Establishment of a Minimum Viable Self-Sovereign Identity Network

Kilian Käslin, 15.06.2020, Final Presentation Master's Thesis

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Outline

1. Introduction
    - Motivation
    - Research Questions
    - Research Approach
    - SSI Fundamentals

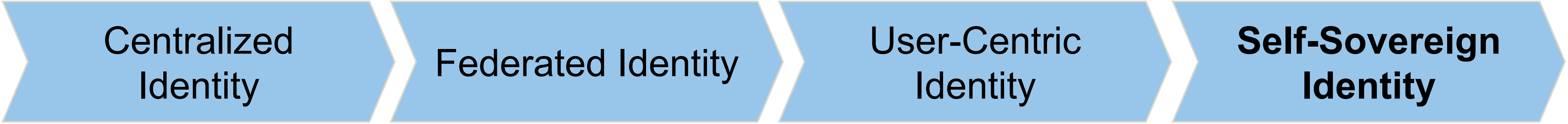
2. RQ3: User Journey

4. RQ5: Use-Cases

5. RQ6: Challenges

6. Conclusion and Future Work

Motivation



Centralized Identity

Federated Identity

User-Centric Identity

Self-Sovereign Identity

Self-Sovereign Identity (SSI): New identity management concept

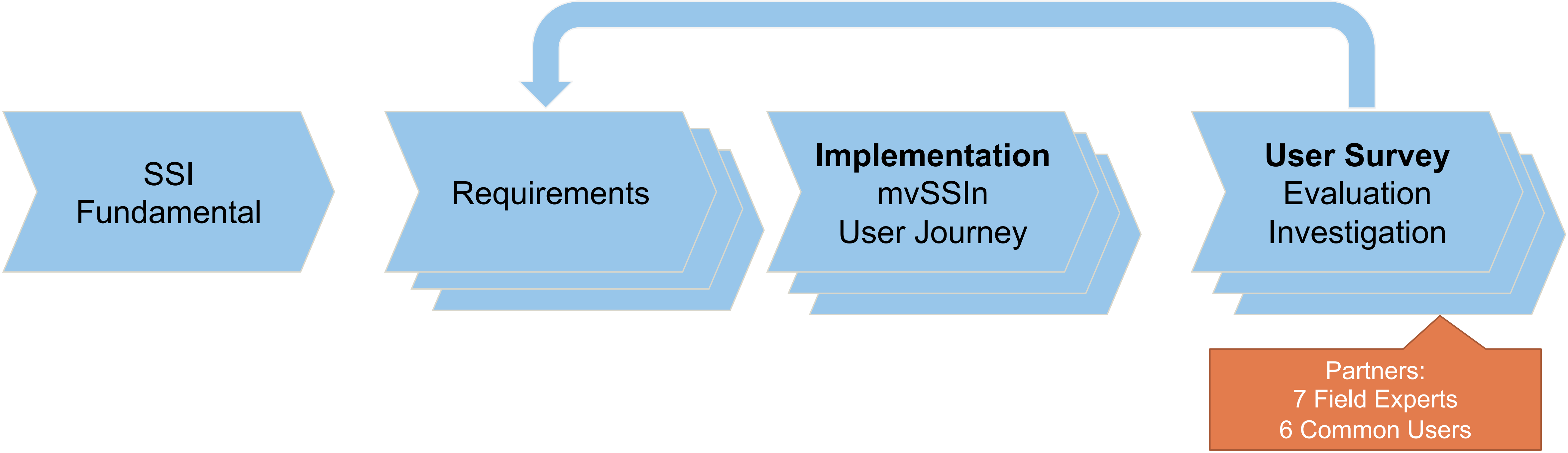
Proclaimed aim: Give the user control over his identity

Investigate the SSI concept:

How can a minimum viable SSI network be established?

The Path to Self-Sovereign Identity: http://www.lifewithalacrity.com/2016/04/the-path-to-self-sovereign-identity.html

Research Approach



SSI Fundamental

Requirements

Implementation
mvSSIn
User Journey

User Survey
Evaluation
Investigation

Partners:
7 Field Experts
6 Common Users

Contributions

Minimum viable Self-Sovereign Identity network:

Implementation ↔ User Journey

User Survey:

- SSI Use-Cases
- Challenges of SSI
- Issuer Desirability Function

Research Questions

RQ1 What are the essential requirements for the implementation of a minimum viable Self-Sovereign Identity network?

RQ2 How can a prototypical minimum viable Self-Sovereign Identity network be implemented?

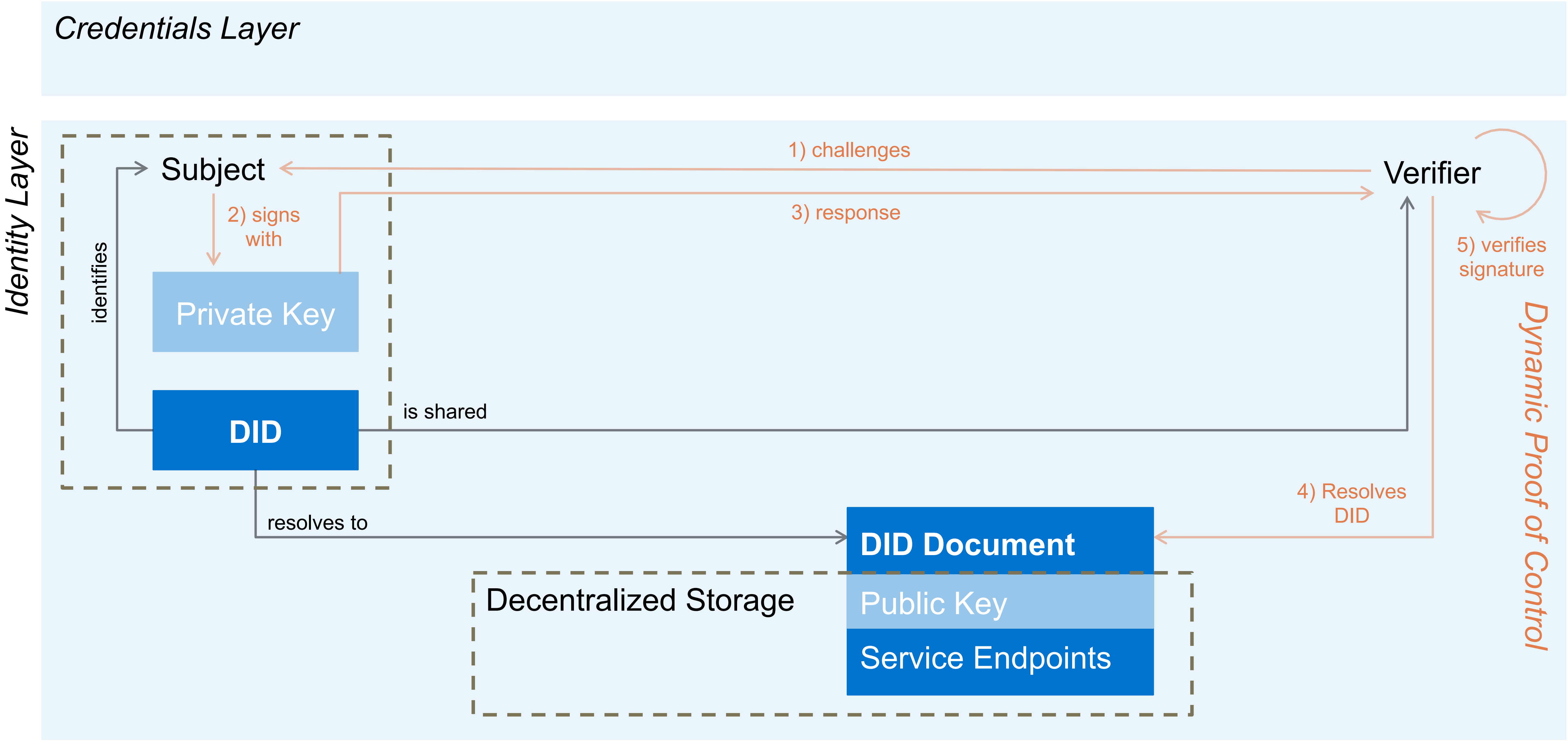
RQ3 What does a user journey taking place in the prototypical implementation of a minimum viable Self-Sovereign Identity network look like?

