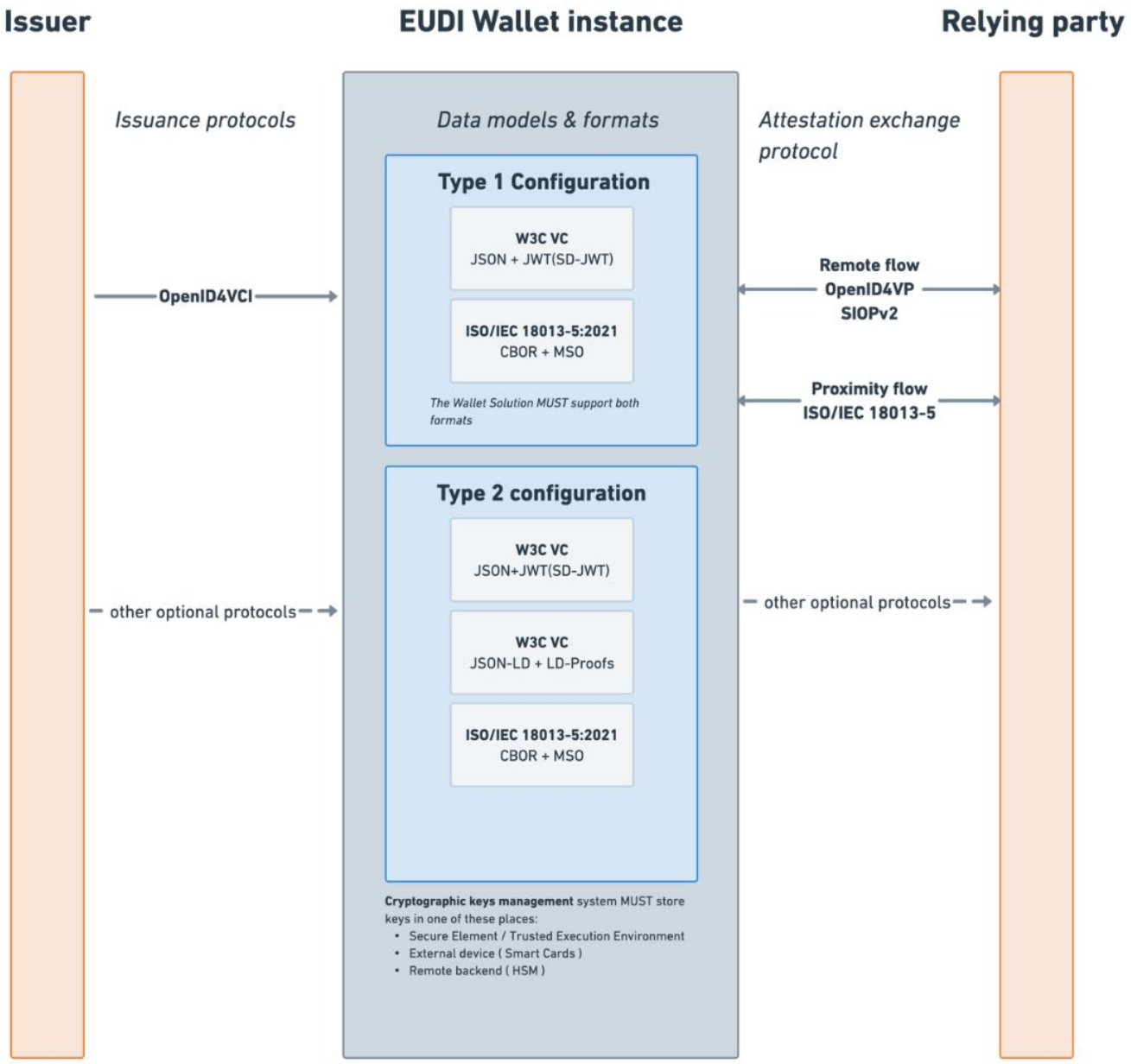
Wallet Security for Identity Ecosystems - OAuth Attestation-based Client Authentication

Paul Bastian & Markus Kreusch,
Bundesdruckerei

eIDAS 2.0 ARF

Motivation

- Decentralized identity ecosystems brings use cases from different domains together
  - Regulated and non-regulated issuers have different security requirements
- eIDAS ARF address these requirements
  - Type 1 Configuration for “high-security credentials” (hardware-bound)
  - Type 2 Configuration for “other credentials” (backup & portability enabled)



Requirements for Identity Credentials

Requirements from Regulations

- eIDAS LoA / TR-03107 Elektronische Identitäten
  - low, substantial, high
- Evaluation factors:
  - Enrolment
    - Proof of identity
    - Issuance security
  - Multi-Factor-Authentication
    - Possession
    - Knowledge
    - Biometry
  - Revocation
  - Communication security
  - Cryptographic algorithms

- Protection according to ISO18045 attack potential
  - ISO29115 attack vectors:
    - Online guessing
    - Offline guessing
    - Credential duplication
    - Credential theft

