Decentralized Identity Overview

Systems Security Lab @ SKKU

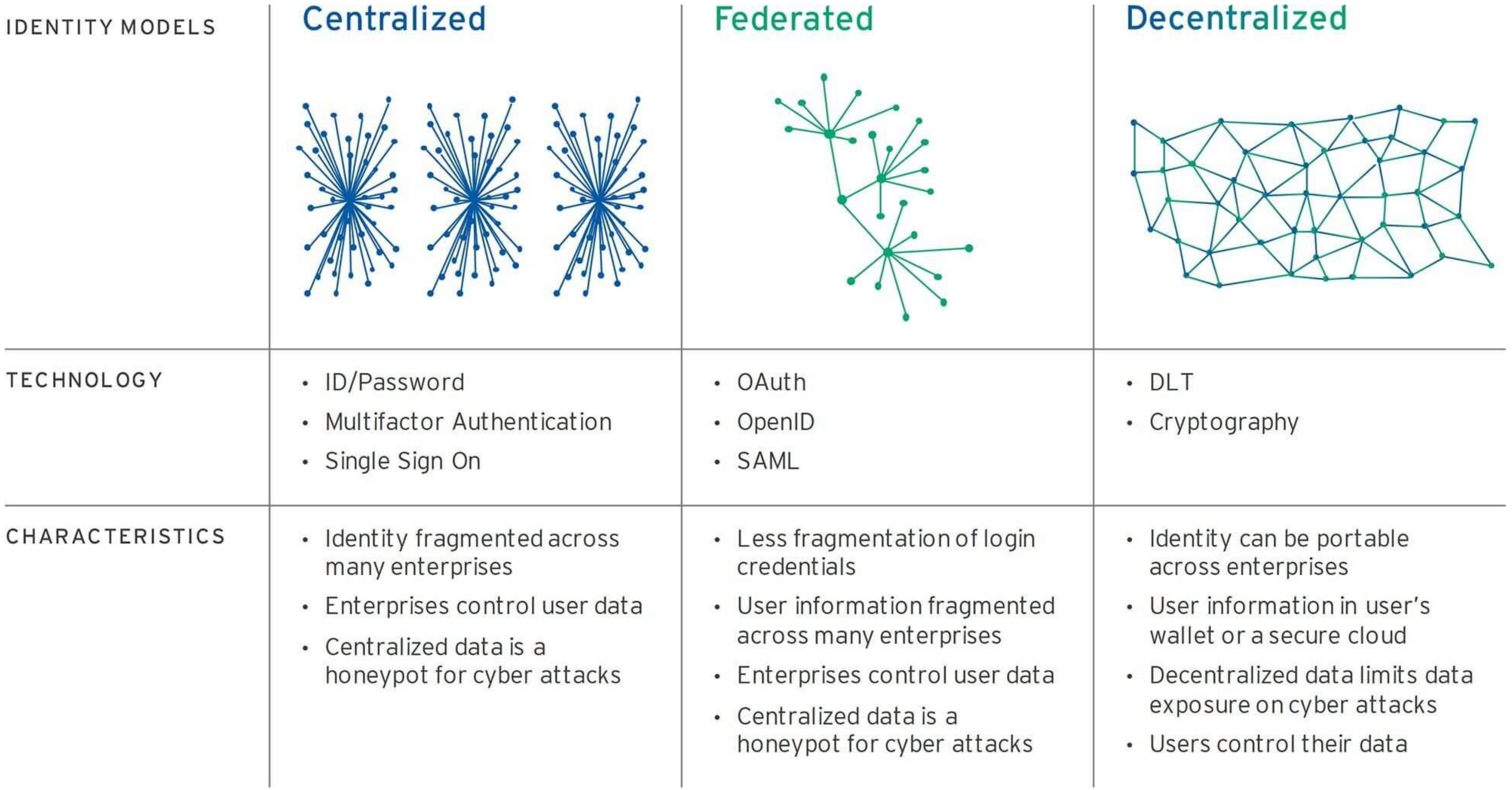
Si Won Heo

Background

- Today’s web
  - Information centralized to few enterprises (e.g. Google, Youtube)
  - Users have to manage different ID-PW pairs for every services

- Decentralized web
  - All the information managed by its owner
  - Decentralized ID is one of the tool for Decentralized Web