RQ4 How can the issuer desirability be modeled?

RQ5 Which Self-Sovereign Identity use-cases do field experts regard as candidates for first real-world Self-Sovereign Identity applications?

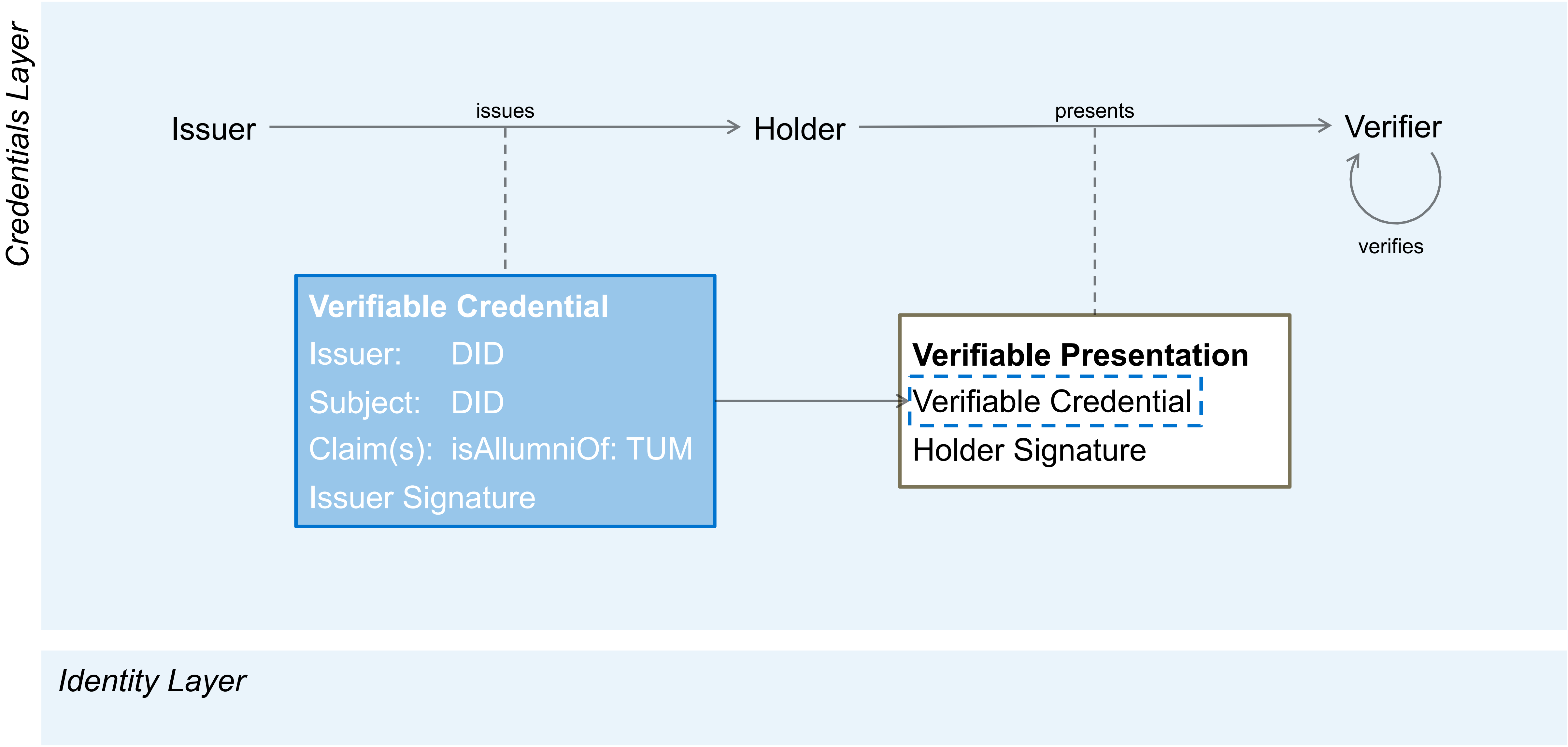
RQ6 Which challenges must be solved by the Self-Sovereign Identity community from the perspective of field experts to mature the Self-Sovereign Identity concept and its applications?

