Introduction to DID Auth

Markus Sabadello

Danube Tech, Decentralized Identity Foundation, Sovrin Foundation, W3C VCWG, W3C CCG, OASIS XDI TC

In a self-sovereign world, how can I prove that "I am me" ?

11th July 2018, markus@danubetech.com

SSIMeetup objectives

1. Empower global SSI communities
2. Open to everyone interested in SSI
3. All content is shared with CC BY SA

Alex Preukschat @SSIMeetup @AlexPreukschat
Coordinating Node SSIMeetup.org

"Who am I?"

"Who are you?"

SIMEETUP Self-Sovereign Identity SSIMeetup.org

"I am me!"

Introduction to DID Auth

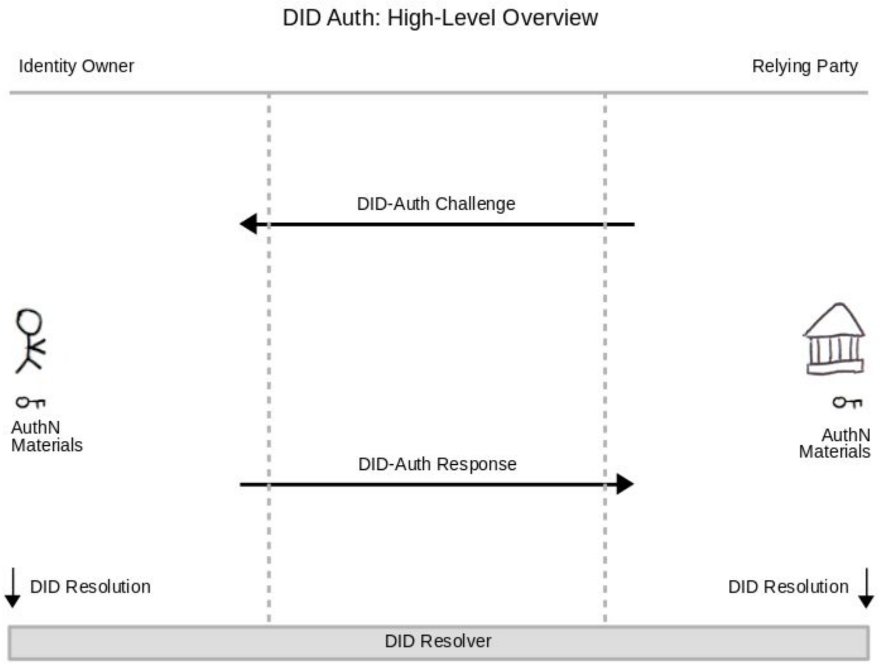
- DIDs = Decentralized Identifiers
- DID Auth = A concept, with different implementation ideas
- 2018 RWoT paper: "Introduction to DID Auth"
  (Markus Sabadello, Kyle Den Hartog, Christian Lundkvist, Cedric Franz, Alberto Elias, Andrew Hughes, John Jordan, Dmitri Zagidulin)
-
- Core idea: **Proving control of a DID**

Introduction to DID Auth

- DID Document contains metadata for authenticating a DID
- Example:

```
{
        "@context": "https://w3id.org/did/v1",
        "id": "did:example:123456789abcdefghi",
        "authentication": [{
                "type": "RsaSignatureAuthentication2018",
                "publicKey": "did:example:123456789abcdefghi#keys-1"
        }, {
                "type": "Ed25519SignatureAuthentication2018",
                "publicKey": "did:example:123456789abcdefghi#keys-2"
        }],
        "publicKey": [{
                "id": "did:example:123456789abcdefghi#keys-1",
                "type": "RsaVerificationKey2018",
                "owner": "did:example:123456789abcdefghi",
                "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"
        }, {
                "id": "did:example:123456789abcdefghi#keys-2",
                "type": "Ed25519VerificationKey2018",
                "owner": "did:example:123456789abcdefghi",
                "publicKeyBase58": "H3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV"
        }],
        "service": {
                "type": "DidAuthService",
                "serviceEndpoint": "https://auth.example.com/did:example:123456789abcdefg"
        }
}
```