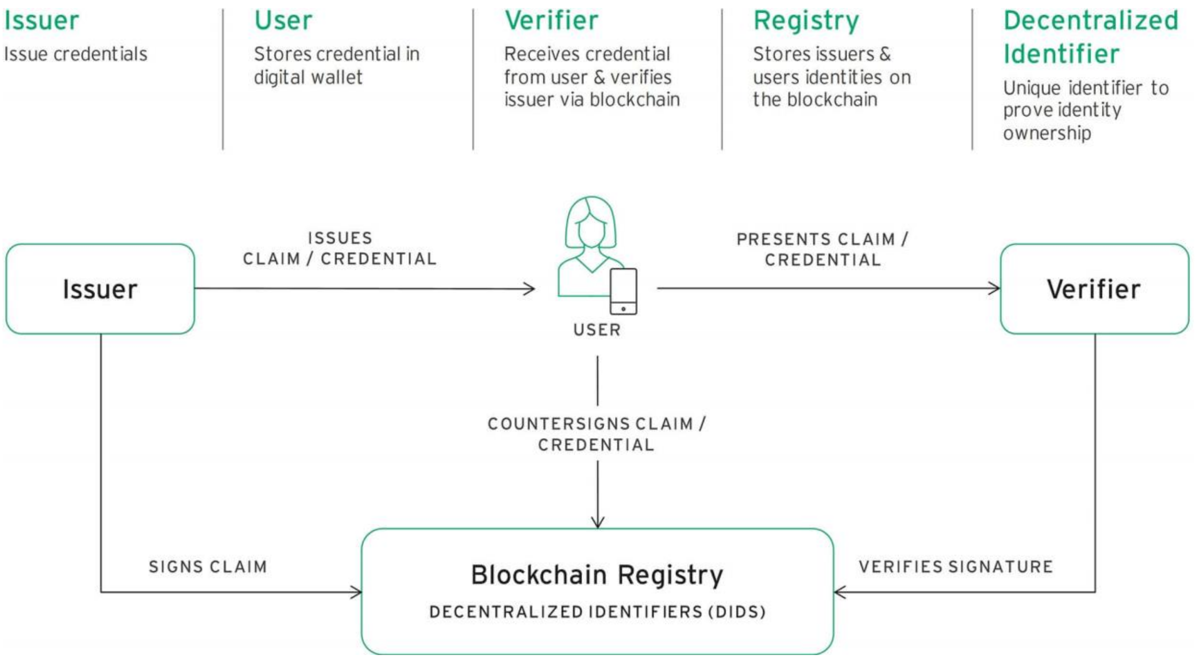
Background



IDENTITY MODELS	Centralized	Federated	Decentralized
TECHNOLOGY	• ID/Password • Multifactor Authentication • Single Sign On	• OAuth • OpenID • SAML	• DLT • Cryptography
CHARACTERISTICS	• Identity fragmented across many enterprises • Enterprises control user data • Centralized data is a honeypot for cyber attacks	• Less fragmentation of login credentials • User information fragmented across many enterprises • Enterprises control user data • Centralized data is a honeypot for cyber attacks	• Identity can be portable across enterprises • User information in user's wallet or a secure cloud • Decentralized data limits data exposure on cyber attacks • Users control their data

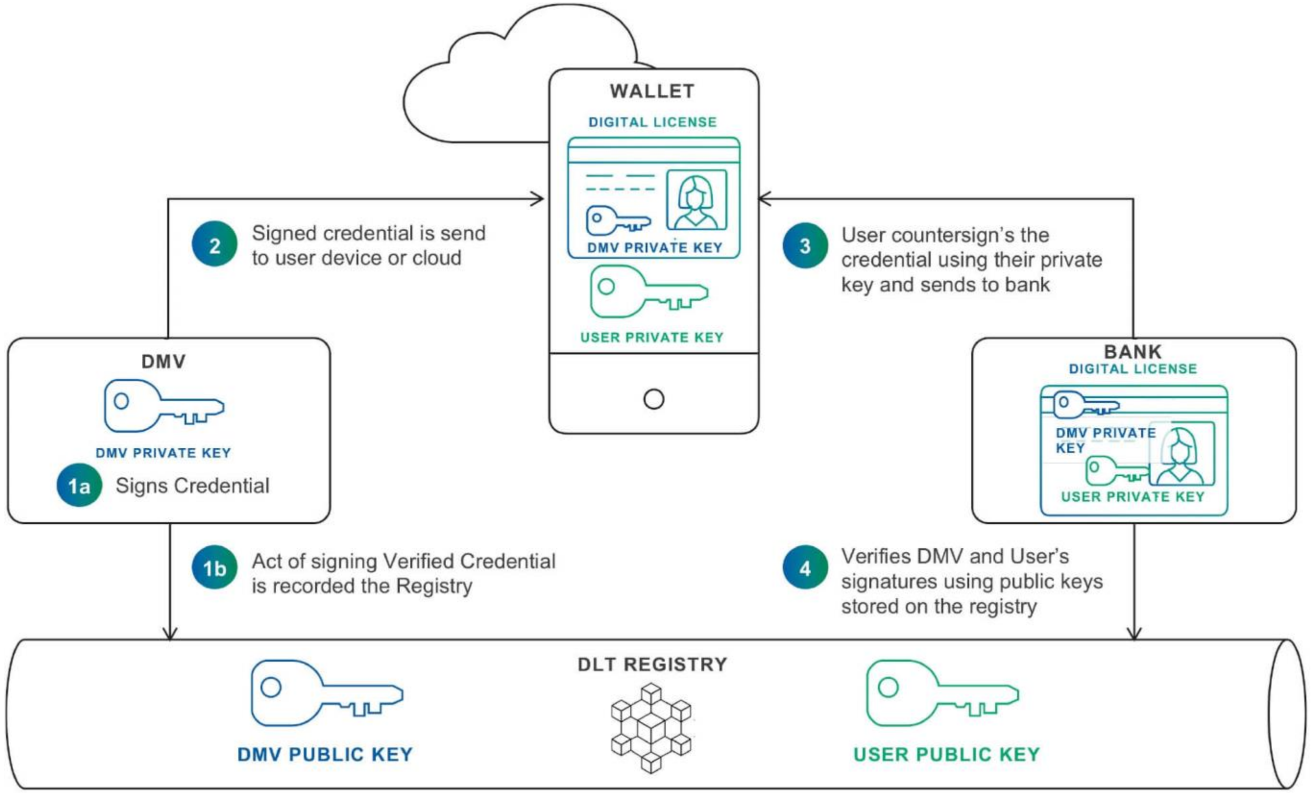
- DID is on the third phase of Identity management system