SSI Fundamentals



Decentralized Identifiers (DIDs) v1.0. https://www.w3.org/TR/did-core/

SSI Fundamentals



Credentials Layer

Issuer —issues→ Holder —presents→ Verifier

verifies

Verifiable Credential

Issuer: DID

Subject: DID

Claim(s): isAllumniOf: TUM

Issuer Signature

Verifiable Presentation

Verifiable Credential

Holder Signature

Identity Layer

Verifiable Credentials Data Model 1.0. https://www.w3.org/TR/vc-data-model/

Outline

1. Introduction
   - Motivation
   - Research Questions
   - Research Approach
   - SSI Fundamentals

2. RQ3: User Journey

4. RQ5: Use-Cases

5. RQ6: Challenges

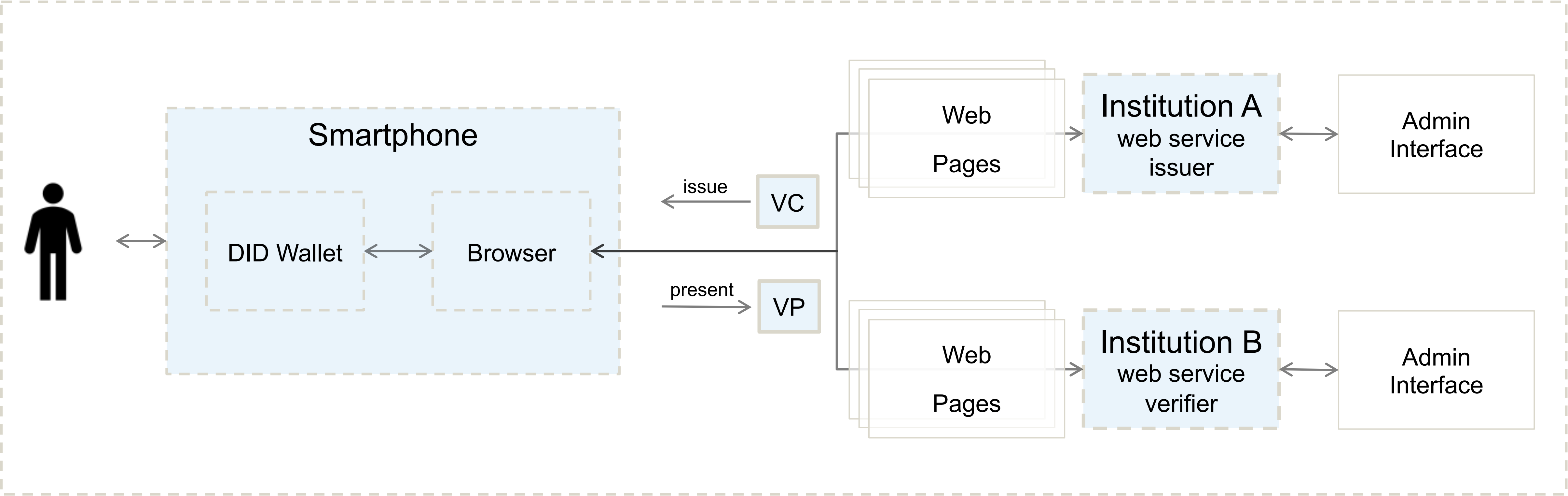
6. Conclusion and Future Work

RQ3: User Journey

Overview

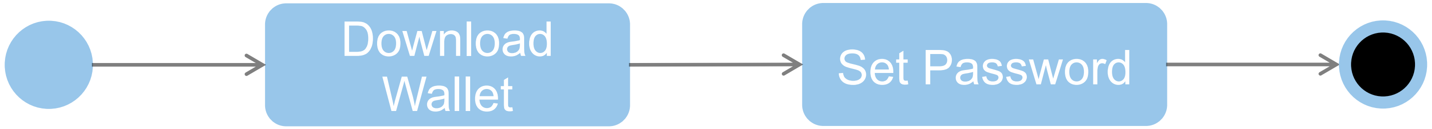
Use-Cases: Login with DID | Credentialed Access Management

Network Overview:



