German OWASP Day 2024

The Crucial Role of Web Protocols and Standards in Digital Wallet Ecosystems

Kristina Yasuda (German Federal Agency for Disruptive Innovation – SPRIND)

Nov. 12th, 2024

eIDAS 2.0 regulation has entered into force

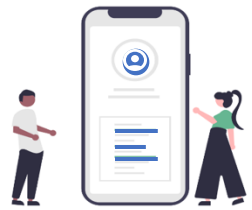
Regulation

In accordance with the revised eIDAS Regulation, **all European Member States are obligated to offer their citizens digital wallets by the end of 2026** – free of charge.

Source: eIDAS 2.0 (Regulation (EU) 2024/1183) https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=OJ:L_202401183#d1e1319-1-1

The Wallet will enable the users to…

- **securely request, obtain, store, delete, and share digital identity and digital documents offline and online**
- use pseudonyms
- sign by means of qualified electronic signatures
- access a dashboard of all transactions with a possibility to report alleged violations of data protection
- interact between wallets



Source: eIDAS 2.0 (Regulation (EU) 2024/1183) Article 5a https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=OJ:L_202401183#d1e1319-1-1

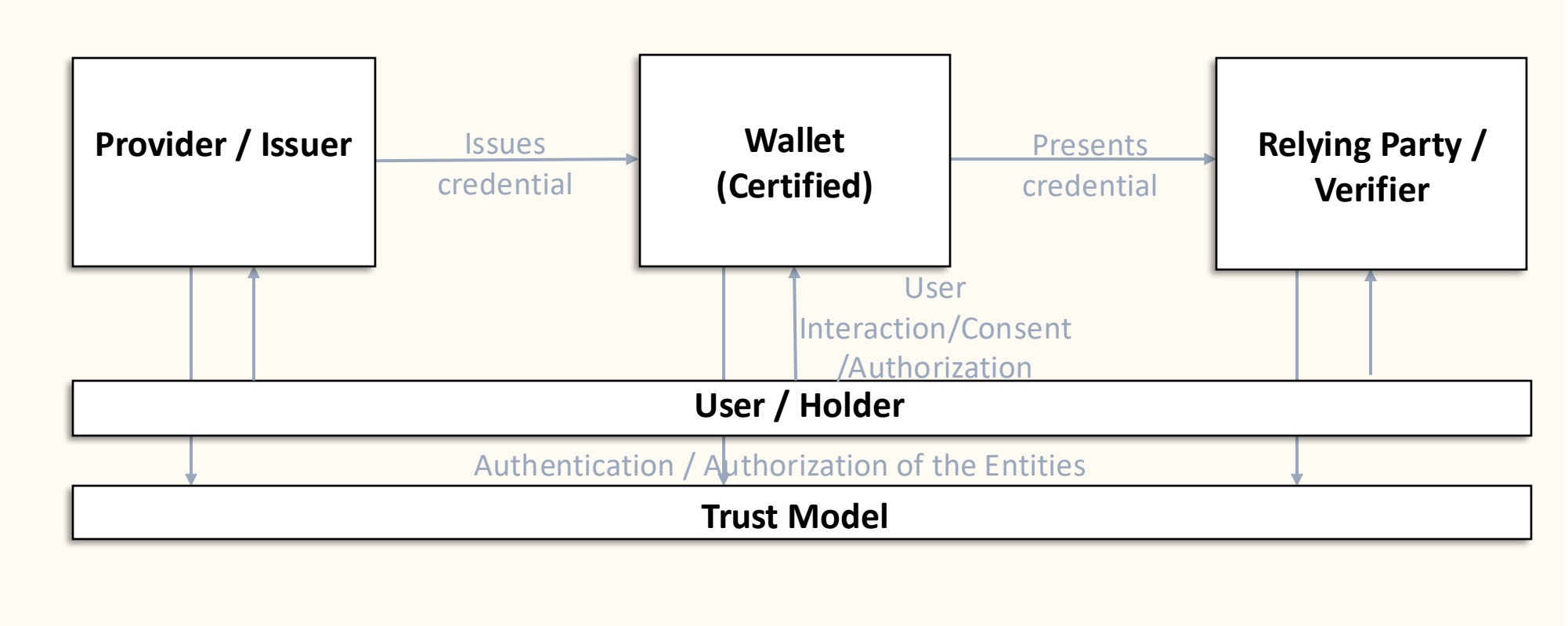
German OWASP Day 2024

Adoption of EUDI Wallets and Use-Cases

"**Very Large Online Platforms designated under the Digital Services Act** (including services such as Amazon, Zalando, Google Shopping, Shein, Temu, and Alibaba AliExpress) and private services that are legally required to authenticate their users will **have to accept the EU Digital Identity Wallet for logging into their online services**."

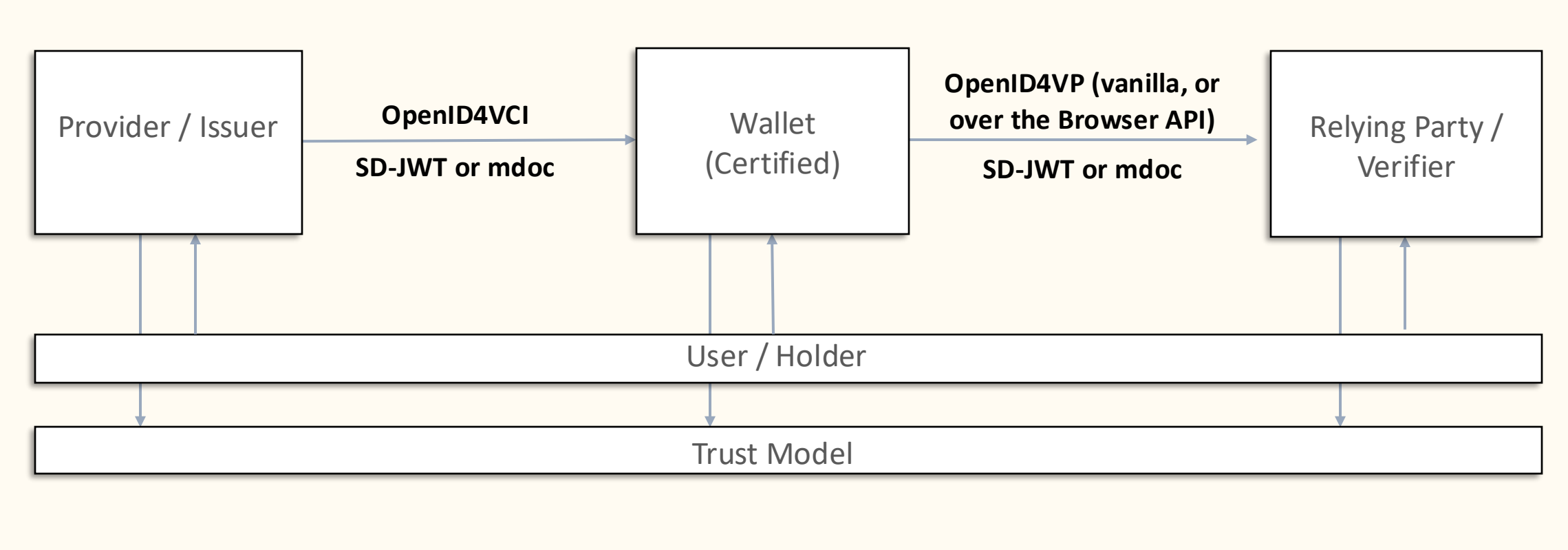
Source: European Commission - Press Release https://ec.europa.eu/commission/presscorner/api/files/document/print/en/ip_23_5651/IP_23_5651_EN.pdf

What is the EUDI Wallet ecosystem?



-> The flow of use data changes

Tech Stack



-> Needs to be privacy-preserving, secure, and interoperable

German OWASP Day 2024

01

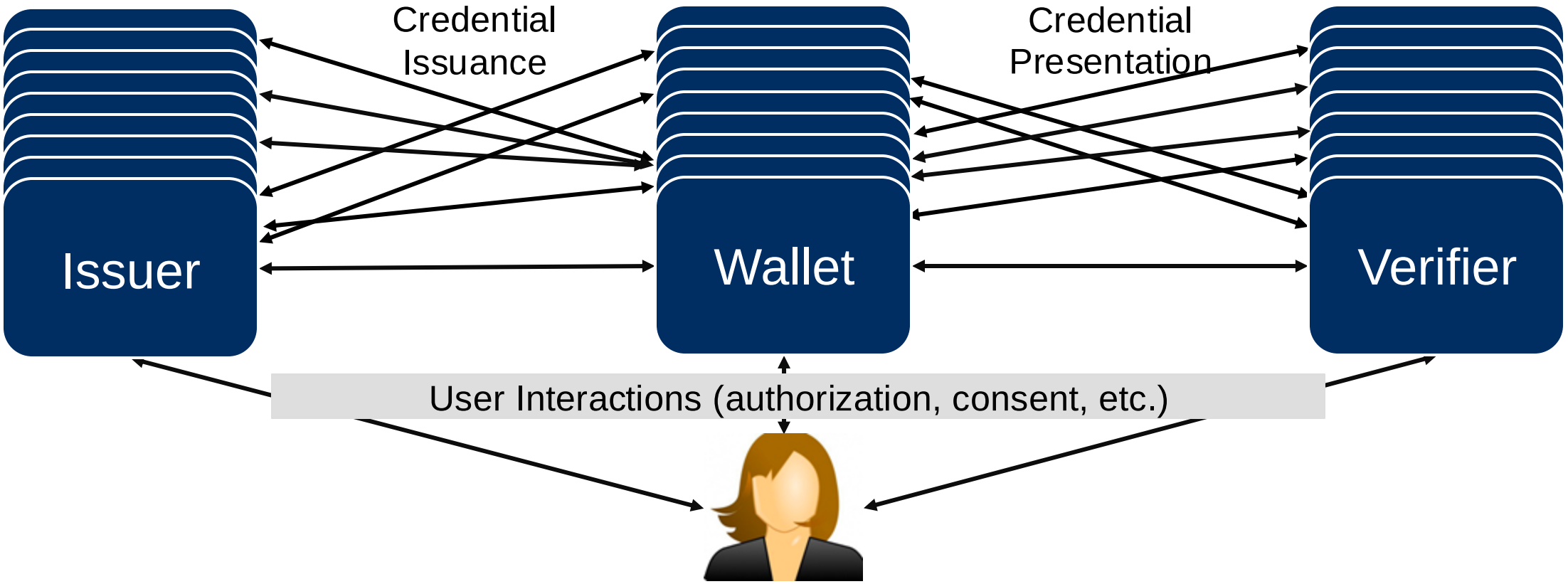
Protocol Layer: OpenID4VC

Design Principles: problems identified and how they were solved

Problem		Solution
A lot of entirely new Protocols. (Hard to get security right, steep learning curve)	⇒	Building upon currently widely used protocols: OAuth 2.0 and OpenID Connect. (Secure, already understood)
No clear winner among Credential Formats	⇒	Designing a protocol agnostic to the Credential Formats. (e.g. works both with ISO mdocs and IETF SD-JWT VC)
No one way to do key management.	⇒	Designing a protocol agnostic to the key management mechanism.
Participating entities cannot typically establish trust upfront, using traditional mechanisms.	⇒	Flexibility in Trust Management. Third Party Trust.

Protocol Layer Interoperability is Crucial

There was a need for the interoperable protocol layer that can support all of the credential formats, key resolution mechanisms and trust frameworks.



German OWASP Day 2024

Open Source libraries (not exhaustive...)

Walt.id

Kotlin:
github.com/walt-id/waltid-ssikit

Kotlin Multiplatform:
shorturl.at/XtEXw

Sphereon

Transcript:
tinyurl.com/2de634na

shorturl.at/yUnkA

shorturl.at/MHW1z

Microsoft

Swift:
tinyurl.com/2jejntsp

Kotlin:
tinyurl.com/4bd5p3bx

Spruce

Rust:
github.com/spruceid/oidc4vci-rs

Rust:
tinyurl.com/rp35fsc8

EBSI

Javascript:
tinyurl.com/y945s5xu

Impierce Technologies

Rust:
github.com/impierce/openid4vc

Animo

Typescript:
github.com/animo/paradym-wallet

Trustbloc

Go:
github.com/trustbloc/vcs

github.com/trustbloc/wallet-sdk

Italian Government

Python:
tinyurl.com/56ft5m34

Python:
shorturl.at/Gxd2D

AltMe

Dart:
github.com/TalaoDAO/AltMe

MOSIP

Kotlin/ Swift/ ReactNative:
github.com/mosip/tuvali

EUDI Reference

Wallet Implementation:
shorturl.at/rD7tf

German OWASP Day 2024

OpenID4VC Security Analysis

„Security and Trust in OpenID for Verifiable Credentials“ document describes the trust architecture in OpenID for Verifiable Credentials specifications, outlines security considerations and requirements for the components in an ecosystem

Master Thesis **„OpenID for Verifiable Credentials: formal security analysis using the Web Infrastructure Model“** published:

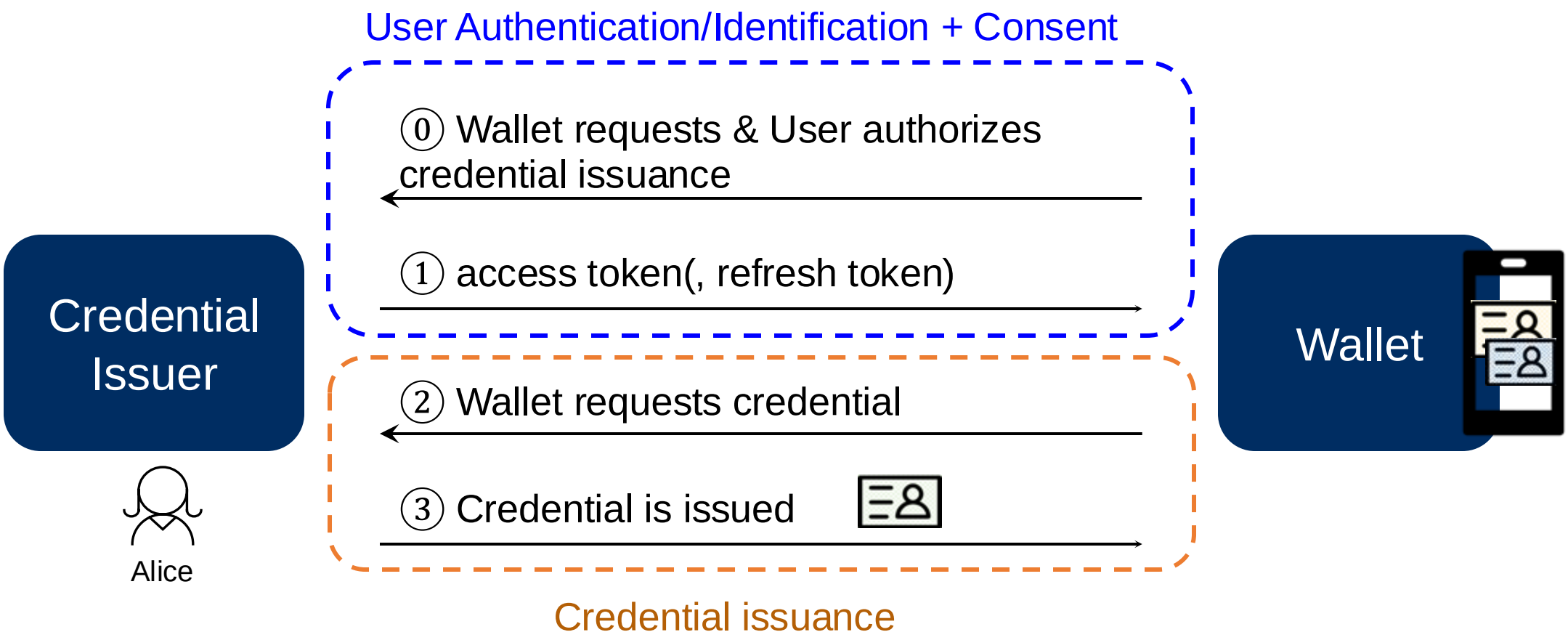
German OWASP Day 2024

02

OpenID for Verifiable Credential Issuance

German OWASP Day 2024

OAuth-protected API



OpenID4VCI can be used in conjunction with any other OAuth extension RFC

German OWASP Day 2024

Authorization Code Flow

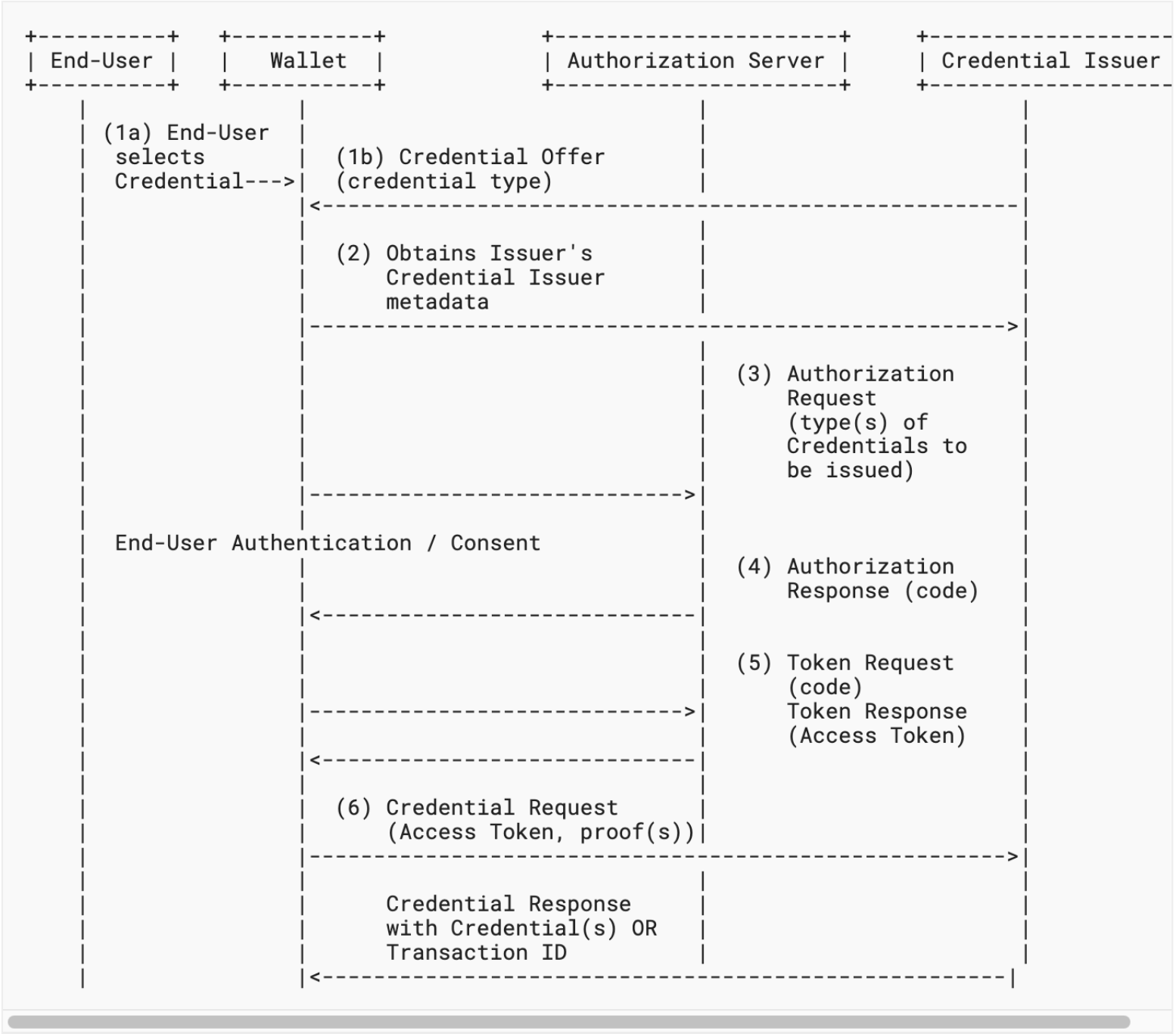


Figure 1: Issuance using Authorization Code Flow

Authorization Code Flow (I)

Credential Offer

```
{
  "credential_issuer": "https://credential-issuer.example.com",
  "credential_configuration_ids": [
    "UniversityDegreeCredential"
  ],
  "grants": {
    "authorization_code": {
      "issuer_state": "eyJhbGciOiJSU0Et...FYUaBy"
    }
  }
}
```

Authorization Request

```
GET /authorize?
  response_type=code
  &client_id=s6BhdRkqt3
  &code_challenge=E9Melhoa2OwvFrEMTJguCHaoeK1t8URWbuGJSstw-cM
  &code_challenge_method=S256
  &authorization_details=%5B%7B%22type%22%3A%20%22openid_credential%22%2C%20%22
    credential_configuration_id%22%3A%20%22UniversityDegreeCredential%22%7D%5D
  &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb
```

Token Request

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

grant_type=authorization_code
&code=SplxlOBeZQQYbYS6WxSbIA
&code_verifier=dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk
&redirect_uri=https%3A%2F%2FWallet.example.org%2Fcb
&client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer
&client_assertion=eyJhbGciOiJSU...
```

Authorization Code Flow (II)

Below is a non-normative example of a Credential Request for two Credential instances in an IETF SD-JWT VC [I-D.ietf-oauth-sd-jwt-vc] format using a Credential instance identifier and key proof type `jwt`:

```
POST /credential HTTP/1.1
Host: server.example.com
Content-Type: application/json
Authorization: Bearer czZCaGRSa3F0MzpnWDFmQmF0M2JW

{
  "credential_identifier": "CivilEngineeringDegree-2023",
  "proofs": {
    "jwt": [
      "eyJ0eXAiOiJvcGVuaWQ0dmNpL...Lb9zioZoipdP-jvh1WlA",
      "eyJraWQiOiJkaWQ6ZXhhbXBsZ...KPxgihac0aW9EkL1nOzM"
    ]
  }
}
```

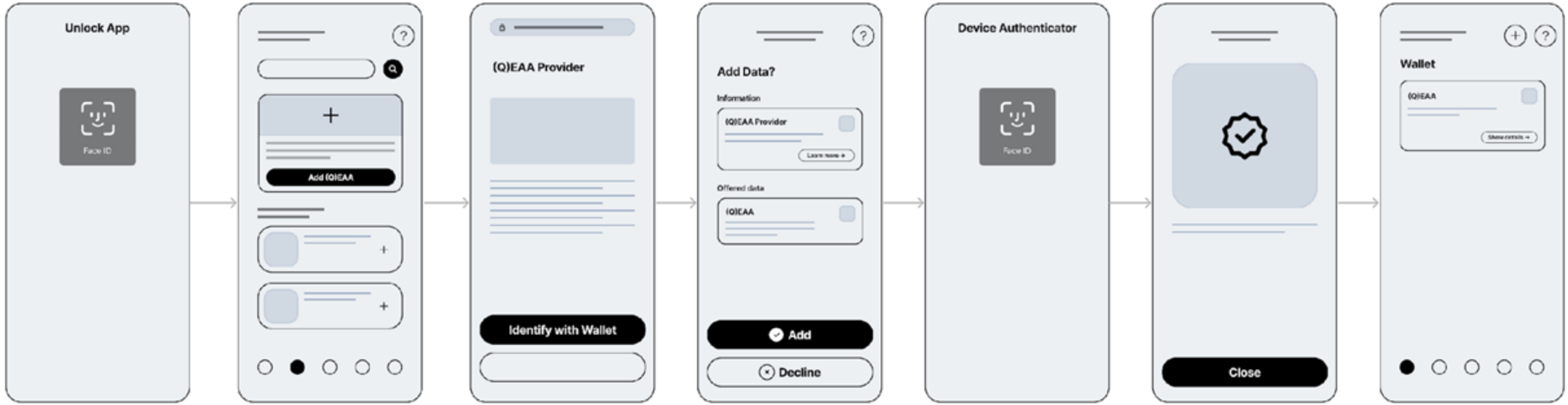
Below is a non-normative example of a Credential Response in an immediate issuance flow for multiple Credential instances in JWT VC format (JSON encoded) with an additional `notification_id` parameter:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "credentials": [
    {
      "credential": "LUpixVCWJk0eOt4CXQe1NXK....WZwmhmn9OQp6YxX0a2L"
    },
    {
      "credential": "YXNkZnNhZGZkamZqZGFza23....29tZTIzMjMyMzIzMjMy"
    }
  ],
  "notification_id": "3fwe98js"
}
```

German OWASP Day 2024

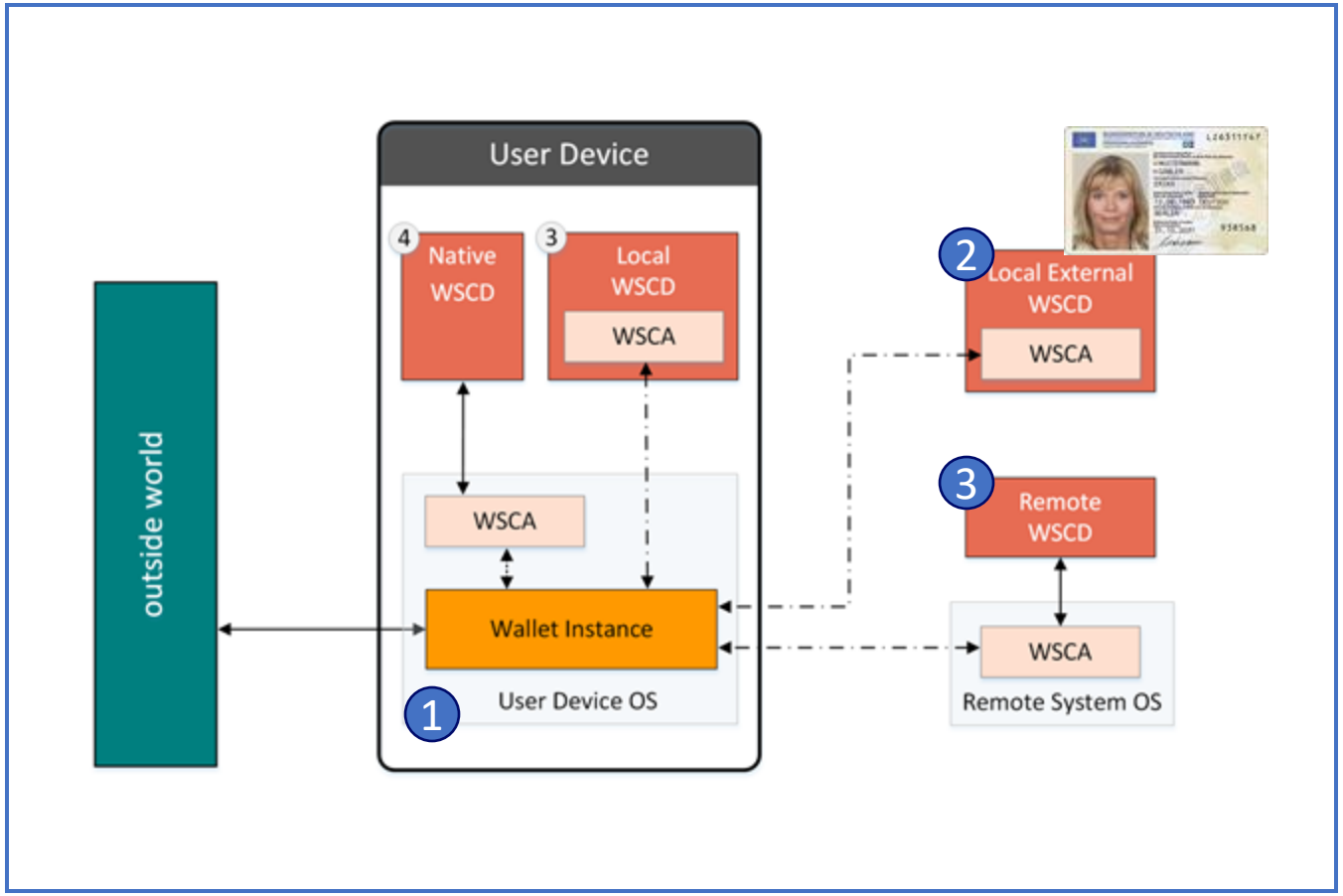
Authorization Code Flow



Security Considerations (selected)

1. Credentials issued into an untrustworthy Wallet
   - **Mitigation:** Credential Issuers use Wallet Attestation and Key Attestation as mechanisms to know what Wallet they are issuing Credentials into and how private keys are managed by that Wallet.
2. Preventing the usage of fraudulent/outdated credentials
   - **Mitigation:** Credential Issuer invalidates fraudulent credentials. Bitmap based Credential status management seems to be most common
3. Issued Credential being bound to a key controlled by an attacker
   - **Mitigation:** Wallet puts Credential Issuer provided nonce into the proof, to ensure proof is bound to the transaction

Where to store the keys?



