your.first.email@orange.net?*

VERIFIABLE CREDENTIALS

Verifiable Credentials (VC) [W3C-VC]

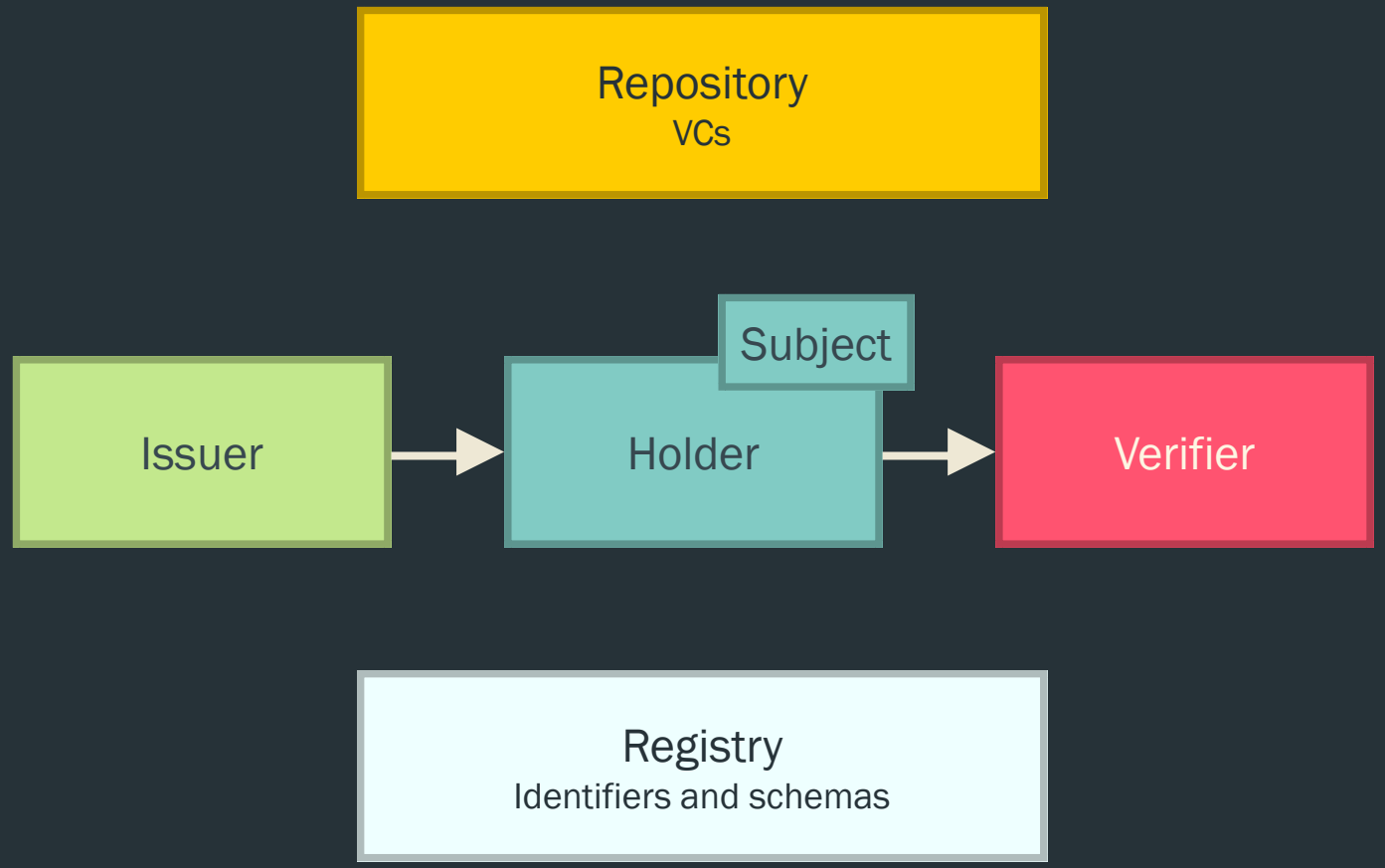
- **Verifiable credentials:** statements made by an issuer in a tamper-evident manner.
- Acting as issuer, holder, or verifier requires neither registration nor approval by any authority, as the trust involved is bilateral between parties. No federation required.





