



FIDO2 Relying Party

cnlab Herbsttagung 2025



Marc Bütikofer
Head of Innovation
Security Solutions Airlock

AIRLOCK[®]

SECURE ACCESS HUB

WAAP



DoS
Protection



Bot
Defense



Web Application
Firewall



API
Security
Gateway

IAM



Adaptive
Authentication



Self
Service



Identity
Management



Standard APIs
Automation

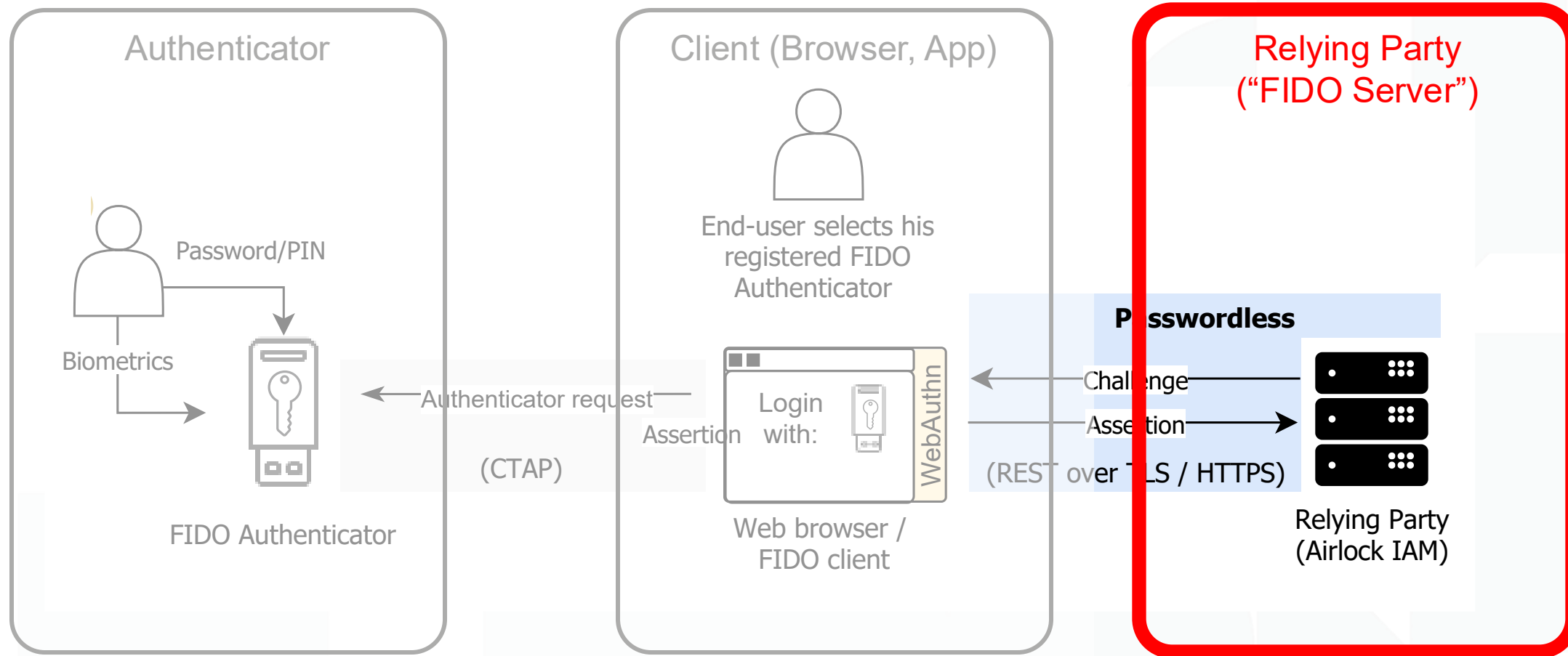


Continuous Adaptive Trust

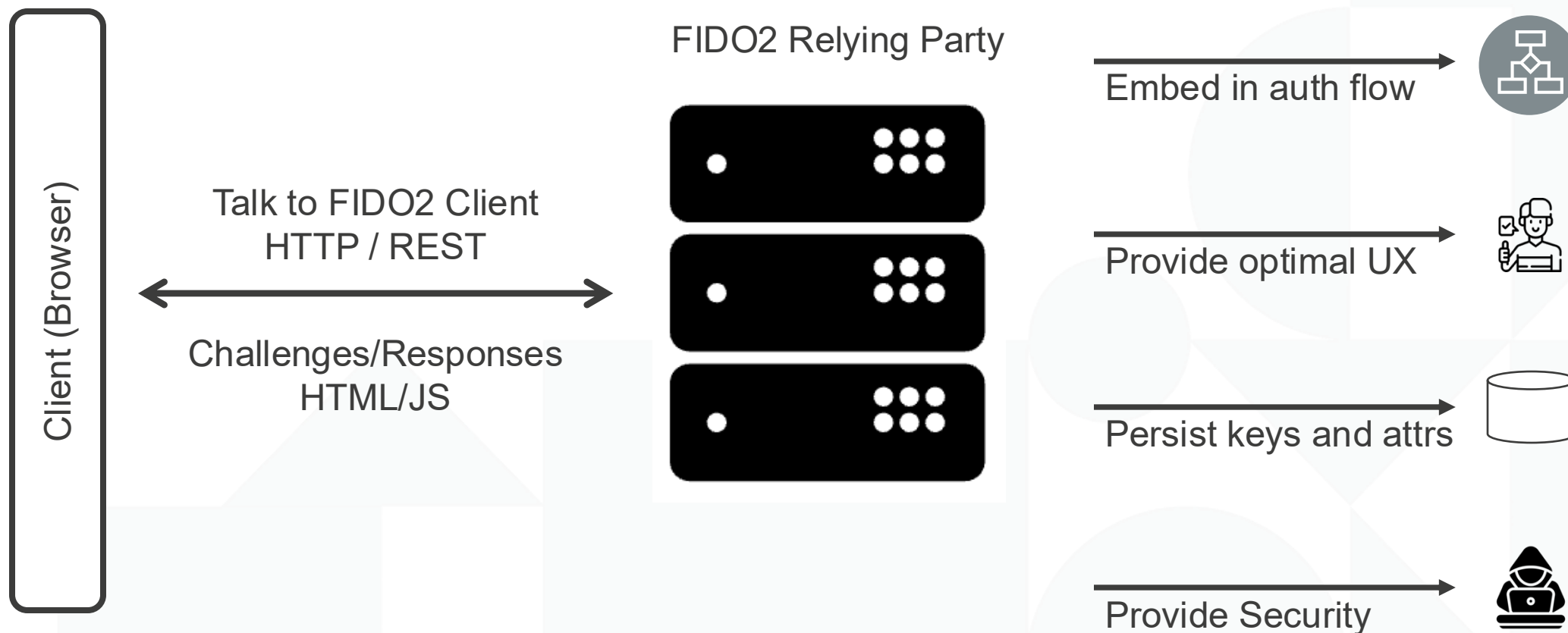
Single Sign-On



FIDO Relying Party

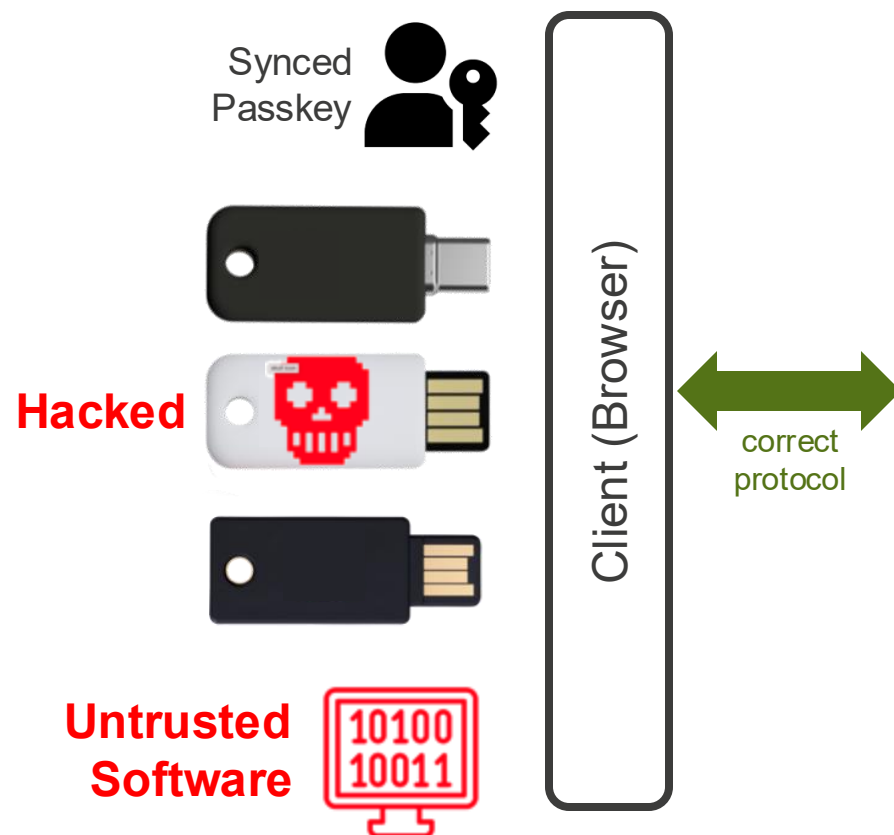


Relying Party – Main Tasks