1 **Secure Element card as Local WSCD**

fully decentralized with offline capability

2 **German ID card as external WSCD**

leverage high security of notified eID

3 **Cloud HSM as Remote WSCD**

high initial market reach and easy backup

Privacy Considerations (selected)

1. Correlation: Issuance/presentation or two presentation sessions by the same End-User can be linked on the basis of unique values encoded in the Credential (End-User claims, identifiers, Issuer signature, etc.) either by colluding Issuer/Verifier or Verifier/Verifier pairs, or by the same Verifier.
   - **Mitigation:** Issue a batch of Credentials with the same Credential Dataset to facilitate the use of a unique Credential per presentation or per Verifier. This approach solely aids in achieving Verifier-to-Verifier unlinkability.
   - **Mitigation:** Use cryptographic schemes that can provide non-correlation.
2. Basic User Privacy Protection: User Consent, Minimum Disclosure, Secure Storage of the Credentials

German OWASP Day 2024

03

OpenID for Verifiable Presentations

German OWASP Day 2024

OpenID for Verifiable Presentations: Same Device Flow

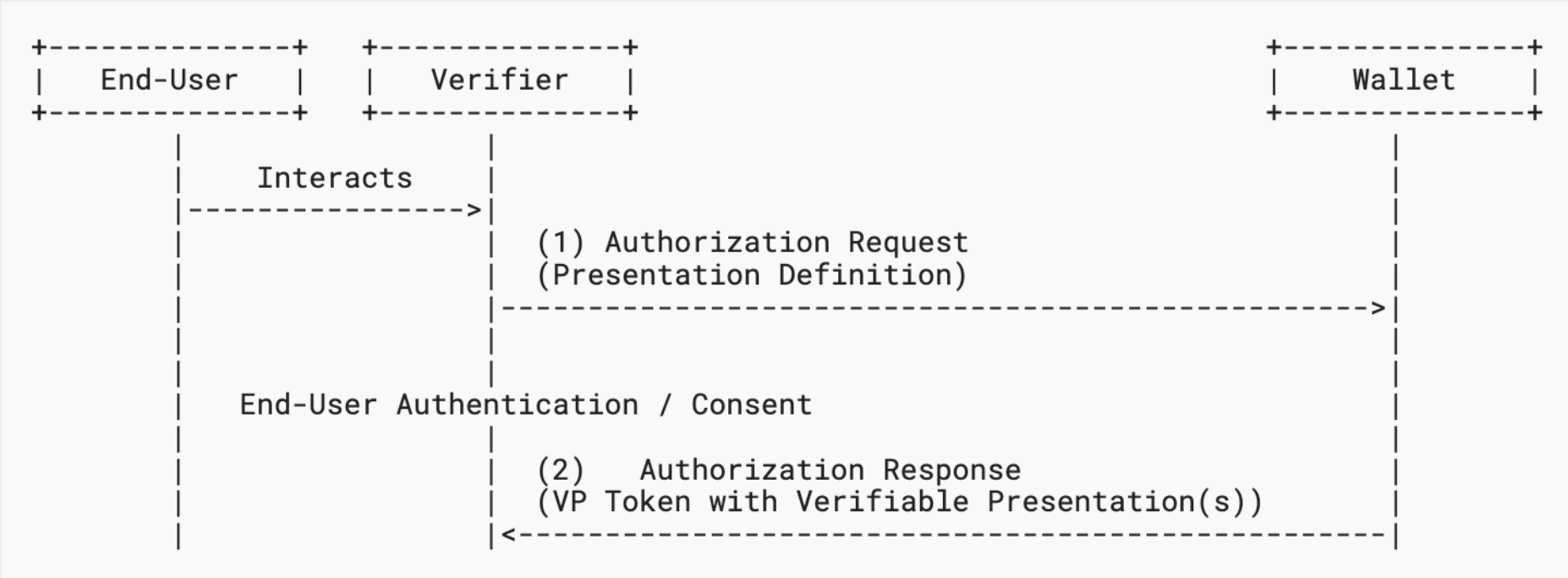


Figure 1: Same Device Flow

OpenID for Verifiable Presentations: Same Device Flow

The following is a non-normative example of an Authorization Request:

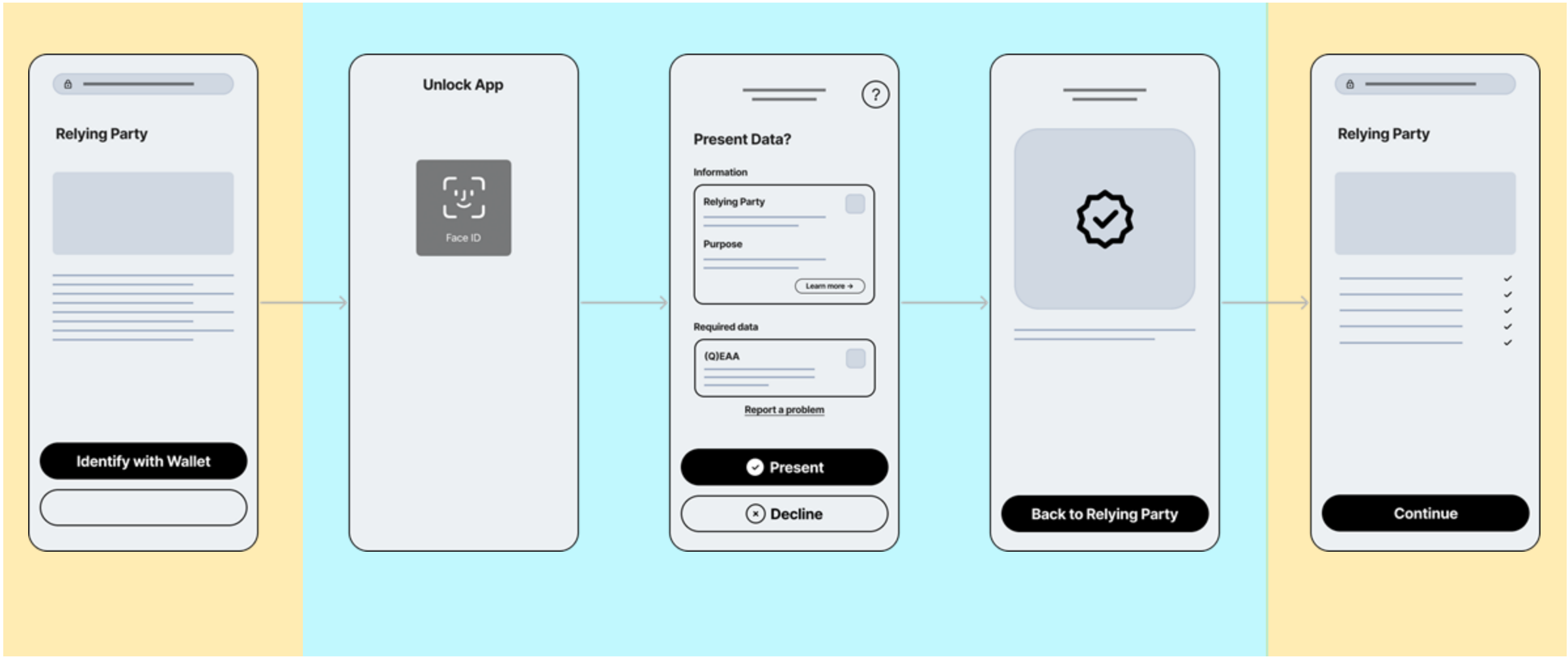
```
GET /authorize?
  response_type=vp_token
  &client_id=redirect_uri:https%3A%2F%2Fclient.example.org%2Fcb
  &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb
  &presentation_definition=...
  &transaction_data=...
  &nonce=n-0S6_WzA2Mj HTTP/1.1
```

The following is a non-normative example of an Authorization Response when the Response Type value in the Authorization Request was `vp_token`:

```
HTTP/1.1 302 Found
Location: https://client.example.org/cb#
  presentation_submission=...
  &vp_token=...
```

German OWASP Day 2024

Same Device Presentation



German OWASP Day 2024

OpenID for Verifiable Presentations: Cross Device Flow

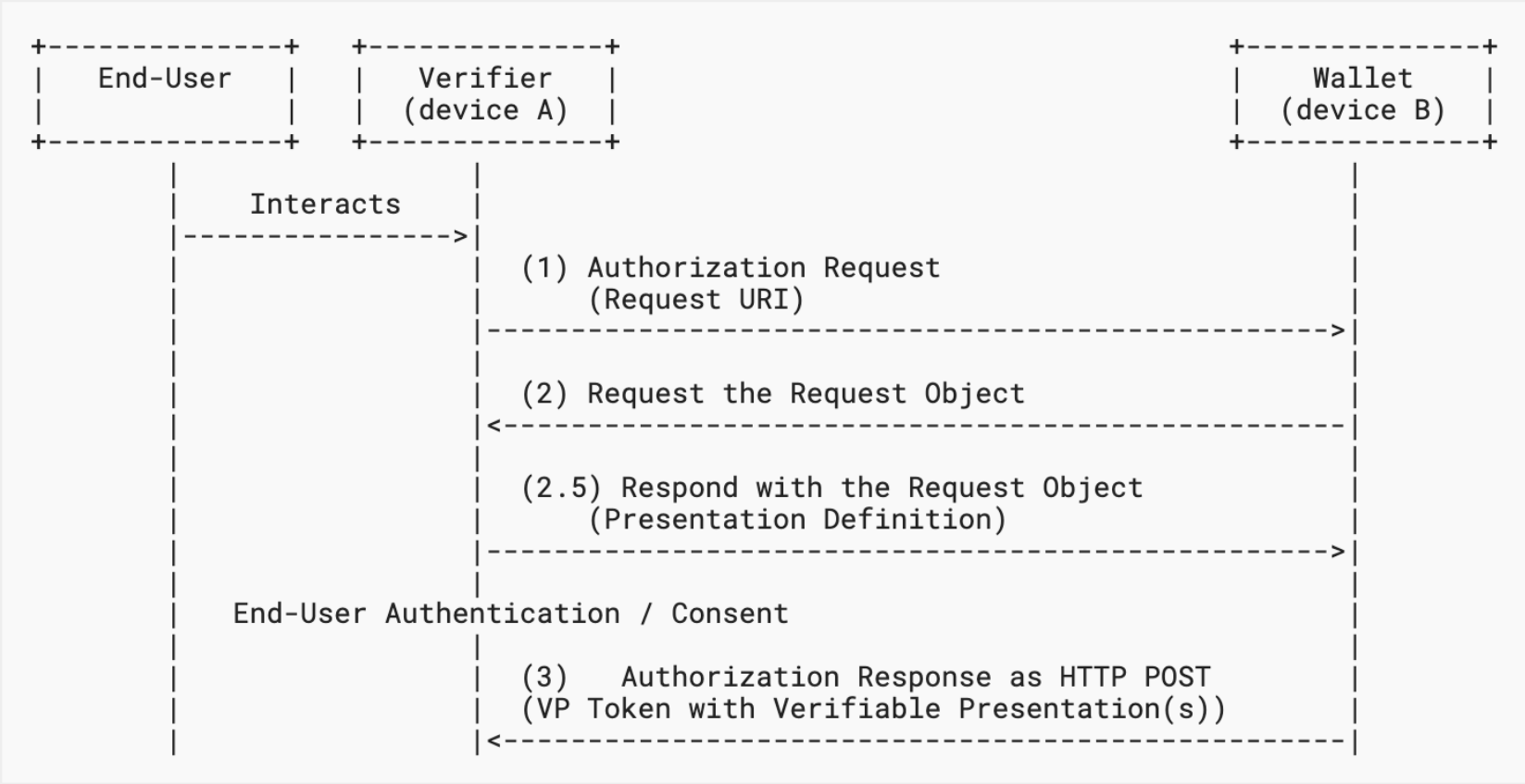


Figure 2: Cross Device Flow

OpenID for Verifiable Presentations: Cross Device Flow

The following non-normative example of an Authorization Request refers to the Authorization Request Object from above through the `request_uri` parameter. The Authorization Request can be displayed to the End-User either directly (as a link) or as a QR Code:

```
https://wallet.example.com?
  client_id=https%3A%2F%2Fclient.example.org%2Fcb
  &request_uri=https%3A%2F%2Fclient.example.org%2F567545564
```

The following is a non-normative example of the payload of a Request Object with Response Mode `direct_post`:

```
{
  "client_id": "redirect_uri:https://client.example.org/post",
  "response_uri": "https://client.example.org/post",
  "response_type": "vp_token",
  "response_mode": "direct_post",
  "presentation_definition": {...},
  "nonce": "n-0S6_WzA2Mj",
  "state": "eyJhb...6-sVA"
}
```

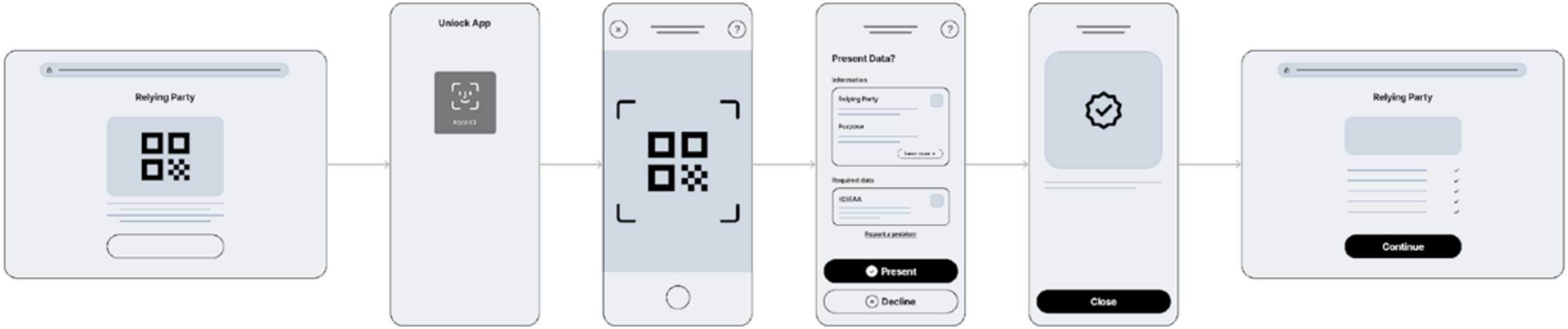
The following is a non-normative example of the Authorization Response that is sent via an HTTP POST request to the Verifier's Response URI:

```
POST /post HTTP/1.1
Host: client.example.org
Content-Type: application/x-www-form-urlencoded

  presentation_submission=...&
  vp_token=...&
  state=eyJhb...6-sVA
```

German OWASP Day 2024

Cross Device Presentation



Security Considerations (selected)

1. VP Token can be replayed
   - **Mitigation:** Credentials in in the VP Token must be bound to the transaction using `nonce` parameter and to the Relying Party using `aud` parameter
2. Session Fixation in response mode `direct_post`
   - **Mitigation:** use response mode `direct_post` followed by a redirect to the verifier front-end, Thiscauses the Wallet to redirect the flow to the Verifier's frontend at the device where the transaction was concluded. Verifier’s backend must ensure only correct verifier frontend can receive the presentation data. (this protection not applicable to cross-device flow.)
3. TLS protection ending before the credential reaches a target verifier application, or attacker stealing a VP Token
   - **Mitigation:** Encrypt the Response.

Privacy Considerations (selected)

1. Fingerprinting of the Wallet requests
   - **Mitigation:** Requests from the Wallet to the Verifier should be sent with the minimal amount of information possible, and in particular, without any HTTP headers identifying the software used for the request (e.g., HTTP libraries or their versions).
2. Basic User Privacy Protection: User Consent, Minimum Disclosure, Secure Storage of the Credentials

German OWASP Day 2024

04

OpenID4VP over W3C Digital Credentials API

Why?