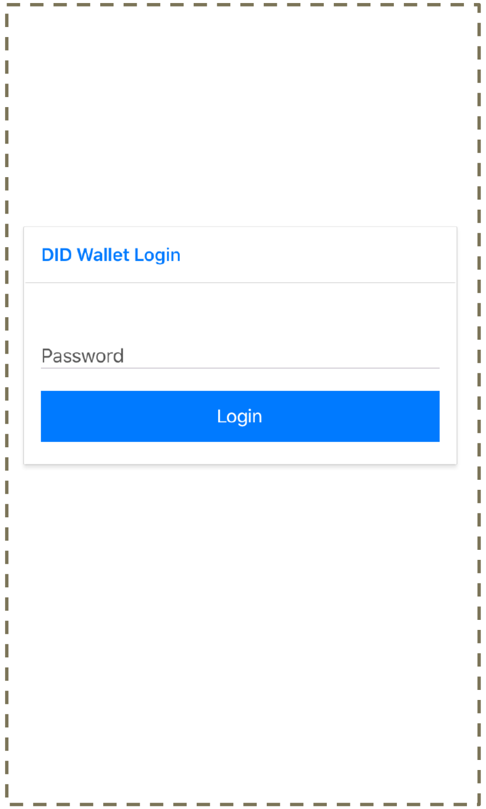
RQ3: User Journey

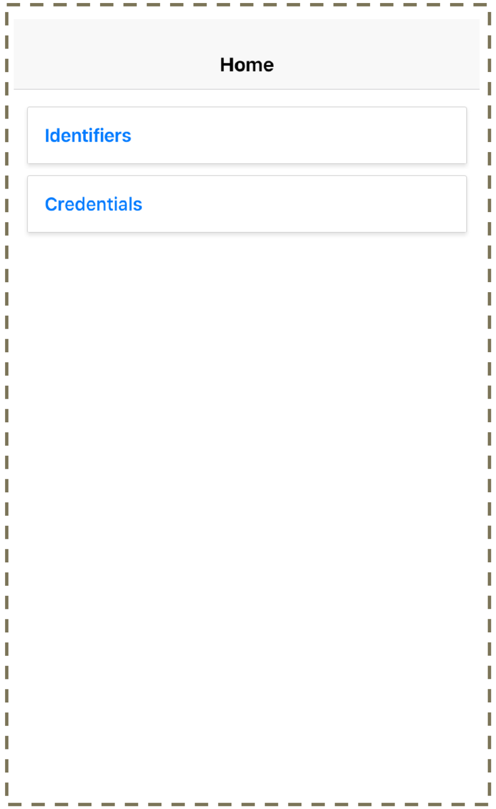
User – Setup



┄ DID Wallet

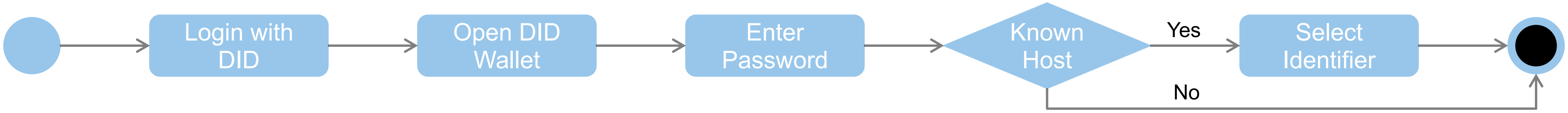
┄ Browser





RQ3: User Journey

User – DID Login



Login with DID

Open DID Wallet

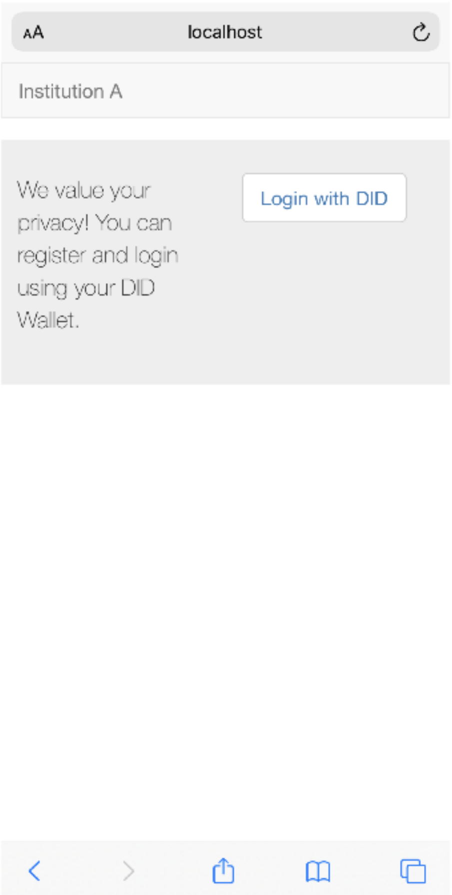
Enter Password

Known Host

Yes

Select Identifier

No

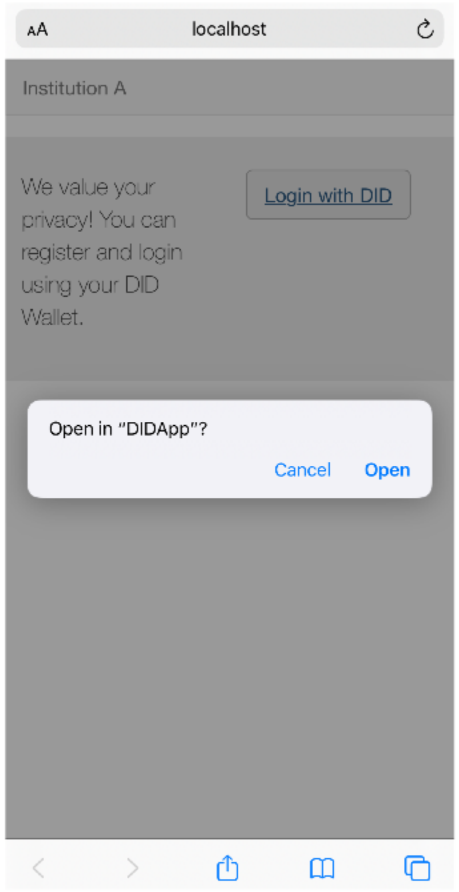


AA localhost

Institution A

We value your privacy! You can register and login using your DID Wallet.

Login with DID



AA localhost

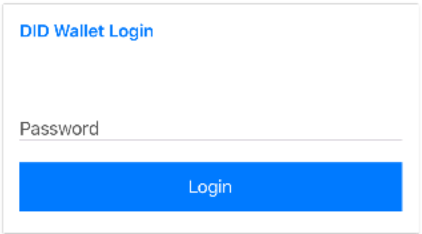
Institution A

We value your privacy! You can register and login using your DID Wallet.

Login with DID

Open in "DIDApp"?

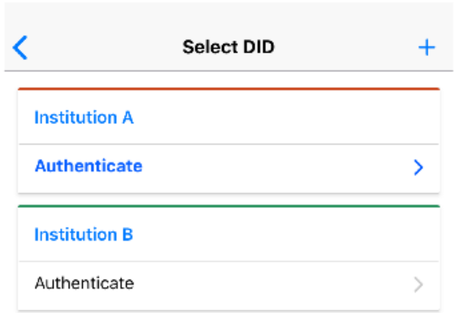
Cancel Open



DID Wallet Login

Password

Login



Select DID

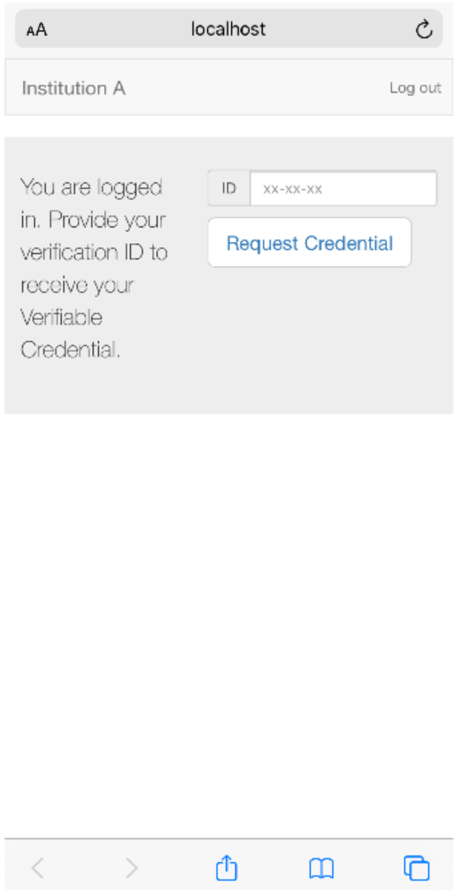
Institution A

Authenticate

Institution B

Authenticate

(optional)



AA localhost

Institution A

Log out

You are logged in. Provide your verification ID to receive your Verifiable Credential.

