ESSIF v1
Architectural / Tech Spec Foundations

ESSIF Vision

ESSIF Datamodeling

ESSIF key Flows

ESSIF (supporting) components

Preview / “under the hood”

- **ESSIF Vision**
  - **The Targeted eco-system**
  - **Targeted Business Use Cases**
  - **Longer Term View <> ESSIF v1**

- **ESSIF Datamodeling**
  - **Identities <> DIDs**
  - **Verifiable IDs <> Attestations**
  - **Links with LoA’s**
  - **Links with Legal Value**
  - **Resulting (flexible) DataModel**

- **ESSIF key Flows:**
  - **DID registrations**
  - **Obtaining a Verifiable ID**
  - **Obtaining a Verifiable Attestation**
  - **Details >> Link SSI and OIDC**
  - **Details >> Link with APIs**

- **ESSIF (supporting) components**
  - **User / Issuer / Relying Party Environments**
  - **Trusted Issuer Ledger / eIDAS bridge**
  - **DID Registrars / Resolvers / Identity Hubs**

- **Preview / “under the hood”**
  - **Technology mapping**

The Targeted eco-system

ESSIF ecosystem: the totality of the actors and systems within the context of ESSIF and according to the rules and standards of the ESSIF-ecosystem.

ESSIF (Trust) Framework: the totality of all policies, guidelines, standards, processes, … which for the "terms and conditions" of membership and/or usage of ESSIF-services.

ESSIF architecture: the definition of ESSIF and all related actors and building blocks at functional level, at level of concepts, at level of resilience/trust requirements, at level of interactions (incl all corresponding technical and operational standards).

ESSIF infrastructure: all supporting capabilities/services which support the functioning of ESSIF and all its members and framework-obiding relying parties, issuers and users.

SSI and blockchain

ESSIF v1

* Build upon / Reuse SSI-community materials

* Use blockchain where useful / added value

* Avoid complex / unstable scenarios

Core required properties:

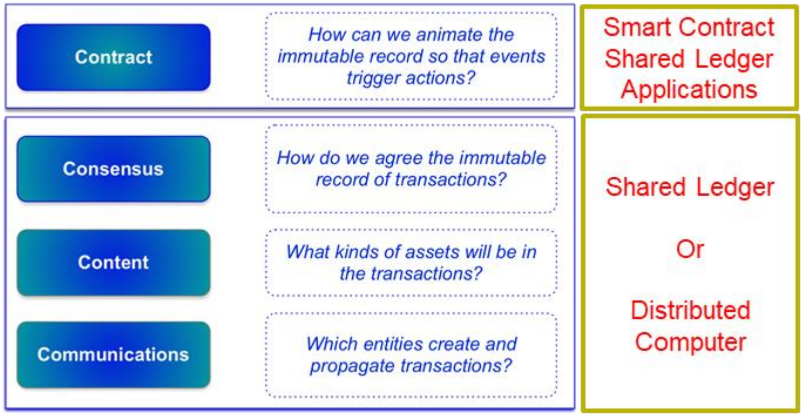
* Reuse knowledge / experience from eIDAS

* Privacy / Data Protecting by Design

* Trusted / Resilient / Secure by design

Leitmotiv:

* Think Long Term… but Act Short Term



Contract

How can we animate the immutable record so that events trigger actions?

Smart Contract Shared Ledger Applications

Consensus

How do we agree the immutable record of transactions?

Content

What kinds of assets will be in the transactions?

Communications

Which entities create and propagate transactions?

Shared Ledger

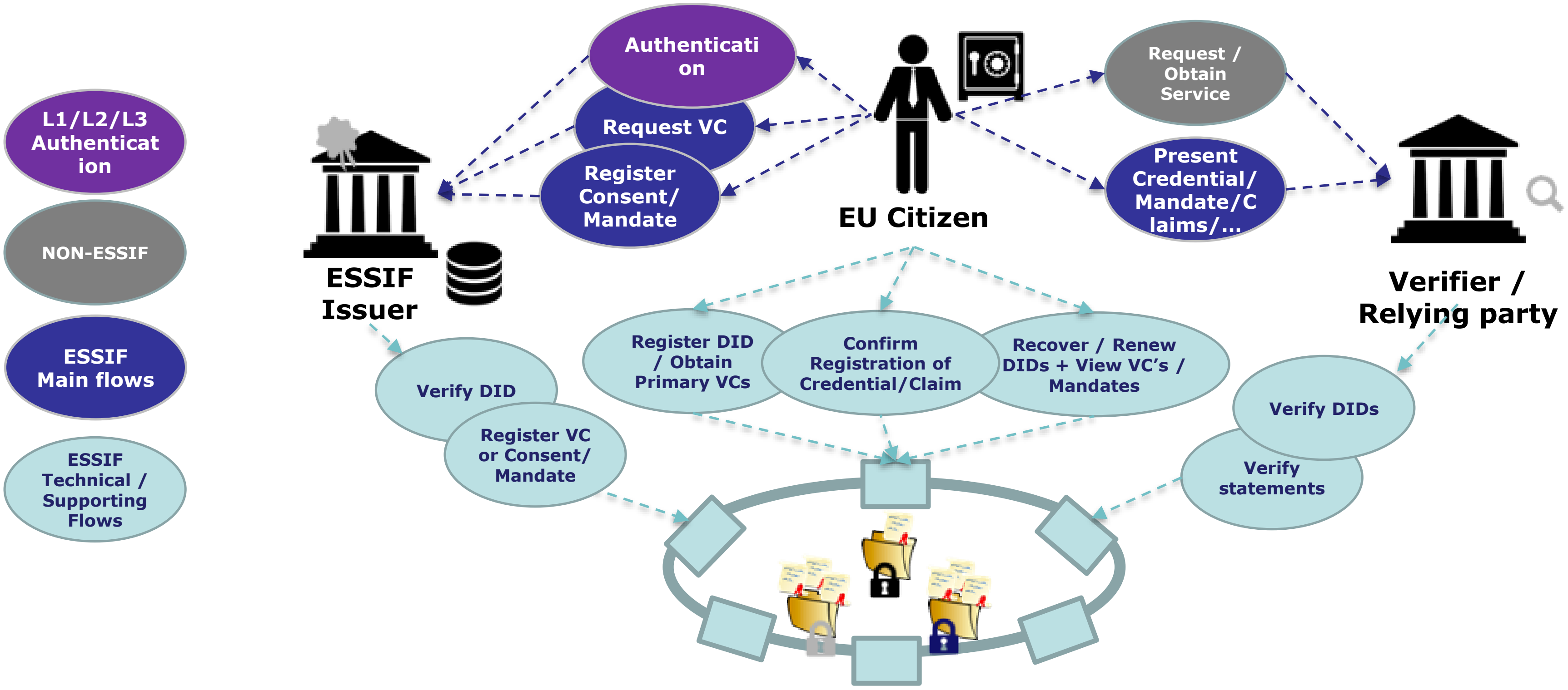
Or

Distributed Computer

"the consensus computer"

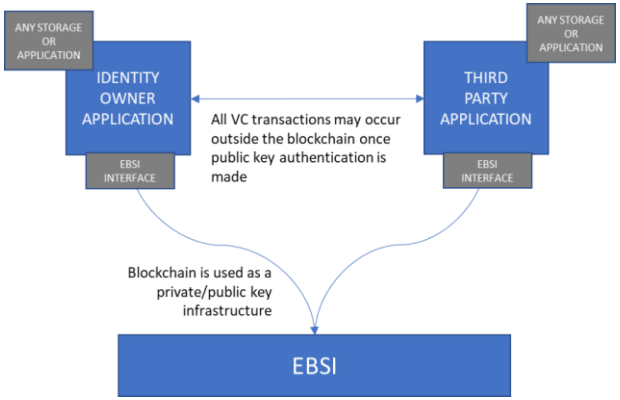
In / Out scope

ESSIF will NOT intervene in the business flow between the EU citizens/entities and relying parties. The requesting of services and the obtaining of those services are out of scope of ESSIF. ESSIF however will allow an EU entity to "obtain" Verifiable Credentials, to "register" Verifiable Mandates/Consents, and to "obtain" Verify Verifiable Claims which then can be use to identify/authenticate towards relying parties and provide those with required claims/attestations.


