WORKSHOP GARR 2023

NET MAKERS

Digital Identity Wallet

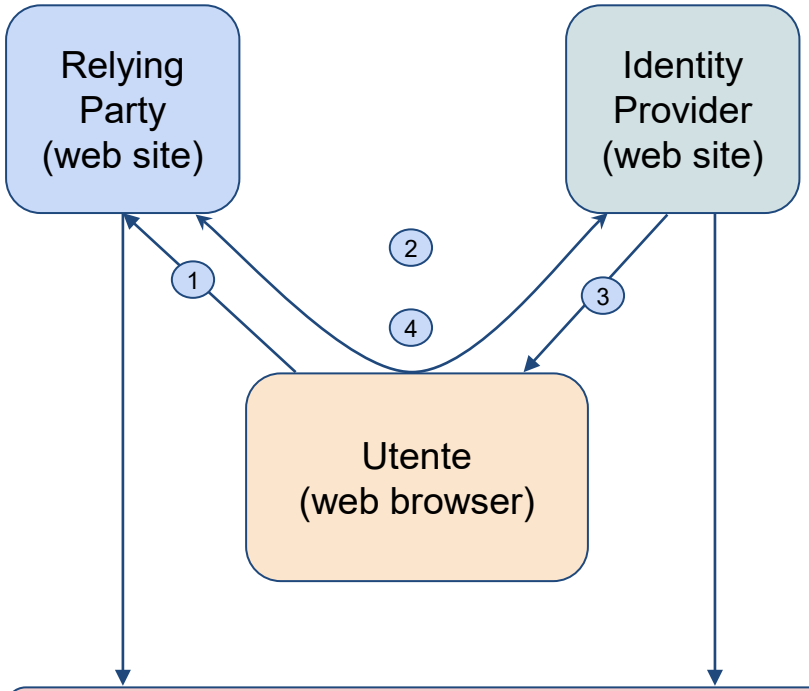
il nuovo paradigma di autenticazione

Giuseppe De Marco

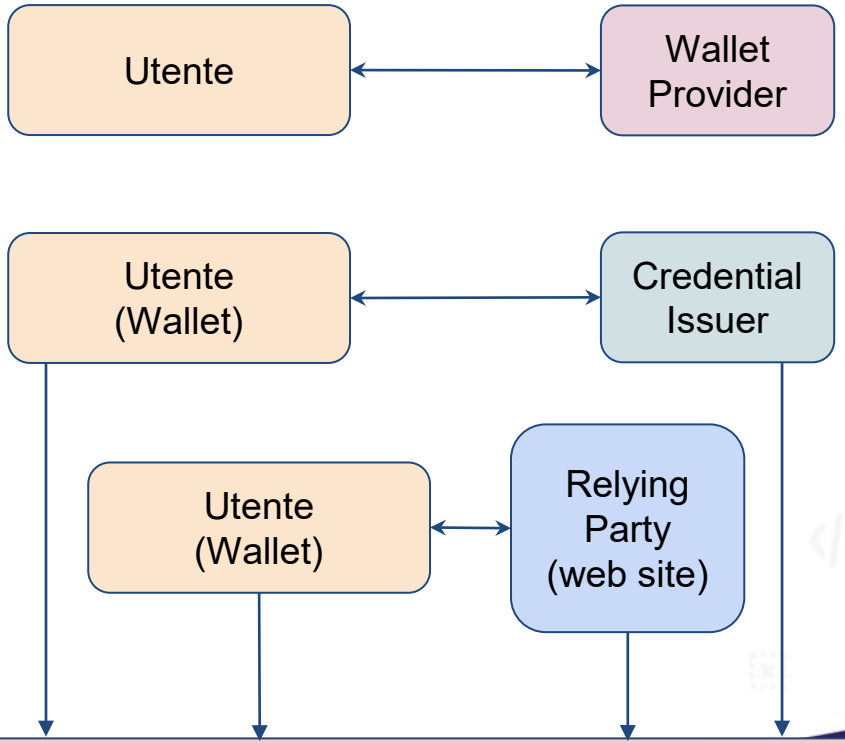
Digital Identity Wallet

- I componenti architetturali dell'ecosistema Identity Wallet
- Differenze con i sistemi di identità digitali attuali basati su SAML2 e OIDC
- Benefici offerti dall'uso del Wallet
- Dispositivi normativi e contesti di adozione
- Formati delle credenziali digitali
- Protocolli e specifiche tecniche, overview ad altissimo livello dei flussi
- Elementi di Trust Model, Sicurezza e garanzie di affidabilità
- Open Points e Rischi

Flusso Saml2/OIDC



Wallet



Definizioni dei partecipanti affidabili e aderenti al Trust Framework
(Federation, Trusted Lists)