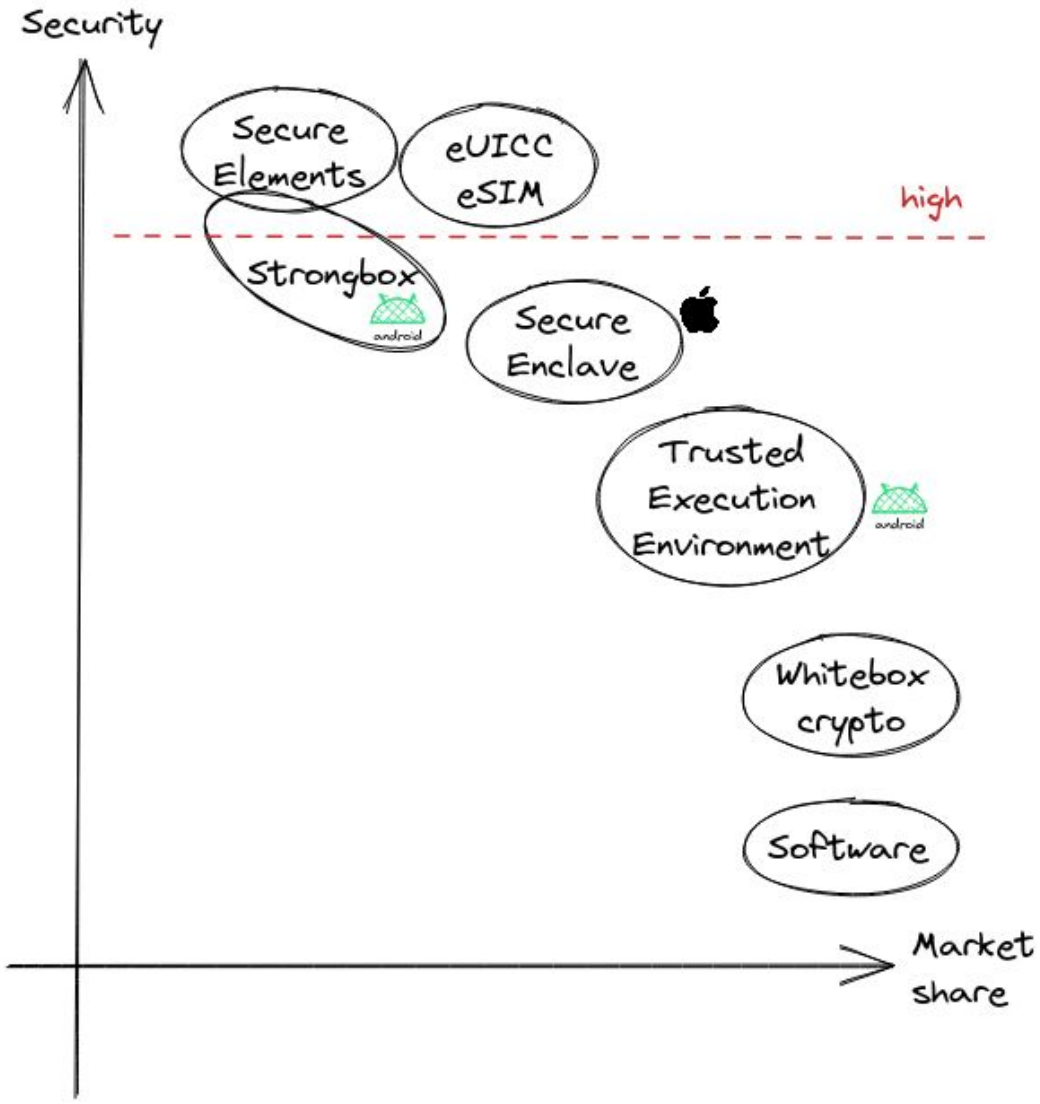
The Existing Tools

Regulatory requirements

- protection against
  - credential duplication/theft (extraction)
  - online/offline guessing (impersonation)
  - others.. (not wallet relevant)
- the wallet enables the issuer to achieve a certain level of assurance (LoA)

Mobile Market

- market of secure cryptographic key storage is very fragmented
- relying (partly) on OS security mechanism



Device binding

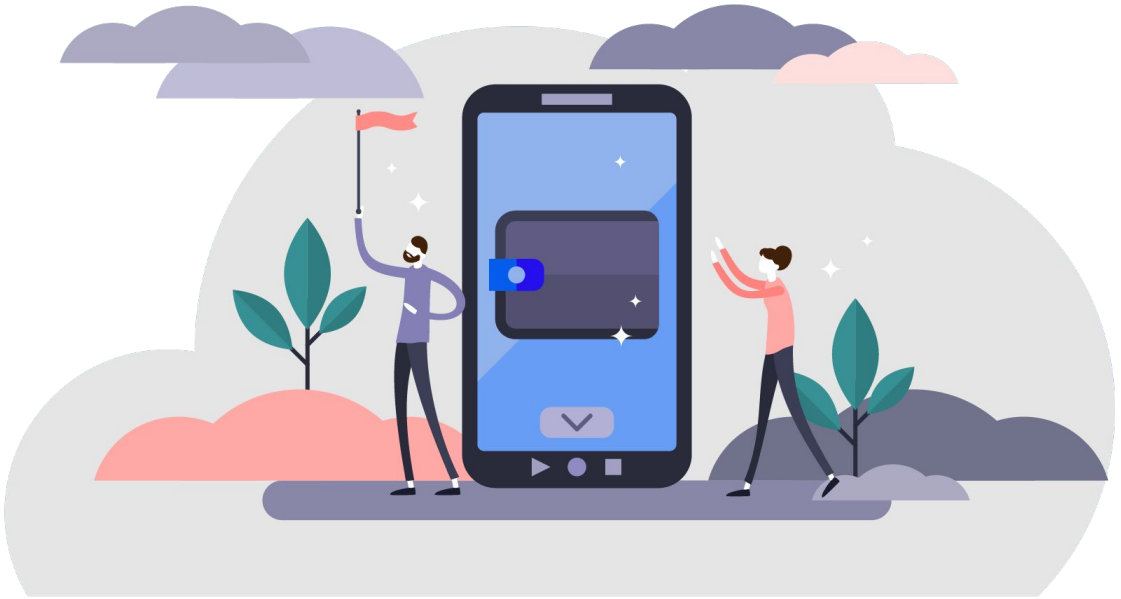
(authentication factor possession)