L1/L2/L3 Authentication
NON-ESSIF
ESSIF Main flows
ESSIF Technical / Supporting Flows
Authentication
Request VC
Register Consent/ Mandate
ESSIF Issuer
EU Citizen
Request / Obtain Service
Present Credential/ Mandate/Claims/…
Verifier / Relying party
Verify DID
Register VC or Consent/ Mandate
Register DID / Obtain Primary VCs
Confirm Registration of Credential/Claim
Recover / Renew DIDs + View VC's / Mandates
Verify DIDs
Verify statements

ESSIF miss-conceptions

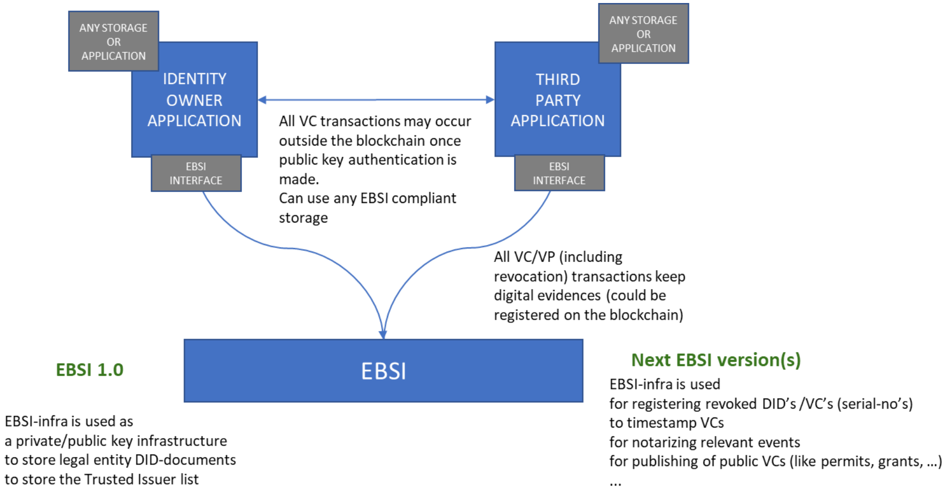
Wrong understanding


ANY STORAGE OR APPLICATION
IDENTITY OWNER APPLICATION
EBSI INTERFACE
ANY STORAGE OR APPLICATION
THIRD PARTY APPLICATION
EBSI INTERFACE
EBSI

All VC transactions may occur outside the blockchain once public key authentication is made

Blockchain is used as a private/public key infrastructure

Right understanding


ANY STORAGE OR APPLICATION
IDENTITY OWNER APPLICATION
EBSI INTERFACE
ANY STORAGE OR APPLICATION
THIRD PARTY APPLICATION
EBSI INTERFACE
EBSI

All VC transactions may occur outside the blockchain once public key authentication is made.
Can use any EBSI compliant storage

All VC/VP (including revocation) transactions keep digital evidences (could be registered on the blockchain)

EBSI 1.0

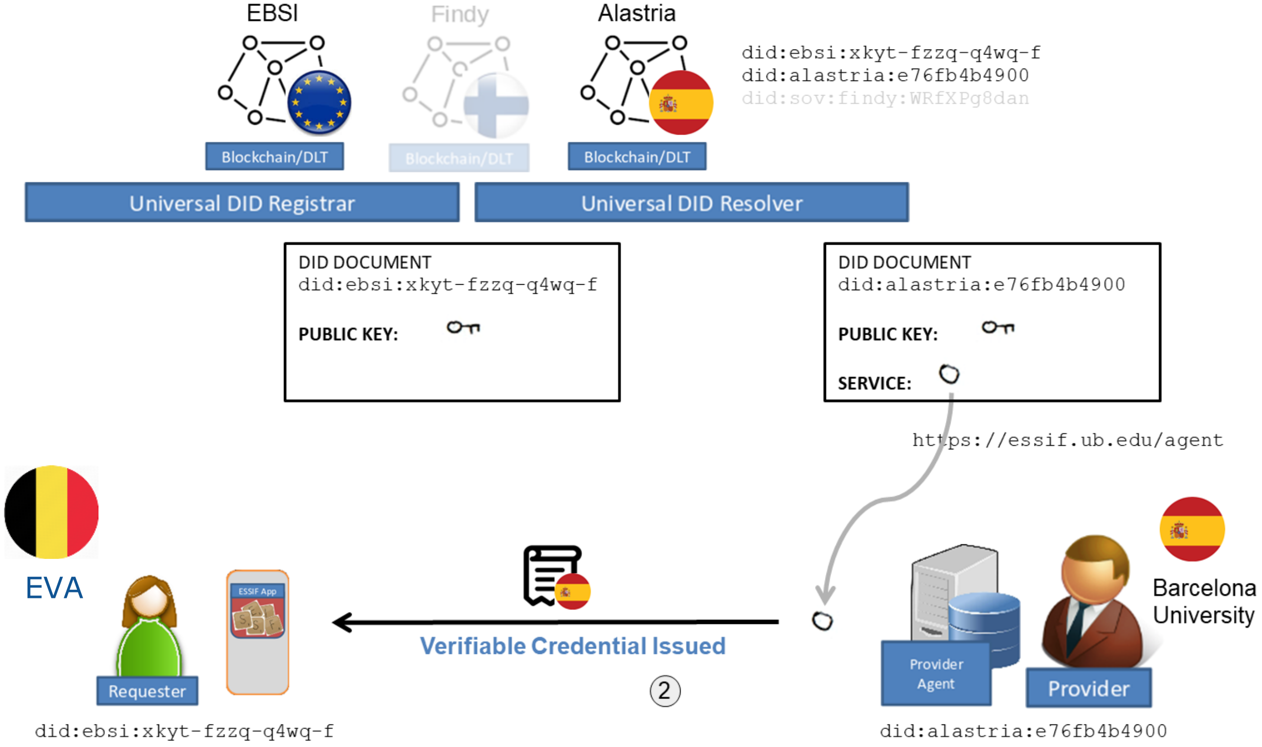
EBSI-infra is used as
a private/public key infrastructure
to store legal entity DID-documents
to store the Trusted Issuer list

Next EBSI version(s)

EBSI-infra is used
for registering revoked DID's /VC's (serial-no's)
to timestamp VCs
for notarizing relevant events
for publishing of public VCs (like permits, grants, …)
…

ESSIF v1 (Verifiable IDs & Attestations)

Barcelona University issues Verifiable Credential to Eva



EBSI

Findy

Alastria

Blockchain/DLT

Blockchain/DLT

Blockchain/DLT

did:ebsi:xkyt-fzzq-q4wq-f
did:alastria:e76fb4b4900
did:sov:findy:WRfXPg8dan

Universal DID Registrar

Universal DID Resolver

DID DOCUMENT
did:ebsi:xkyt-fzzq-q4wq-f

PUBLIC KEY:

DID DOCUMENT
did:alastria:e76fb4b4900

PUBLIC KEY:

SERVICE:

https://essif.ub.edu/agent

EVA

ESSIF App

Requester

Verifiable Credential Issued

2

Barcelona University

Provider Agent

Provider

did:ebsi:xkyt-fzzq-q4wq-f

did:alastria:e76fb4b4900

DID Private Keys – generated and stored in wallet

PRIVATE KEY:

Verifiable Credentials – stored in wallet, associated with DIDs

3

DID Private Keys – generated and stored in wallet

PRIVATE KEY:

Verifiable Credentials – issued and signed by DID's private key

1

Warning

Due to time/resource limitations ESSIF v1 / EBSI v1 does not fully reflect the architectural / technical specifications listed here.

The specifications should be read as “target” and ESSIF v1 / EBSI v1 should be understood s a non-production “demonstrator”

In reality these specifications will be fine-tuned in light of ESSIF v2 / EBSI v2 and taking into account the lessons learned from v1 + input from the use cases + legal considerations.

- ESSIF Vision
    - The Targeted eco-system
    - Targeted Business Use Cases
    - Longer Term View <> ESSIF v1

- **ESSIF Datamodeling**
    - **Identities <> DIDs**
    - **Verifiable IDs <> Attestations**
    - **Links with LoA's**
    - **Links with Legal Value**
    - **Resulting (flexible) DataModel**

- ESSIF key Flows:
    - DID registrations
    - Obtaining a Verifiable ID
    - Obtaining a Verifiable Attestation
    - Details >> Link SSI and OIDC
    - Details >> Link with APIs

- ESSIF (supporting) components
    - User / Issuer / Relying Party Environments
    - Trusted Issuer Ledger / eIDAS bridge
    - DID Registrars / Resolvers / Identity Hubs

- Preview / "under the hood"
    - Technology mapping

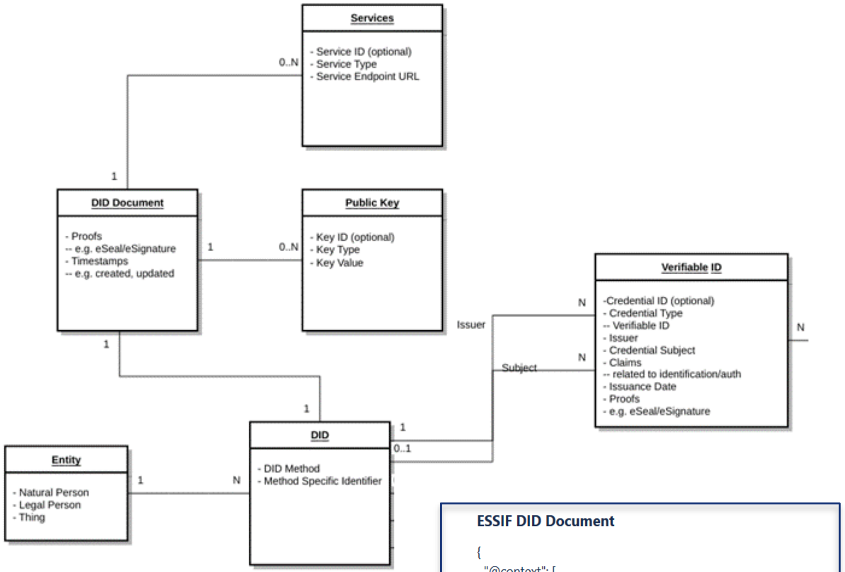
Identities <> DIDs

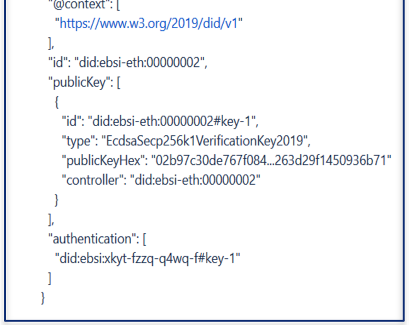
Key Properties:

- One entity can have multiple DIDs!
- DIDs of Issuers / Relying Parties will be anchored on ledger.