Introduction to DID Auth



Challenges, Responses, Transports

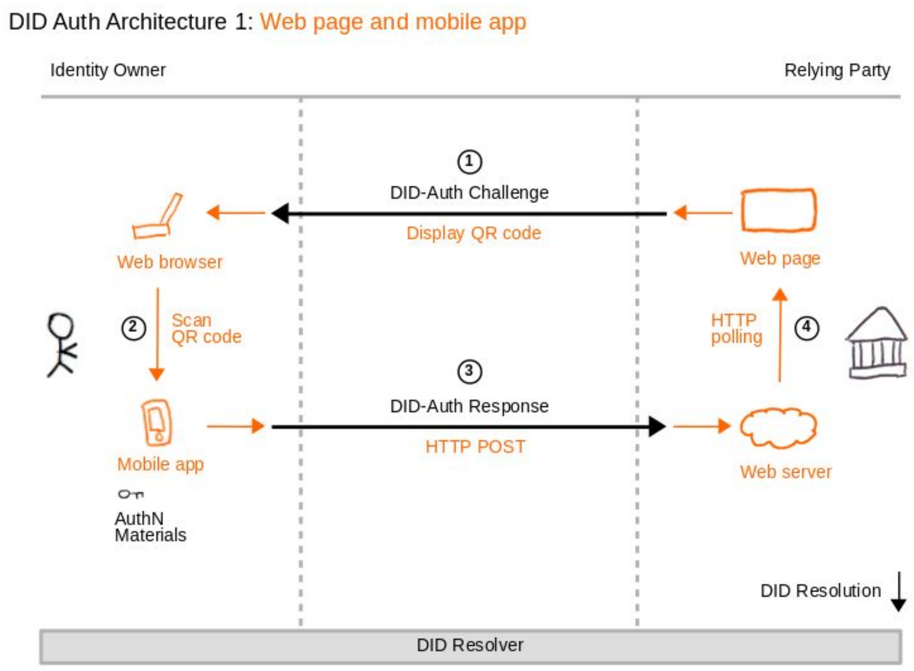
- Challenge–response cycle in which an identity owner proves to a relying party that they are in control of a DID.

- **Challenge:**
  - Identity owner’s DID may or not be known.
  - May or may not contain proof of control of a DID of the relying party.

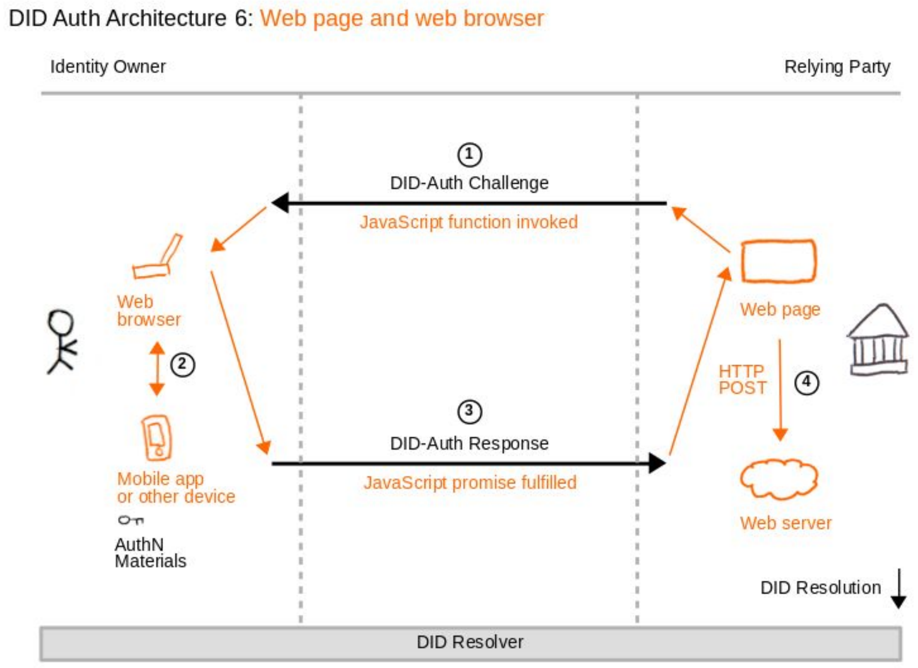
- **Response:**
  - Linked to a challenge (e.g. using a nonce).
  - Contains proof of control of a DID of the identity owner.

- **Transports:** HTTP POST, QR code, Mobile deep link, JavaScript browser API, Bluetooth, NFC, etc.
- Transports may require additional information such as endpoint URIs that may be included in the challenge, or discoverable from a DID.