Decentralized Identification : How?


Issuer
Issue credentials
User
Stores credential in digital wallet
Verifier
Receives credential from user & verifies issuer via blockchain
Registry
Stores issuers & users identities on the blockchain
Decentralized Identifier
Unique identifier to prove identity ownership
Issuer
ISSUES CLAIM / CREDENTIAL
USER
PRESENTS CLAIM / CREDENTIAL
Verifier
COUNTERSIGNS CLAIM / CREDENTIAL
SIGNS CLAIM
Blockchain Registry
DECENTRALIZED IDENTIFIERS (DIDS)
VERIFIES SIGNATURE

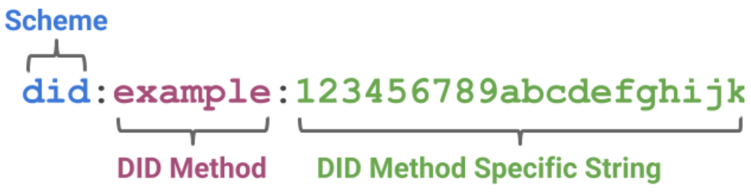
- User stores signed credential in Blockchain Registry or DLT
- Decentralized Identifier(DID) proves user’s identity ownership

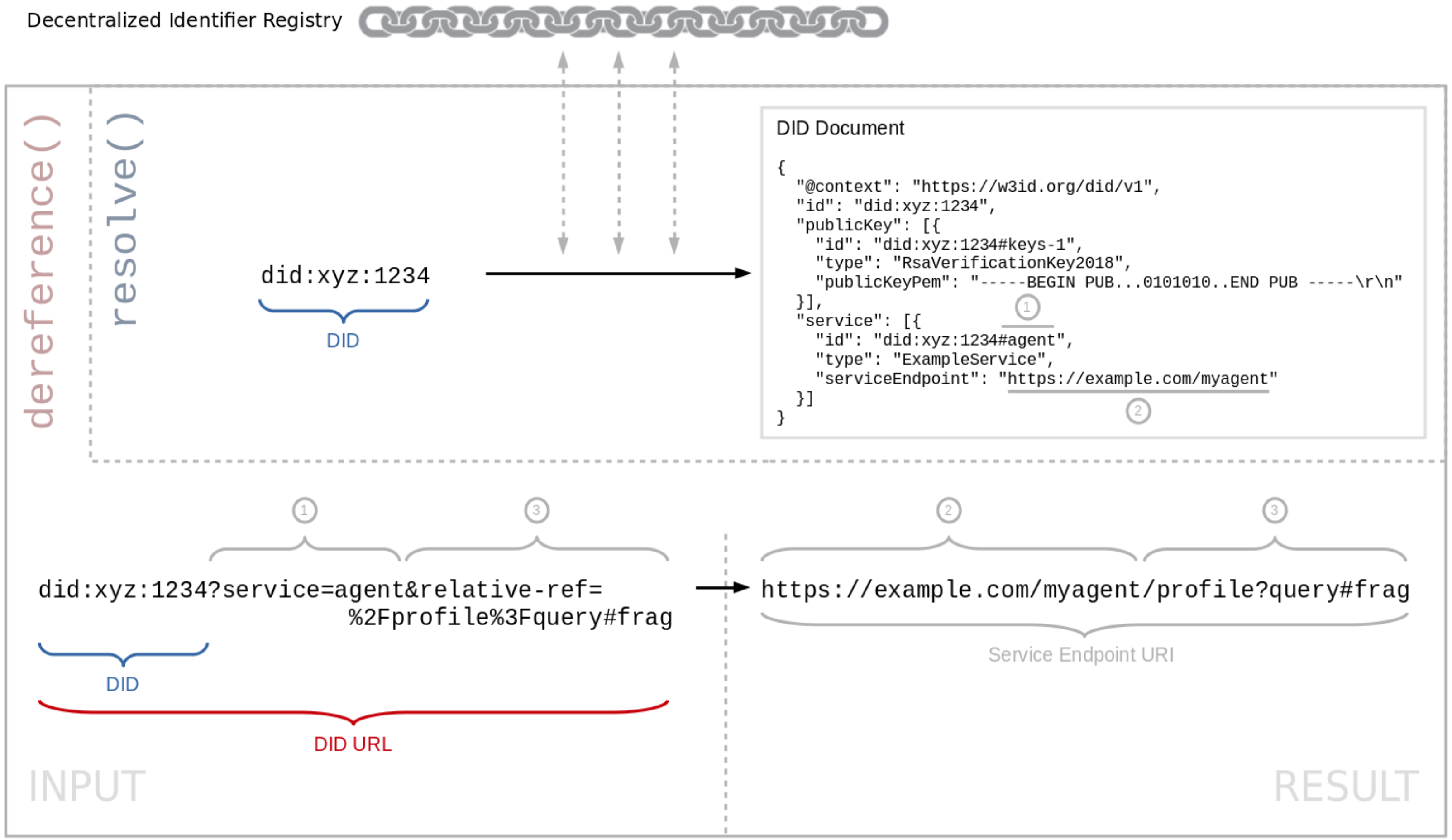
Decentralized Identification : How?


WALLET
DIGITAL LICENSE
DMV PRIVATE KEY
USER PRIVATE KEY
2 Signed credential is send to user device or cloud
3 User countersign's the credential using their private key and sends to bank
DMV
DMV PRIVATE KEY
1a Signs Credential
BANK
DIGITAL LICENSE
DMV PRIVATE KEY
USER PRIVATE KEY
1b Act of signing Verified Credential is recorded the Registry
4 Verifies DMV and User's signatures using public keys stored on the registry
DLT REGISTRY
DMV PUBLIC KEY
USER PUBLIC KEY

- Public-Private Key Cryptography
- Private-key Signed Credentials & User's Public Key stored in DLT
- Verified by Public Key of User

What is DID?