VC logical view

- **Repository**: a program, such as a storage **vault** or personal verifiable credential **wallet**, that stores and protects access to holders' verifiable credentials.
- Verifiable data **Registry: s**ystem mediating creation and verification of **identifiers**, **keys**, VC schemas, **revocation registries**, issuer public keys, etc



VC Trust Model

- **Verifier** trusts **Issuer** to issue the credential that it received.
- All entities trust the verifiable data **Registry** to be tamper-evident and to be a correct record of which data is controlled by which entities.
- **Holder** and **Verifier** trust **Issuer** to issue true (not false) credentials about the **Subject**, and to **revoke** them quickly when appropriate.
- **Holder** trusts the **Repository** to store credentials securely, to not release them to anyone other than the holder, and to not corrupt or lose them while they are in its care.
- This trust model differentiates itself from other trust models by ensuring the:
  - ***Issuer*** *and the* ***Verifier*** *do* ***not*** *need to trust the* ***Repository***
  - ***Issuer*** *does* ***not*** *need to know or trust the* ***Verifier****.*

VC implementations

- JSON Web Tokens [RFC 7519] secured using JSON Web Signatures [RFC 7515]
- Linked Data Signatures [https://w3c-dvcg.github.io/ld-signatures/]
- Camenisch-Lysyanskaya Zero-Knowledge Proofs [CL]

VC data model [W3C-VC]


```
{  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "id": "http://example.edu/credentials/1872",
  "type": ["VerifiableCredential", "AlumniCredential"],
  "issuer": "https://example.edu/issuers/565049",
  "issuanceDate": "2010-01-01T19:73:24Z",
  "credentialSubject": {
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    "alumniOf": {
      "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
      "name": [{
        "value": "Example University",
        "lang": "en"
      }, {
        "value": "Exemple d'Université",
        "lang": "fr"
      }] } },
```

```
"proof": {
    "type": "RsaSignature2018",
    "created": "2017-06-18T21:19:10Z",
    "proofPurpose": "assertionMethod",
    "verificationMethod":
"https://example.edu/issuers/keys/1",
    "jws":
"eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQi
OlsiYjY0II19
..
TCYt5XsITJX1CxPCT8yAV-
TVkIEq_PbChOMqsLfRoPsnsgw5WEuts01mq-
pQy7UJiN5mgRxD-
WUcX16dUEMGlv50aqzpqh4Qktb3rk-
BuQy72IFLOqV0G_zS245-
kronKb78cPN25DGlcTwLtjPAYuNzVBAh4vGHSrQyH
UdBBPM"  }}
```

DECENTRALIZED IDENTIFIERS

Decentralized Identifier (DID) [W3C-DID]