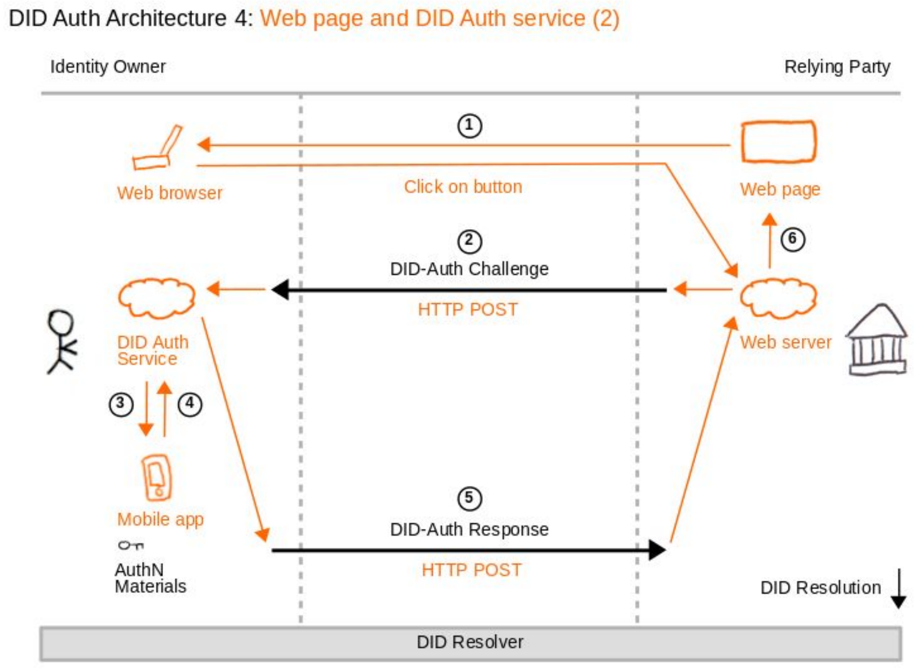
DID Auth Architectures



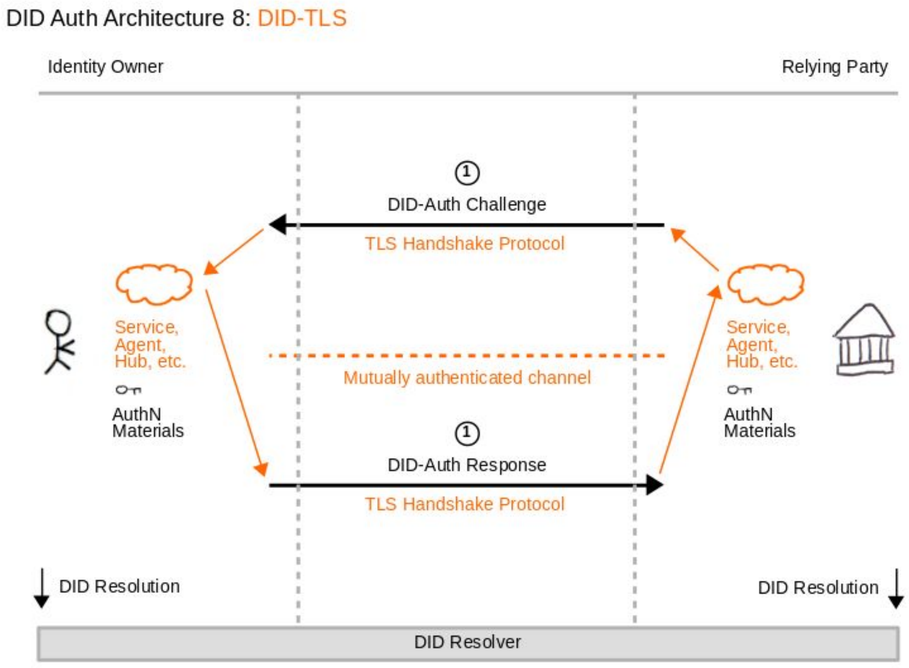
DID Auth Architectures



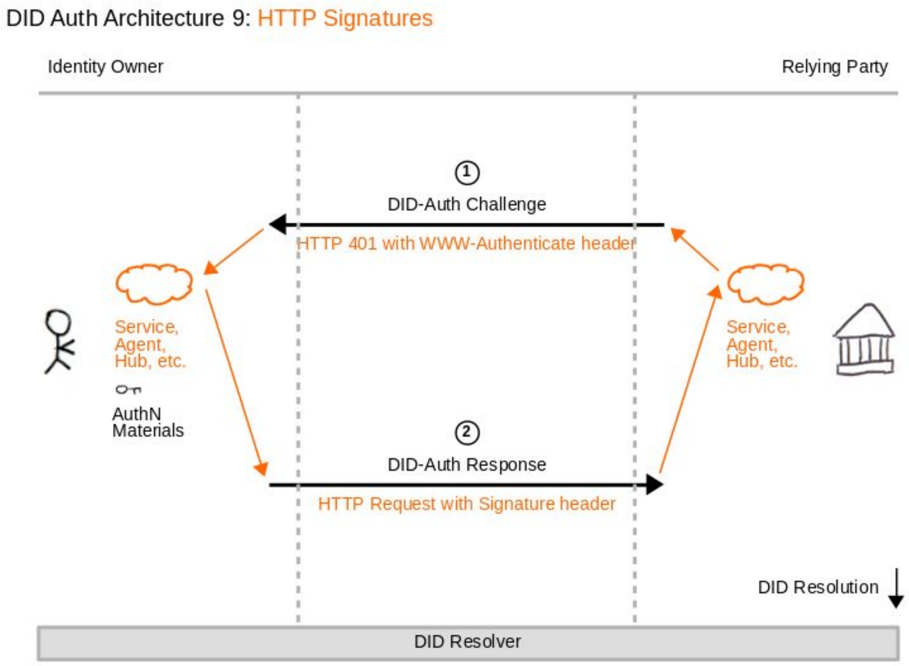
DID Auth Architectures



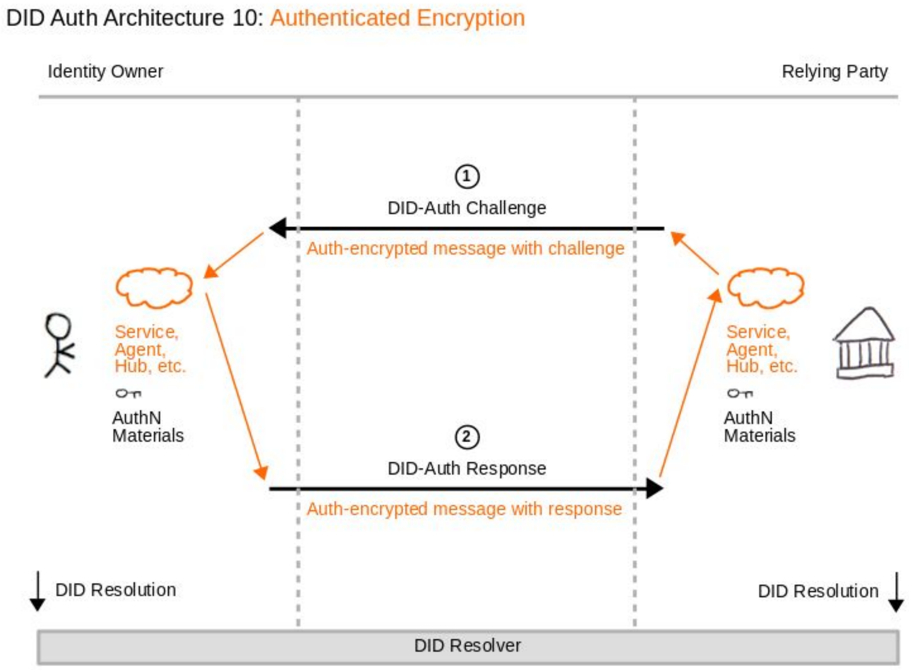
DID Auth Architectures



DID Auth Architectures


DID Auth Architecture 9: HTTP Signatures
Identity Owner
Relying Party
1
DID-Auth Challenge
HTTP 401 with WWW-Authenticate header
Service, Agent, Hub, etc.
AuthN Materials
Service, Agent, Hub, etc.
2
DID-Auth Response
HTTP Request with Signature header
DID Resolution
DID Resolver

DID Auth Architectures


DID Auth Architecture 10: Authenticated Encryption
Identity Owner
Relying Party
1
DID-Auth Challenge
Auth-encrypted message with challenge
Service, Agent, Hub, etc.
AuthN Materials
Service, Agent, Hub, etc.
AuthN Materials
2
DID-Auth Response
Auth-encrypted message with response
DID Resolution
DID Resolution
DID Resolver

Example Formats:

- Example JWT:

```
{
        "header": {
                "typ": "JWT",
                "alg": "ES256"
        },
        "payload": {
                "iss": "did:example:123456789abcdefg",
                "sub": "did:example:123456789abcdefg",
                "iat": 1479850830,
                "exp": 1511305200,
        },
        "signature": "..."
}
```

Example Formats:

- Example JSON-LD Verifiable Credential:

```
{
		"type": ["Credential"],
		"issuer": "did:example:123456789abcdefghi",
		"issued": "2018-03-07",
		"claim": {
			"id": "did:example:123456789abcdefghi",
			"publicKey": "did:example:123456789abcdefghi#keys-2"
		},
		"proof": {
			"type": "Ed25519Signature2018",
			"created": "2018-01-01T21:19:10Z",
			"creator": "did:example:123456789abcdefghi#keys-2",
			"nonce": "c0ae1c8e-c7e7-469f-b252-86e6a0e7387e",
			"signatureValue": "..."
		}
}
```

Example Formats:

- Example HTTP Signature:

```
POST /api/v1/issuerservices HTTP/1.1
Host: testhost.gov.bc.ca
User-Agent: curl/7.58.0
Accept: */*
Authorization: Signature
keyId="did:example:123456789abcdefghi#keys-1",algorithm="rsa-sha256",headers="(request-target) accept
user-agent",signature="214BeK0YJ9P2wmMXBjZNNXDMT4prNlc32ZkslillkJYkJeLp3zbz4r1WfgCltd103m7Ay
Y734qbau+GsWENDXaqxeTaP6LSMLWr6FexWMVgBbMzH1KDMhJlozTMFPkMsGlbuDpRKwEPqnX1Yy6ld
HLe8mlJfSAEUy5P/Hf3y1b1kI8XyHNVbChFJLiUkOocF7XsFuTfoB+MJSEUqJDnuKibiF+Ap9rxI7J7Uroe6EjaV
YqLXnGbpu8j8Oxn5QzGBZFCA/j6XgHy4NK9fG9pcCyyAPGzSYi1RWjDWFyS0RDQAXFBBNgyskXAgssKuV
S2AFwPvXcHb5mhvKFUYMvMESg=="
```

DID Auth and Verifiable Credentials

- Three ways to think about it:

1) DID Auth and Verifiable Credentials exchange are separate.
2) Verifiable Credentials exchange is an extension to (or part of) DID Auth.
3) DID Auth is a certain kind of Verifiable Credential.

DID Auth and Object Capabilities

- Object Capabilities:
  Authorization model where you can do something not because of who you are, but because of something you possess.
- DID Auth and Verifiable Credentials alone are not sufficient in an authorization decision.
- But: DID Auth and Verifiable Credentials can play a role in the process when Object Capabilities are issued or invoked.

DID Auth and Biometrics

- Unique and specific to an identity owner, and available to every human being.
- Matching a non-reversible biometric template against biometric input data.
- Various aspects:
  1) Biometrics can protect an identity owner’s physical device (e.g. phone).
  2) Biometric protocols such as IEEE 2410-2015 "BOPS" or Web Authentication.
  3) Direct exchange of biometric input data between identity owner and relying party.
- “Six Principles for Self-Sovereign Biometrics”

DID Auth and OpenID Connect, Web Authentication

- OpenID Connect:
  - Common web-based authentication protocol.
  - Use OIDC / OAuth 2.0 request and response formats, but with DIDs as identifiers.
  - Personal OIDC Provider can be discovered from DID Document.
  - OIDC can use DID Auth as a "local authentication method".
- Web Authentication:
  - JavaScript API to use FIDO authentication in the browser.
  - Separate registration and login flows for every "origin".
  - Relying party associates DIDs instead of public keys with an identity owner.

DID Auth and existing PKI Applications and Services

- PGP, SSH, etc.
- Could support DIDs instead of (or in addition to) static public keys.
- E.g. imagine a ~/.ssh/authorized_dids file.

Security and Privacy Considerations

- Directed Identity
  - Pairwise-pseudonymous DIDs on microledgers
- Identity owner vs. controller
  - Digital Guardianship
- Single logout
  - DID revocation

Semantics

- How do you express "I am me"?
- DID Auth based on JWT, DID-TLS, HTTP Signatures, Authenticated Encryption:
  - No real semantics, just proof of control of a DID.
- JSON-LD Verifiable Credentials:
  - "This is my public key."

```
"claim": {
          "id": "did:example:123456789abcdefghi",
          "publicKey": "did:example:123456789abcdefghi#keys-2"
}
```

- XDI local root nodes, relative identifiers:
  - "This is my DID."
  - "I am Markus."
  - "I am me."

```
/$is$ref/(=!:did:example:123456789abcdefghi)

/$is$ref/(=~markus)

/$is$ref/($self)
```

Introduction to DID Auth

Markus Sabadello

Danube Tech, Decentralized Identity Foundation, Sovrin Foundation, W3C VCWG, W3C CCG, OASIS XDI TC

In a self-sovereign world, how can I prove that "I am me"?

11th July 2018, markus@danubetech.com