Benefici all'uso dei Digital Identity Wallet

- Propone di **migliorare la Privacy**
  - Il credential Issuer non conosce l'uso delle credenziali da parte dell'utente
    - Rischio: i meccanismi di controllo delle revoche possono impattare negativamente
- **Selective Disclosure** (possibilità di rilasciare sub set di attributi contenuti in una credenziale)
  - Requisito (non più scelta implementativa come in IDP SAML2/OIDC)
- Autenticazioni remote (web) e in prossimità fisica, online e **offline**
- ~~Predicati~~ -> algoritmi avanzati non ancora standard. Alternative possibili:
  - attestazioni statiche opache, eg: **age_over_18** (boolean).
  - Linked hashes possibili con algoritmi standard, ma non adottati in nessuna specifica.
- Maggiore controllo sul ciclo di vita delle credenziali
  - **Miglioramento della UX** su revoche e consultazione uso delle credenziali

Contesti

Nel mondo

- **Standardization bodies**:
  - W3C, IETF, OpenID Foundation, ISO, ENISA …
- Il paradigma basato sugli ID Wallet facilita l'interoperabilità cross-border e propone nuovi casi d'uso, tuttavia:

  Tecnologie e standard non bastano, servono piattaforme sovranazionali condivise e armonizzare i regolamenti nazionali in conformità a queste piattaforme.

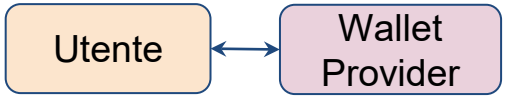
In Europa

- Revisione di eIDAS (The electronic IDentification Authentication and Signature Regulation)
  - In 2020 a revision has started by European Parliament, taking into account technological, market and legal developments in relation to the paradigms of **Decentralised Identity** and **Self-Sovereign Identity**. The regulation is not definitive yet.

- Testo approvato 8 Novembre 2023 (Parlamento Europeo e Consiglio Europeo), adozione prevista per il 2024.
- EUDI Wallet Architectural Reference Framework disponibile online e in continua evoluzione.
- **Obbligatorio per i servizi della Pubblica Amministrazione** e volontario per i cittadini.

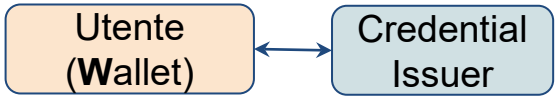
Flussi - ad alto livello

Inizializzazione (…)



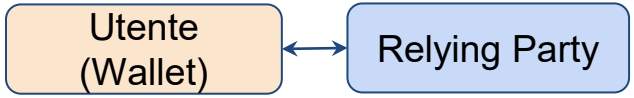
1. l'Utente accede al Wallet
2. Il Wallet fornisce al Wallet Provider la prova di possedere i livelli richiesti di sicurezza e garanzia
3. Il Wallet ottiene una attestazione di conformità dal Wallet Provider (o altra terza parte fidata).

Issuance (OpenID4VCI)



1. L'Utente chiede una credenziale ad un Issuer.
2. **W** valuta conformità Issuer
3. **W** fornisce attestazione di conformità (client auth) nella richiesta all'Issuer
4. Autenticazione Utente
5. Emissione credenziale
6. Credenziale conservata nel Wallet (Holder)

Presentation (OpenID4VP)



1. RP fornisce richiesta di presentazione
2. Wallet valuta conformità RP
3. Consenso utente (SD)
4. Presentazione credenziali
5. RP controlla validità e revoche credenziali

Formati e modelli dati delle credenziali

Selective Disclosure for JWTs (**SD-JWT**)

- Definito in IETF
  draft-ietf-oauth-selective-disclosure-jwt

- Basato su IETF JWT e JOSE

- Casi d'uso: **online** e offline

- Serializzazione: compact (JWT) e JSON

- Data model:
  - Schema comune definito nel formato
  - Esteso da SD-JWT-VC
  - Attributi utente custom

ISO **MDOC CBOR**

- Definito in **ISO/IEC 18013-5:2021** (mobile driving license)