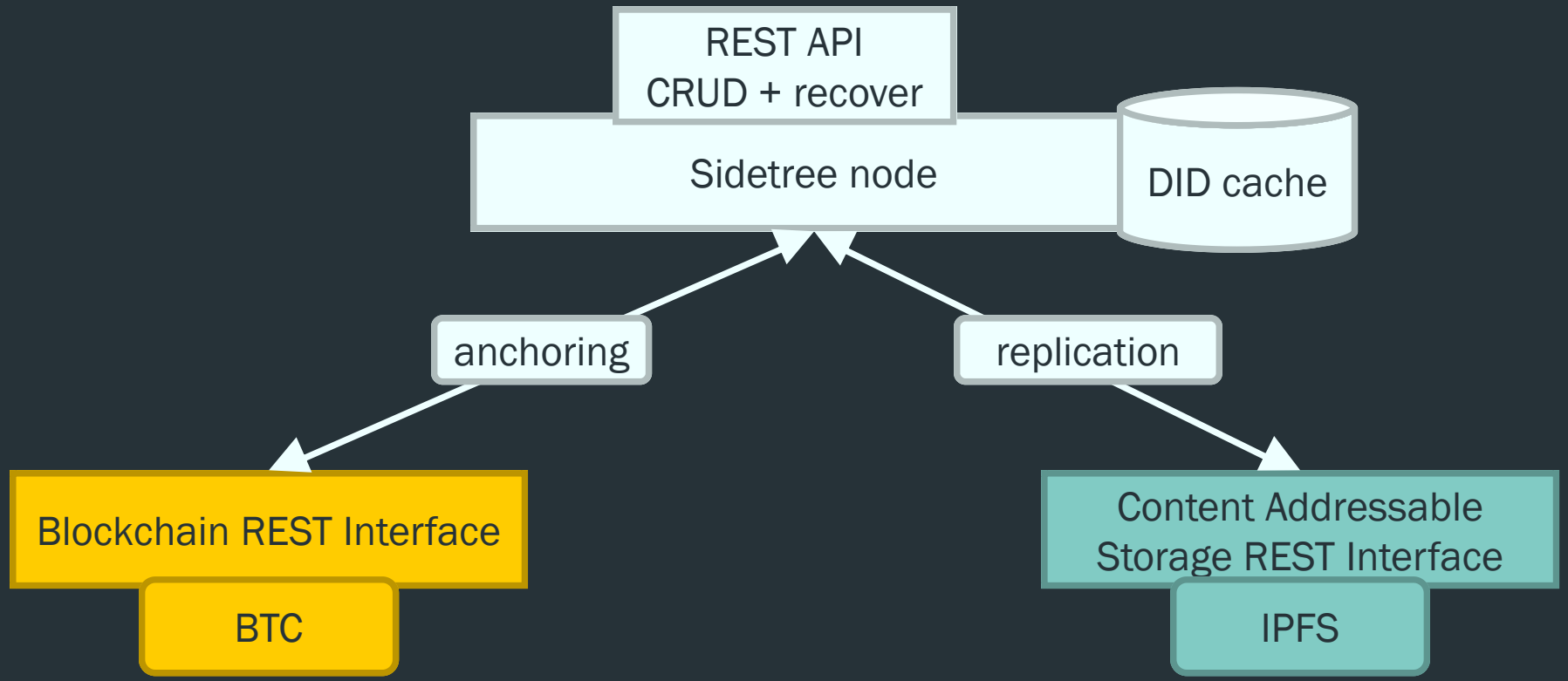
- Uniform Resource Identifier:
  - *URI = **scheme :** [//authority] **path** [?query][#fragment]*
- Decentralized Identifier
  - *DID = "did" : method-name : method-specific-id*
  - `did:sov:base58(16-byte uuid)`
  - `did:ethr:[ethr-network:]ethr-address`
- See https://w3c-ccg.github.io/did-method-registry
- DID **method**: definition of implementation, including precise methods to resolve and deactivate DID, write and update DID documents
- DID **document**: data describing DID subject, including authentication mechanisms
- DID **resolver**: A system capable of retrieving a DID document for a given DID.

DID document


```
{  "@context": "https://www.w3.org/ns/did/v1",
  "id": "did:example:123456789abcdefghi",
  "authentication": [{
    "id": "did:example:123456789abcdefghi#keys-1",
    "type": "RsaVerificationKey2018",
    "controller": "did:example:123456789abcdefghi",
    "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC
KEY-----\r\n"
  }],
  "service": [{
    "id":"did:example:123456789abcdefghi#vcs",
    "type": "VerifiableCredentialService",
    "serviceEndpoint": "https://example.com/vc/"
  }]}
```

ION – Identity Overlay Network

- [ION] is a public and permissionless instantiation of sidetree [SdT] on bitcoin:



Sovrin - `did:sov:`

- Public and permissioned service based on Indy https://github.com/hyperledger/indy-node
  - *[RBFT] consensus https://github.com/hyperledger/indy-plenum*
  - *[BLS] signatures for consensus (aggregation)*
  - *[CL] signatures for credentials*
  - *Claim: anonymous credentials to prove predicates about selected attributes*
    - Credential schema on ledger contains attributes (integers) $\{m\_j\}$
    - Verifier asks Holder for proof that some m_k $\{<, =, >\}$ t
    - Holder presents unlinkable proof
    - Use case: proof of legal age