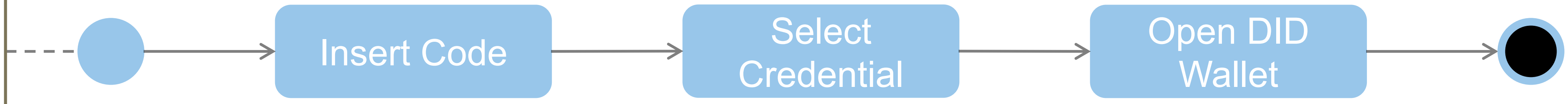
ID xx-xx-xx

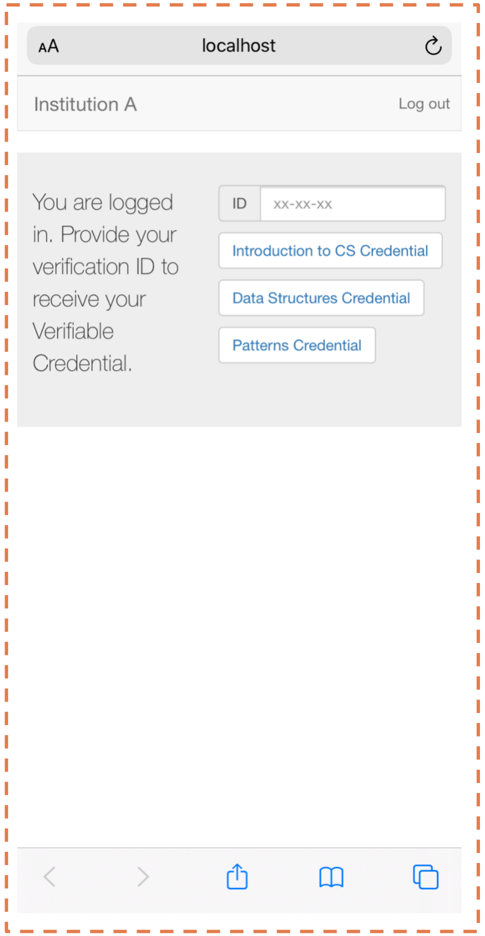
Request Credential

RQ3: User Journey

User – Request Credential

Prerequisites:
- Logged into Institution A
- Received Authentication Code




AA localhost

Institution A Log out

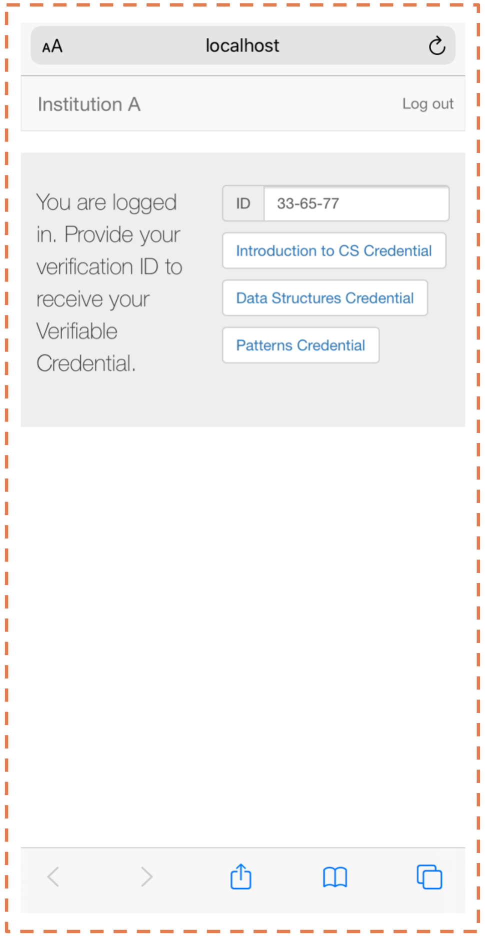
You are logged in. Provide your verification ID to receive your Verifiable Credential.

ID xx-xx-xx

Introduction to CS Credential

Data Structures Credential

Patterns Credential


AA localhost

Institution A Log out

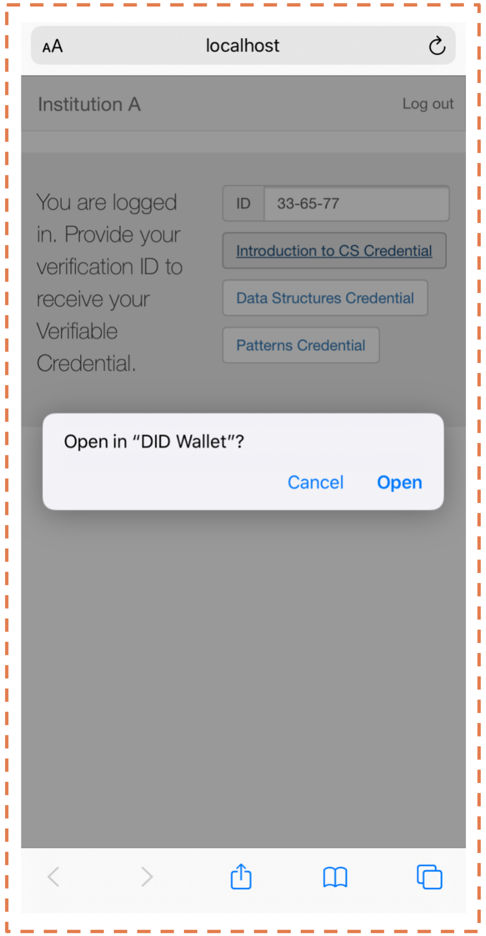
You are logged in. Provide your verification ID to receive your Verifiable Credential.

ID 33-65-77

Introduction to CS Credential

Data Structures Credential

Patterns Credential


AA localhost

Institution A Log out

You are logged in. Provide your verification ID to receive your Verifiable Credential.

ID 33-65-77

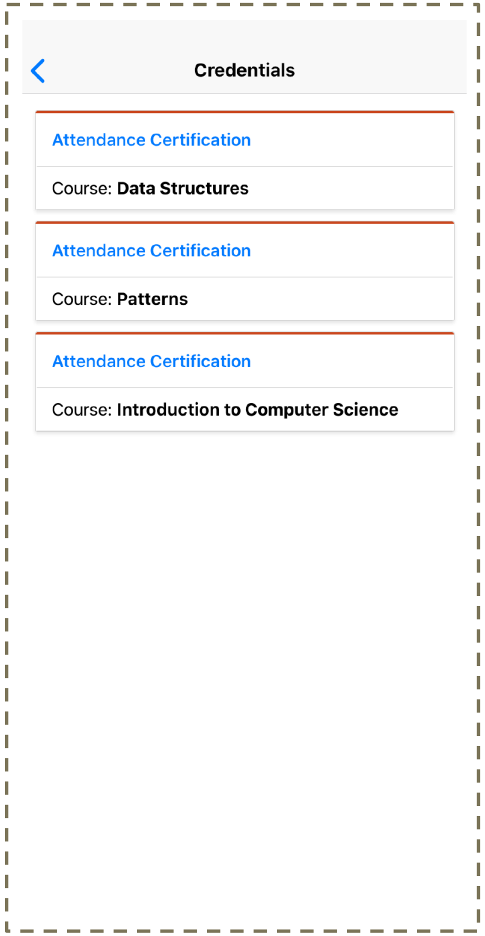
Introduction to CS Credential

Data Structures Credential

Patterns Credential

Open in “DID Wallet”?

Cancel Open


Credentials

Attendance Certification

Course: **Data Structures**

Attendance Certification

Course: **Patterns**

Attendance Certification

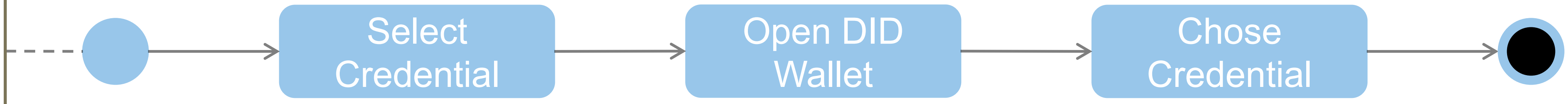
Course: **Introduction to Computer Science**