- Basato su IETF CBOR e COSE

- Casi d'uso: **offline** e online

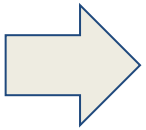
- Serializzazione: AF Binary

- Data model:
  - Schema comune
  - Attributi utente custom

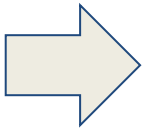
SD-JWT - Issuer Signed JWT

<**Issuer-signed JWT**>~<B64(Disclosure 1)>~...~<B64(Disclosure N)>~<optional KB-JWT>


```
{
  "sub": "user_42",
  "given_name": "John",
  "family_name": "Doe",
  "email": "johndoe@example.com",
  "phone_number": "+1-202-555-0101",
  "phone_number_verified": true,
  "address": {
    "street_address": "123 Main St",
    "locality": "Anytown",
    "region": "Anystate",
    "country": "US"
  },
  "birthdate": "1940-01-01",
}
```


```
{
  "_sd": [
                "CrQe7S5kqBAH …", …
  ],
  "iss": "https://issuer.example.com",
  "iat": 1683000000,
  "exp": 1883000000,
  "sub": "user_42",
  "_sd_alg": "sha-256",
  "cnf": {
                "jwk": {
                    "kty": "EC",
                    "crv": "P-256",
                    "x":
"TCAER19Zvu3OHF4 …",
                    "y":
"ZxjiWWbZMQGHVW …"
                }
  }
}
```

Signed JWT

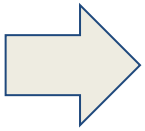
o

JWS JSON Serialization

SD-JWT - Disclosure

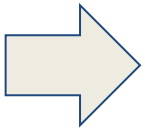
<Issuer-signed JWT>~<**B64url(Disclosure 1)**>~...~<**B64url(Disclosure N)**>~<optional KB-JWT>

```
[
  "2GLC42sKQveCfGfryNRN9w",
  "given_name",
  "John"
]
```



```
WyIyR0xDNDJzS1F2ZUNm
R2ZyeU5STjl3IiwgImdpdmV
uX25hbWUiLCAiSm9o
biJd
```

```
eyJhbGciOiAiRVMyNTYiLCAidHlwIjogImtiK2p3dC
J9.eyJub25jZSI
6ICIxMjM0NTY3ODkwIiwgImF1ZCI6ICJodHRwcz
ovL3ZlcmlmaWVyLmV4YW1wbGUub3J
nIiwgImlhdCI6IDE2OTgwNzc3OTAsICJfc2RfaGF
zaCI6ICIzNHQ4dkNDX2NfdlZMbk9
hZEJ0d2g0ZEZ2QkVyU2w5ektPcXdtNmloVF9VIn
0.ZlotfwqF9NUTRAShrd8jGSJEB6e
3Z3EKm-AD5udfzggxfK-
lQM4TCKbHK81eV088YTKl-
UfM7WSyQpx5wpNpZw
```



```
{
  "alg": "ES256",
  "typ": "kb+jwt"
}
{
  "nonce": "1234567890",
  "aud": "https://verifier.example.org",
  "iat": 1698077790,
  "_sd_hash": "34t8vCC_c_vVLnOadBtw …"
}
```

ISO MDOC CBOR

Non è nato per il web, viene trasportato in formato binario e serializzato in AF Binary

```
{ "version": "1.0",
        "documents": [
         {
           "docType": "org.iso.18013.5.1.mDL",
           "issuerSigned": {
        "nameSpaces": {
        "org.iso.18013.5.1": [
                        {
                                "digestID": 0,
                                "random": h'8798645b20ea200e19ffabac92 …',
                                "elementIdentifier": "family_name",
                                "elementValue": "Doe",
                        },
                        ...
        ],
        },
        "issuerAuth": $MSO ,
        },
        "deviceSigned": { ... }, }, }, ],
        "status": 0}
```

ISO MDOC CBOR - Mobile Security Object

MSO è un COSE Sign1 object. Nei suoi unprotected header vi è la X509 Chain dell'Issuer


```
{
 'version': '1.0',
 'digestAlgorithm': 'SHA-256',
 'valueDigests': {
                'org.iso.18013.5.1': {
                                0: b'u\x16s3\xb4{I+\xfb\x86\xec\xcc\x1fC\x8c\xf5z\xf0U7\x1a\xc5^\x1e5\x9e \xf2T\xad\xce\xbf',
   … }},
 'deviceKeyInfo': {
        'deviceKey': {
        1: 2,
                -1: 1,
                -2: b'\x961=Ic\xe2N3rt+\xfd\xb1\xa3;\xa2\xc8\x97\xdc\xd6\x8a\xb8\xc7S\xe4\xfb\xd4\x8d\xcak\x7f\x9a',
                -3:
b'\x1f\xb3&\x9e\xddA\x88W\xde\x1b9\xa4\xe4\xa4K\x92\xfaHL\xaar,"\x82\x88\xf0\x1d\x0c\x03\xa2\xc3\xd6'}},
 'docType': 'org.iso.18013.5.1.mDL',
 'validityInfo': {'signed': datetime.datetime(2020, 10, 1, 13, 30, 2, tzinfo=datetime.timezone.utc),
 'validFrom': datetime.datetime(2020, 10, 1, 13, 30, 2, tzinfo=datetime.timezone.utc),
 'validUntil': datetime.datetime(2021, 10, 1, 13, 30, 2, tzinfo=datetime.timezone.utc)}
}
```