DID-docs in ESSIF v1:

- DID Subject: This is the subject (individual, organization, thing, animal, etc.) identified by the DID.
- Public Keys: Public Keys associated with a DID are a prerequisite for secure and authenticated communication between DID Subjects.
- Authentication: The Authentication block in a DID Document simply references the DID Document’s Public Key (see above) that is intended for proving control/ownership of a DID. This is used when two parties (e.g. a Holder and a Verifier) connect and exchange data and messages.
- Proof: This can be added to a DID Document to prove integrity or correctness or other security and trust aspects of a DID Document.





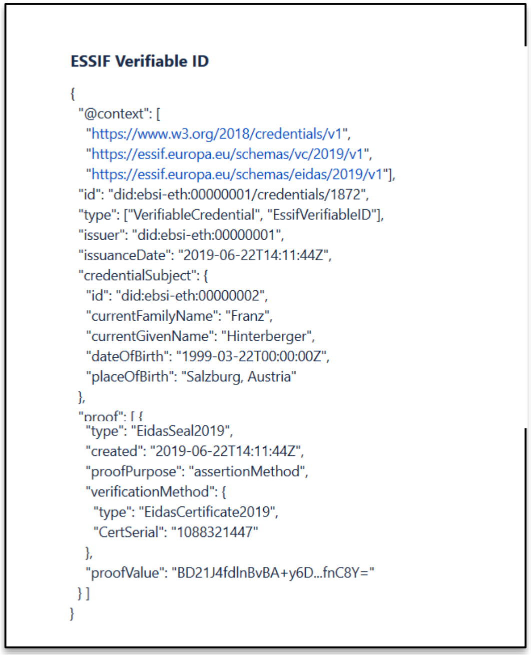
ESSIF DID Document

```
{
  "@context": [
    "https://www.w3.org/2019/did/v1"
  ],
  "id": "did:ebsi-eth:00000002",
  "publicKey": [
    {
      "id": "did:ebsi-eth:00000002#key-1",
      "type": "EcdsaSecp256k1VerificationKey2019",
      "publicKeyHex": "02b97c30de767f084...263d29f1450936b71"
      "controller": "did:ebsi-eth:00000002"
    }
  ],
  "authentication": [
    "did:ebsi:xkyt-fzzq-q4wq-f#key-1"
  ]
}
```

Verifiable IDs

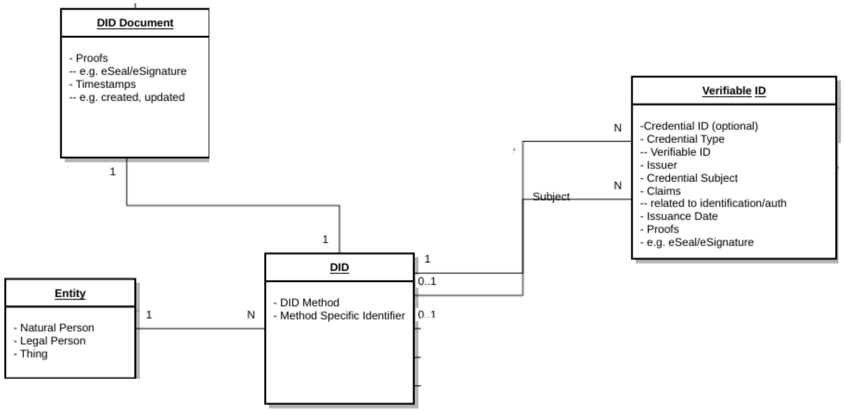
Key Properties:

- One DID can have multiple Verifiable IDs
- Verifiable IDs can have official identifiers



ESSIF Verifiable ID

```
{
 "@context": [
   "https://www.w3.org/2018/credentials/v1",
   "https://essif.europa.eu/schemas/vc/2019/v1",
   "https://essif.europa.eu/schemas/eidas/2019/v1"],
 "id": "did:ebsi-eth:00000001/credentials/1872",
 "type": ["VerifiableCredential", "EssifVerifiableID"],
 "issuer": "did:ebsi-eth:00000001",
 "issuanceDate": "2019-06-22T14:11:44Z",
 "credentialSubject": {
   "id": "did:ebsi-eth:00000002",
   "currentFamilyName": "Franz",
   "currentGivenName": "Hinterberger",
   "dateOfBirth": "1999-03-22T00:00:00Z",
   "placeOfBirth": "Salzburg, Austria"
 },
 "proof": [ {
   "type": "EidasSeal2019",
   "created": "2019-06-22T14:11:44Z",
   "proofPurpose": "assertionMethod",
   "verificationMethod": {
     "type": "EidasCertificate2019",
     "CertSerial": "1088321447"
   },
   "proofValue": "BD21J4fdlnBvBA+y6D...fnC8Y="
 } ]
}
```



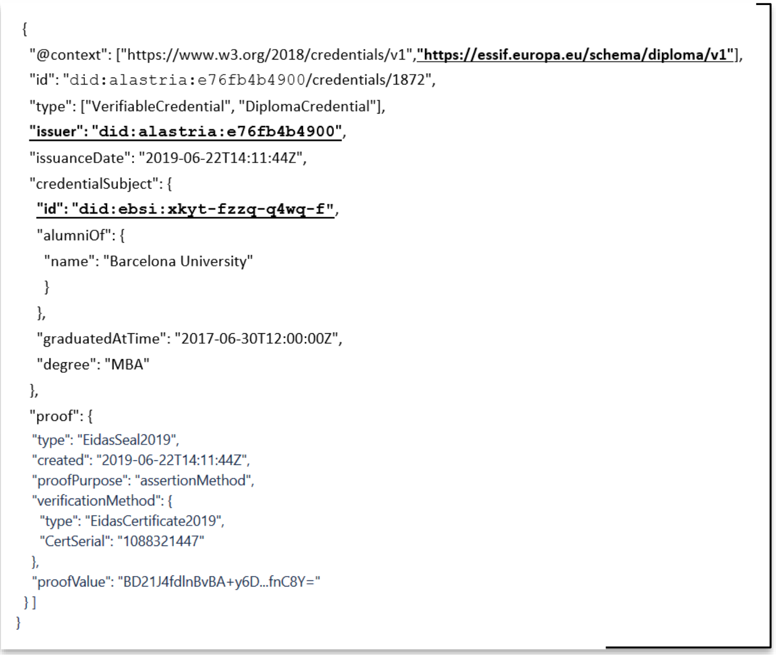
Verifiable IDs in ESSIF v1:

- eIDAS minimal data set + Optional national / University ID
- The following Basic Concepts of the W3C specification ARE used: Contexts, Identifiers, Types, Issuers, Credential Subject, Issuer, Issuance Date, Proofs
- Attention Points: (national) identifiers, LoA-Information, linked eSeal

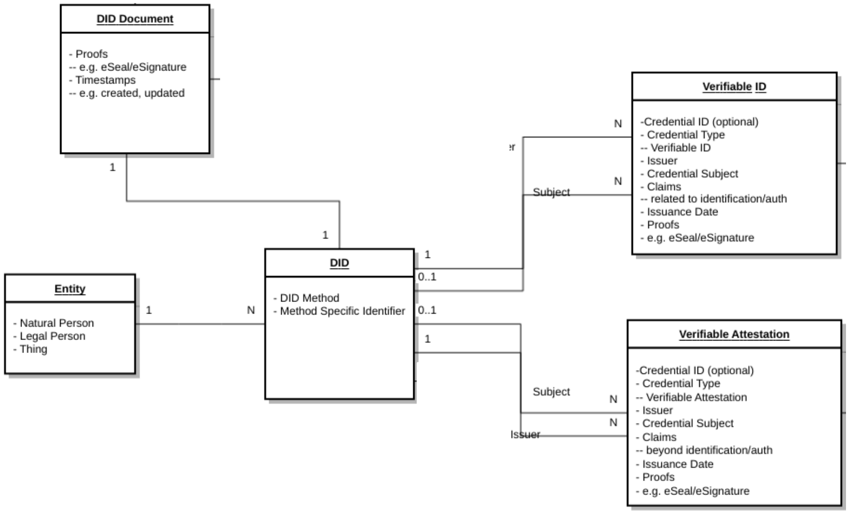
Verifiable Attestations

Key Properties:

- One DID can have multiple Verifiable Attestations
- Verifiable Attestations can inherit attributes from “parenting” Verifiable IDs



```
{
  "@context": ["https://www.w3.org/2018/credentials/v1","https://essif.europa.eu/schema/diploma/v1"],
  "id": "did:alastria:e76fb4b4900/credentials/1872",
  "type": ["VerifiableCredential", "DiplomaCredential"],
  "issuer": "did:alastria:e76fb4b4900",
  "issuanceDate": "2019-06-22T14:11:44Z",
  "credentialSubject": {
   "id": "did:ebsi:xkyt-fzzq-q4wq-f",
   "alumniOf": {
    "name": "Barcelona University"
    }
   },
   "graduatedAtTime": "2017-06-30T12:00:00Z",
   "degree": "MBA"
  },
  "proof": {
  "type": "EidasSeal2019",
  "created": "2019-06-22T14:11:44Z",
  "proofPurpose": "assertionMethod",
  "verificationMethod": {
   "type": "EidasCertificate2019",
   "CertSerial": "1088321447"
  },
  "proofValue": "BD21J4fdlnBvBA+y6D...fnC8Y="
 }]
}
```



Verifiable Attestations in ESSIF v1:

•The following Basic Concepts of the W3C specification ARE used: Contexts, Identifiers, Types, Issuers, Credential Subject, Issuer, Issuance Date, Proofs