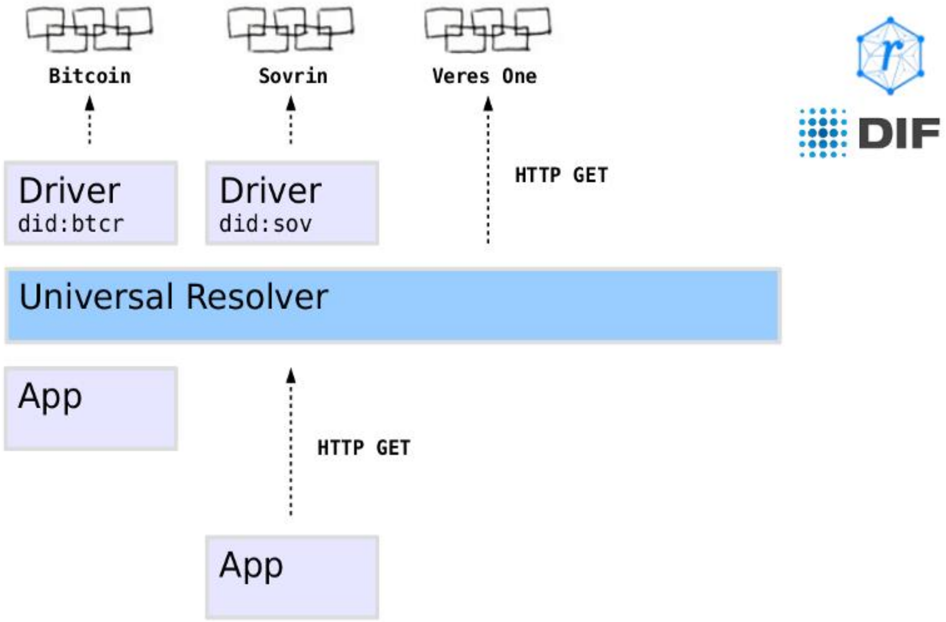


Decentralized Identifier Registry
dereference()
resolve()
did:xyz:1234
DID
DID Document
1
2
1
3
2
3
did:xyz:1234?service=agent&relative-ref=
%2Fprofile%3Fquery#frag
https://example.com/myagent/profile?query#frag
DID
DID URL
Service Endpoint URI
INPUT
RESULT

```
{
  "@context": "https://w3id.org/did/v1",
  "id": "did:xyz:1234",
  "publicKey": [{
    "id": "did:xyz:1234#keys-1",