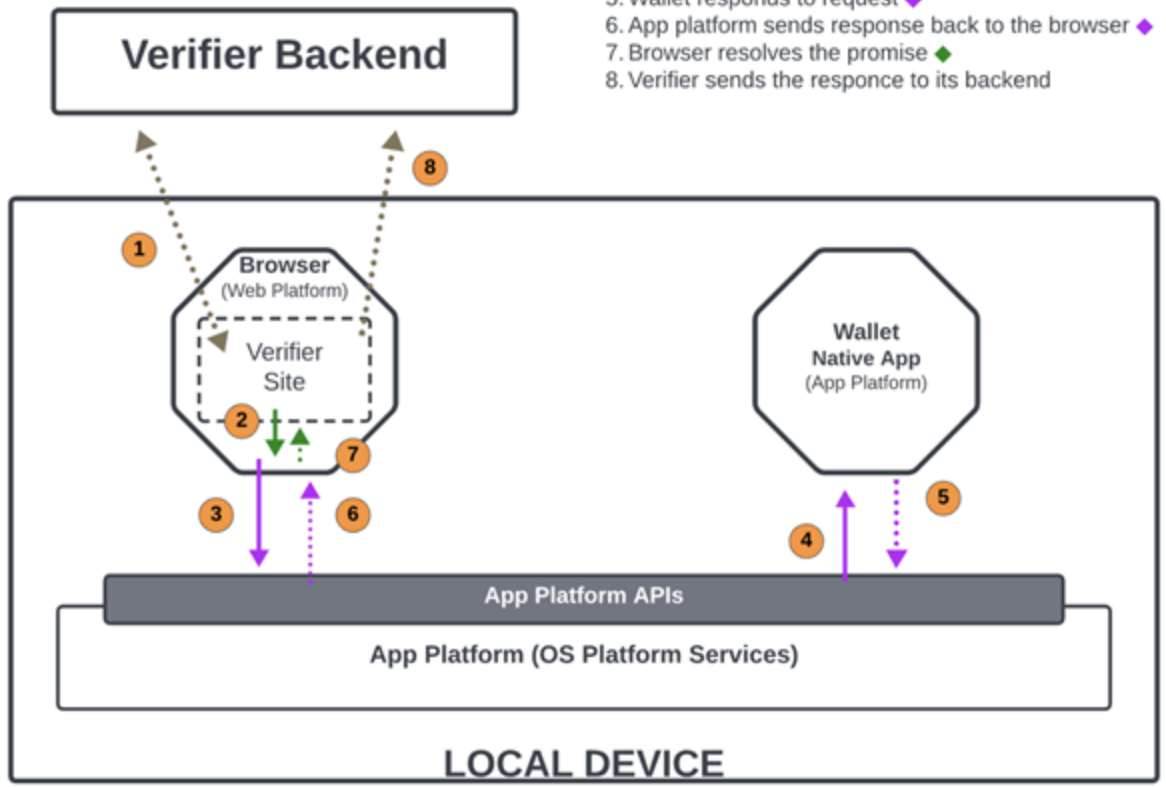
- Security:
  - Secure cross device, and even cross-platform, presentation of credentials.
  - The web platform provides the calling origin (or the app package if calling from an native app) that can be used as additional data point by the Wallet.
- UX:
  - Enabling privacy preserving model for Wallet selection based on request data, and getting rid of custom schemes in favor of a flexible and.
  - Guarantee that the user will end up on the same browser, where it started.

German OWASP Day 2024

Digital Credentials API Overview

SCENARIO
same-device
web-based verifier
native app wallet

1. Verifier site loaded in browser, request initiated
2. Web platform API request initiated ◆
3. Browser processes request and routes to the app platform ◆
4. App platform processes request and routes to wallet ◆
5. Wallet responds to request ◆
6. App platform sends response back to the browser ◆
7. Browser resolves the promise ◆
8. Verifier sends the responce to its backend



standardized API (W3C)
standardized API (Other)
platform-specific function API
platform-specific web translation API
protocol-specific

OID4VP request sent over W3C digital credentials API

- OID4VP request can be signed or unsigned
  - Unsigned request is needed to meet eIDAS 2.0 regulation.
- W3C Digital Credentials API work is happening in WICG, with a plan to be moved to the Federated Identity WG

```
let dcResponse = await navigator.credentials.get({
    signal: controller.signal,
    mediation: "required",
    digital: {
        requests: [{
            protocol: "openid4vp",
            data: presReqData
        }]
    }
});
```

German OWASP Day 2024

05

Credential Format Layer: IETF SD-JWT VC

SD-JWT & SD-JWT VC

Selective Disclosure for JWTs

using a simple, salted-hash based format
— enabling selective disclosure and key binding for JWS/JWT for verifiable credentials and more.

German OWASP Day 2024

Selective Disclosure

Issuer issued a whole set of claims:


```
{
  "iss": "https://server.example.com",
  "sub": "some-user-identifier",
  "aud": "s6BhdRkqt3",
  "given_name": "John",
  "family_name": "Doe",
  "email": "johndoe@example.com",
  "phone_number": "+1-202-555-0101",
  "address": {
     "street_address": "123 Main St",
     "locality": "Anytown",
     "region": "Anystate",
     "country": "US"
  },
  "birthdate": "1940-01-01"
}
```


✓ signed by Issuer

But **Verifier** only needs a subset in a given request:


```
{
  "iss": "https://server.example.com",
  "sub": "some-user-identifier",
  "aud": "s6BhdRkqt3",
  "given_name": "John",
  "family_name": "Doe",
  "email": "johndoe@example.com",
  "
  "address": {
  },
  "
}
```


✓ signed by Issuer

Design Principles (selected)

Algorithms	**Standard cryptography**: JWS Signature + Hash function
Security	Security-by-design Easy to understand & verify **Hardware binding possible** Cryptographic agility
Availability	**Widely-available JWT libraries can be leveraged** Already five independent implementations
Use Cases	Universal (beyond identity use cases)

German OWASP Day 2024

06

Issuance of IETF SD-JWT VC

SD-JWT in 5 Simple Steps

Step 1: Prepare User Data


```
{
 "iss": "https://example.com",
 "type": "IdentityCredential",
 "cnf": {"jwk": {"kty": "RSA","n": "0vx....Kgw","e": "AQAB" } },

 "given_name": "Max",
 "family_name": "Mustermann",
 "email": "mustermann@example.com",
 "address": {
   "street_address": "Musterstr. 23",
   "locality": "Berlin",
   "country": "DE"
  }
}
```

SD-JWT in 5 Simple Steps

Step 2: Create Disclosures

```
{
 "iss": "https://example.com",
 "type": "IdentityCredential",
 "cnf": {"jwk": {"kty": "RSA","n": "0vx....Kgw","e": "AQAB" } },

 "given_name": "Max",
 "family_name": "Mustermann",
 "email": "mustermann@example.com",
 "address": {
   "street_address": "Musterstr. 23",
   "locality": "Berlin",
   "country": "DE"
 }
```

salt | claim name | claim value

```
["GO0r26nO-iW50ZcAoOilFw", "given_name", "Max"]
["cSlbR135i0NjhsouMxrjjg", "family_name", "Mustermann"]
["oHDt43Vwuhpo8mzaprgCcw", "email", "mustermann@example.com"]
["rGc0KtY6WmflywTTKEWIEQ", "street_address", "Musterstr. 23"]
["pGQMQx-2tH2XwC_eQCFn4g", "locality", "Berlin"]
["TI15M8G5UIxPiWNZ-VLYBA", "country", "DE"]
```

SD-JWT in 5 Simple Steps

Step 3: Hash Disclosures & Replace Original Claims

```
{
 "iss": "https://example.com",
 "type": "IdentityCredential",
 "cnf": {"jwk": {"kty": "RSA","n": "0vx....Kgw","e": "AQAB" } },

          "_sd": [ "EW1o0egqa5mGcbytT5S-kAubcEjYEUwRkXlu2vC5l20",      ← ["GO0r26nO-iW50ZcAoOilFw", "given_name", "Max"]
                   "FEx-ITHt41I8_cn0SS-hvoLneX_RGlJo_8o2xRNhfdk",      ← ["cSlbR135i0NjhsouMxrjjg", "family_name", "Mustermann"]
                   "igg7H5fn2eBEMIEkE5Ckbm23QuwDJlTYoKRip08dYIc" ],    ← ["oHDt43Vwuhpo8mzaprgCcw", "email", "mustermann@example.com"]

 "address": {
            "_sd": [ "gqB5kmAwyry88aHjaAeO-USX6JOMaojukKsheo38O0c",     ← ["rGc0KtY6WmflywTTKEWIEQ", "street_address", "Musterstr. 23"]
                     "w8InvxsPXdKoowuVpyBMgI1b9_R2b6Xpa3OYOIjgQro",     ← ["pGQMQx-2tH2XwC_eQCFn4g", "locality", "Berlin"]
                     "vOnlYtcjr872fP3Wa75Ozl7c-6_MOVdIUNtwLKKxZw0" ]    ← ["TI15M8G5UIxPiWNZ-VLYBA", "country", "DE"]

 }
```

German OWASP Day 2024

SD-JWT in 5 Simple Steps

Step 4: Sign SD-JWT & Encode for Transport

{

eyJhbGciOiAiUlMyNTYiLCAia2lkIjogImNBRUlVcUowY21MekQxa3pHemhlaUJhZzBZ
UkF6VmRsZnhOMjgwTmdIYUEifQ.eyJpc3MiOiAiaHR0cHM6Ly9leGFtcGxlLmNvbS9pc
3N1ZXIiLCAiY25mIjogeyJqd2siOiB7Imt0eSI6ICJSU0EiLCAibiI6ICIwdng3YWdvZ
WJHY1FTdS4uLi4tY3NGQ3VyLWtFZ1U4YXdhcEp6S25xREtndyIsICJlIjogIkFRQUIif
X0sICJ0eXBlIjogIklkZW50aXR5Q3JlZGVudGlhbCIsICJjcmVkZW50aWFsU3ViamVjd
CI6IHsiX3NkIjogWyJFVzFvMGVncWE1bUdjYnl0VDVTLWtBdWJjRWlZRlV3UmtYbHUyd
kM1bDIwIiwgIkZFeC1JVEh0NDFJOF9jbjBTUy1odm9MbmVYX1JHbEpvXzhvMnhSTmhmZ
GsiLCAiUXhKVi0yVjFIOG1jbHRSNnZWQzRtM3JlVTVhTkg5d2RKejJVZG1Sb0kxRSIsI
CJhdFVuMVRZd1JBbDRHUTdQZUV0WGFNdzJmNHVJVGlKclg0ODV3TTh2NjdFIiwgImZUT
XczdmtrRUx3TDFYTnVZSzhlN3pCS0NIdV91aWY2MFNsRzFweVhJVVEiLCAiaWdnN0g1Z
m4yZUJFTUlFa0U1Q2tibTIzUXV3REpsVFlvS1JpcDA4ZFlJYyIsICJ0cFV0bDcwaHBVX
3hucnZaaTBHaEdvUllxam10MXpZZ3Z2NUlZMEF4N0tjIl0sICJhZGRyZXNzIjogeyJfc
2QiOiBbImdxQjVrbUF3eXJ5ODhhSGphQWVPLVVTWDZKT01hb2p1a0tzaGVvMzhPMGMiL
CAidk9ubFl0Y2pyODcyZlAzV2E3NU96bDdjLTZfTU9WZElVTnR3TEtLeFp3MCIsICJ3O
EludnhzUFhkS29vd3VWcHlCTWdsMWI5X1IyYjZYcGEzT1lPSWpnUXJvIl19fSwgImlhd
CI6IDE1MTYyMzkwMjIsICJleHAiOiAxNTE2MjQ3MDIyLCAic2RfZGlnZXN0X2RlcmI2Y
XRpb25fYWxnIjogInNoYS0yNTYifQ.1UHEPtLLUXOT51jH3gg-3C-ZidWzsB9Un-VxmM
VdQtTbLLhwDTB6HJtt15p43yCXTzdpiZxtDI6fr07Tp0Dy_Umg3Q5_FxFj4WHnsVuVzu
ASU8cFlGPi6xgH9D3w1G2hqepBS8DyQ5bA_p5kN_tKJVoP1xWhcQujRJ8kkEKQsRia4F
hrBldl8f41wgu_ipPqh1Ix4BVl7GJClZNx94nWPT7JUFkI6Y6JkahLf3S6gB0MxtmLAe
Y0qkuz8VeOZNfl_CDog55kVTkArorfoL6D6TEjI__-w6YyU0PnIRJXJ0wrYfoyhNl8LK
AP38QYMpdR7z_rsvHpQHzFAPTmevnHDg

["GO0r26nO-iW50ZcAoOilFw", "given_name", "Max"]

["gSlbR135i0NjhsouMxrjjg", "family_name", "Mustermann"]

["oHDt43Vwuhpo8mzaprgCcw", "email", "mustermann@example.com"]

["rGc0KtY6WmflywTTKEWIEQ", "street_address", "Musterstr. 23"]

["pGQMQx-2tH2XwC_eQCFn4g", "locality", "Berlin"]

["Tl15M8G5UlxPiWNZ-VLYBA", "country", "DE"]

},

SD-JWT in 5 Simple Steps

Step 5: Base64url-encode Disclosures for Transport

{

eyJhbGciOiAiUlMyNTYiLCAia2lkIjogImNBRUlVcUowY21MekQxa3pHemhlaUJhZzBZ
UkF6VmRsZnhOMjgwTmdIYUEifQ.eyJpc3MiOiAiaHR0cHM6Ly9leGFtcGxlLmNvbS9pc
3N1ZXIiLCAiY25mIjogeyJqd2siOiB7Imt0eSI6ICJSU0EiLCAibiI6ICIwdng3YWdvZ
WJHY1FTdS4uLi4tY3NGQ3VyLWtFZ1U4YXdhcEp6S25xREtndyIsICJlIjogIkFRQUIif
X0sICJ0eXBlIjogIklkZW50aXR5Q3JlZGVudGlhbCIsICJjcmVkZW50aWFsU3ViamVjd
CI6IHsiX3NkIjogWyJFVzFvMGVncWE1bUdjYnl0VDVTLWtBdWJjRWpZRVV3UmtYbHUyd
kM1bDIwIiwgIkZFeC1JVEh0NDFJOF9jbjBTUy1odm9MbmVYX1JHbEpvXzhvMnhSTmhmZ
GsiLCAiUXhKVi0yVjFIOG1jbHRSNnZWQzRtM3JlVTVhTkg5d2RKejJVZG1Sb0kxRSIsI
CJhdFVuMVRZd1JBbDRHUTdQZUV0WGFNdzJmNHVJVGlKclg0ODV3TTh2NjdFIiwgImZUT
XczdmtrRUx3TDFYTnVZSzhlN3pCS0NIdV91aWY2MFNsRzFweVhJVVEiLCAiaWdnN0g1Z
m4yZUJFTUlFa0U1Q2tibTIzUXV3REpsVFlvS1JpcDA4ZFlJYyIsICJ0cFV0bDcwaHBVX
3hucnZaaTBHaEdvUlIxam10MXpZZ3Z2NUlZMEF4N0tjIl0sICJhZGRyZXNzIjogeyJfc
2QiOiBbImdxQjVrbUF3eXJ5ODhhSGphQWVPLVVTWDZKT01hb2p1a0tzaGVvMzhPMGMiL
CAidk9ubFl0Y2pyODcyZlAzV2E3NU96bDdjLTZfTU9WZElVTnR3TEtLeFp3MCIsICJ3O
EludnhzUFhkS29vd3VWcHlCTWdsMWI5X1IyYjZYcGEzT1lPSWpnUXJvIl19fSwgImlhd
CI6IDE1MTYyMzkwMjIsICJleHAiOiAxNTE2MjQ3MDIyLCAic2RfZGlnZXN0X2Rlcml2Y
XRpb25fYWxnIjogInNoYS0yNTYifQ.1UHEPtLLUXOT51jH3gg-3C-ZidWzsB9Un-VxmM
VdQtTbLLhwDTB6HJtt15p43yCXTzdpiZxtDI6fr07Tp0Dy_Umg3Q5_FxFj4WHnsVuVzu
ASU8cFlGPi6xgH9D3w1G2hqepBS8DyQ5bA_p5kN_tKJVoP1xWhcQujRJ8kkEKQsRia4F
hrBldI8f41wgu_ipPqh1Ix4BVI7GJClZNx94nWPT7JUFkI6Y6JkahLf3S6gB0MxtmLAe
Y0qkuz8VeOZNfl_CDog55kVTkArorfoL6D6TEjI__-w6YyU0PnIRJXJ0wrYfoyhNI8LK
AP38QYMpdR7z_rsvHpQHzFAPTmevnHDg
~WyJHTzByMjZuTy1pVzUwWmNBb09pbEZ3IiwgImdpdmVuX25hbWUiLCAiTWF4Il0
~WyJjU2xiUjEzNWkwTmpoc291TXhyampnIiwgImZhbWlseV9uYW1lIiwgIk11c3Rlcm1hbm4iX
Q
~WyJvSER0NDNWd3VocG84bXphcHJnQ2N3IiwgImVtYWlsIiwgIm11c3Rlcm1hbm5AZXhhbXB
sZS5jb20iXQ
~WyJyR2MwS3RZNldtZmx5d1RUS0VXSUVRIiwgInN0cmVldF9hZGRyZXNzIiwgIk11c3RlcnN0ci
4gMjMiXQ
~WyJwR1FNUXgtMnRIMlh3Q19lUUNGbjRnIiwgImxvY2FsaXR5IiwgIkJlcmxpbiJd
~WyJUSTE1TThHNVVJeFBpV05aLVZMWUJBIiwgImNvdW50cnkiLCAiREUiXQ

→ Done!