User binding

(authentication factor knowledge/biometry)

Wallet authentication

(integrity and authencity of the wallet)



Solution Components

Device Binding

- hardware-backed crypto systems are very restrained
  - NIST P256 with ECDSA-SHA256 as the smallest common denominator
  - simple, well-understood crypto system
  - SD-JWT, crypto agility by JOSE, (theoretically) PQC ready
- No backup & recovery strategy possible
- ZKP in mobile hardware is not available and might take 5-10 (?) years

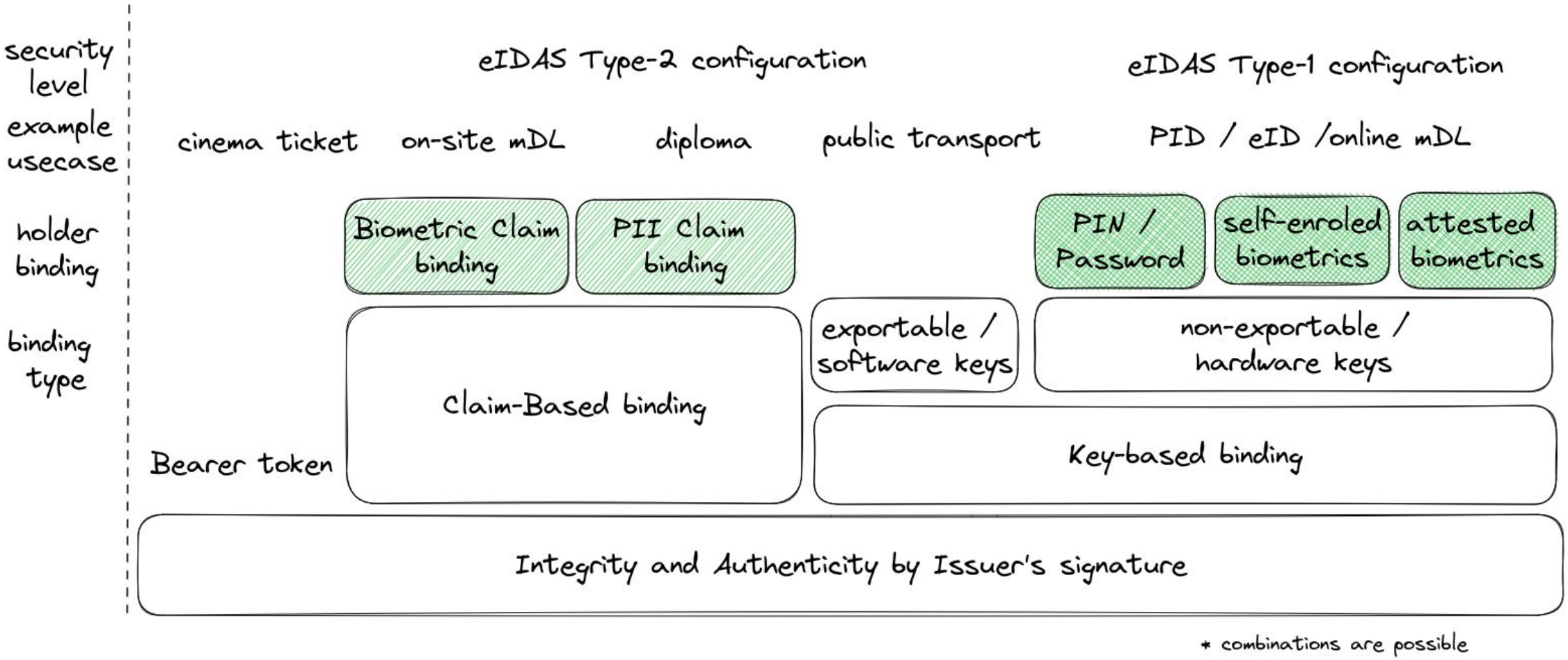
User/Holder Binding

- Local, on-device authentication
- Biometrics have many challenges and security issues (weak sensors, unknown FAR/FRR, attested enrolment, privacy..)
- Regulators are still in favour of PINs (some problems here as well, System-PIN vs App vs SE-PIN)