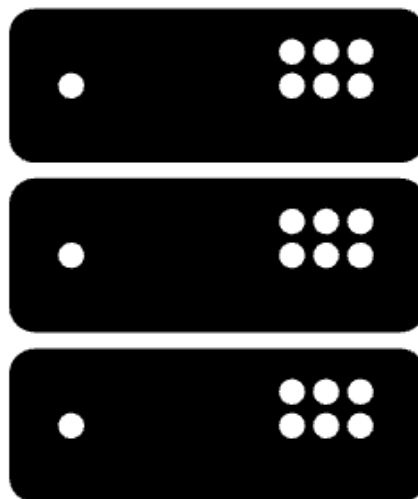


BYOA, FIDO & Attestations

Bring your own authenticator



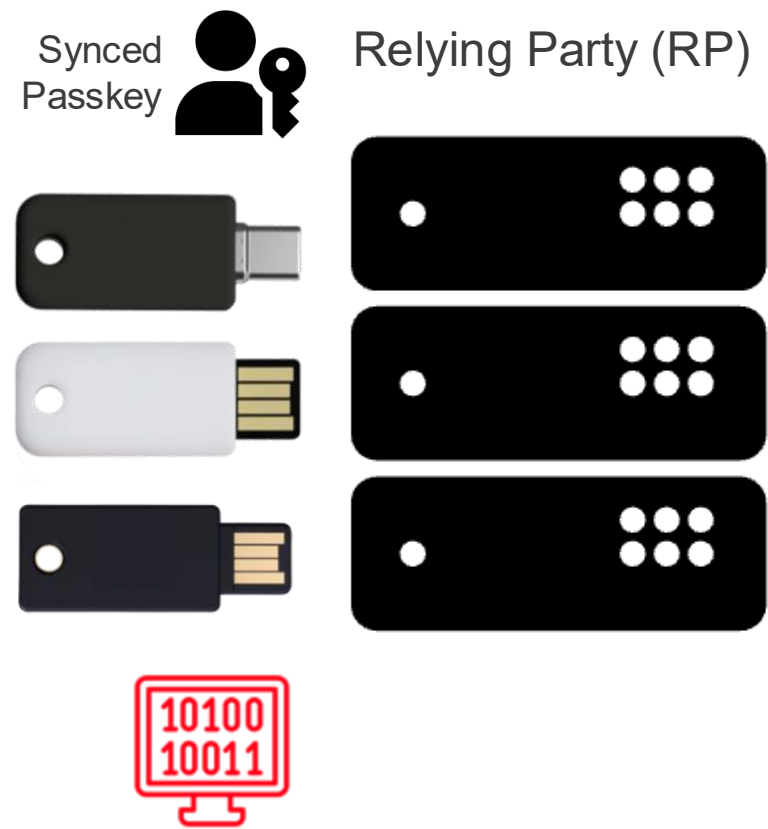
Relying Party (RP)



- Trust the authenticator?
- Vulnerable key models?
- Synced passkeys?
- Untrusted software?