•Attention Points: (national) identifiers, LoA-Information, linked eSeal

Links with LoA’s

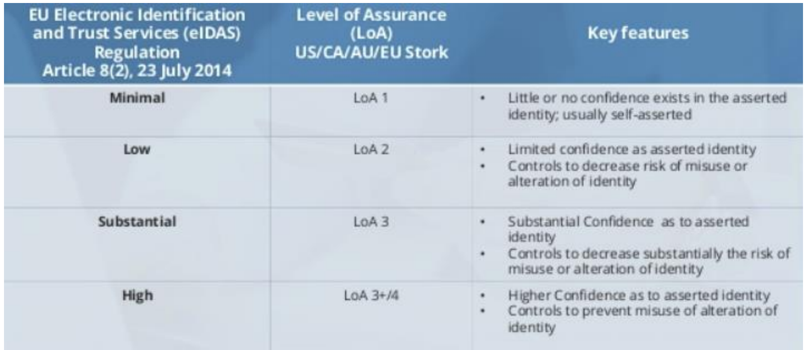
In context of Identification/Authentication:

- **When doing an eIDAS identification / authentication:**
  * Need to control proof of DID
  + “Authentication strength”
  * Need to rely on eIDAS-Identification
  + “Authentication strength”

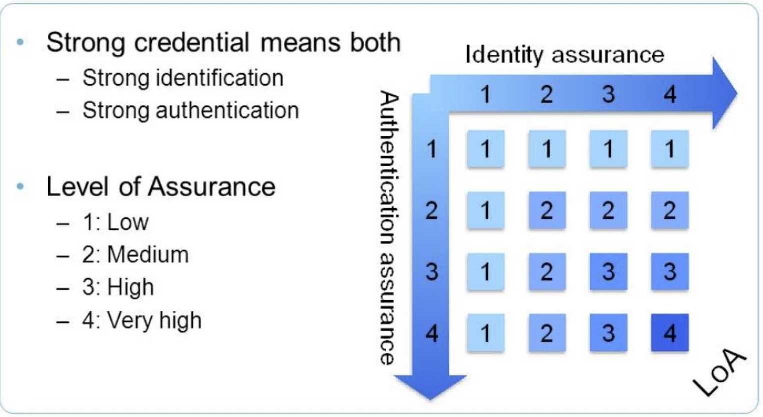
- **When presenting a Verifiable ID:**
  * Need to verify the “presented VC”,
  incl the LoA claimed by the Issuer
  incl the possibly present eSeals.
  * Need to check the “presence” of the issuer
  + “Authentication strength”

In context of presenting Attestations:

- **When presenting a Verifiable Attestations:**
  * Need to verify the “presented VCs”,
  incl the LoA claimed by the Issuer
  incl the possibly present eSeals.



EU Electronic Identification and Trust Services (eIDAS) Regulation Article 8(2), 23 July 2014	Level of Assurance (LoA) US/CA/AU/EU Stork	Key features
Minimal	LoA 1	• Little or no confidence exists in the asserted identity; usually self-asserted
Low	LoA 2	• Limited confidence as asserted identity • Controls to decrease risk of misuse or alteration of identity
Substantial	LoA 3	• Substantial Confidence as to asserted identity • Controls to decrease substantially the risk of misuse or alteration of identity
High	LoA 3+/4	• Higher Confidence as to asserted identity • Controls to prevent misuse of alteration of identity



- Strong credential means both
  - Strong identification
  - Strong authentication
- Level of Assurance
  - 1: Low
  - 2: Medium
  - 3: High
  - 4: Very high

Identity assurance

	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	2	3	3
4	1	2	3	4

Authentication assurance

LoA

Links with Legal Value

Legal value of Verifiable IDs / Attestations:

- Some VCs need to provide assurance wrt Authenticity / Integrity / Non-repudiation
- REUSE eIDAS eSeals
- Legal Value = “Issued by OrganizationX”
- IN SCOPE of ESSIF v1

Legal value of Presentations

•When presenting a Verifiable Attestations it might be legally required to know who submitted the VCs

•REUSE eIDAS eSignatures

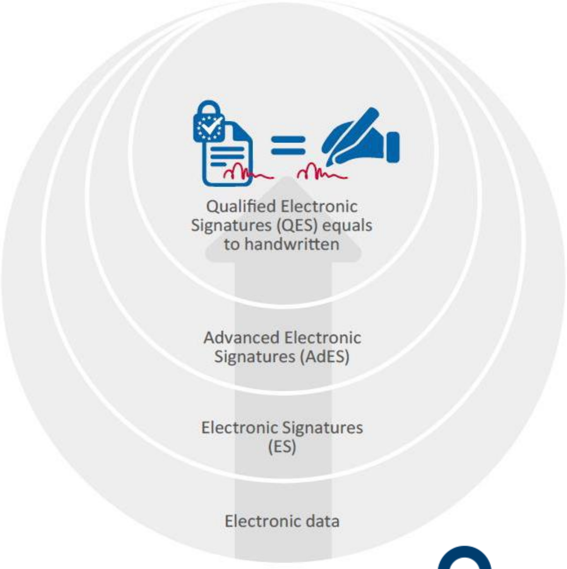
•Legal Value of QeS = same as handwritten signatures.

•OUT of SCOPE for ESSIF v1

Identity <> eSeal/Signing-keys

•DID <> eSeal-key – probably cumbersome

•DID <> Verifiable ID – flexible


Qualified Electronic Signatures (QES) equals to handwritten
Advanced Electronic Signatures (AdES)
Electronic Signatures (ES)
Electronic data

eIDAS-compliant

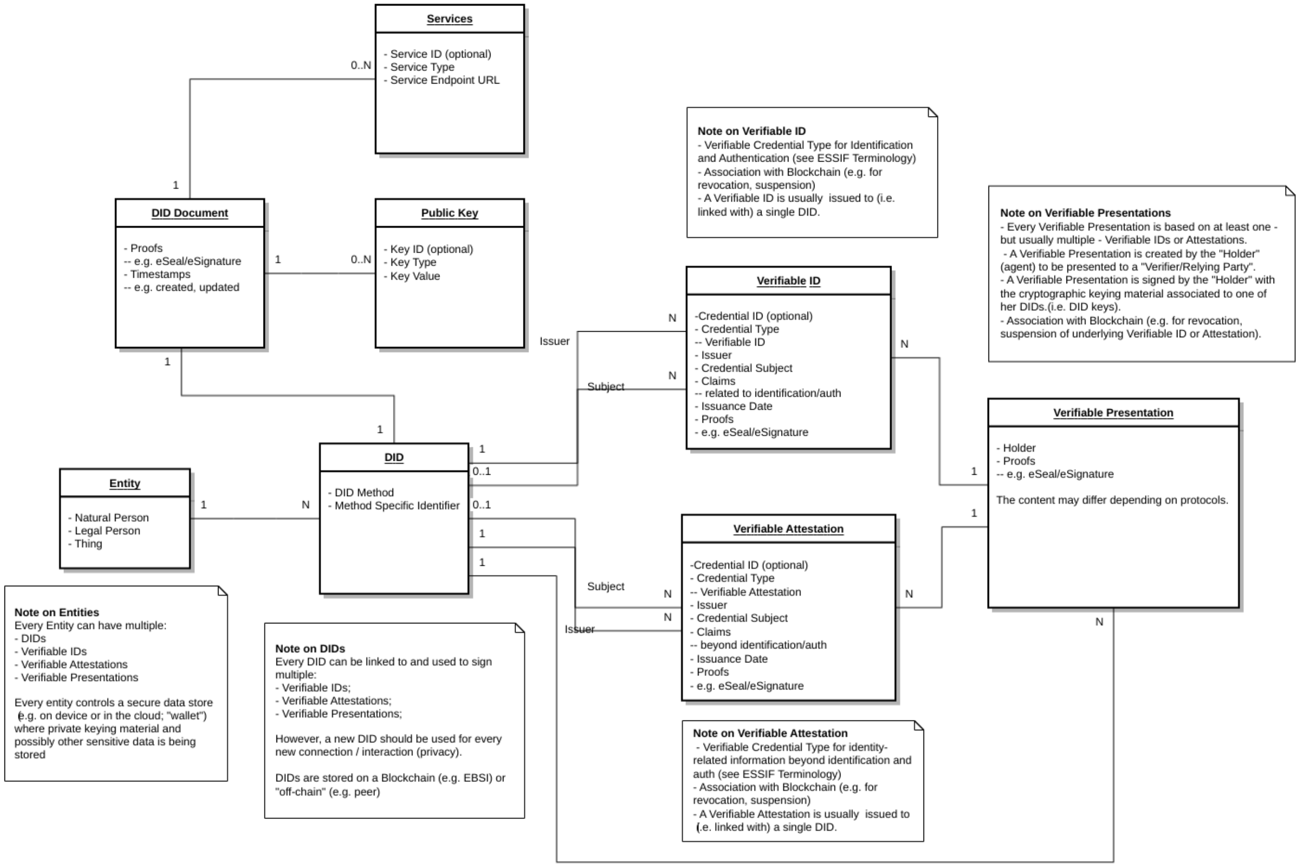
Resulting (flexible) DataModel

Key Properties:

- One entity CAN have multiple DIDs
- One DID can have multiple Verifiable IDs
- One DID can have have multiple Verifiable Attestations

Attributes:

- Verifiable IDs can be linked to eg GOV IDs
- Verifiable Attestations can inherit attributes from the “parenting” Verifiable ID



Note on Verifiable ID
- Verifiable Credential Type for Identification and Authentication (see ESSIF Terminology)
- Association with Blockchain (e.g. for revocation, suspension)
- A Verifiable ID is usually issued to (i.e. linked with) a single DID.

Note on Verifiable Presentations
- Every Verifiable Presentation is based on at least one - but usually multiple - Verifiable IDs or Attestations.
- A Verifiable Presentation is created by the "Holder" (agent) to be presented to a "Verifier/Relying Party".
- A Verifiable Presentation is signed by the "Holder" with the cryptographic keying material associated to one of her DIDs.(i.e. DID keys).
- Association with Blockchain (e.g. for revocation, suspension of underlying Verifiable ID or Attestation).