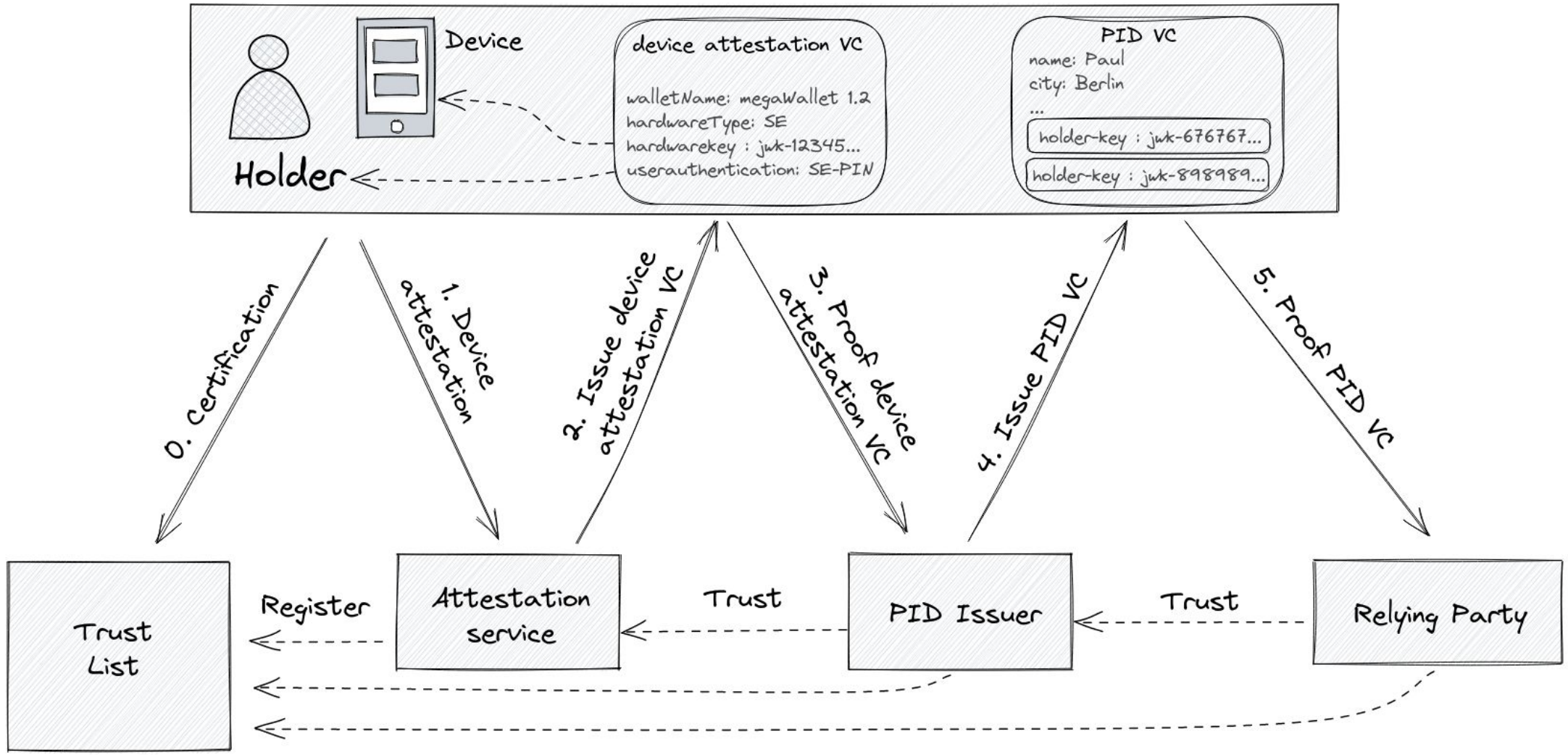
Wallet Authentication

- mobile OS presents a less-trusted, complex layer in front of trusted, high secure hardware key storage
- Use existing technology by mobile OS: iOS Device Check, Android SafetyNet/Integrity API
- Use Key attestations (not available on iOS)

• User Binding in depth



• Trust Model


Device
Holder
device attestation VC
walletName: megaWallet 1.2
hardwareType: SE
hardwarekey : jwk-12345...
userauthentication: SE-PIN
PID VC
name: Paul
city: Berlin
...
holder-key : jwk-676767...
holder-key : jwk-898989...
0. certification
1. Device attestation
2. Issue device attestation VC
3. Proof device attestation VC
4. Issue PID VC
5. Proof PID VC
Trust List
Register
Attestation service
Trust
PID Issuer
Trust
Relying Party

The Wallet Attestation Concept

Advantages

- Point of Interoperability is the attestation VC schema
  - Not the attestation process and protocols between wallet and attestation service
  - Future Proof mechanism independent from specific technology
- Simplify attestations for issuers and verifiers
  - Issuers do not need to parse and analyze complex OS-specific attestation statements
  - Easy integration into existing issuance protocols
- Design respects privacy of the holder, scaling and limits of attestations

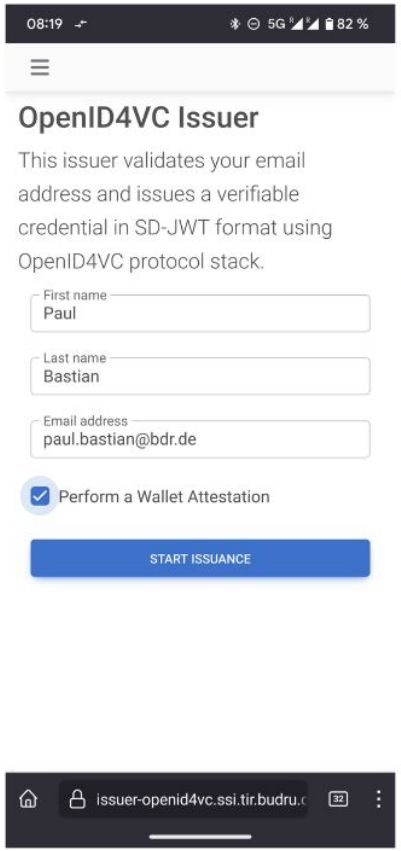
 Official Paper from HMD Journal (german)

Translated paper in English

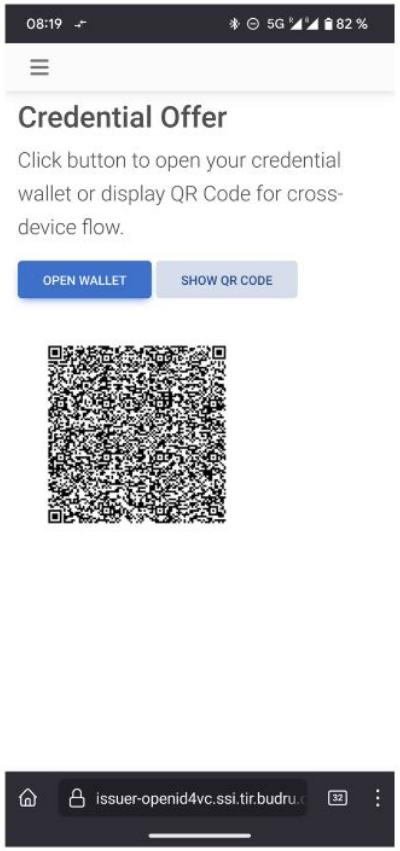
(the paper was submitted by 09/22, so some details have changed)