},

German OWASP Day 2024

06

Presentation of IETF SD-JWT VC

German OWASP Day 2024

Example Presentation

eyJhbGciOiAiRVMyNTYifQ.eyJfc2QiOiBbIkNyUWU3UzVrcUJBSHQtbk1ZWGdjNmJkd
DJTSDVhVFkxc1VfTS1QZ2tqUEkiLCAiSnpZakg0c3ZsaUgwUjNQeUVNZmVadTZKdDY5d
TVxZWhabzdGN0VQWWxTRSIsICJQb3JGYnBLdVZ1Nnh5bUphZ3ZrRnNGWEFiUm9jMkpHb
EFVQTJCQTRvN2NJIiwgIlRHZjRvTGJnd2Q1SlFhSHILVlFaVTlVZEdFMHc1cnREc3Jae
mZVYW9tTG8iLCAiWFFfM2tQS3QxWHlYN0tBTmtxVll2eVoyVmE1TnJQSXZQWWJ5TXZSS
0JNTSIsICJYekZyendzY002R242Q0pEYzZ2Vks4QmtNbmZHOHZPU0tmcFBJWmRBZmRFI
iwgImdiT3NJNEVkcTJ4Mkt3LXc1d1BFemFrb2I5aFYxY1JEMEFUTjNvUUw5Sk0iLCAia
nN1OXlWdWx3UVFsaEZsTV8zSmx6TWFTRnpnbGhRRzBEcGZheVF3TFVLNCJdLCAiaXNzI
jogImh0dHBzOi8vaXNzdWVyLmV4YW1wbGUuY29tIiwgImlhdCI6IDE2ODMwMDAwMDAsI
CJleHAiOiAxODgzMDAwMDAwLCAic3ViIjogInVzZXJfNDliLCAibmF0aW9uYWxpdGllc
yI6IFt7Ii4uLiI6ICJwRm5kamtaX1ZDem15VGE2VWpsWm8zZGgta284YUlLUWM5RGxHe
mhhVllvIn0sIHsiLi4uIjogIjdDZjZKa1B1ZHJ5M2xjYndIZ2VaOGtoQXYxVTFPU2xIc
lAwVmtCSnJXWjAifV0sICJfc2RfYWxnIjogInNoYS0yNTYiLCAiY25mIjogeyJqd2siO
iB7Imt0eSI6ICJFQyIsICJjcnYiOiAiUC0yNTYiLCAieCI6ICJUQ0FFUjE5WnZ1M09IR
jRqNFc0dmZTVm9ISVAxSUxpbERsczd2Q2VHZW1jIiwgInkiOiAiWnhqaVdXYlpNUUdlV
ldLVlE0aGJTSWlyc1ZmdWVjQ0U2dDRqVDlGMkhaUSJ9fX0.OeQrinudSFTXNysz2NuNQ
rwWJv-P9gQ-Ce3wWEYZkxngeA4GKfPfApdNzBa40dH1urt8tXhW2WQl-I00v8teuw~Wy
JlbHVWNU9nM2dTTklJOEVZbnN4QV9BIiwgImZhbWIseV9uYW1lIiwgIkRvZSJd~WyJBS
ngtMDk1VlBycFR0TjRRTU9xUk9BIiwgImFkZHJlc3MiLCB7InN0cmVldF9hZGRyZXNzI
jogljEyMyBNYWluIFN0IiwgImxvY2FsaXR5IjogIkFueXRvd24iLCAicmVnaW9uIjogI
kFueXN0YXRlIiwgImNvdW50cnkiOiAiVVMifV0~WyIyR0xDNDJzS1F2ZUNmR2ZyeU5ST
jl3IiwgImdpdmVuX25hbWUiLCAiSm9obiJd~WyJsa2x4RjVqTVlsR1RQVW92TU5JdkNB
IiwgIlVTIl0~eyJhbGciOiAiRVMyNTYiLCAidHlwIjogImtiK2p3dCJ9.eyJub25jZSI
6ICIxMjM0NTY3ODkwIiwgImF1ZCI6ICJodHRwczovL3ZlcmlmaWVyLmV4YW1wbGUub3J
nIiwgImlhdCI6IDE2OTgwNzc3OTAsICJfc2RfaGFzaCI6ICIzNHQ4dkNDX2NfdlZMbk9
hZEJ0d2g0ZEZ2QkVyU2w5ektPcXdtNmloVF9VIn0.ZlotfwqF9NUTRAShrd8jGSJEB6e
3Z3EKm-AD5udfzggxfK-lQM4TCKbHK81eV088YTKl-UfM7WSyQpx5wpNpZw

Issuer-signed SD-JWT~Disclosures~KB-JWT

Key-Binding JWT Body:


```
{
  "nonce": "1234567890",
  "aud": "https://verifier.example.org",
  "iat": 1698077790,
  "sd_hash":
"34t8vCC_c_vVLnOadBtwh4dFvBErSl9zKOqwm6ihT_U"
}
```

Reconstructing the Original Data


```
eyJhbGciOiAiRVMyNTYifQ.eyJfc2QiOiBbIkNyUWU3UzVrcUJBSHQtbk1ZWGdjNmJkd
DJTSDVhVFkxc1VfTS1QZ2tqUEkiLCAiSnpZakg0c3ZsaUgwUjNQeUVNZmVadTZKdDY5d
TVxZWhabzdGN0VQWWxTRSIsICJQb3JGYnBLdVZ1Nnh5bUphZ3ZrRnNGWEFiUm9jMkpHb
EFVQTJCQTRvN2NJIiwgIlRHZjRvTGJnd2Q1SlFhSHILVlFaVTlVZEdFMHc1cnREc3Jae
mZVYW9tTG8iLCAiWFFfM2tQS3QxWHlYN0tBTmtxVll2eVoyVmE1TnJQSXZQWWJ5TXZSS
0JNTSIsICJYekZyendzY002R242Q0pEYzZ2Vks4QmtNbmZHOHZPU0tmcFBJWmRBZmRFl
iwgImdiT3NJNEVkcTJ4Mkt3LXc1d1BFemFrb2l5aFYxY1JEMEFUTjNvUUw5Sk0iLCAia
nN1OXlWdWx3UVFsaEZsTV8zSmx6TWFTRnpnbGhRRzBEcGZheVF3TFVLNCJdLCAiaXNzI
jogImh0dHBzOi8vaXNzdWVyLmV4YW1wbGUuY29tIiwgImlhdCI6IDE2ODMwMDAwMDAsI
CJleHAiOiAxODgzMDAwMDAwLCAic3ViIjogInVzZXJfNDIiLCAibmF0aW9uYWxpdGllc
yI6IFt7Ii4uLiI6ICJwRm5kamtaX1ZDem15VGE2VWpsWm8zZGgta284YUlLUWM5RGxHe
mhhVIlvIn0sIHsiLi4uIjogIjdDZjZKa1B1ZHJ5M2xjYndIZ2VaOGtoQXYxVTFPU2xlc
IAwVmtCSnJXWjAifV0sICJfc2RfYWxnIjogInNoYS0yNTYiLCAiY25mIjogeyJqd2siO
iB7Imt0eSI6ICJFQyIsICJjcnYiOiAiUC0yNTYiLCAieCI6ICJUQ0FFUjE5WnZ1M09IR
jRqNFc0dmZTVm9ISVAxSUxpbERsczd2Q2VHZW1jIiwgInkiOiAiWnhqaVdXYlpNUUdlV
ldLVlE0aGJTSWlyc1ZmdWVjQ0U2dDRqVDIGMkhaUSJ9fX0.OeQrinudSFTXNysz2NuNQ
rwWJv-P9gQ-Ce3wWEYZkxngeA4GKfPfApdNzBa40dH1urt8tXhW2WQI-I00v8teuw~Wy
JIbHVWNU9nM2dTTklJOEVZbnN4QV9BIiwgImZhbWIseV9uYW1lIiwgIkRvZSJd~WyJBS
ngtMDk1VlBycFR0TjRRTU9xUk9BIiwgImFkZHJlc3MiLCB7InN0cmVldF9hZGRyZXNzI
jogljEyMyBNYWluIFN0IiwgImxvY2FsaXR5IjogIkFueXRvd24iLCAicmVnaW9uIjogI
kFueXN0YXRlIiwgImNvdW50cnkiOiAiVVMifV0~WyIyR0xDNDJzS1F2ZUNmR2ZyeU5ST
jI3IiwgImdpdmVuX25hbWUiLCAiSm9obiJd~WyJsa2x4RjVqTVlsR1RQVW92TU5JdkNB
IiwgIlVTII0~eyJhbGciOiAiRVMyNTYiLCAidHlwIjogImtiK2p3dCJ9.eyJub25jZSI
6ICIxMjM0NTY3ODkwIiwgImF1ZCI6ICJodHRwczovL3ZlcmlmaWVyLmV4YW1wbGUub3J
nIiwgImlhdCI6IDE2OTgwNzc3OTAsICJfc2RfaGFzaCI6ICIzNHQ4dkNDX2NfdlZMbk9
hZEJ0d2g0ZEZ2QkVyU2w5ektPcXdtNmloVF9VIn0.ZIotfwqF9NUTRAShrd8jGSJEB6e
3Z3EKm-AD5udfzggxfK-IQM4TCKbHK81eV088YTKI-UfM7WSyQpx5wpNpZw
```


decode, check signature, check validity


```
{
  "_sd": [
    "fOBUSQvo46yQO-wRwXBcGqvnbKIueISEL961_Sjd4do"
  ],
  "iss": "https://issuer.example.com",
  "iat": 1683000000,
  "exp": 1883000000,
  "sub": "6c5c0a49-b589-431d-bae7-219122a9ec2c",
  "_sd_alg": "sha-256",
  "cnf": {
    "jwk": {
      "kty": "EC",
      "crv": "P-256",
      "x": "TCAER19Zvu3OHF4j4W4vfSVoHIP1ILilDls7vCeGemc",
      "y": "ZxjiWWbZMQGHVWKVQ4hbSIirsVfuecCE6t4jT9F2HZQ"
    }
  }
}
```


decode


```
["2GLC42sKQveCfGfryNRN9w", "address", {"street_address":
"Schulstr. 12", "locality": "Schulpforta", "region": "Sachsen-Anhalt", "country": "DE"}]
```


hash


```
fOBUSQvo46yQO-wRwXBcGqvnbKIueISEL961_Sjd4do
```


decode, check signature, check sd_hash, check nonce & aud


```
{
  "nonce": "1234567890",
  "aud": "https://verifier.example.org",
  "iat": 1721206049,
  "sd_hash": "GpLqgSVPUPqeetF7H4jxV3GkPn3BkT1pf3yY2I4G4ew"
}
```

Security Considerations (I)

Signature verification: Verifiers could verify the signature inadequately/partially and accept tampered credentials

Mitigating measures:

- Simple processing model, specified in detail in the standard
- Established algorithms enable the use of existing implementations

Manipulation of disclosures: If the hashes of the disclosures are not checked by the verifier, manipulated plaintext values could be accepted.

Mitigating measures:

- Design: Generally no assignment to the document possible without hash calculation
- Processing model specified in detail

Security Considerations (II)

Missing check of key binding: Verifiers could accept credentials without key binding

Mitigating measures:

- Different formats with/without key binding
- Differentiation in terminology
- Detailed discussion in the standard

Privacy Considerations

Unlinkability (“unlinkability”): Several presentations of the same credential can be traced back to the same person (due to the same hash values).

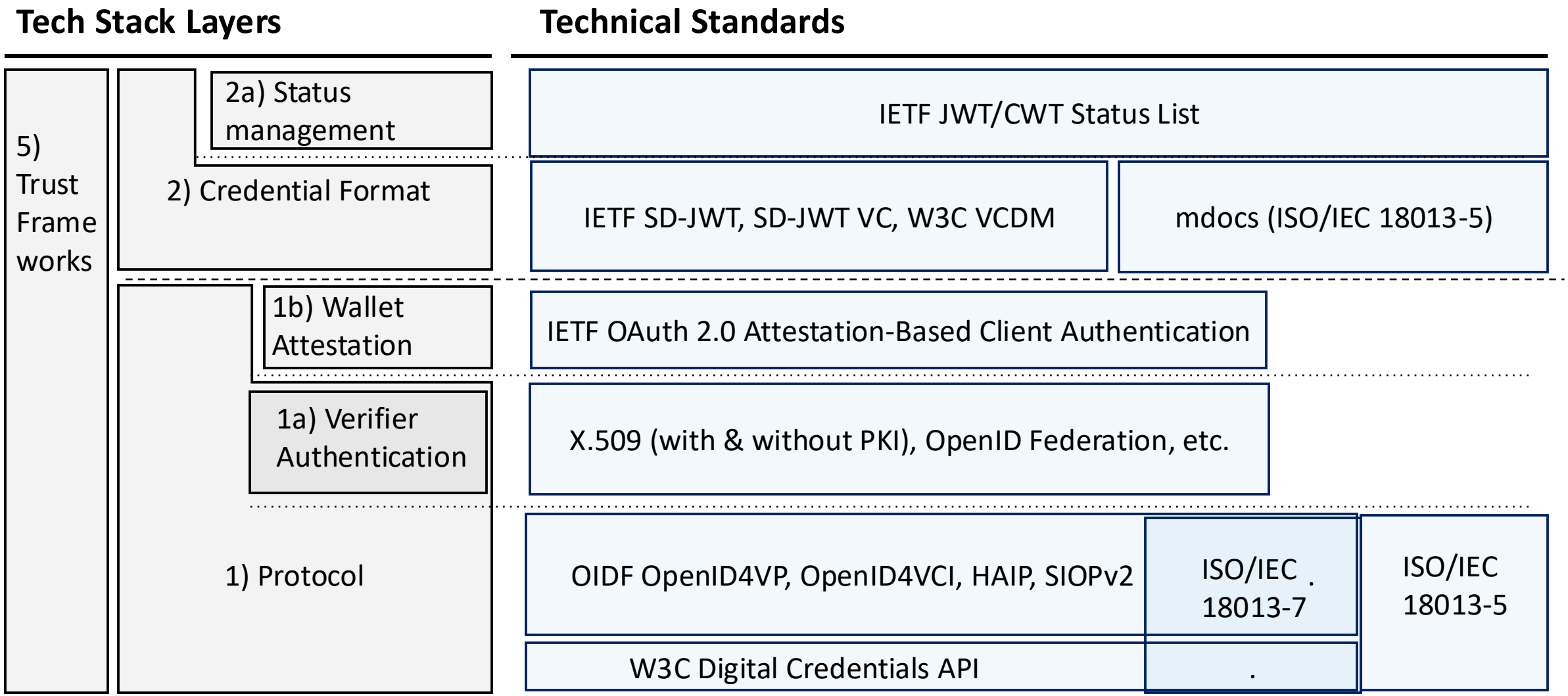
Mitigating measures:

- Single use: Credentials are always issued in groups - same data, different salt values. Each individual credential is then only used once.