Note on Entities
Every Entity can have multiple:
- DIDs
- Verifiable IDs
- Verifiable Attestations
- Verifiable Presentations

Every entity controls a secure data store (e.g. on device or in the cloud; "wallet") where private keying material and possibly other sensitive data is being stored

Note on DIDs
Every DID can be linked to and used to sign multiple:
- Verifiable IDs;
- Verifiable Attestations;
- Verifiable Presentations;

However, a new DID should be used for every new connection / interaction (privacy).

DIDs are stored on a Blockchain (e.g. EBSI) or "off-chain" (e.g. peer)

Note on Verifiable Attestation
- Verifiable Credential Type for identity-related information beyond identification and auth (see ESSIF Terminology)
- Association with Blockchain (e.g. for revocation, suspension)
- A Verifiable Attestation is usually issued to (i.e. linked with) a single DID.

- ESSIF Vision
  - The Targeted eco-system
  - Targeted Business Use Cases
  - Longer Term View <> ESSIF v1
- ESSIF Datamodeling
  - Identities <> DIDs
  - Verifiable IDs <> Attestations
  - Links with LoA's
  - Links with Legal Value
  - Resulting (flexible) DataModel
- **ESSIF key Flows:**
  - **DID registrations**
  - **Obtaining a Verifiable ID**
  - **Obtaining a Verifiable Attestation**
  - **Details >> Link SSI and OIDC**
  - **Details >> Link with APIs**
- ESSIF (supporting) components
  - User / Issuer / Relying Party Environments
  - Trusted Issuer Ledger / eIDAS bridge
  - DID Registrars / Resolvers / Identity Hubs
- Preview / "under the hood"
  - Technology mapping

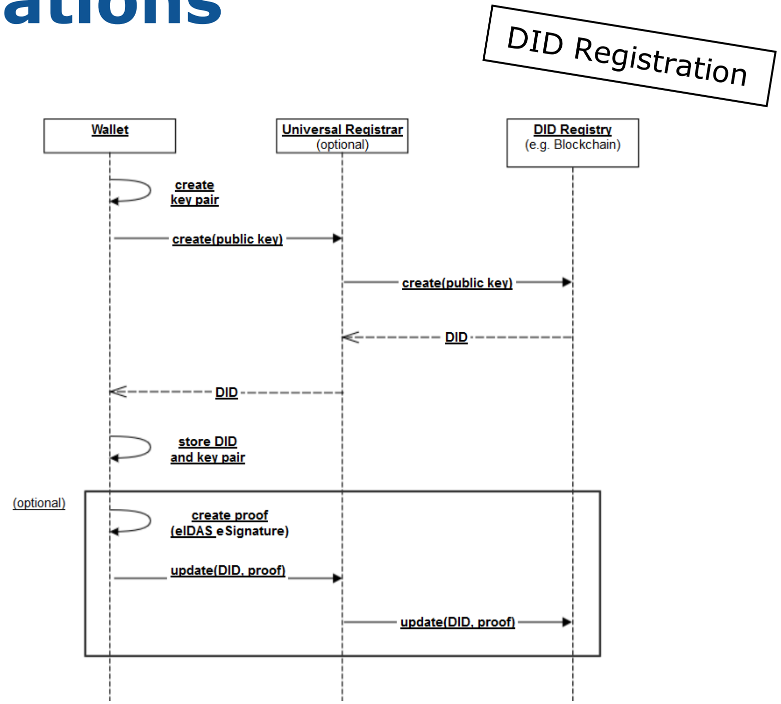
DID registrations

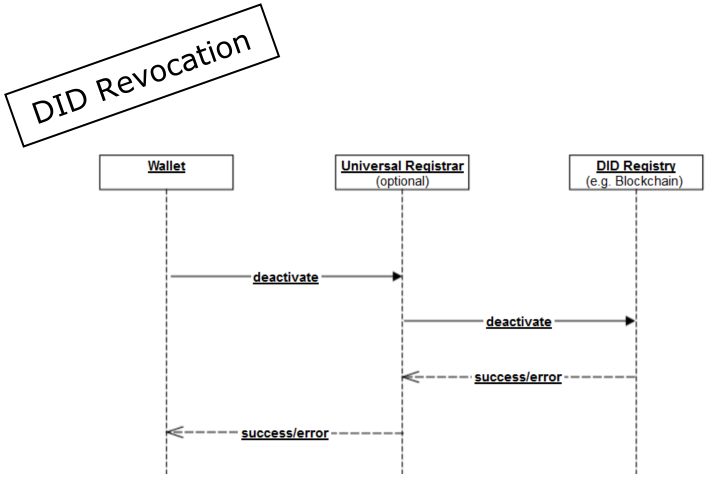
DID(-key) registration in ESSIF v1

- For Issuer / Relying Parties – ON LEDGER
- For Holders / Subjects – OFF LEDGER

DID(-key) revoaction in ESSIF v1

- For Issuer / Relying Parties – ON LEDGER
- For Holders / Subjects – ON LEDGER





Attention Points

- "Gating" of Registrations / Updates
- Authoritive ledgers for DID(-type)s

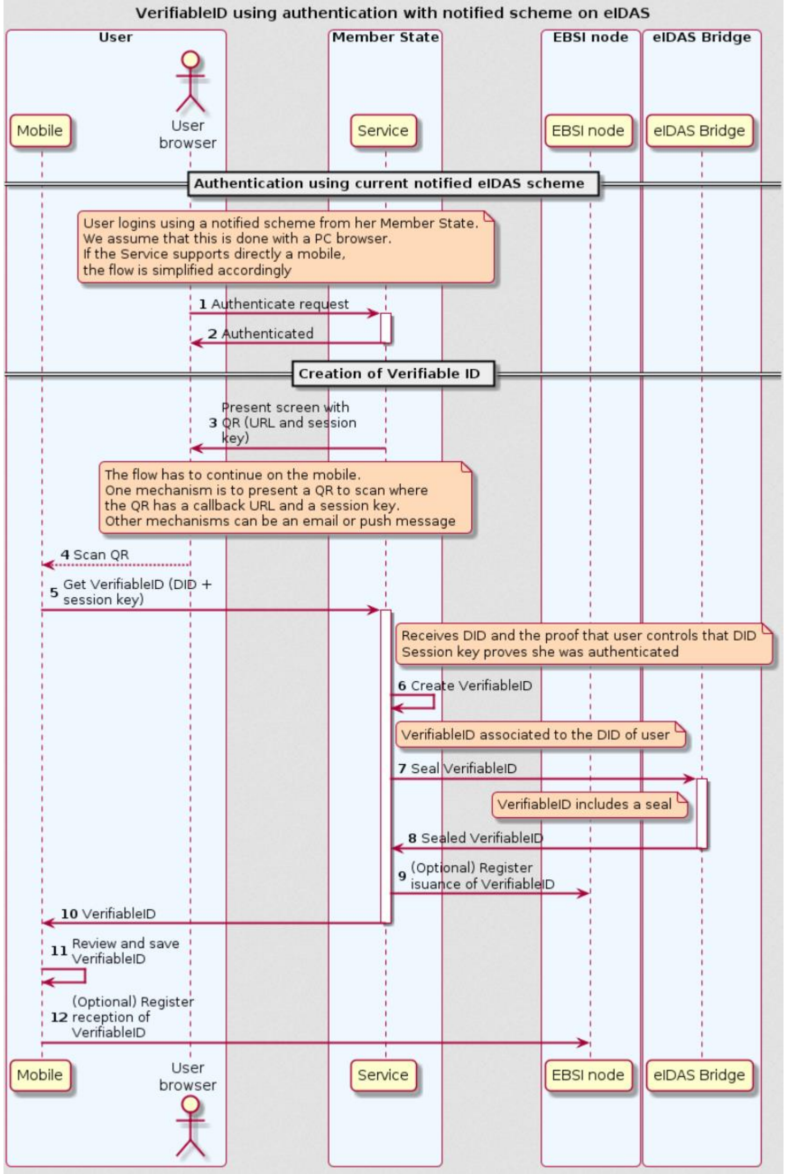
Obtaining a Verifiable ID

Flow:

- Trusted Issuer can be requested for a Verifiable ID
- Trusted Issuer can rely on eIDAS-authentication service to authenticate the holder/subject
- Trusted issuer needs to mind the LoA stated by the authentication service
- Trusted Issuer must check ownership (and strength) of the DID(-keys)

Properties:

- Verifiable ID should state LoAs
- Linked with (Qualified) Trusted Issuers
- Should be eSealed by Trusted Issuer in case of "High LoA"



VerifiableID using authentication with notified scheme on eIDAS

Authentication using current notified eIDAS scheme

User logins using a notified scheme from her Member State.
We assume that this is done with a PC browser.
If the Service supports directly a mobile,
the flow is simplified accordingly

1 Authenticate request

2 Authenticated

Creation of Verifiable ID

3 Present screen with QR (URL and session key)

The flow has to continue on the mobile.
One mechanism is to present a QR to scan where
the QR has a callback URL and a session key.
Other mechanisms can be an email or push message

4 Scan QR

5 Get VerifiableID (DID + session key)

Receives DID and the proof that user controls that DID
Session key proves she was authenticated

6 Create VerifiableID

VerifiableID associated to the DID of user

7 Seal VerifiableID

VerifiableID includes a seal

8 Sealed VerifiableID

9 (Optional) Register isuance of VerifiableID

10 VerifiableID

11 Review and save VerifiableID

12 (Optional) Register reception of VerifiableID

Obtaining a Verifiable Attestation

Flow:

•Issuer can be requested for a Verifiable Attestation

•Issuer should identify / authenticate the holder/subject relying on his/her Verifiable ID

•Issuer needs to validate the Verifiable ID (eSeal, DID-ownership, relevant attributes)

•Issuer to do any additional checks needed before generating a Verifiable Attestation

•Issuer generated Attestation

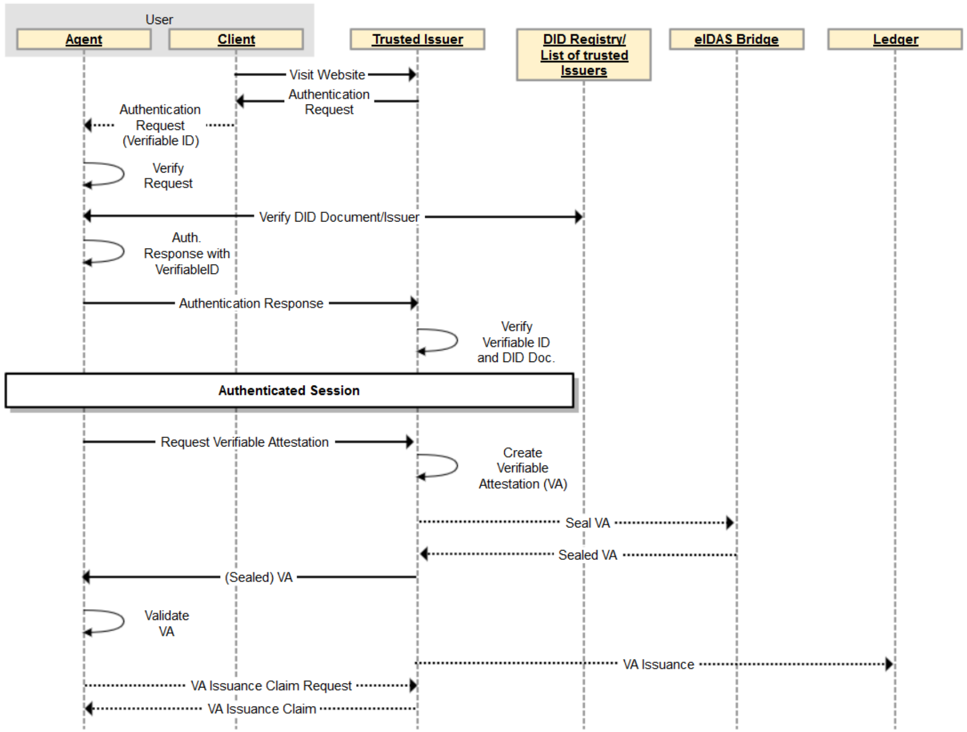
Properties:

•Verifiable Attestation should include type-info, LoA-info, …

•Should be linked with (Qualified / Trusted) Issuers

•Should be eSealed by Trusted Issuer in case of “High LoA”

Verifiable Attestation with Verifiable ID Authentication



User

Agent

Client

Trusted Issuer

DID Registry/ List of trusted Issuers

eIDAS Bridge

Ledger

Visit Website

Authentication Request

Authentication Request (Verifiable ID)

Verify Request

Verify DID Document/Issuer

Auth. Response with VerifiableID

Authentication Response

Verify Verifiable ID and DID Doc.

Authenticated Session

Request Verifiable Attestation

Create Verifiable Attestation (VA)

Seal VA

Sealed VA

(Sealed) VA

Validate VA

VA Issuance

VA Issuance Claim Request

VA Issuance Claim

European
Commission

Details >> Link SSI and OIDC

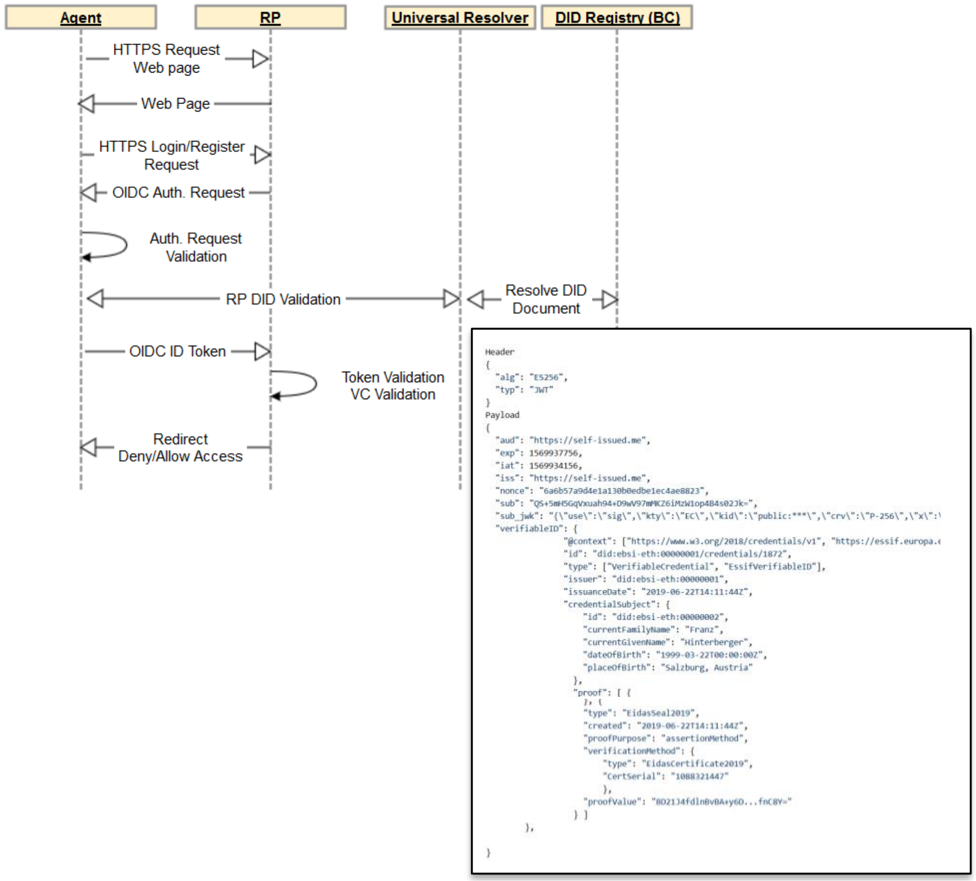
Point of Departure

- OIDC is the standard lots of online services use today
- In OIDC Relying Parties redirect users to an IDP to authenticate/identify users.
- IDP's provide IDtokens to Relying Parties.

Linking ID Tokens and V.IDs

- Proposal is to inject V.IDs into IDtoken and generated "self-declared" IDtokens
- Relying party can decide to trust V.IDs of certain LoAs of certain Trusted Issuers.
- Relying Party OIDC-client must be enabled to "consume" such tokens >> specific library needed to consume "self issued IDs"

OIDC SSI Authentication



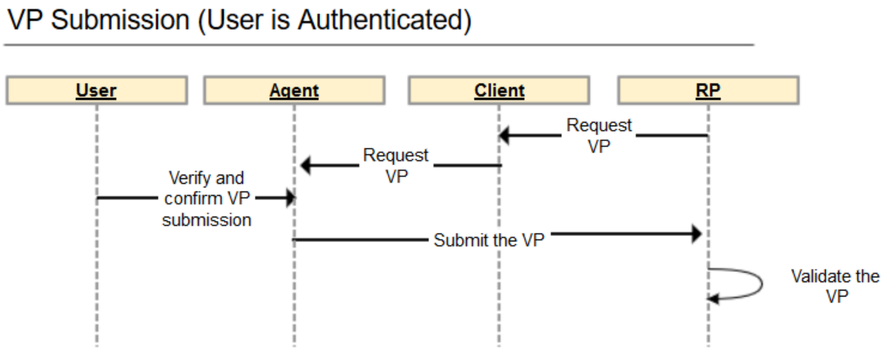
```
Header
{
  "alg": "ES256",
  "typ": "JWT"
}
Payload
{
  "aud": "https://self-issued.me",
  "exp": 1569937756,
  "iat": 1569934156,
  "iss": "https://self-issued.me",
  "nonce": "6a6b57a9d4e1a130b0edbe1ec4ae8823",
  "sub": "QS+5mH5GqVxuah94+D9wV97mMKZ6iMzW1op4B4s02Jk=",
  "sub_jwk": "{\"use\":\"sig\",\"kty\":\"EC\",\"kid\":\"public:***\",\"crv\":\"P-256\",\"x\":\
  "verifiableID": {
              "@context": ["https://www.w3.org/2018/credentials/v1", "https://essif.europa.e
              "id": "did:ebsi-eth:00000001/credentials/1872",
              "type": ["VerifiableCredential", "EssifVerifiableID"],
              "issuer": "did:ebsi-eth:00000001",
              "issuanceDate": "2019-06-22T14:11:44Z",
              "credentialSubject": {
                "id": "did:ebsi-eth:00000002",
                "currentFamilyName": "Franz",
                "currentGivenName": "Hinterberger",
                "dateOfBirth": "1999-03-22T00:00:00Z",
                "placeOfBirth": "Salzburg, Austria"
               },
               "proof": [ {
                 }, {
                "type": "EidasSeal2019",
                "created": "2019-06-22T14:11:44Z",
                "proofPurpose": "assertionMethod",
                "verificationMethod": {
                    "type": "EidasCertificate2019",
                    "CertSerial": "1088321447"
                    },
                "proofValue": "BD21J4fdlnBvBA+y6D...fnCBY="
               } ]
          },

}
```

Submitting an Attestation
(in an authenticated session)

Flow:

- User has authenticated the RP and RP has authenticated the user
- RP ask to (instead of filling in a form) to submit certain Verifiable Credentials
- User decided to provide (or not) the VCs and constructs needed Verifiable Presentation(s)
- User submits (over API) the VPs
- The RP checks if the VP-signature matches the Authenticated user.
- The RP checks the VCs (including type/version, eSeals, LoA's, …) and if needed relationship with the submitting user.