• IDunion Demo

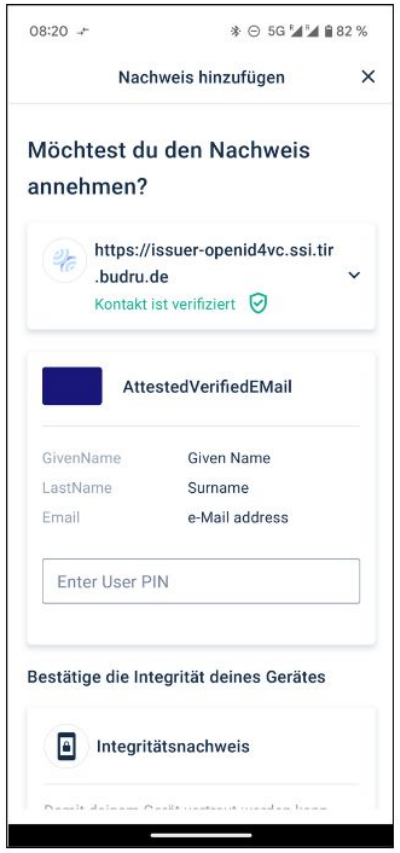
Demo of End-to-End Issuance with Wallet Attestation and Device Binding



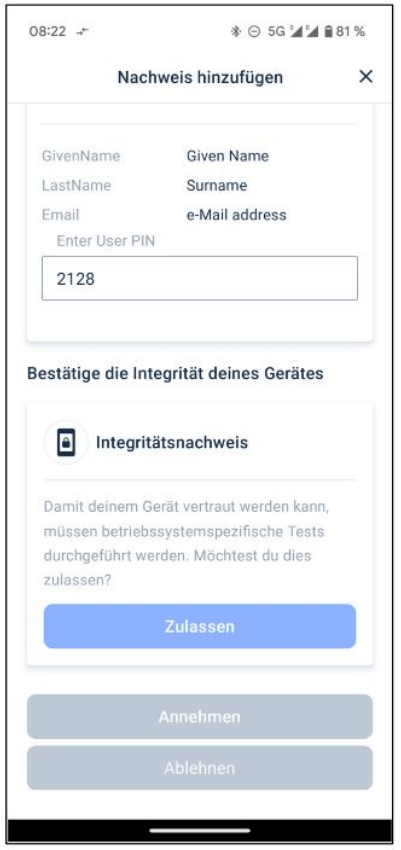
issuer's website as a starting point



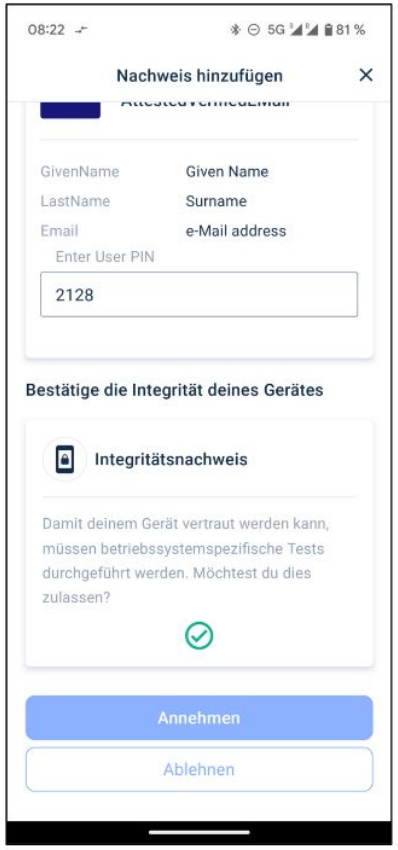
Wallet invocation with deeplink or QR-Code