THANK YOU!

Mail: kristina.yasuda@sprind.org

Technical stack



OID4VCI Privacy Considerations (II)

3. Information in the credential identifying a particular Credential Issuer, such as a Credential Issuer Identifier, issuer's certificate, or issuer's public key may reveal information about the End-User.
   - **Mitigation: A** group of organizations may elect to use a common Credential Issuer, such that any credentials issued by this Issuer cannot be attributed to a particular organization through identifiers of the Credential Issuers alone. A group signature scheme may also be used instead of an individual signature.
4. Leaking information about the Wallet to third parties when the Wallet reacts to a Credential Offer.
   - **Mitigation: T**he Wallet requiring user interaction or establish trust in the Issuer before fetching any credential_offer_uri or acting on the received Credential Offer.

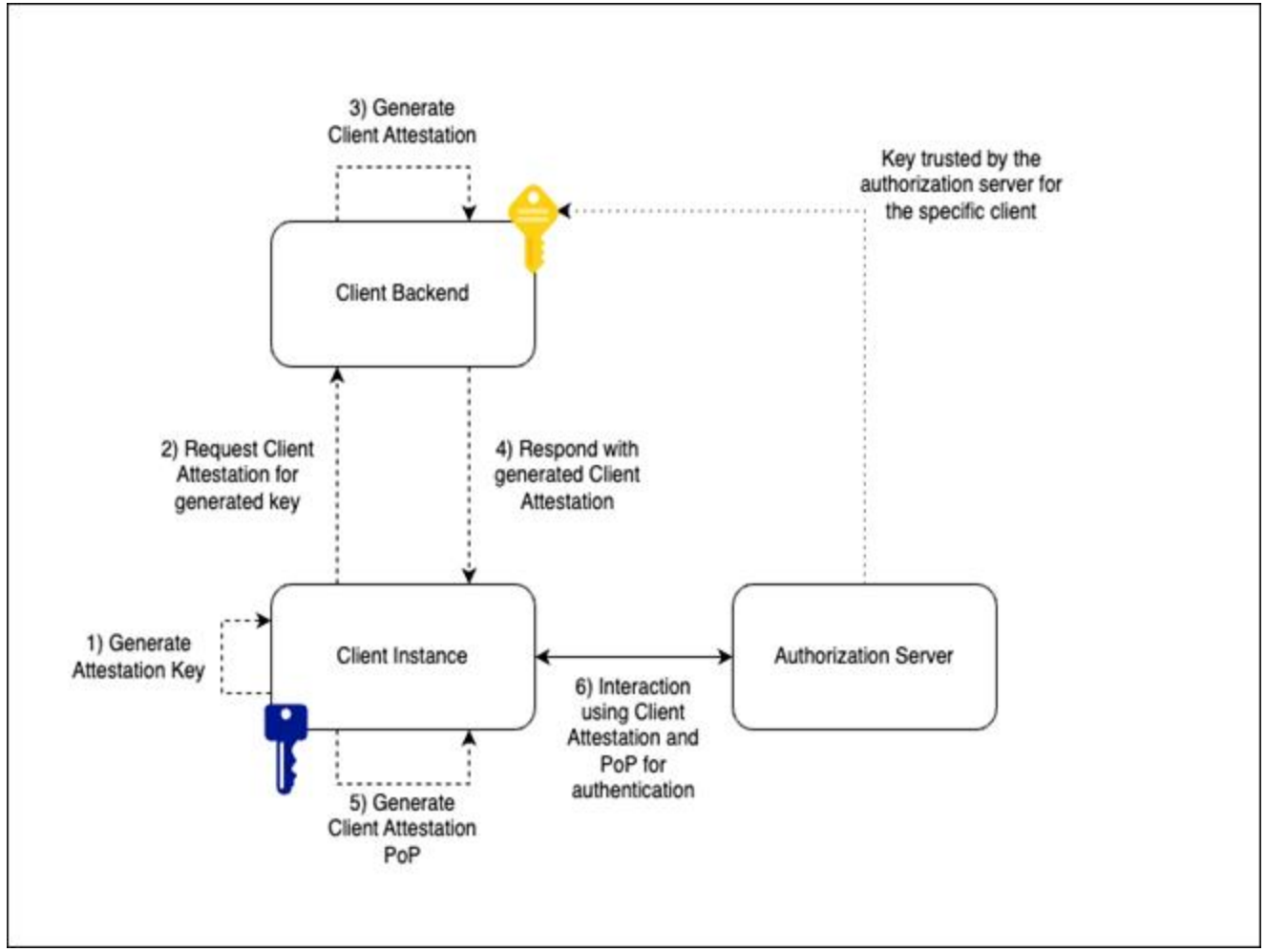
German OWASP Day 2024

Wallet Attestation

German OWASP Day 2024

Wallet Attestation Architecture

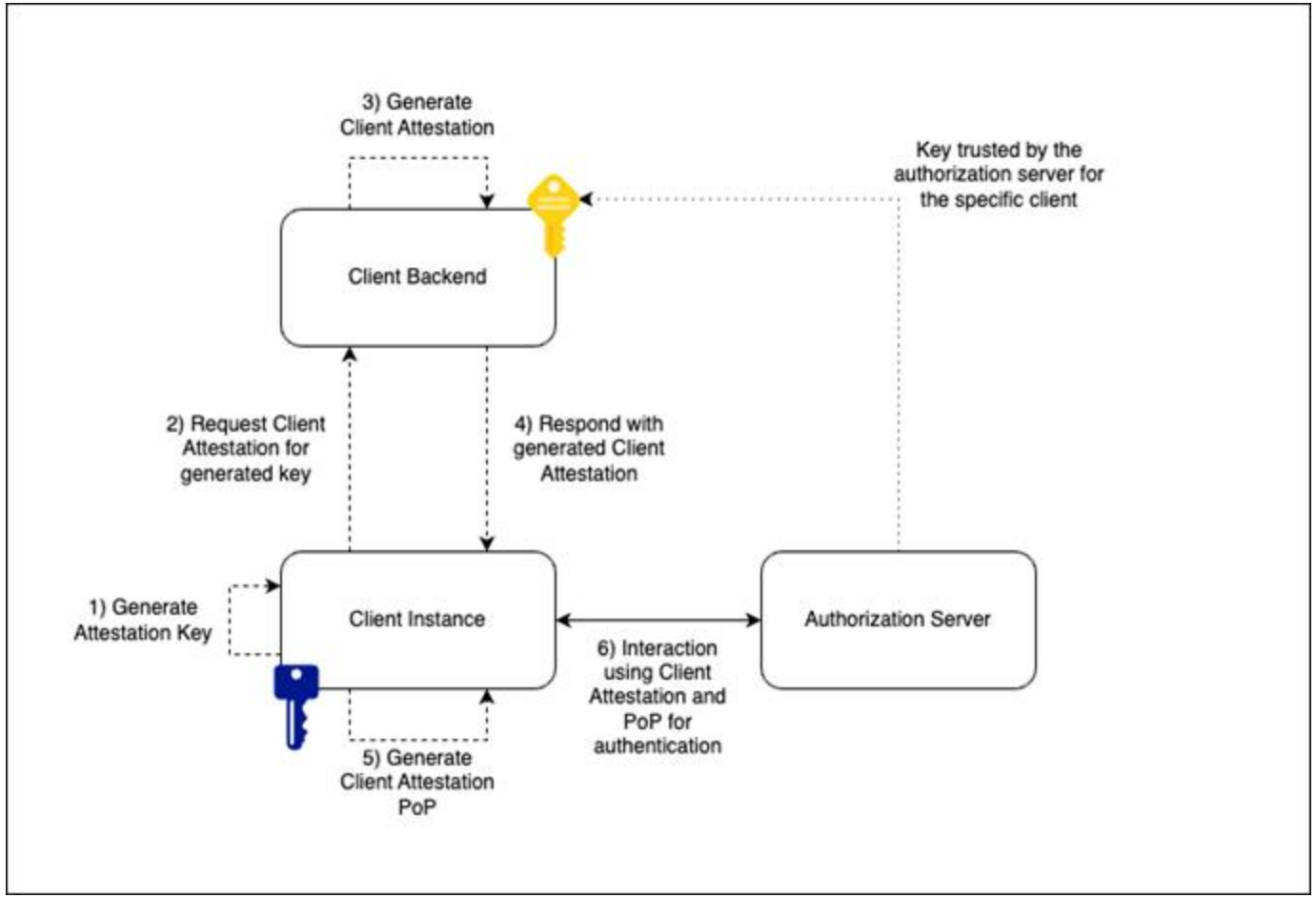
- Differentiate Client and Client Instance
- Client Backend attests a Client Instance
- Client backend may perform any number of security checks before issuing a key-bound attestation JWT to the client instance, however, steps 2 and 4 are out of scope
  - Mechanisms of authentication
  - Issuance process
- Trust mechanism for the Client Backend public key is out of scope



German OWASP Day 2024

Wallet Attestation Architecture

- 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



Example - Wallet Attestation

```
{
    "alg": "ES256",
    "kid": "11"
}
.
{
  "iss": "https://client.example.com",
  "sub": "https://client.example.com",
  "nbf": 1300815780,
  "exp": 1300819380,
  … //other claims, e.g. key type, user authentication, LoA
  "cnf": {
    "jwk": {
      "kty": "EC",
      "crv": "P-256",
      "x": "18wHLeIgW9wVN6VD1Txgpqy2LszYkMf6J8njVAibvhM",
      "y": "-V4dS4UaLMgP_4fY4j8ir7cl1TXlFdAgcx55o7TkcSA"
    }
  }
}
```

Key used to verify the Client Attestation PoP

OID4VC High Assurance Interoperability Profile with SD-JWT VC

OPENID4VC is a framework that requires profiling

- Interoperability requires instantiation of OpenID4VC with concrete
  - Definition of “Mandatory to Implement” elements of the protocols, i.e., grant types, response types, etc.
  - Definition of how wallet invocation is made (i.e., custom scheme, browser API, etc.)
  - Definition of authentication mechanisms for Verifiers and Wallets
  - Credential Format(s) with
    - issuer identification and key resolution
    - holder key binding
  - Crypto algorithms
- Instantiation designated as “Profile”

OID4VC High Assurance Interoperability Profile with SD-JWT VC

German OWASP Day 2024

Protocols

- OID4VP
  - custom scheme
  - credential profile
  - client id scheme
- OID4VCI
  - custom scheme
  - credential profile
  - wallet attestation scheme

- Authenticated issuer identifiers as basis for trust management
- Trust Management Mechanism can be defined on top

Credential profile: SD-JWT VC

- SD-JWT VC
  IETF OAuth WG adopted draft
  - crypto suites
  - issuer key resolution
- JWT/CWT Statuslist
  - crypto suites
  - issuer key resolution

Wallet Attestation Scheme

- Attestation based Client Authentication
  IETF OAuth WG adopted draft
  - crypto suites
  - issuer key resolution

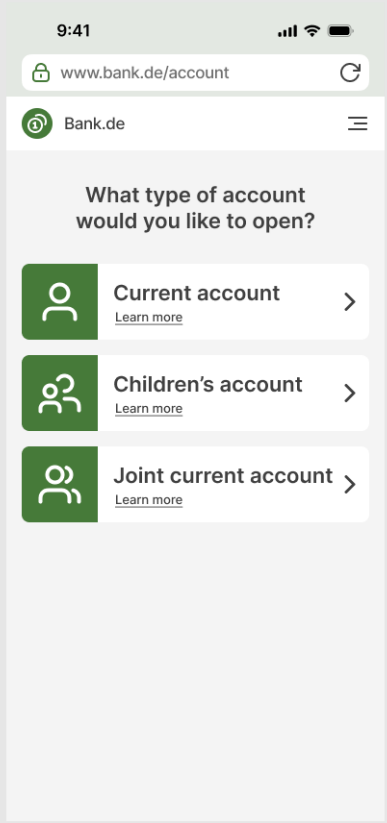
Basic Choices

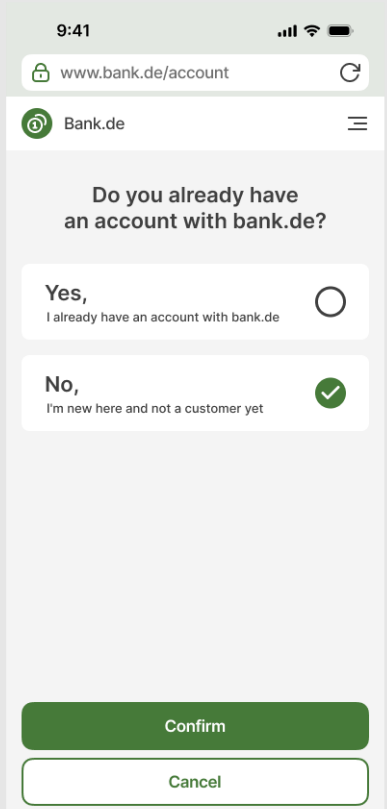
- Custom Scheme: **haip://**
- issuer key resolution: **x509**
- Verifier authentication: **x509**
- Crypto Suites: **P-256(secp256r1), SHA256**

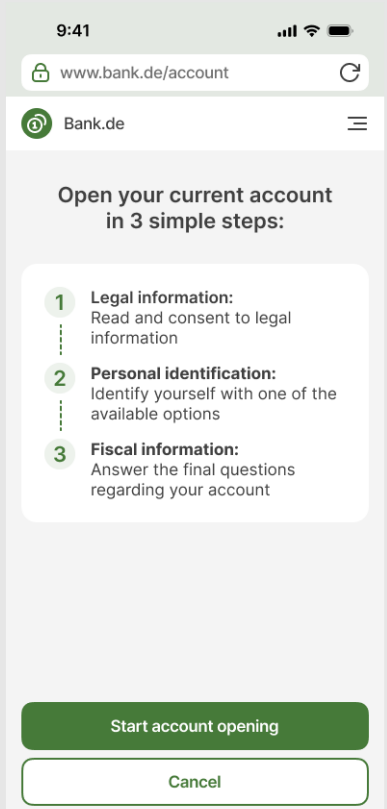
Note: Similar profile for ISO mdoc is being worked on in ISO/IEC 18013-7