    "type": "RsaVerificationKey2018",
    "publicKeyPem": "-----BEGIN PUB...0101010..END PUB -----\r\n"
  }],
  "service": [{
    "id": "did:xyz:1234#agent",
    "type": "ExampleService",
    "serviceEndpoint": "https://example.com/myagent"
  }]
}
```

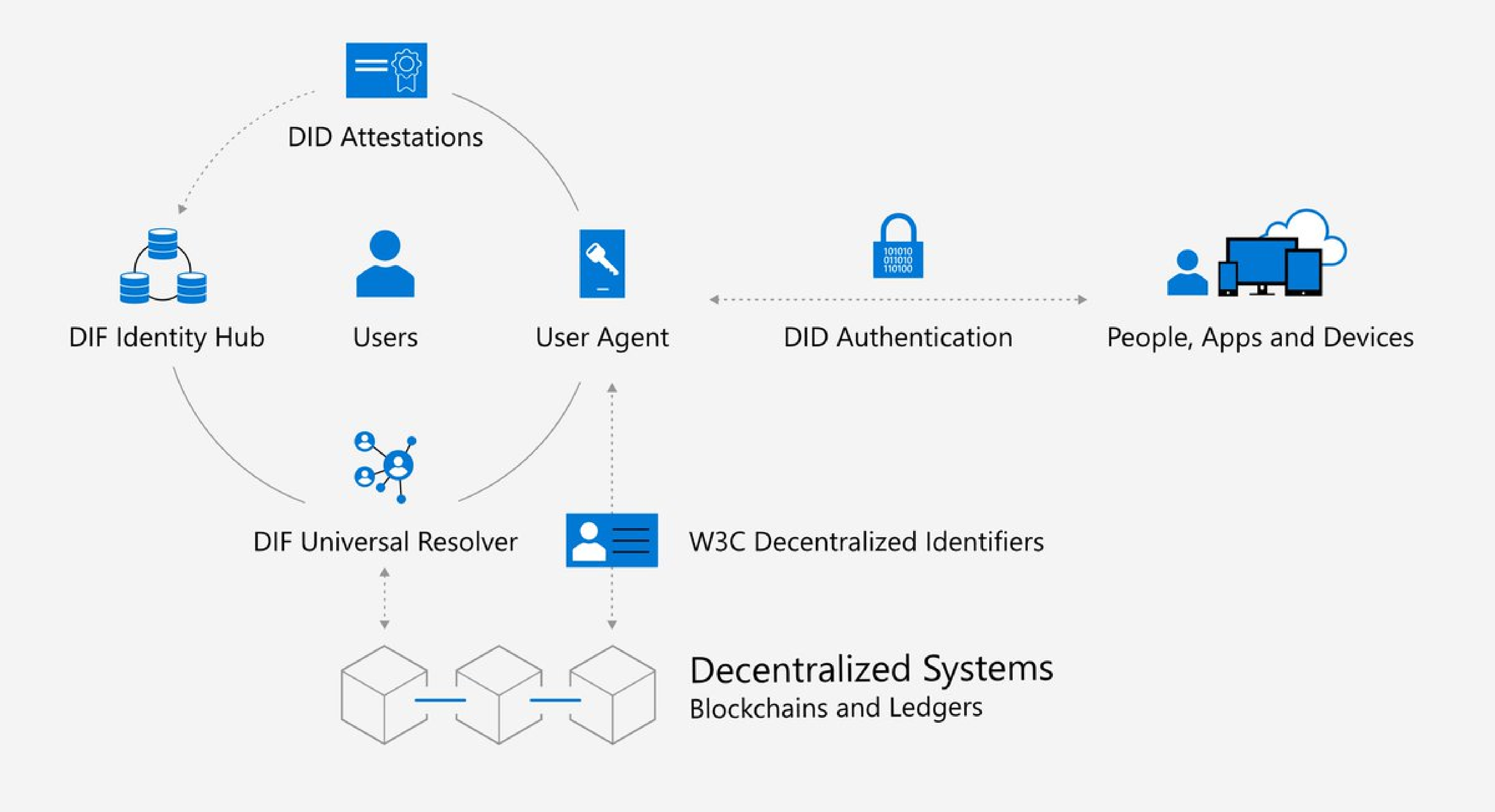
- URI form identifier
- Method : Platform of DID service
- Specific String : Resolved to DID document