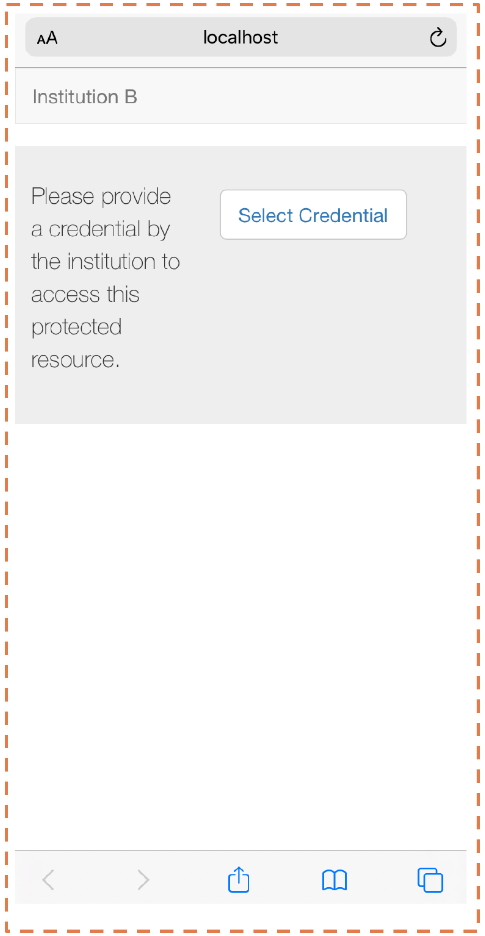
RQ3: User Journey

User – Present Credential

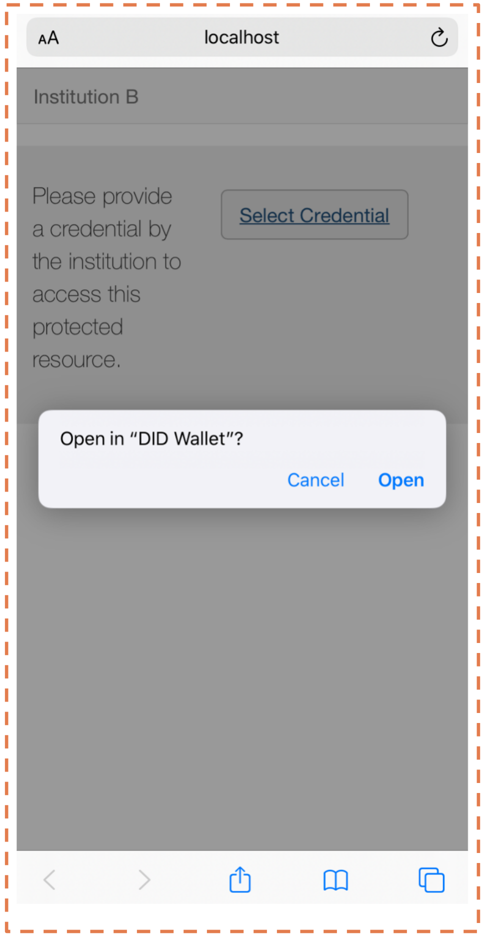
Prerequisites:

- Logged into Institution B
- Received Credential



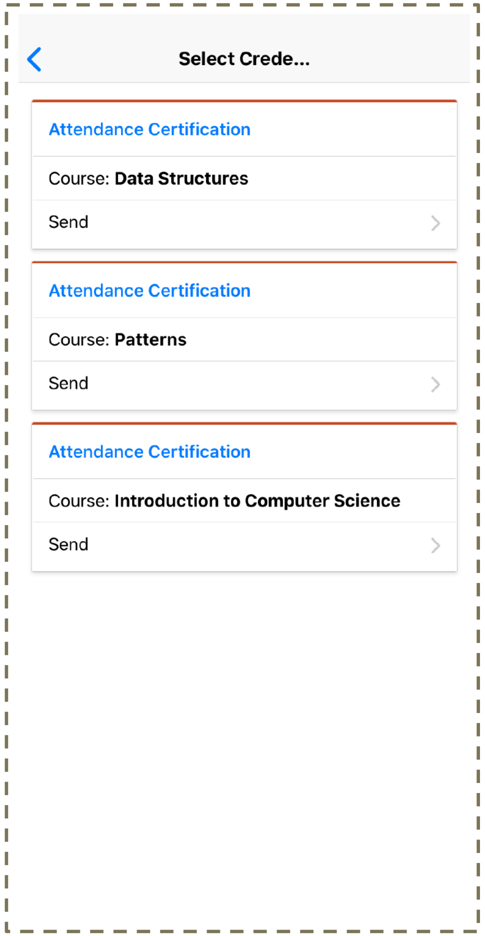


Please provide a credential by the institution to access this protected resource.



Please provide a credential by the institution to access this protected resource.

Open in “DID Wallet”?



Attendance Certification

Course: **Data Structures**

Send

Attendance Certification

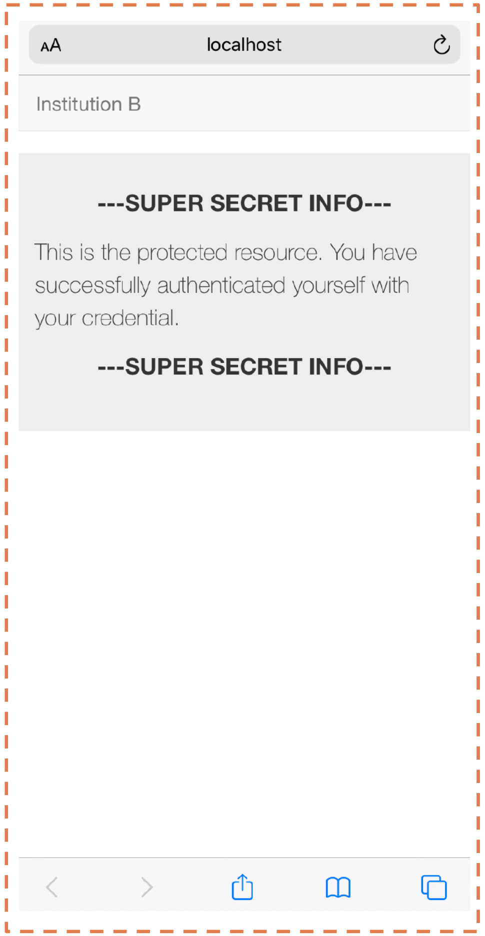
Course: **Patterns**

Send

Attendance Certification

Course: **Introduction to Computer Science**

Send



---SUPER SECRET INFO---

This is the protected resource. You have successfully authenticated yourself with your credential.