Issuer Authentication with eIDAS 1 QWAC or EV certificates



UserPIN as a security feature of OpenID4VCI



Wallet Attestation for eIDAS Type-1 high assurance credential

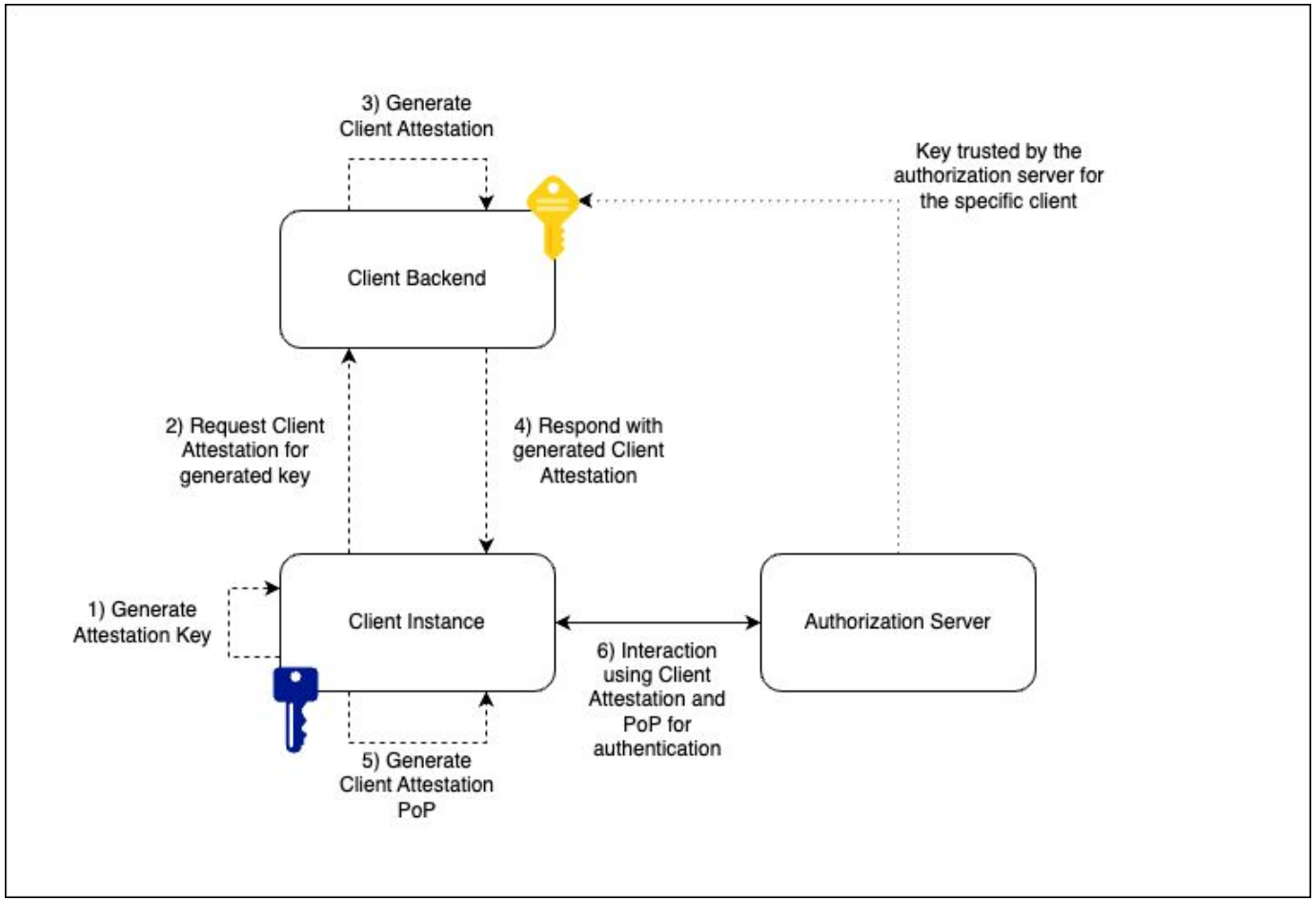
W3C SD-JWT VC issued after validation of wallet attestation