indy - issues

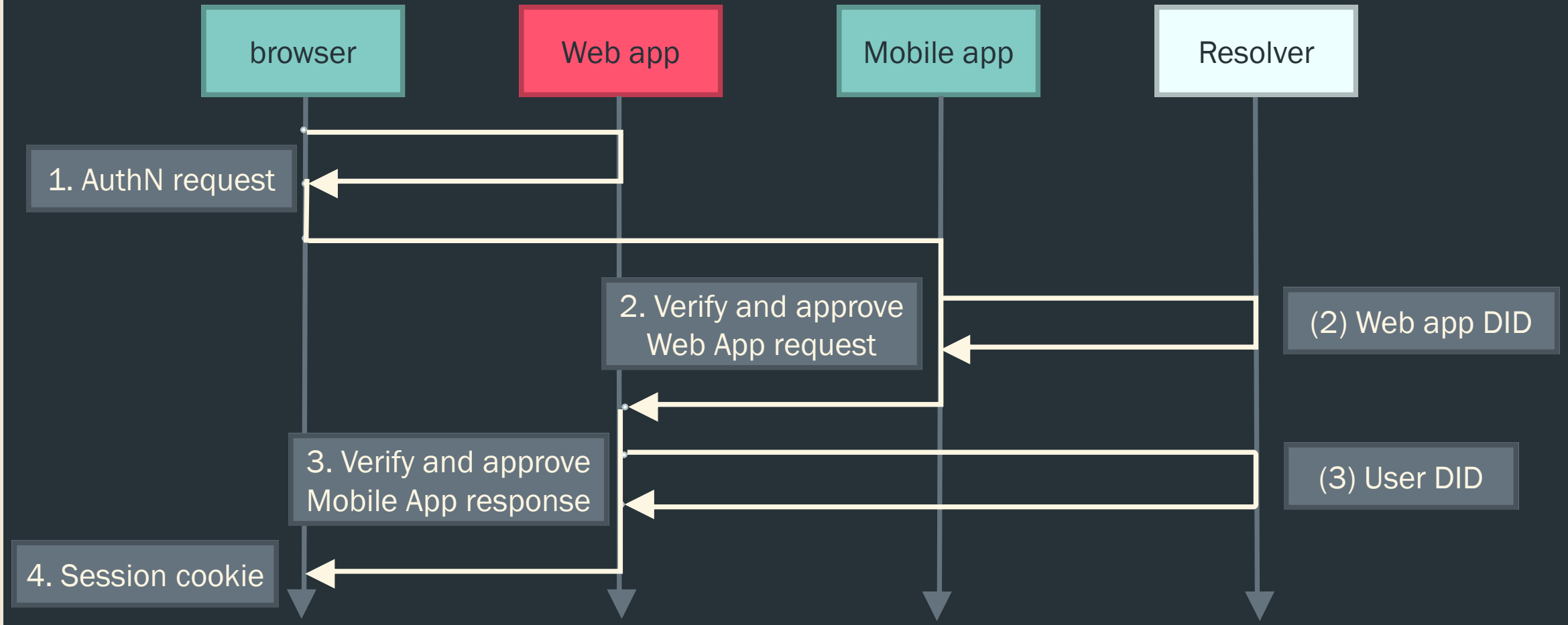
- Interoperability of credentials
- Immutable schemas
- Permissioning – users cannot create own DID without endorser, endorser can deactivate DID (no updates)
- Pricing, e.g. Sov:

DID Write	$10
Schema	$50
Credential Definition	$25
Revocation Registry	$20
Revocation Update	$0.10