How can RP be sure of the used authenticator type?

FIDO Attestations



FIDO2 Attestation (during key registration)

- Process for verifying the **authenticity of the authenticator** (not the user)
- **Provides information** about the registered authenticator
- **Different types** of attestation (Basic, Self, AttCA/AnonCA, Enterprise, None)
- Attestation object may or may not be **signed**
- Attestations may be a **privacy** issue

FIDO Attestations – Apple iCloud passkey example



```
"attestationObject": {  
  "fmt": "none",  
  "attStmt": {},  
  "authData": {  
    "rpIdHash": "e9a88899f0e3ab9b98f693e",  
    "flags": {  
      "userPresent": true,  
      "reserved1": false,  
      "userVerified": true,  
      "backupEligibility": true,  
      "backupState": true,  
      "reserved2": false,  
      "attestedCredentialData": true,  
      "extensionDataIncluded": false  
    },  
    "signCount": 0,  
    "attestedCredentialData": {  
      "aaguid": "fbfc3007-154e-4ecc-8c0b-6e020557d7bd",  
      "credentialId": "ca05e1b89caa87c1d34971d234b",  
      "credentialPublicKey": {  
        "kty": "EC",  
        "alg": "ECDSA_w_SHA256",  
        "crv": "P-256",  
        "x": "MJeoVq+Cv3wzhxxhrTLZ9FmndB/hUv1b0PN",  
        "y": "C7mn4+5klzc3WyhXuJQTeKBA3zWhmQNTu7ZHdK6mDOU="
```

No signature

Information not verifiable → only use for stats or UX but **not for security**

flags

Useful information on registration for RP

aaguid: fbfc3007-1...

Used to look up authenticator model and its properties

FIDO Attestations – YubiKey 5 Example



```
"attestationObject": {  
  "fmt": "packed",  
  "attStmt": {  
    "alg": -7,  
    "sig": "3046022100a69a62b15c...",  
    "x5c": [  
      "308202bd308201a5a0030201...",  
    ]  
  },  
},  
"authData": {  
  "rpIdHash": "e9a88899f0e3ab9b98f693e0738...",  
  "flags": {  
    "userPresent": true,  
    "reserved1": false,  
    "userVerified": true,  
    "backupEligibility": false,  
    "backupState": false,  
    "reserved2": false,  
    "attestedCredentialData": true,  
    "extensionDataIncluded": false  
  },  
  "signCount": 1,  
  "attestedCredentialData": {  
    "aaguid": "2fc0579f-8113-47ea-b116-bb5...",  
    "credentialId": "35fe9a83e634421e3da8b...",  
    "credentialPublicKey": {  
      "kty": "EC",  
      "alg": "ECDSA_w_SHA256",  
      "crv": "P-256",  
      "x": "cw9J1L7oVemaPtjkJhJ+c1ejTtCUpfCTKRvf2x/GnKM=",  
      "y": "L5PAg02WhUrEwoT3tdXq/mwWgETsF8//bPsNg1kg8cU="
```

→ Attestation data is **verifiable**
→ Eligible for **security-related decisions**

sig: 3046022100a69a62b15c...
Digital signature on attested data

x5c:
308202bd308201a5a0030201...
Issuer certificate chain

Attestation signature verification

1. Attestation signature verification

Standard, well-known crypto operations

2. Establish in trust chain

Signature is useless if cert issuer is not trusted.

Option 1:

Configure trust roots in RP

→ works for small “closed” set of authenticator models.