OAuth 2.0 Attestation-Based Client Authentication

Tobias Looker, Paul Bastian

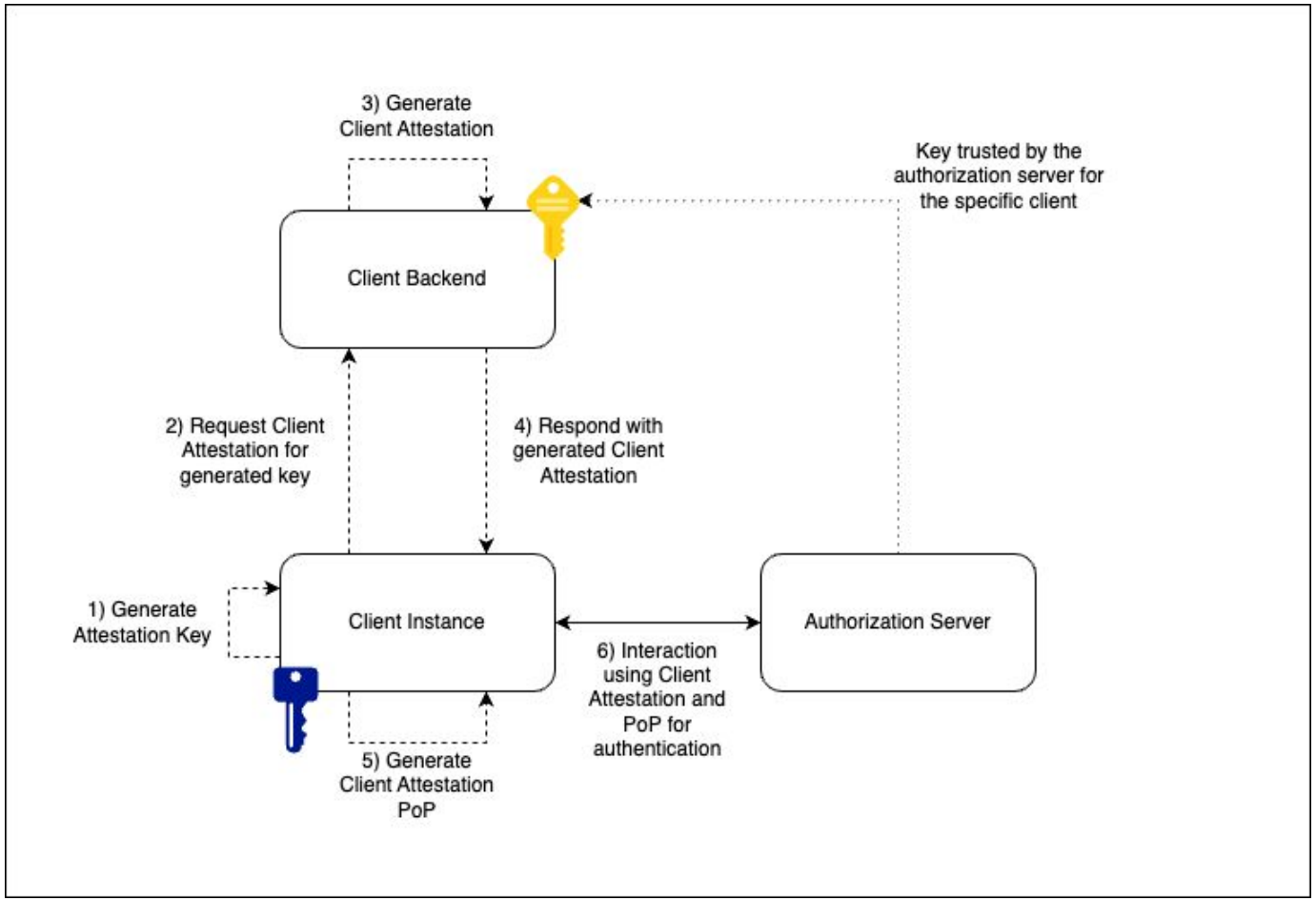
Proposed Solution

- Extends the established framework of RFC7521 for a new form of client authentication
- Client instance obtains an attestation from client backend
- Client backend may perform any number of security checks before issuing a key-bound attestation JWT to the client instance
- Client instance authenticates towards Authorization server during a token or PAR request
- **Note -** how the client communicates with the client backend in steps 2&4 are out of scope

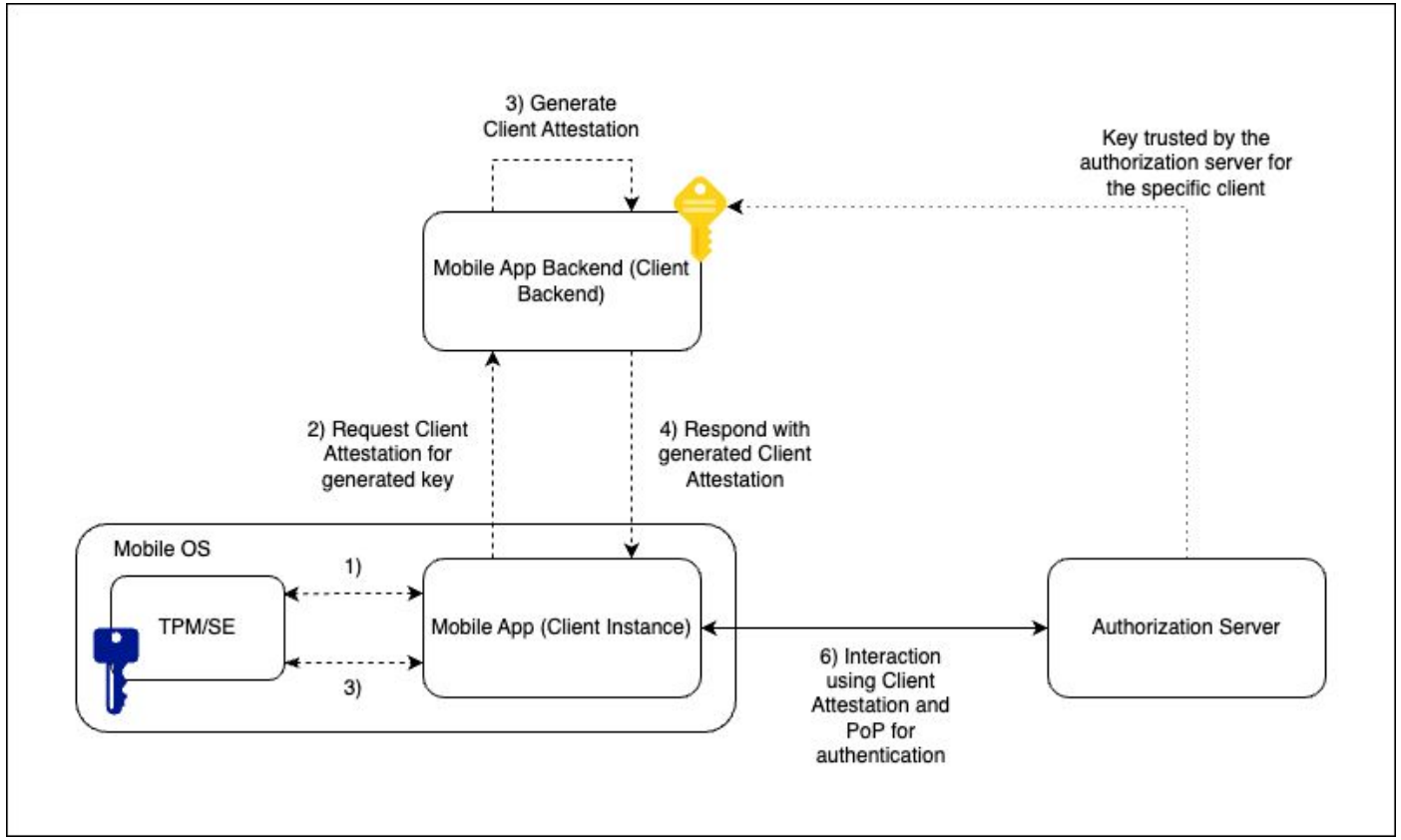


Key Callouts

- Proof of possession enabled client authentication method
- Can be used to authenticate the key used to bind to an access token via DPoP
- Direct mode of authentication between the client instance and the authorization server rather than a backend for front end pattern
- Avoids the client instance from having to register with the AS via DCR



Native App Example



Example - Token Request

```
POST /token HTTP/1.1
Host: as.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&
code=n0esc3NRze7LTCu7iYzS6a5acc3f0ogp4&
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3A
client-assertion-type%3Ajwt-client-attestation&
client_assertion=eyJhbGciOiJSUzI1NiIsImtpZCI6IjIyIn0.
eyJpc3Mi[...omitted for brevity...].
cC4hiUPo[...omitted for brevity...]~eyJzI1NiIsImtphbGciOimtpZCI6IjIyIn0.
IjIyIn0[...omitted for brevity...].
iOiJSUzI1[...omitted for brevity...]
```