Specifiche tecniche di riferimento

Proprietary Standards:

- ISO 18013-5 (esteso dal draft ISO 18013-7 include i seguenti OpenID)

Proprietary *Drafts*:

- ISO 23220-3 e ISO 23220-4 (issuance e presentation, includono i seguenti OpenID)

Open Standardization *Drafts*:

- Flows
  - OAuth 2.0 Attestation-Based Client Authentication (non ancora adottato in IETF WG)
  - OpenID for Verifiable Credential Issuance
  - OpenID for Verifiable Presentations
  - SIOPv2 (solo per self-issued attestations, eg: pseudonyms -> autocertificazione!)
- Credenziali
  - IETF SD-JWT (esteso da IETF SD-JWT-VC)

- oid4vc-haip-sd-jwt-vc (non ancora adottato in IETF WG)
- draft-ietf-oauth-status-list (controllo revoche)

Trust Model – Wallet Instance – Requisiti e rischi

Una terza parte fidata deve attestare la sicurezza del Wallet.

- Le **chiavi private non devono essere esportate** e di conseguenza rubate:
  - Un dispositivo rooted/jailbroken non garantisce la sicurezza al punto precedente
    - Chi attesta la sicurezza dei Wallet deve aggiungere controlli custom(?)
  - Non esistono ad oggi specifiche consolidate per i controlli di affidabilità dei dispositivi
- La fiducia in Hardware ed API proprietarie diventa requisito … oppure …
- È possibile usare **Hardware Security Modules (HSM) esterni per**:
  - facilitare la **migrazione delle credenziali da dispositivo a dispositivo**
  - **evitare il trust incondizionato verso i mobile device vendors**
    - per esempio: una Smart Card è un HSM
      - Numero limitato di chiavi -> usare sempre la medesima chiave rischia di facilitare il tracciamento su Issuers e RP diversi (Privacy)
      - Si possono generare chiavi derivate da una chiave master usando passphrases. UX da studiare -> Open Point.

Trust Model – Generale e Issuance

Wallet Provider, RP e Issuer devono aderire a regole condivise e darne prova:

- Un modello federativo è auspicabile

I Credential Issuers:

- Devono essere eleggibili all'emissione di specifici tipologie di credenziali:
  - Un issuer di Patente di Guida non può emettere una Attestazione di Diploma
  - I Wallet devono usare meccanismi automatici di valutazione dell'affidabilità degli Issuer in relazione all'emissione di specifiche credenziali
- Un bogus Issuer può danneggiare sia gli Utenti che i Relying Party

Security framework, policy e constraints:

- Servono framework che definiscono regole e protocolli di sicurezza ad alto livello
- Servono meccanismi automatici di controllo delle policy all'interno delle implementazioni

Trust Model – Presentazione

I Relying Party devono:

- essere compliant al trust framework e darne prova agli utenti:
  - la responsabilità non può essere degli Utenti soltanto, il Wallet deve proteggerli con meccanismi automatici di valutazione di affidabilità dei Relying Party

- chiedere agli utenti dati personali non eccedenti agli scopi del loro servizio -> Chi certifica i Relying Party sulla base delle categorie di servizio e dati ottenibili? (Policy)

- controllare lo stato di revoca di una credenziale non oltre un tempo limite e non oltre i limiti del consenso dell'Utente

- non essere in grado di ripudiare una richiesta di presentazione, di negare di aver chiesto ed ottenuto i dati dell'Utente.

Profilo implementativo Italiano

Draft
https://italia.github.io/eudi-wallet-it-docs/versione-corrente/en/

Repository
https://github.com/italia/eudi-wallet-it-docs

Scopo: Creare un profilo implementativo per l'ecosistema Wallet, con una sensibilità particolare sulla Privacy e la Sicurezza. Audience: gli implementatori che hanno deadlines e desiderano una lettura armonizzata di tutte le specifiche (non hanno bisogno di ridurre i requisiti!).
Costantemente aggiornata, al passo con l'evoluzione dei draft OpenID e IETF.

Particolarità: OpenID Federation per l'infrastruttura della fiducia (X.509 e Trusted Lists tuttora non chiare/definite in ARF).
Ricerca alternative e privacy preserving per la verifica dello stato della revoca delle credenziali.
OpenID for Identity Assurance per livelli di garanzia delle fonti dei dati (Authentic Sources).
Emissione delle credenziali *hardenizzato*.

Grazie!

Ci sono domande?

WORKSHOP GARR 2023

NET MAKERS

Streams and Timeline