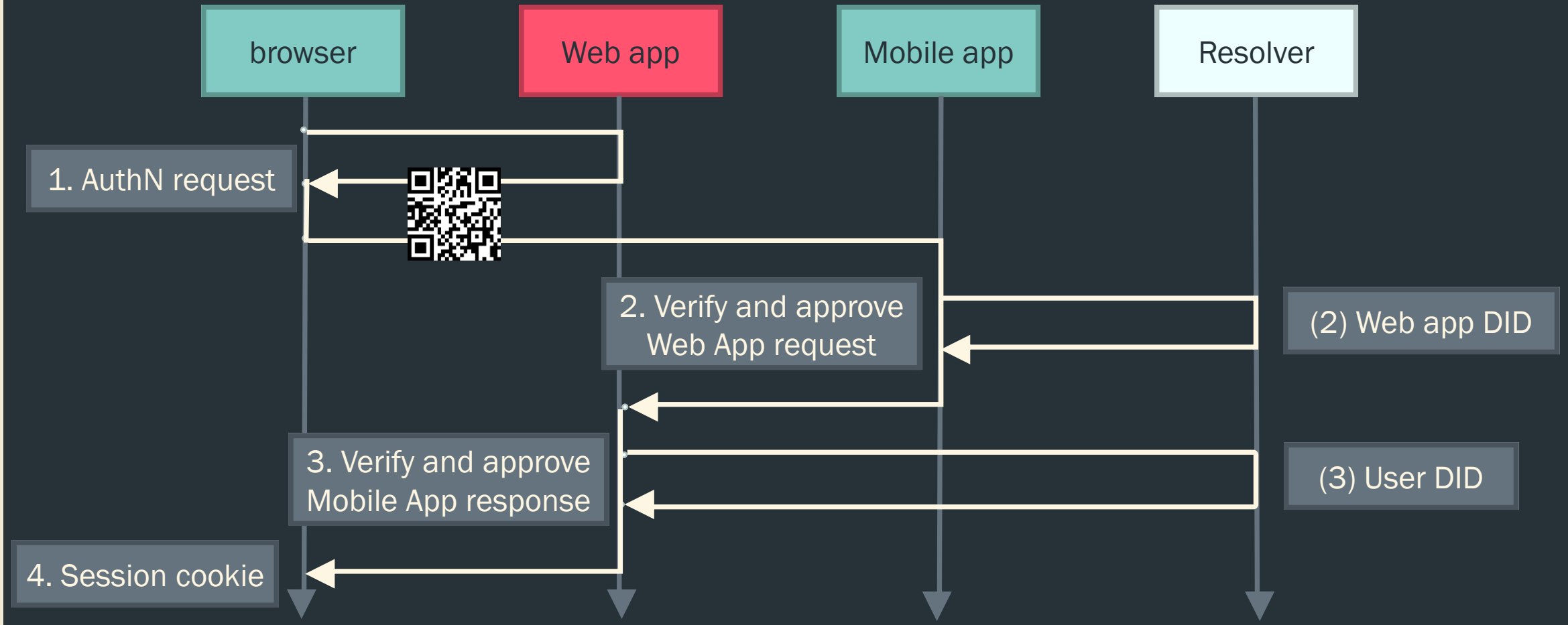
uPort

- `did:ethr:[ethr-network:]ethr-address`
- Any Ethereum address is automatically a valid DID and owner
- Ethereum Registry Contract ERC 1056 deployed to networks: registry for key and attribute management
- ethr-did-resolver to CRUD DID
- Unpredictable and significant transaction costs: chain is used as a registry of keys but verification is off-chain, if possible

[DID-auth] - OIDC



[DID-auth] - OIDC



DID is a WIP

- Authentication
  - *No established protocol yet*
- Recovery
  - *No authority can issue new authenticators for DID if lost. Key rotation, delegation control, and recovery are critical (compare with gvt-issued eID).*
  - *Shamir secret sharing?*
- Trust
  - *Can we tie DID to eID effectively while respecting privacy?*
- Governance
  - *Should DID creation be permissioned?*
  - *Should there be identity proofing on creation?*
  - *Should there be security vetting of client software trusted to facilitate DID creation?*

References

- [BLS] "Short Signatures from the Weil Pairing". doi: 10.1007/s00145-004-0314-9. url: https://crypto.stanford.edu/~dabo/pubs/papers/BLSmultisig.html
- [CL] Camenisch, Lysyanskaya: "A Signature Scheme with Efficient Protocols". doi: 10.1007/3-540-36413-7_20. url: https://groups.csail.mit.edu/cis/pubs/lysyanskaya/cl02b.pdf.
- [DID-auth] "Application Sign-In Protocol with OpenID Connect Self-Issued ID Tokens and DID Signatures". https://github.com/decentralized-identity/did-auth-jose/blob/master/docs/OIDCAuthentication.md
- [ION] Identity Overlay Network https://github.com/decentralized-identity/ion

References

- [RBFT] Aublin, Ben Mokhtar, Quéma: “Redundant Byzantine Fault Tolerance”. doi: 10.1109/ICDCS.2013.53.
- [RFC 5280] IETF RFC: “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”. https://tools.ietf.org/html/rfc5280
- [RFC 7515] IETF RFC: “JSON Web Signature (JWS)”. https://tools.ietf.org/html/rfc7515
- [RFC 7519] IETF RFC: “JSON Web Token (JWT)”. https://tools.ietf.org/html/rfc7519
- [RFC 7797] IETF RFC: “JSON Web Signature (JWS) Unencoded Payload Option”. https://tools.ietf.org/html/rfc7797
- [RFC 8555] IETF RFC: “Automatic Certificate Management Environment (ACME)”. https://tools.ietf.org/html/rfc8555

References

- [SdT] Sidetree https://github.com/decentralized-identity/sidetree
- [SP 800-63-3] NIST “Digital Identity Guidelines”. doi: 10.6028/NIST.SP.800-63-3. url: https://pages.nist.gov/800-63-3/
- [W3C-DID] W3C Working Draft: “Decentralized Identifiers (DIDs) v1.0”. https://www.w3.org/TR/did-core/
- [W3C-VC] W3C Recommendation: “Verifiable Credentials Data Model 1.0”. https://www.w3.org/TR/vc-data-model/
- [WebAuthN] W3C recommendation: https://www.w3.org/TR/webauthn/, demo source code: https://github.com/duo-labs/webauthn.io