Example - Token Request

```
POST /token HTTP/1.1
Host: as.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&
code=n0esc3NRze7LTCu7iYzS6a5acc3f0ogp4&
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3A
client-assertion-type%3Ajwt-client-attestation&
client_assertion=eyJhbGciOiJSUzI1NiIsImtpZCI6IjIyIn0.
eyJpc3Mi[...omitted for brevity...].
cC4hiUPo[...omitted for brevity...]~eyJzI1NiIsImtphbGciOimtpZCI6IjIyIn0.
IjIyIn0[...omitted for brevity...].
iOiJSUzI1[...omitted for brevity...]
```

New assertion type

Example - Token Request

```
POST /token HTTP/1.1
Host: as.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&
code=n0esc3NRze7LTCu7iYzS6a5acc3f0ogp4&
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3A
client-assertion-type%3Ajwt-client-attestation&
client_assertion=eyJhbGciOiJSUzI1NiIsImtpZCI6IjIyIn0.
eyJpc3Mi[...omitted for brevity...].
cC4hiUPo[...omitted for brevity...]~eyJzI1NiIsImtphbGciOimtpZCI6IjIyIn0.
IjIyIn0[...omitted for brevity...].
iOiJSUzI1[...omitted for brevity...]
```

Two JWTs concatenated via a '~' character

- Client Attestation
- Client Attestation PoP

Example - Client Assertion

Client Attestation

eyJhbGciOiAiRVMyNTYiLCJraWQiOiAiMTEifQ.eyJpc3MiOiJodHRwczovL2NsaWVudC5leGFtcGxlLmNvbSIsInN1YiI6Imh0dHBzOi8vY2xpZW50LmV4YW1wbGUuY29tIiwibmJmIjoxMzAwODE1NzgwLCJleHAiOjEzMDA4MTkzODAsImNuZiI6eyJqd2siOnsia3R5IjoiRUMiLCJ1c2UiOiJzaWciLCJjcnYiOiJQLTI1NiIsIngiOiIxOHdITGVJZ1c5d1ZONlZEMVR4Z3BxeTJMc3pZa01mNko4bmpWQWlidmhNIiwieSI6Ii1WNGRTNFVhTE1nUF80Zlk0ajhpcjdjbDFUWGxGZEFnY3g1NW83VGtjU0EifX19.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c~eyJhbGciOiJFUzI1NiJ9.eyJpc3MiOiJodHRwczovL2NsaWVudC5leGFtcGxlLmNvbSIsImF1ZCI6Imh0dHBzOi8vYXMuZXhhbXBsZS5jb20iLCJuYmYiOjEzMDA4MTU3ODAsImV4cCI6MTMwMDgxOTM4MH0.coB_mtdXwvi9RxSMzbIey8GVVQLv9qQrBUqmc1qj9Bs

Client Attestation PoP

Note signatures are invalid

Client Attestation JWT

```
{
  "typ": "wallet-attestation+jwt",
  "alg": "ES256",
  "kid": "1"
}
```

- Wallet Provider maps given key type and user authentication to Authenticator Assurance Level (aal)
- Wallet Instance Attestation is issued to Wallet and proven to the PID Issuer

```
{
  "iss": "https://attester.example.com",
  "sub": "https://client.example.com",
  "iat": 1516247022,
  "exp": 1541493724,
  "aal" : "https://trust-list.eu/aal/high",
  "cnf": {
    "jwk": {
      "kty": "EC",
      "crv": "P-256",
      "x": "TCAER19Zvu3OHF4j4W4vfSVoHIP1ILilDls7vCeGemc",
      "y": "ZxjiWWbZMQGHVWKVQ4hbSIirsVfuecCE6t4jT9F2HZQ"
    },
    "key_type": "STRONGBOX", //optional
    "user_authentication": "SYSTEM_PIN", //optional
  }
}
```

Client Attestation PoP JWT

```
{
  "typ": "kb+jwt",
  "alg": "ES256",
  "kid": "1"
}
```

```
{
  "iss": "https://client.example.com",
  "aud": "https://as.example.com",
  "nbf": 1516247022,
  "exp": 1541493724,
  "nonce" : "d25d00ab-552b-46fc-ae19-98f440f25064"
}
```

- PoP JWT binds the client attestation to the Authorization Server

PID SD-JWT

- PID Issuer takes Authenticator Assurance Level (aal) from Wallet Attestation
- PID Issuer adds Identification Assurance Level (ial) from his enrolment process
- Verifiers can query presentations constrained to IAL/AAL values


```
{
  "iss": "https://credential-issuer.example.com",
  "iat": 1541493724,
  "exp": 1541494724,
  "type": "Identity",
  "given_name": "Erika",
  "family_name": "Mustermann",
  "cnf": {
    "jwk": {
        "kty": "EC",
        "crv": "P-256",
        "x": "TCAER19Zvu3OHF4j4W4vfSVoHIP1ILilDls7vCeGemc",
        "y": "ZxjiWWbZMQGHVWKVQ4hbSIirsVfuecCE6t4jT9F2HZQ"
    }
  },
  "status": {
   "idx": "0,",
   "uri": "https://example.com/statuslists/1"
 },
  "assurance_level": {
     "aal": "https://trust-list.eu/aal/high",
     "ial": "https://trust-list.eu/ial/high"
   }
```

Questions / Feedback?

- Which method for replay attack prevention?
- How to provide nonces for the PAR endpoint?

Thanks!

Paul Bastian, Bundesdruckerei GmbH
paul.bastian@bdr.de

@idunion

@IDunion_SCE

contact@idunion.org

https://www.idunion.org/