Option 2:

Use FIDO MDS

→ next Slides

```
"attestationObject": {
  "fmt": "packed",
  "attStmt": {
    "alg": -7,
    "sig": "3046022100a69a62b15c0a1824c975c44dc40bce7e",
    "x5c": [
      "308202bd308201a5a00302010202041e8f8734300d06092"
    ]
  },
  "authData": {
    "rpIdHash": "e9a88899f0e3ab9b98f693e0738a37725a3ef",
    "flags": {
      "userPresent": true,
      "reserved1": false,
      "userVerified": true,
      "backupEligibility": false,
      "backupState": false,
      "reserved2": false,
      "attestedCredentialData": true,
      "extensionDataIncluded": false
    },
    "signCount": 1,
    "attestedCredentialData": {
      "aaguid": "2fc0579f-8113-47ea-b116-bb5a8db9202a",
      "credentialId": "35fe9a83e634421e3da8b6c07716f61",
      "credentialPublicKey": {
        "kty": "EC",
        "alg": "ECDSA_w_SHA256",
        "crv": "P-256",
        "x": "cw9J1L7oVemaPtjkJhJ+c1ejTtCUpfCTKRvf2x/G",
        "y": "L5PAg02WhUrEwoT3tdXq/mwWgETsF8//bPsNg1kg"
      }
    }
  },
}
```

FIDO2 MDS (Meta Data Service)



- Large JWT (signed JSON structure)
- Many attributes of > 300 FIDO authenticator models
- Attestation certificate roots
- FIDO Certification level
- Key protection type
- + many more attributes

```
{
  "authId": "2fc8579f-8113-47ea-b116-bb5ab09202a",
  "metaData": {
    "legalHeader": "Submission of this statement and retrieval and use of this statement",
    "authId": "2fc8579f-8113-47ea-b116-bb5ab09202a",
    "description": "Public Key Series with NFC",
    "authenticatorVersion": 328706,
    "protocolFamily": "FIDO2",
    "schema": 3,
    "upv": {
      "major": 1,
      "minor": 0
    },
    "authenticationAlgorithm": {
      "sec256r1_ecdsa_sha256_raw",
      "ed25519_ecdsa_sha256_raw"
    },
    "publicKeyAndEncodings": {
      "cose"
    },
    "attestationTypes": {
      "basic_full"
    },
    "userVerificationDetails": {
      {
        "userVerificationMethod": "passcode_external",
        {
          "userVerificationMethod": "presence_internal",
          "base": 64,
          "minLength": 4,
          "maxRetries": 8,
          "blackoutdown": 0
        }
      },
      {
        "userVerificationMethod": "passcode_external",
        {
          "userVerificationMethod": "presence_internal",
          "base": 64,
          "minLength": 4,
          "maxRetries": 8,
          "blackoutdown": 0
        }
      },
      {
        "userVerificationMethod": "none"
      }
    },
    "keyProtection": {
      "hardware",
      "secure_element"
    },
    "matcherProtection": {
      "on_chip"
    },
    "cryptoStrength": 128,
    "attestationHint": {
      "external",
      "wired",
      "wireless",
      "nfc"
    },
    "display": {
      "attestationRootCertificates": {
        "MIDHjCCAggAwIBAgIIGBIB79zMBgqhkhIGdWBAQIFADAUMSwKgTDOVOOEYhZdKjpt728yTJGIFj",
        "icon": "data:image/png;base64,iVBORw0KGgoAAANSURUUAACAAAAAFCAYAAACGVsHMAAAWNS"
      },
      "authenticatorGetInfo": {
        "version": {
          "u2f_v2",
          "FIDO_2_0",
          "FIDO_2_1_PSE"
        },
        "extensions": {
          "credProtect",
          "hmac-secret"
        },
        "authId": "2fc8579f811347eab16bb5ab09202a",
        "options": {
          "platform": false,
          "rk": true,
          "clientPin": false,
          "up": true
        },
        "credentialMgmtPreview": true
      },
      "maxMsgSize": 1200,
      "pinAuthProtocols": {
        2,
        1
      },
      "maxCredentialCountInList": 0,
      "maxCredentialIdLength": 128,
      "transports": {
        "nfc",
        "usb"
      },
      "algorithms": {
        {
          "type": "public-key",
          "alg": -7
        },
        {
          "type": "public-key",
          "alg": -8
        }
      },
      "minPINLength": 4,
      "firmwareVersion": 328706
    },
    "statusReports": {
      {
        "status": "FIDO_CERTIFIED_L1",
        "effectiveDate": "2020-05-12",
        "certificationDescription": "FIDO Key S NFC Series",
        "certificationNumber": "FIDO2020190826082",
        "certificationPolicyVersion": "1.1.1",
        "certificationRequirementsVersion": "1.3"
      },
      {
        "status": "FIDO_CERTIFIED",
        "effectiveDate": "2020-05-12"
      }
    },
    "timeOfLastStatusChange": "2020-05-12"
  },
}
```