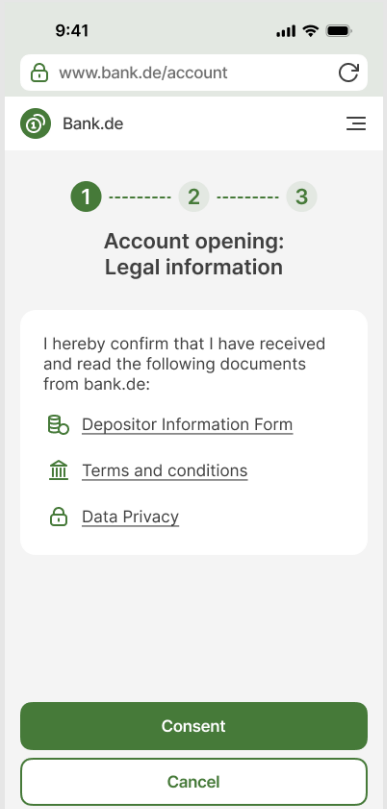
German OWASP Day 2024

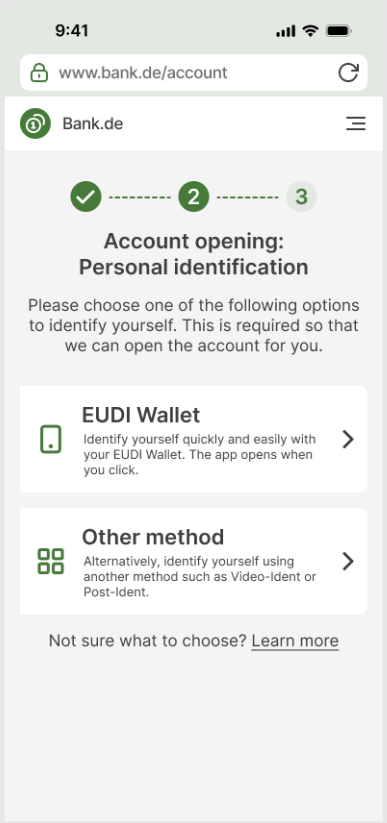
SAMPLE USER JOURNEY: BANK ACCOUNT OPENING (1/2)

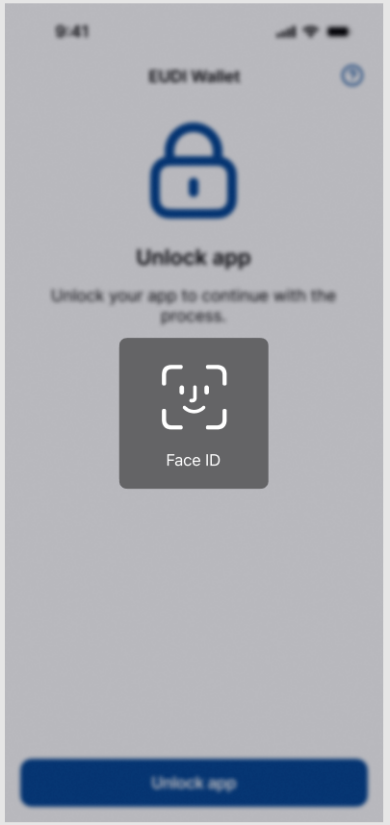




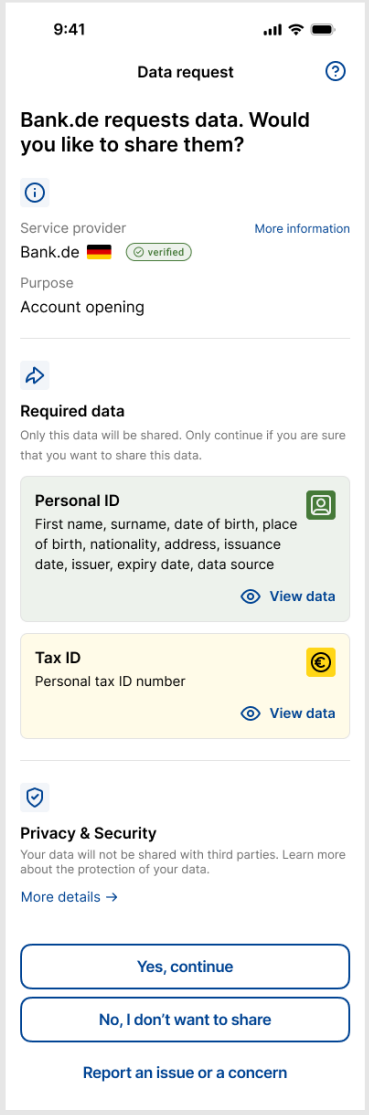


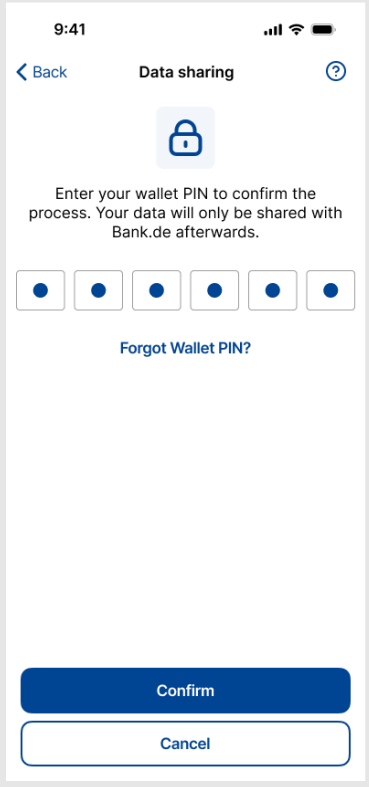


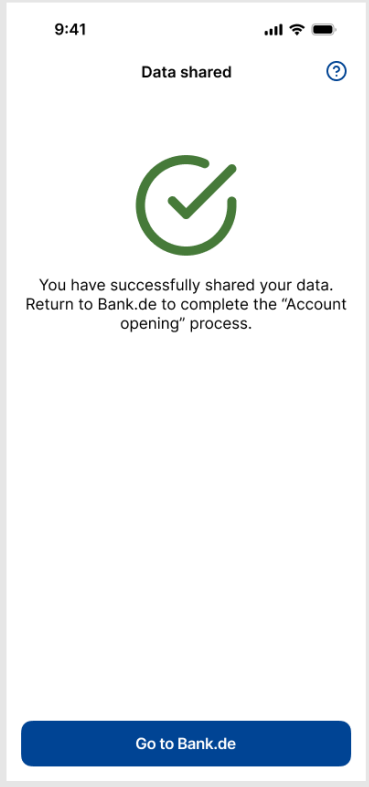


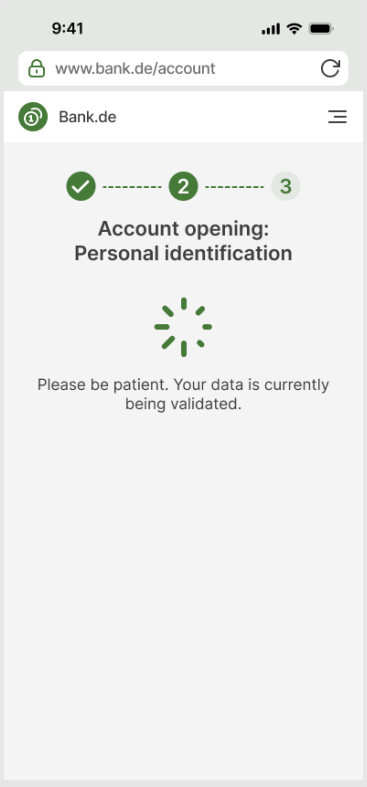


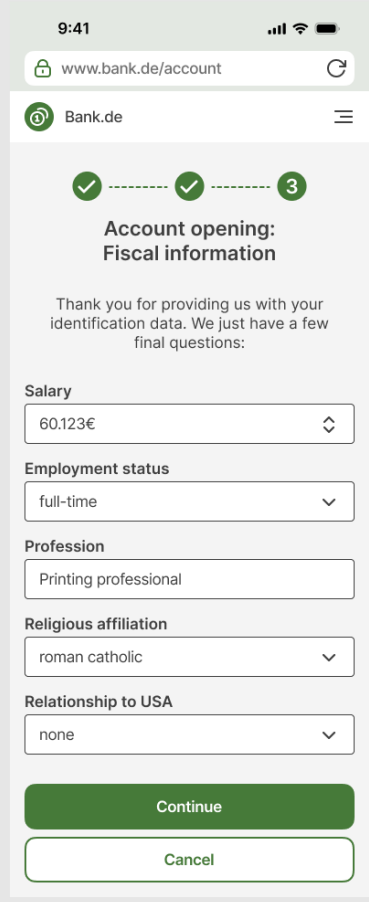
SAMPLE USER JOURNEY: BANK ACCOUNT OPENING (2/2)

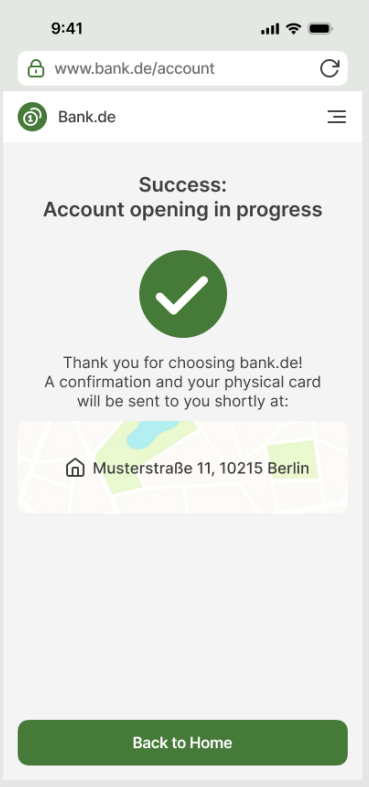








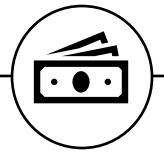




VALUE PROPOSITION FOR THE RELYING PARTIES/VERIFIERS

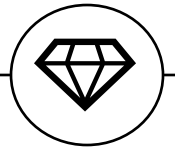
Fraud Reduction

Safeguard **personal information** from **cybercrime** and **fraud**



Better Conversions

Seamless, fast and simple ID verification and authentication



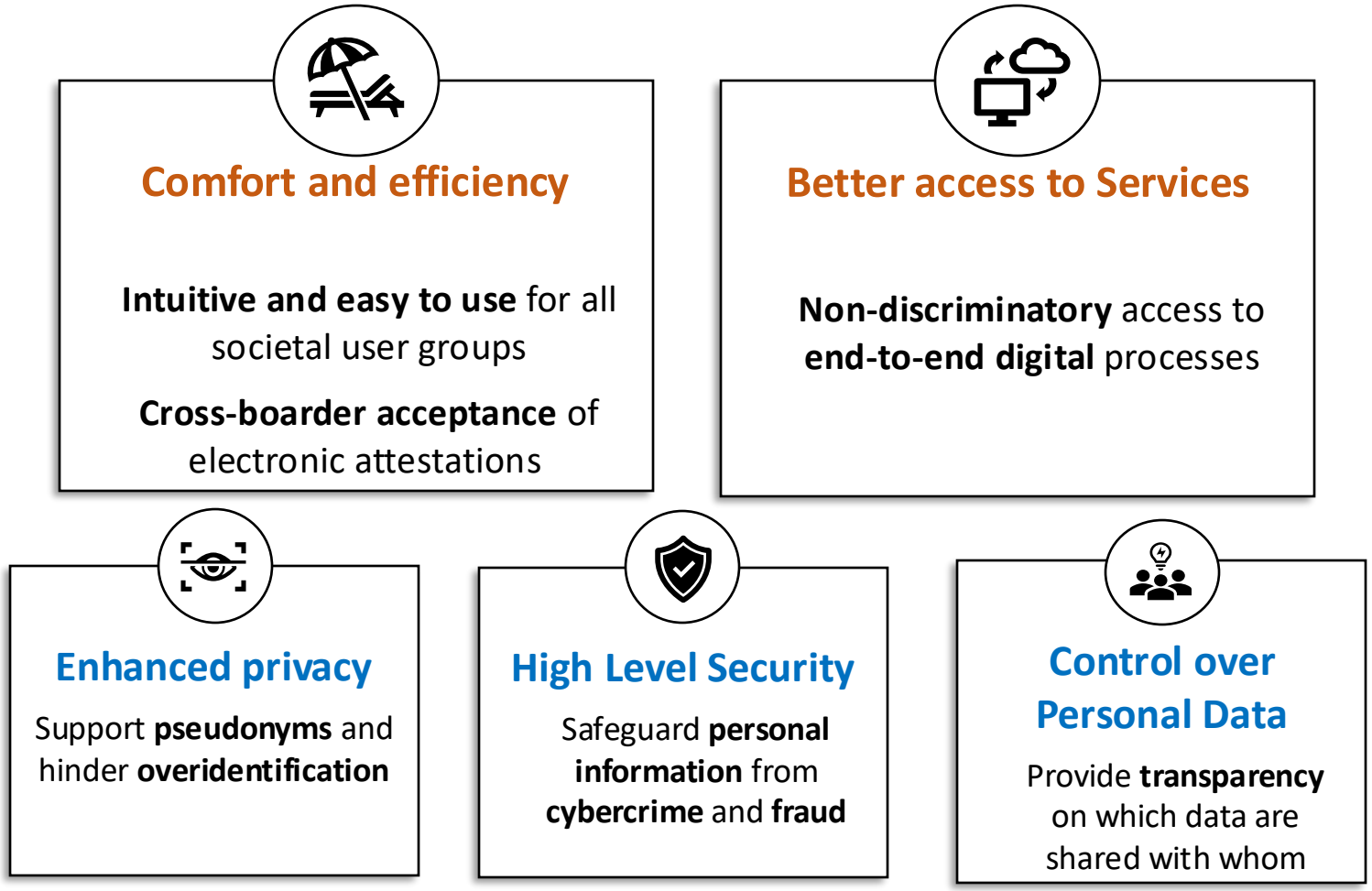
Increase Customer Satisfaction

Create economic value by enabling **new digital products and services**

Legal and regulatory compliance

Privacy and **Security** at the heart of EUDIW

VALUE PROPOSITION FOR THE USERS



HYPOTHESIS: EUDIW USAGE IN ECOMMERCE

1 Buyers & Sellers Onboarding

- KYC & age verification
- Proofs of specific attestations (ex. students, doctors, social benefits, etc.)
- EUDIW is for natural and legal persons
- Free of charge QES to sign contracts