VP Submission (User is Authenticated)
User
Agent
Client
RP
Request VP
Request VP
Verify and confirm VP submission
Submit the VP
Validate the VP

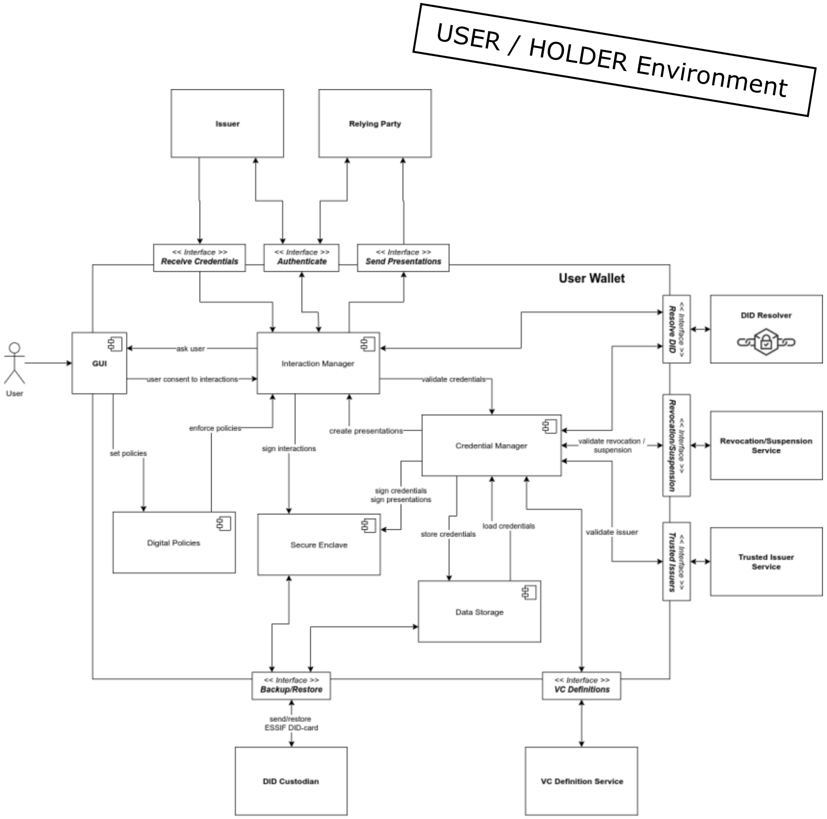
Properties:

- Verifiable Presentation must be signed with DID-key of the submittor and might be e-signed by the submittor
- RP should be be able to inform user which VCs will be accepted (issuer, type/version, LoA, …)

- ESSIF Vision
  - The Targeted eco-system
  - Targeted Business Use Cases
  - Longer Term View <> ESSIF v1
- ESSIF Datamodeling
  - Identities <> DIDs
  - Verifiable IDs <> Attestations
  - Links with LoA’s
  - Links with Legal Value
  - Resulting (flexible) DataModel
- ESSIF key Flows:
  - DID registrations
  - Obtaining a Verifiable ID
  - Obtaining a Verifiable Attestation
  - Details >> Link SSI and OIDC
  - Details >> Link with APIs
- **ESSIF (supporting) components**
  - **User / Issuer / Relying Party Environments**
  - **Trusted Issuer Ledger / eIDAS bridge**
  - **DID Registrars / Resolvers / Identity Hubs**
- Preview / “under the hood”
  - Technology mapping

User / Issuer / Relying Party Environments

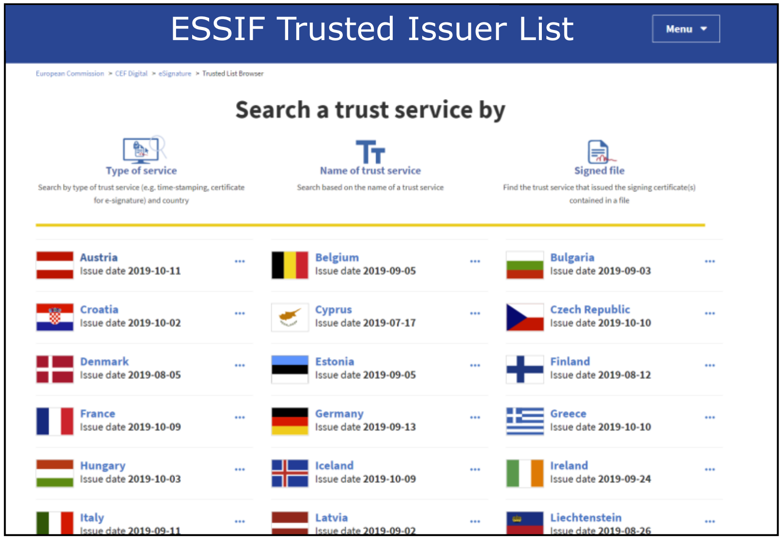
Graphical User Interface	**Interactions with the user**
Secure element	**Storing the private key in a secure manner**
	Exposing API endpoints for generating digital signatures, decrypting data encrypted for the private key contained in the secure element.
	Expose API endpoints for deriving additional key pairs, and for extracting the associated public keys.
Data Storage	**Offer generic storage capabilities (e.g. for storing digital policies, records of previous interactions, etc.)**
Credential Manager	**Creating VCs**
	process VCs
	Validating VCs
	Create VPs
Interaction Manager	**Create Interactions**
	Validate Interactions
	Create Interaction Responses
Digital Policies/Preferences	**store users policies/preferences**
	enforce digital polices within interactions


USER / HOLDER Environment

Trusted Issuer Ledger / eIDAS bridge

Trusted Issuer Ledger
in ESSIF v1 "on ledger"
Identity of Trusted Issuer
Types of VC and LoAs allowed
Registered by Issuer-Registrars

eIDAS Bridge
in ESSIF v1 only advanced eSeals
NO HSM or QeSCD
? OV-certificates ?
Link captured in issuer's V.ID



ESSIF Trusted Issuer List

European Commission > CEF Digital > eSignature > Trusted List Browser

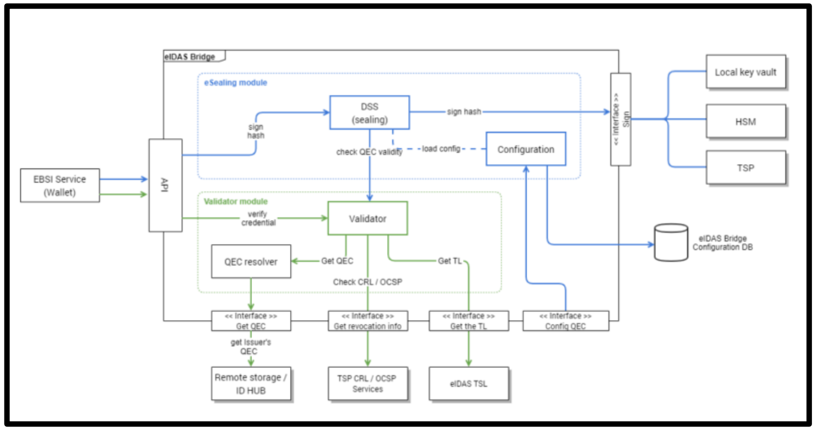
Search a trust service by

Type of service
Search by type of trust service (e.g. time-stamping, certificate for e-signature) and country

Name of trust service
Search based on the name of a trust service

Signed file
Find the trust service that issued the signing certificate(s) contained in a file

Austria Issue date 2019-10-11 ···	**Belgium** Issue date 2019-09-05 ···	**Bulgaria** Issue date 2019-09-03 ···
Croatia Issue date 2019-10-02 ···	**Cyprus** Issue date 2019-07-17 ···	**Czech Republic** Issue date 2019-10-10 ···
Denmark Issue date 2019-08-05 ···	**Estonia** Issue date 2019-09-05 ···	**Finland** Issue date 2019-08-12 ···
France Issue date 2019-10-09 ···	**Germany** Issue date 2019-09-13 ···	**Greece** Issue date 2019-10-10 ···
Hungary Issue date 2019-10-03 ···	**Iceland** Issue date 2019-10-09 ···	**Ireland** Issue date 2019-09-24 ···
Italy Issue date 2019-09-11 ···	**Latvia** Issue date 2019-09-02 ···	**Liechtenstein** Issue date 2019-08-26 ···