Extensible Driver Network : Universal Resolver

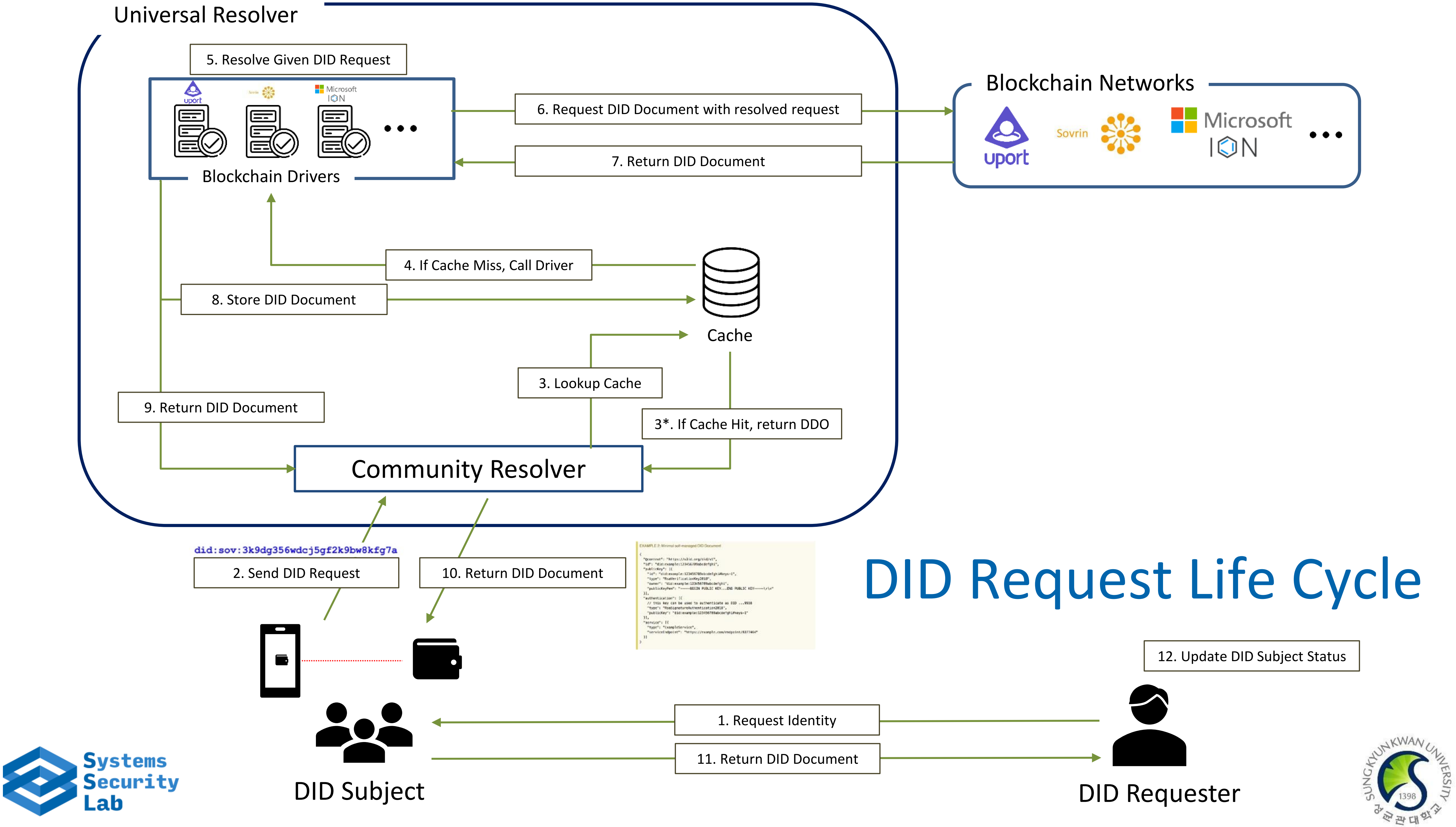


- Unified interface resolves any kind of DID
- “Driver” of each type comm. with DLT
- Drivers’ job may be different each other