2 Customer Journey

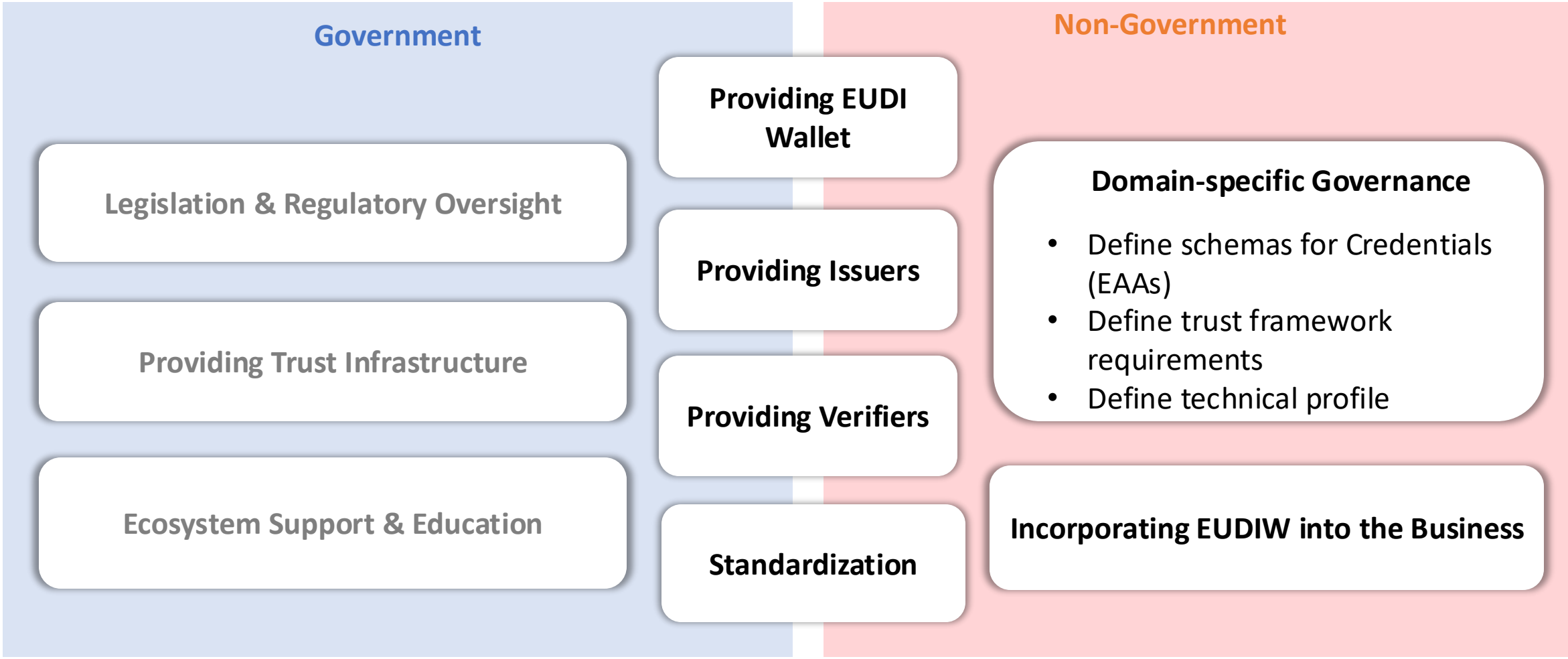
- Seamless, fast and simple authentication
- Boost conversion rates
- Increased consumer trust

3 Checkout

- Reduce fraud
- Additional checks for high-value transactions
- Getting discounts, and services based on specific attestations

SPRIN-D

RESPONSIBILITY OF NON-GOVERNMENT ACTORS

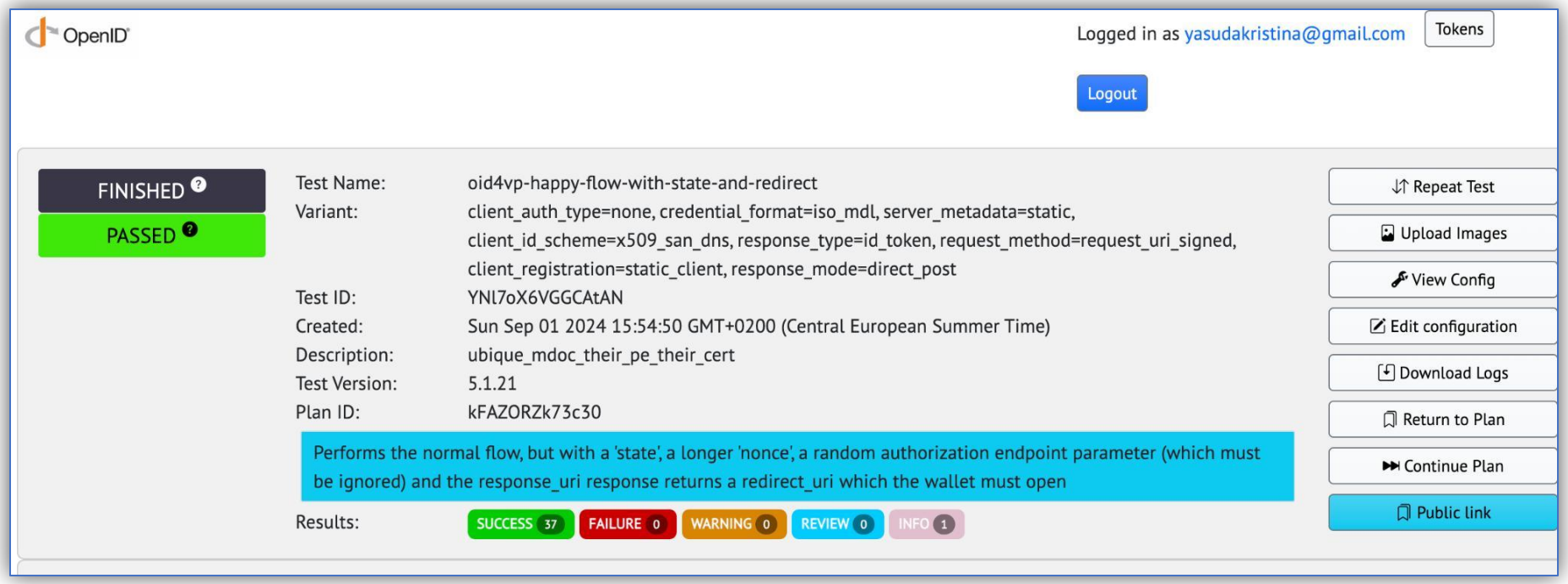


SPRIN–D

CREDENTIALS AVAILABLE IN THE MARKET (POTENTIAL LARGE SCALE PILOT)

Use-Case	Credentials		
Use Case 1: eGov Services: This aims to provide citizens with a secure digital ID for online citizenship procedures.	PID (digitizing National ID)	Power of Attorney	
Use Case 2: Bank Account Opening: It enables the use of a digital ID to open bank accounts across Europe.		Tax ID Attestation	Certificate of Residence
Use Case 3: SIM Card Registration: This use case supports the online activation of mobile contracts.		Verified Phone Number	
Use Case 4: Mobile Driving License: This digital version of the driving license will be recognized across Europe.		Mobile Driving License	
Use Case 5: Qualified eSignature: It allows for the remote signing of documents across Europe using a secure digital signature.			
Use Case 6: ePrescription: A digital method for managing prescriptions throughout Europe		Health Insurance Number	

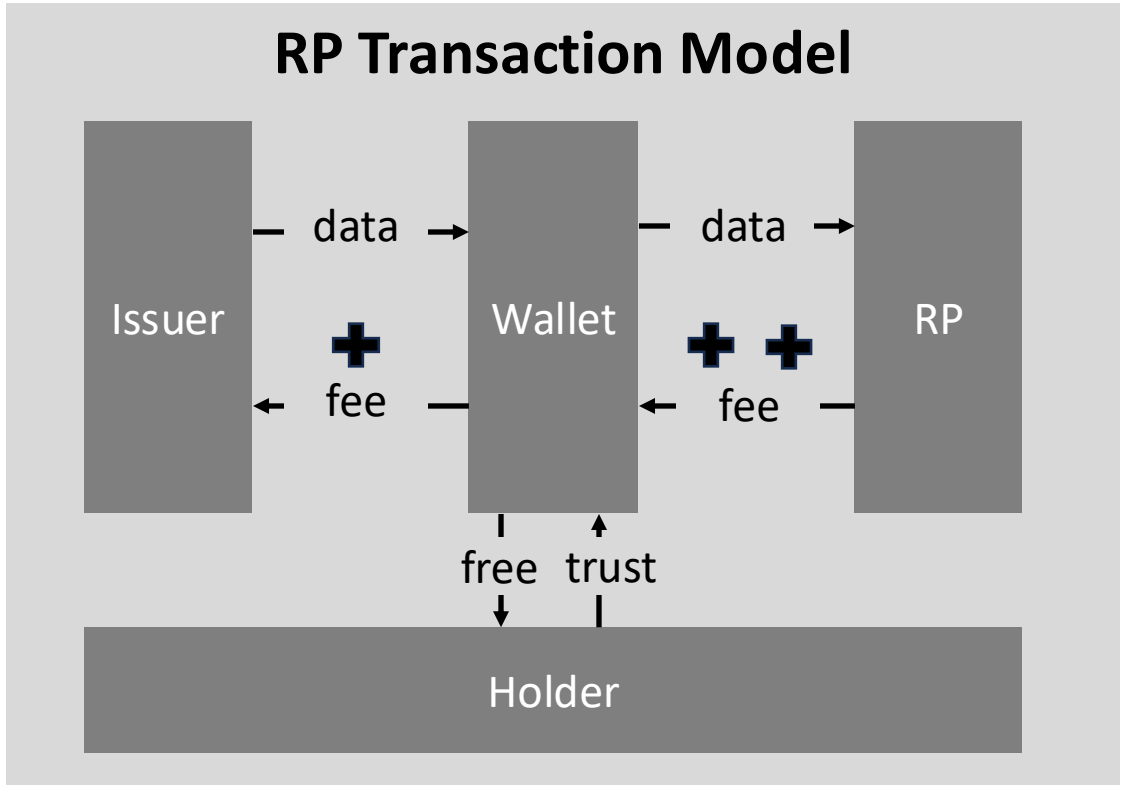
Ensuring interoperability when implementing



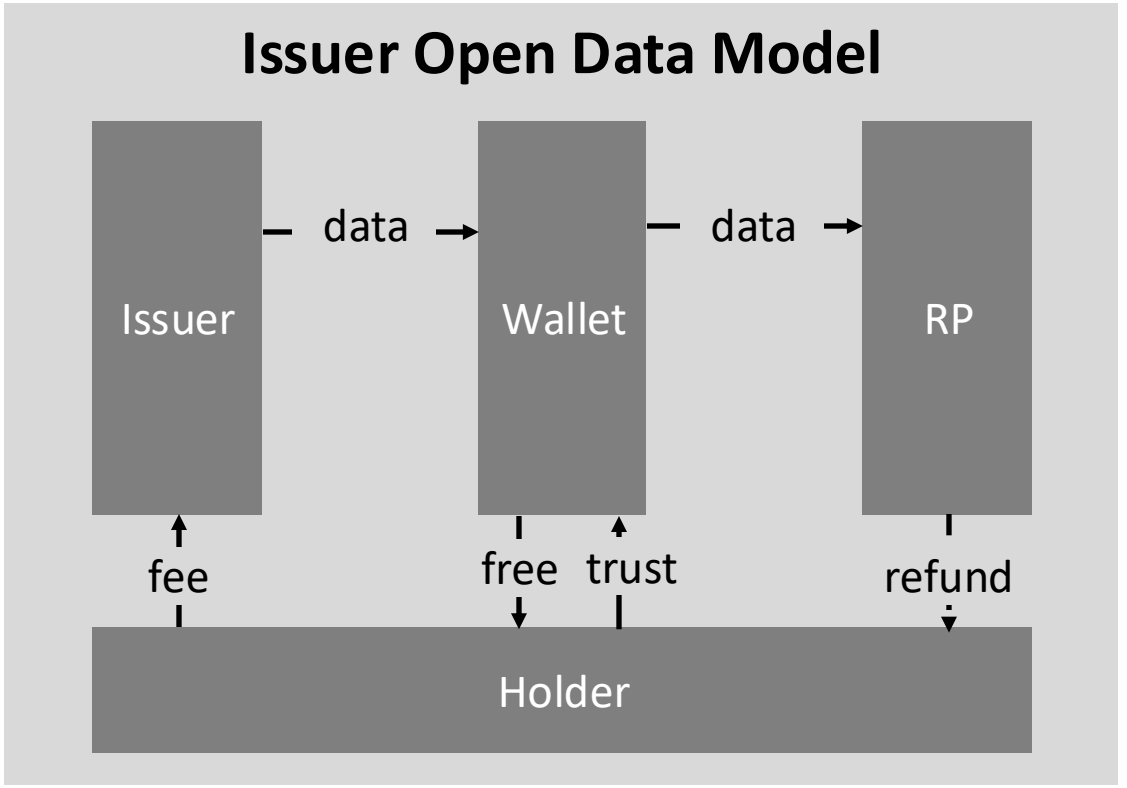
A lot of tools available already. Including…

- OpenID Foundation conformance tests are available to ensure interoperability among various implementations and help during development

We have identified two schools of thought regarding operating models



project focus



eIDAS ARTICLE 5a

4. European Digital Identity Wallets shall enable the user, in a manner that is user-friendly, transparent, and traceable by the user, to:

(a) securely request, obtain, select, combine, store, delete, share and present, under the sole control of the user, person identification data and, where applicable, in combination with electronic attestations of attributes, to authenticate to relying parties online and, where appropriate, in offline mode, in order to access public and private services, while ensuring that selective disclosure of data is possible;

(b) generate pseudonyms and store them encrypted and locally within the European Digital Identity Wallet;

(c) securely authenticate another person's European Digital Identity Wallet, and receive and share person identification data and electronic attestations of attributes in a secured way between the two European Digital Identity Wallets;

(d) access a log of all transactions carried out through the European Digital Identity Wallet via a common dashboard enabling the user to:

  (i) view an up-to-date list of relying parties with which the user has established a connection and, where applicable, all data exchanged;

  (ii) easily request the erasure by a relying party of personal data pursuant to Article 17 of the Regulation (EU) 2016/679;

  (iii) easily report a relying party to the competent national data protection authority, where an allegedly unlawful or suspicious request for data is received;

(e) sign by means of qualified electronic signatures or seal by means of qualified electronic seals;

(f) download, to the extent technically feasible, the user's data, electronic attestation of attributes and configurations;

(g) exercise the user's rights to data portability.

SPRIN-D

THE GERMAN ARCHITECTURE & CONSULTATION PROCESS FOR EUDI WALLETS

ARCHITECTURE

Building a **concept for a digital wallet ecosystem** developed by **experts** and the **public**

The concept is **developed in iterations** with a focus on different parts of the ecosystem

ARCHITECTURE PROPOSAL VERSION 2

Considerations on **data protection** and **security** issues

Establishing **operational methods**

Design and **governance** of operating models

FULL PROPOSAL

CONSULTATION

Three workshops and **five Open Online Consultations** have taken place

Over **220 "Issues" submitted** on OpenCoDE

Civil society groups are **actively approached**

GET INVOLVED

SPRIND FUNKE

„Funke" aims to track down **ground-breaking innovations** by teams **competing** with each other

In this **three-stage competition**, the **team's work is evaluated** and **only the best** remain in the competition

THREE KEY FEATURES

Issuance and Presentation of **PIDs**

Issue and submission of **EAAs**

Pseudonymous login, QES and payment use-cases