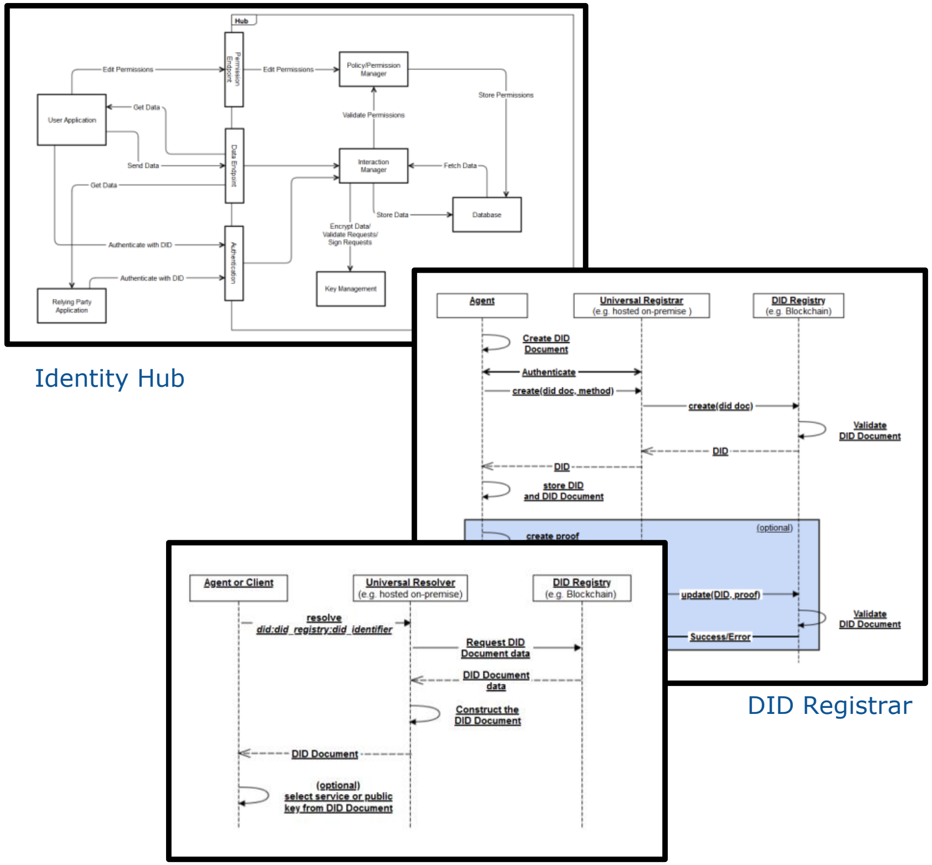
DID Registrars / Resolvers / Identity Hubs

DIF Identity Hub

- Providing users with a personal data store, allowing fully GDPR compliant storage of personal documents / info
- Can be "in the cloud" allowing the user to access one's DIDs / VCs anywhere and anytime
- Access by other agents can be provided subject to owner's or holder's consent.
- In ESSIF v1: only access by "owner"

Universal DID-resolver

- Allow user to interact with multiple DID-schemes and to "find" required info and endpoints
- Allows to provide DID-documents in format user agent understands.
- In ESSIF v1: EBSI-ledger only



Identity Hub

DID Registrar

DID resolver

- ESSIF Vision
  - The Targeted eco-system
  - Targeted Business Use Cases
  - Longer Term View <> ESSIF v1
- ESSIF Datamodeling
  - Identities <> DIDs
  - Verifiable IDs <> Attestations
  - Links with LoA's
  - Links with Legal Value
  - Resulting (flexible) DataModel
- ESSIF key Flows:
  - DID registrations
  - Obtaining a Verifiable ID
  - Obtaining a Verifiable Attestation
  - Details >> Link SSI and OIDC
  - Details >> Link with APIs
- ESSIF (supporting) components
  - User / Issuer / Relying Party Environments
  - Trusted Issuer Ledger / eIDAS bridge
  - DID Registrars / Resolvers / Identity Hubs
- **Preview / "under the hood"**
  - **Technology mapping**

Technology mapping

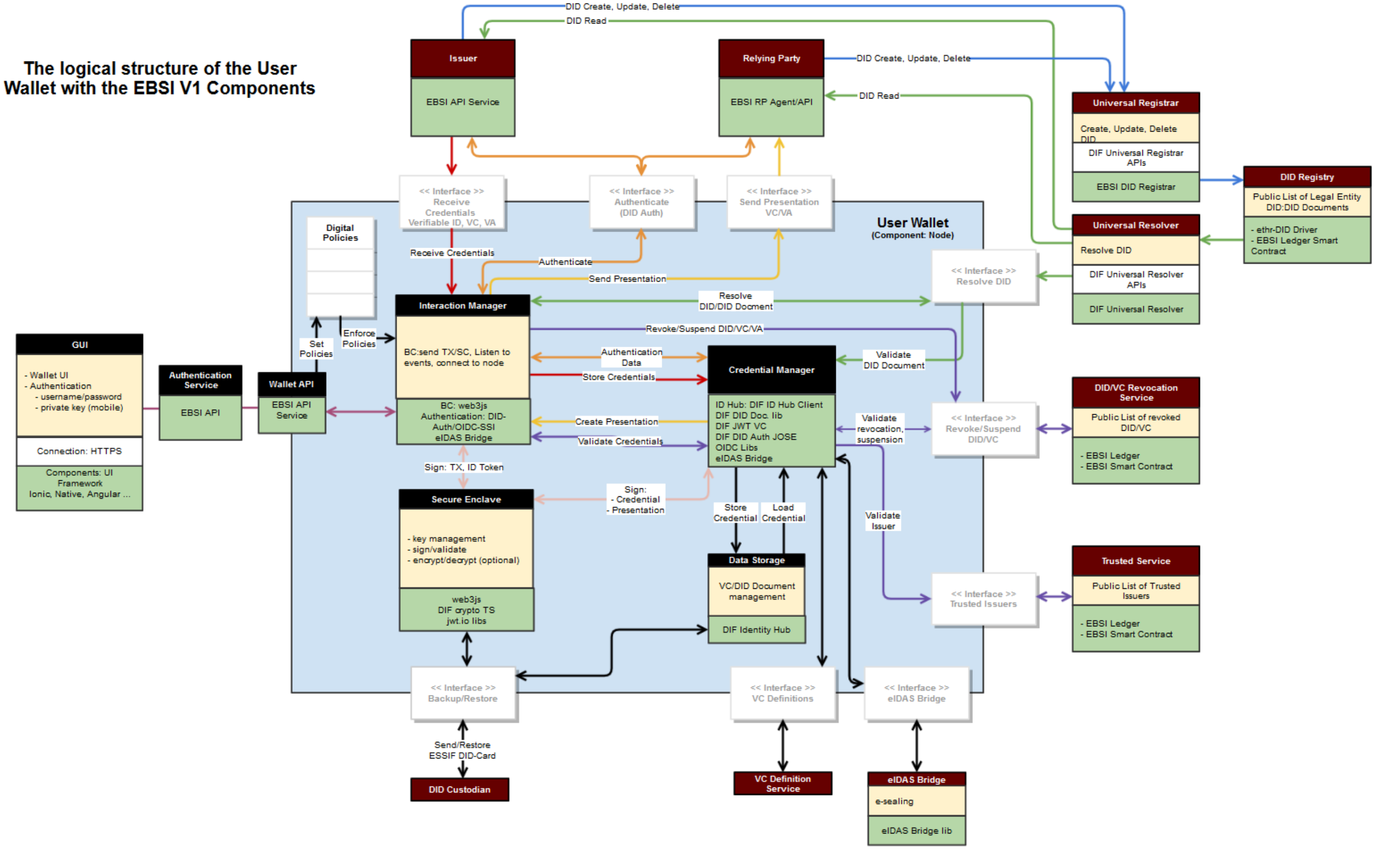
Reusing for ESSIF v1 lots of available code / libraries:

- For User Environment
- For Issuer Environment
- For Relying Party Environment

As well as:

- For Identity hub
- For DID Resolver
- For Ledger Anchoring

The logical structure of the User Wallet with the EBSI V1 Components



Issuer
EBSI API Service
DID Create, Update, Delete
DID Read
Relying Party
EBSI RP Agent/API
DID Create, Update, Delete
DID Read
Universal Registrar
Create, Update, Delete DID
DIF Universal Registrar APIs
EBSI DID Registrar
DID Registry
Public List of Legal Entity DID:DID Documents
- ethr-DID Driver
- EBSI Ledger Smart Contract
Universal Resolver
Resolve DID
DIF Universal Resolver APIs
DIF Universal Resolver
<< Interface >> Receive Credentials Verifiable ID, VC, VA
<< Interface >> Authenticate (DID Auth)
<< Interface >> Send Presentation VC/VA
User Wallet
(Component: Node)
Digital Policies
Receive Credentials
Authenticate
Send Presentation
Resolve DID/DID Docment
<< Interface >> Resolve DID
Revoke/Suspend DID/VC/VA
Set Policies
Enforce Policies
Interaction Manager
BC:send TX/SC, Listen to events, connect to node
BC: web3js
Authentication: DID-Auth/OIDC-SSI
eIDAS Bridge
Authentication Data
Store Credentials
Create Presentation
Validate Credentials
Validate DID Document
Credential Manager
ID Hub: DIF ID Hub Client
DIF DID Doc. lib
DIF JWT VC
DIF DID Auth JOSE
OIDC Libs
eIDAS Bridge
Validate revocation, suspension
<< Interface >> Revoke/Suspend DID/VC
DID/VC Revocation Service
Public List of revoked DID/VC
- EBSI Ledger
- EBSI Smart Contract
GUI
- Wallet UI
- Authentication
- username/password
- private key (mobile)
Connection: HTTPS
Components: UI Framework Ionic, Native, Angular ...
Authentication Service
EBSI API
Wallet API
EBSI API Service
Sign: TX, ID Token
Secure Enclave
- key management
- sign/validate
- encrypt/decrypt (optional)
web3js
DIF crypto TS
jwt.io libs
Sign:
- Credential
- Presentation
Store Credential
Load Credential
Validate Issuer
Data Storage
VC/DID Document management
DIF Identity Hub
<< Interface >> Trusted Issuers
Trusted Service
Public List of Trusted Issuers
- EBSI Ledger
- EBSI Smart Contract
<< Interface >> Backup/Restore
Send/Restore ESSIF DID-Card
DID Custodian
<< Interface >> VC Definitions
VC Definition Service
<< Interface >> eIDAS Bridge
eIDAS Bridge
e-sealing
eIDAS Bridge lib