DID Standardization



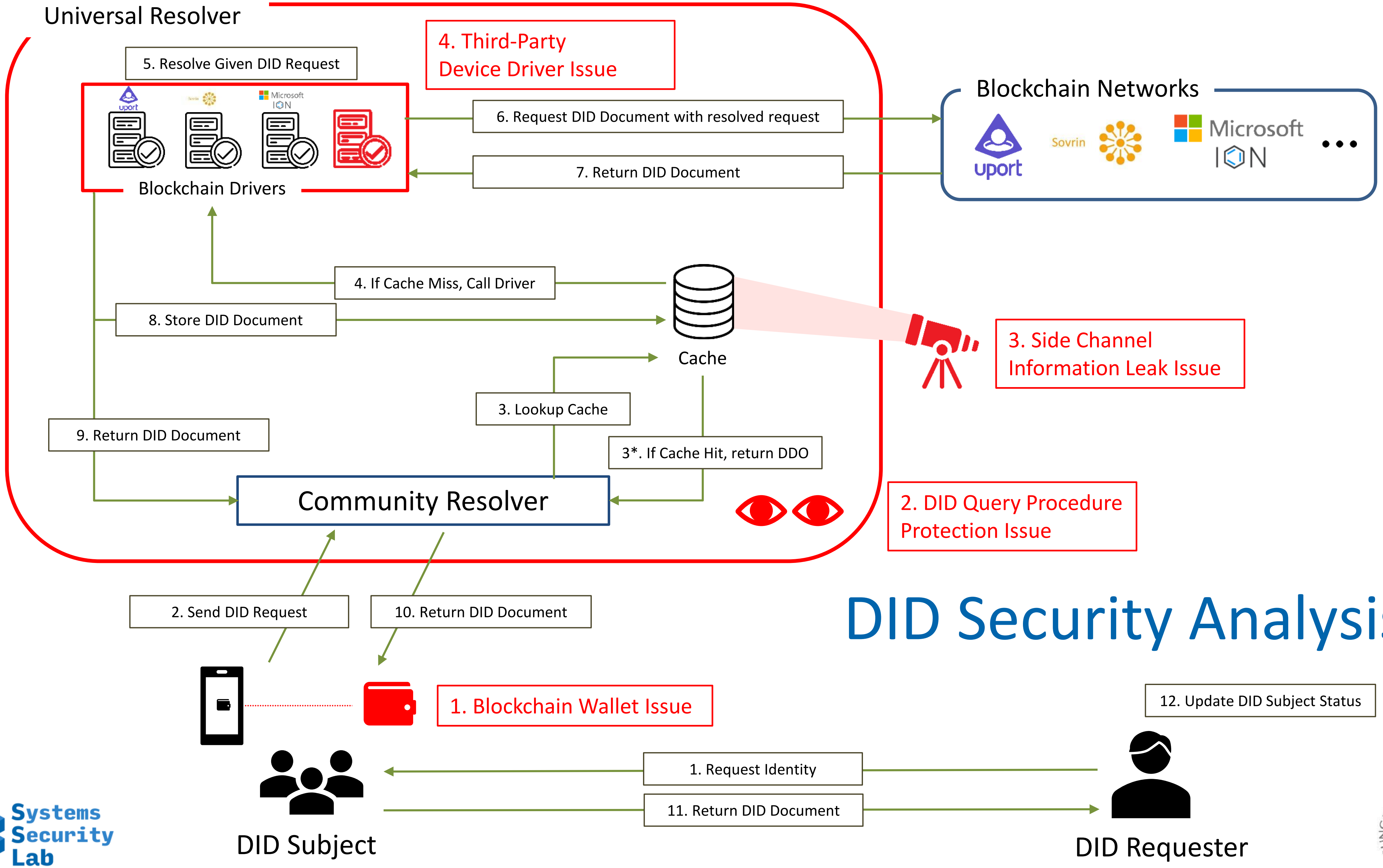
DID Request Life Cycle



EXAMPLE 2: Minimal self-managed DID Document

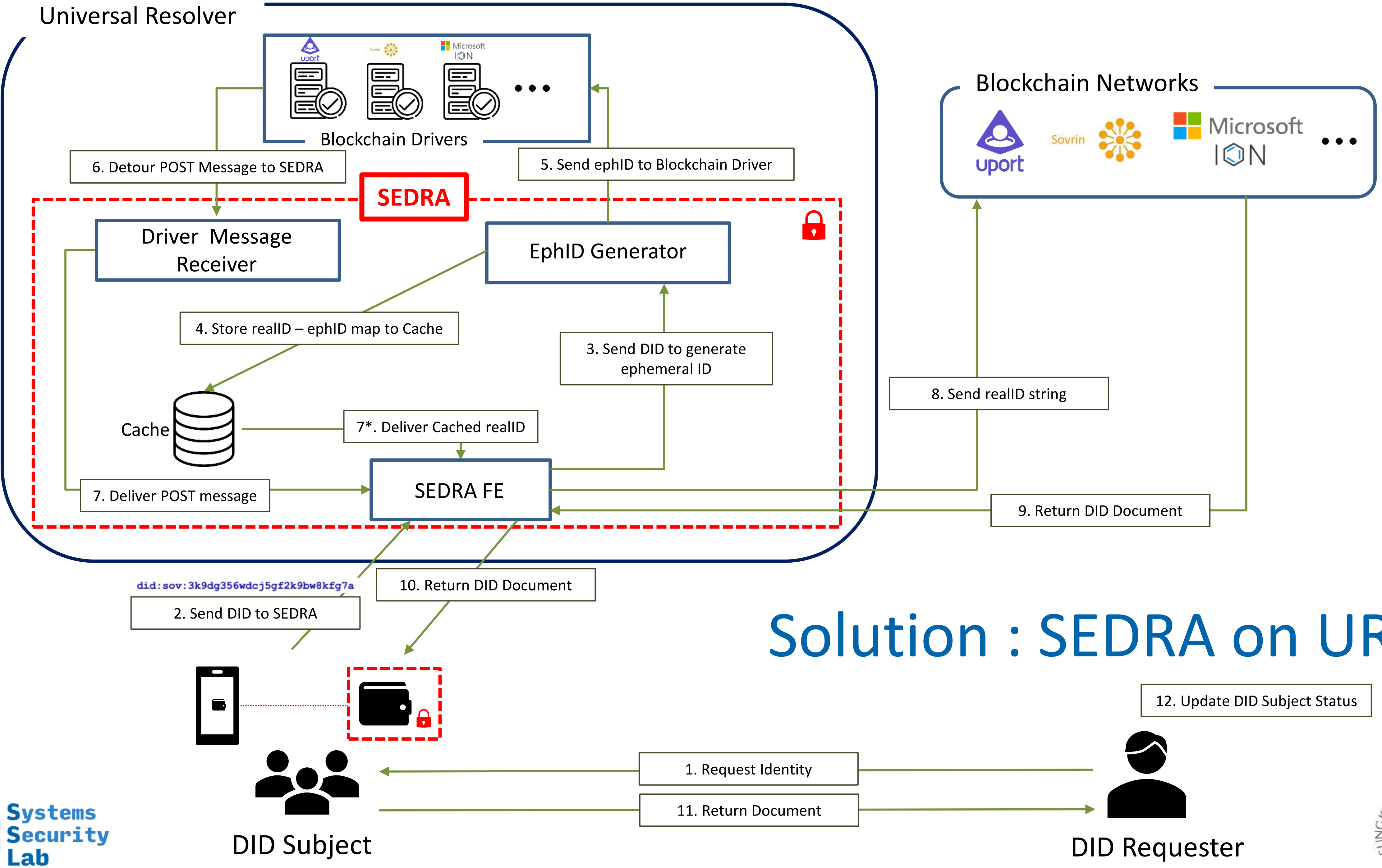
```
{
  "@context": "https://w3id.org/did/v1",
  "id": "did:example:123456789abcdefghi",
  "publicKey": [{
    "id": "did:example:123456789abcdefghi#keys-1",
    "type": "RsaVerificationKey2018",
    "owner": "did:example:123456789abcdefghi",
    "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"
  }],
  "authentication": [{
    // this key can be used to authenticate as DID ...9938
    "type": "RsaSignatureAuthentication2018",
    "publicKey": "did:example:123456789abcdefghi#keys-1"
  }],
  "service": [{
    "type": "ExampleService",
    "serviceEndpoint": "https://example.com/endpoint/8377464"
  }]
}
```

DID Security Analysis



Systems Security Lab

Solution : SEDRA on UR



Universal Resolver

Blockchain Drivers

Blockchain Networks

SEDRA

Driver Message Receiver

EphID Generator

Cache

SEDRA FE

1. Request Identity

2. Send DID to SEDRA

did:sov:3k9dg356wdcj5gf2k9bw8kfg7a

3. Send DID to generate ephemeral ID

4. Store realID – ephID map to Cache

5. Send ephID to Blockchain Driver

6. Detour POST Message to SEDRA

7. Deliver POST message

7*. Deliver Cached realID

8. Send realID string

9. Return DID Document

10. Return DID Document

11. Return Document

12. Update DID Subject Status

DID Subject

DID Requester

Systems Security Lab