---SUPER SECRET INFO---

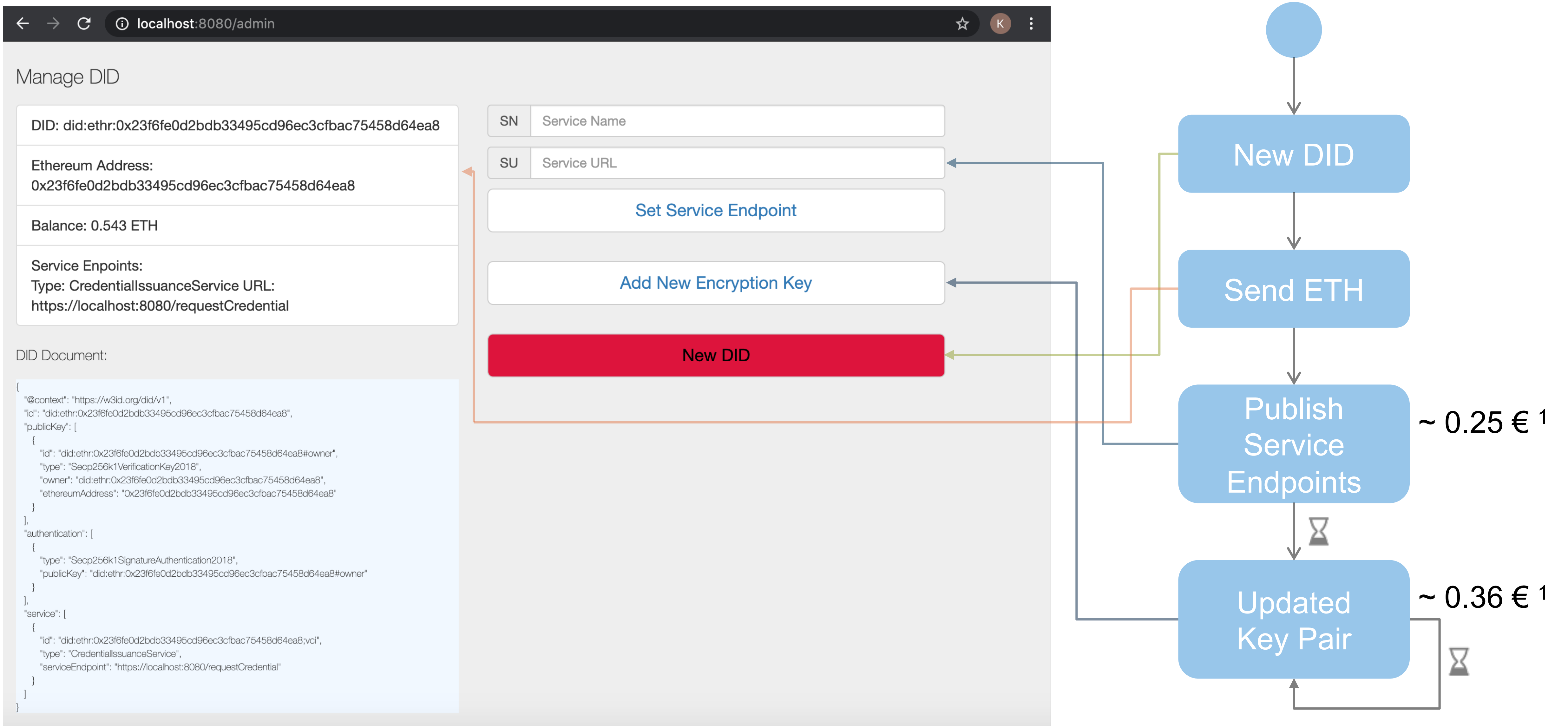
RQ3: User Journey

Institution - Preliminary

	Issuer	Verifier
Preliminary Questions	▪ What Credentials to issue? ▪ Who to issue Credentials to? ▪ How to connect DIDs to real-world identities?	▪ Which issuers are trusted? ▪ Which claims are expected?
Implementation	⇒ Attendance Certifications ⇒ Students attending a course ⇒ One-time code	⇒ Institution A ⇒ Attends Course: “Introduction to CS” | ”Data Structures” | “Patterns”

RQ3: User Journey

Institution



localhost:8080/admin

Manage DID

DID: did:ethr:0x23f6fe0d2bdb33495cd96ec3cfbac75458d64ea8

Ethereum Address:
0x23f6fe0d2bdb33495cd96ec3cfbac75458d64ea8

Balance: 0.543 ETH

Service Enpoints:
Type: CredentialIssuanceService URL:
https://localhost:8080/requestCredential

DID Document:

```
{
  "@context": "https://w3id.org/did/v1",
  "id": "did:ethr:0x23f6fe0d2bdb33495cd96ec3cfbac75458d64ea8",
  "publicKey": [
    {
      "id": "did:ethr:0x23f6fe0d2bdb33495cd96ec3cfbac75458d64ea8#owner",
      "type": "Secp256k1VerificationKey2018",
      "owner": "did:ethr:0x23f6fe0d2bdb33495cd96ec3cfbac75458d64ea8",
      "ethereumAddress": "0x23f6fe0d2bdb33495cd96ec3cfbac75458d64ea8"
    }
  ],
  "authentication": [
    {
      "type": "Secp256k1SignatureAuthentication2018",
      "publicKey": "did:ethr:0x23f6fe0d2bdb33495cd96ec3cfbac75458d64ea8#owner"
    }
  ],
  "service": [
    {
      "id": "did:ethr:0x23f6fe0d2bdb33495cd96ec3cfbac75458d64ea8;vci",
      "type": "CredentialIssuanceService",
      "serviceEndpoint": "https://localhost:8080/requestCredential"
    }
  ]
}
```

~ 0.25 € [1]

~ 0.36 € [1]

[1] *Conversion Rate: Gas:* ***34 GWEi, ETH 185*** *source:* https://ethgasstation.info/, https://coinmarketcap.com, accessed: 27.05.20

Extensions

We do not provide:

Key-recovery scenario | Password lost scenario

Identity import / export scenario to other wallets (Vendor Lock-In)

Institution GUI for VC creation and VP verification

In-app explanations

Outline

1. Introduction
   - Motivation
   - Research Questions
   - Research Approach
   - SSI Fundamentals

2. RQ3: User Journey

4. RQ5: Use-Cases

5. RQ6: Challenges

6. Conclusion and Future Work

RQ5: Use-Cases

User-Centric

Government Credentials
Issuer: Government
Holder: Citizens
Verifier: Government / Businesses / Citizens
VC: Government Documents

Educational Credentials
Issuer: Education Institutions
Holder: Students / Alumni
Verifier: Government / Businesses / Citizens
VC: Student / Graduation Credentials

KYC in Banking
Issuer: Banks
Holder: Bank Customers
Verifier: Banks
VC: Identification Credentials