“FIDO2 **attestation** – especially using the MDS – is a **must** for meaningful RP policies in **high-security** use-cases.”

Anything else?

More crucial relying party tasks

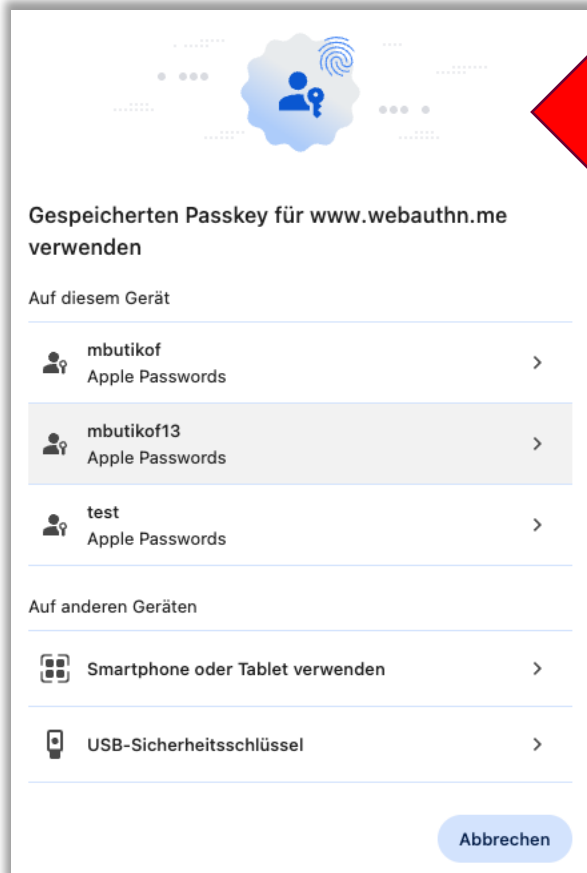




Flexibility

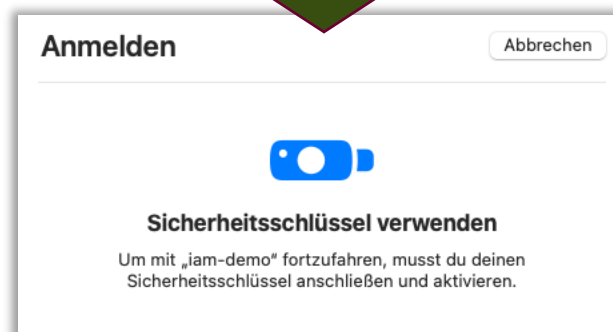
- ✓ Flexible Authentication workflows
- ✓ Allow coexistence of passkeys with other auth factors
- ✓ Migrate end-users from passwords to passkeys
- ✓ Self-services for passkeys
- ✓ Flexible FIDO policy options

Optimize User Experience (UX)



Bad UX:
Too many choices

Good UX:
Relying party
limits choices



- This is just one of many example
- The FIDO2 relying party has many options to influence end-user UX
- Passkey UX can be challenging for unexperienced end-users
- Optimized UX makes the the day in consumer use-cases



KEY LEARNINGS

- Relying party is crucial for security
- Verification of authenticator authenticity plays a major role
- Achieving good UX is not that easy
- IAM (RP) must be flexible and mature



AIRLOCK®

Adopting passkeys? We know how!

Talk to us.

www.airlock.com
marc.buetikofer@ergon.ch
www.linkedin.com/in/bueti/



WAAP & CIAM since
more than 20 years

650 Airlock customers in
30 countries

> 50'000 protected
applications

Saas and on-prem