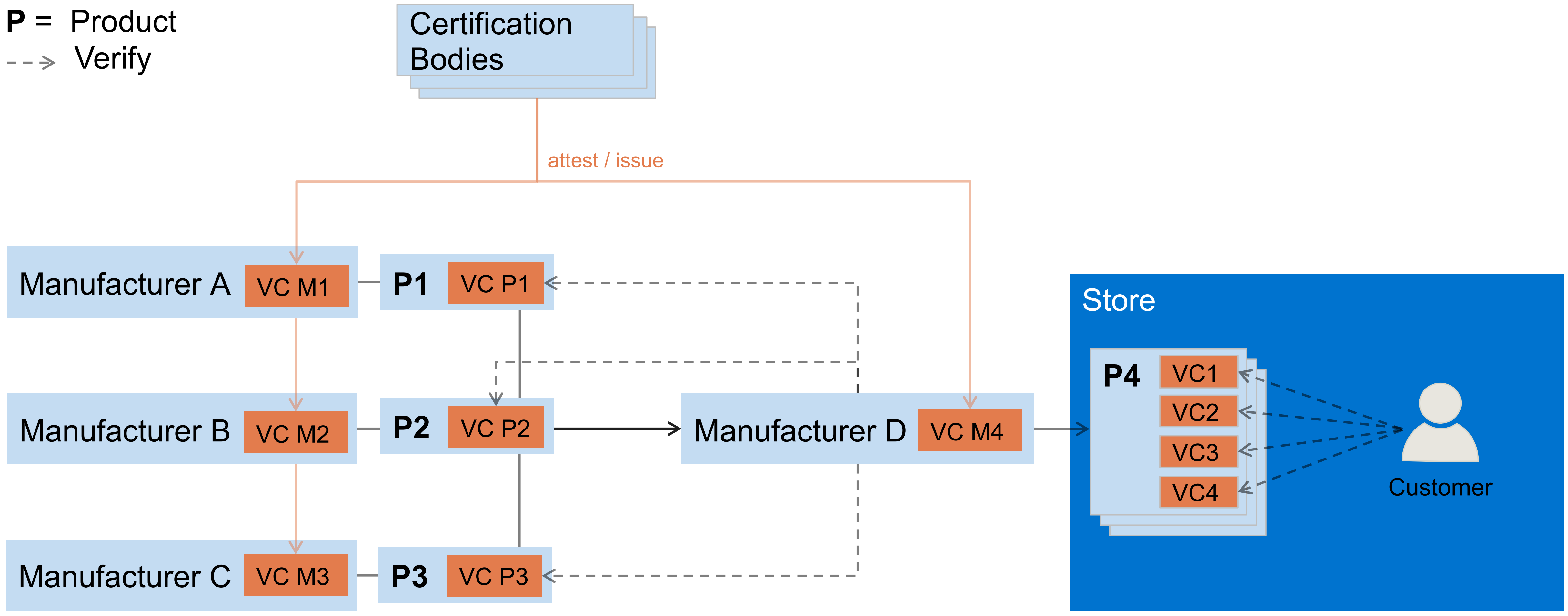
Enterprise

Government Credentials
Issuer: Government
Holder: Businesses
Verifier: Government / Businesses / Citizens
VC: Government Documents

Supply Chain Credentials
Issuer: Governments / Certification Bodies
Holder: Businesses / Products
Verifier: Government / Businesses / Consumers
VC: Production / Product Credentials

RQ5: Use-Cases

Supply Chain Credentials



Outline

1. Introduction
   - Motivation
   - Research Questions
   - Research Approach
   - SSI Fundamentals

2. RQ3: User Journey

4. RQ5: Use-Cases

5. RQ6: Challenges

6. Conclusion and Future Work

RQ6: Challenges

	Security and Trust	Regulations	User Experience & Market Acceptance
Issues	▪ What are the threat models of SSI and its applications? ▪ How to implement key recovery for common user?	▪ Compliance with GDPR: ▪ What can be stored on the Blockchain?	▪ How to map to the user's Identity Model? ▪ How to motivate institutions to issue VCs? ▪ Will market participants accept VPs and ZKPs?
Approach	▪ First prototypes with low level assurance and little Personally Identifiable Information	▪ Collaboration with Regulators ▪ Gold standard application	▪ Iterative approach to UI ▪ Market education ▪ Trust through gold standard application

Outline

1. Introduction
   - Motivation
   - Research Questions
   - Research Approach
   - SSI Fundamentals

2. RQ3: User Journey

4. RQ5: Use-Cases

5. RQ6: Challenges

6. Conclusion and Future Work

Conclusion

Implementation: It is possible to implement a mvSSIn based on current documentation, specification and open-source free-to-use software.

User Journey: It is possible to create a user-friendly user journey in a mvSSIn.

Centralization: The reliance on centralized trust anchors is often inherent to SSI use-cases.

Use-Cases: The SSI concept is applicable to a broad spectrum of use-cases. Interest from governments and companies exists to realize use-cases based on the SSI concept.

Challenges: Security, trust, regulatory, user experience and market acceptance challenges remain.

Future Work

- Implementation:
  - Extend the implementation by a key recovery system, identity import / export (interoperability)

- Investigate SSI use-cases:
  - Government
  - Educational
  - Supply Chain Credentials

- Investigate challenges:
  - Threat models
  - Regulations
  - Market acceptance

- Investigate Zero-Knowledge Proofs

Kilian Käslin

kilian.kaeslin@tum.de

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for Business
Information Systems

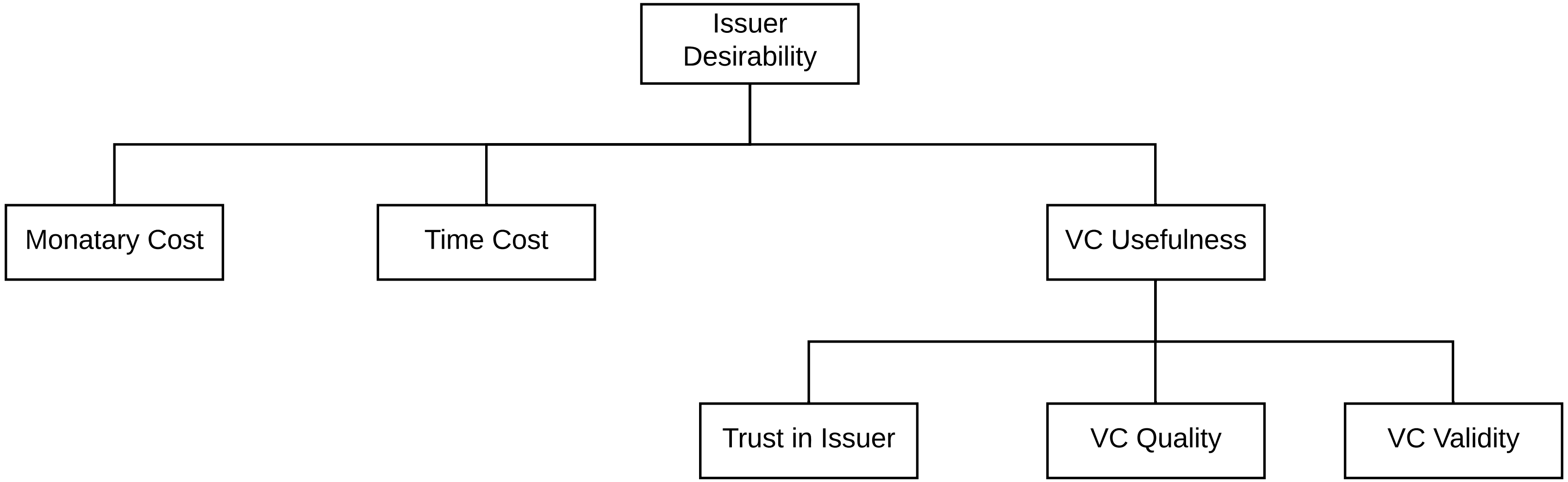
Boltzmannstraße 3
85748 Garching bei München

Tel +49.89.289.17132
Fax +49.89.289.17136

matthes@in.tum.de
wwwmatthes.in.tum.de

Backup Slides

RQ3: Issuer Desirability Function



$$\text{IssuerDesirability} = \mathit{MonetaryCost} \ast 0.125 + \mathit{TimeCost} \ast 0.125 + \mathit{TrustInIssuer} \ast 0.487 + \mathit{VCQuality} \ast 0.216 + \mathit{VCValidity} \ast 0.0616$$

Limitations

- Assumes use-case with issuer choice
- Assumes static use-case
- Issuer characteristics need to be translated to numeric values

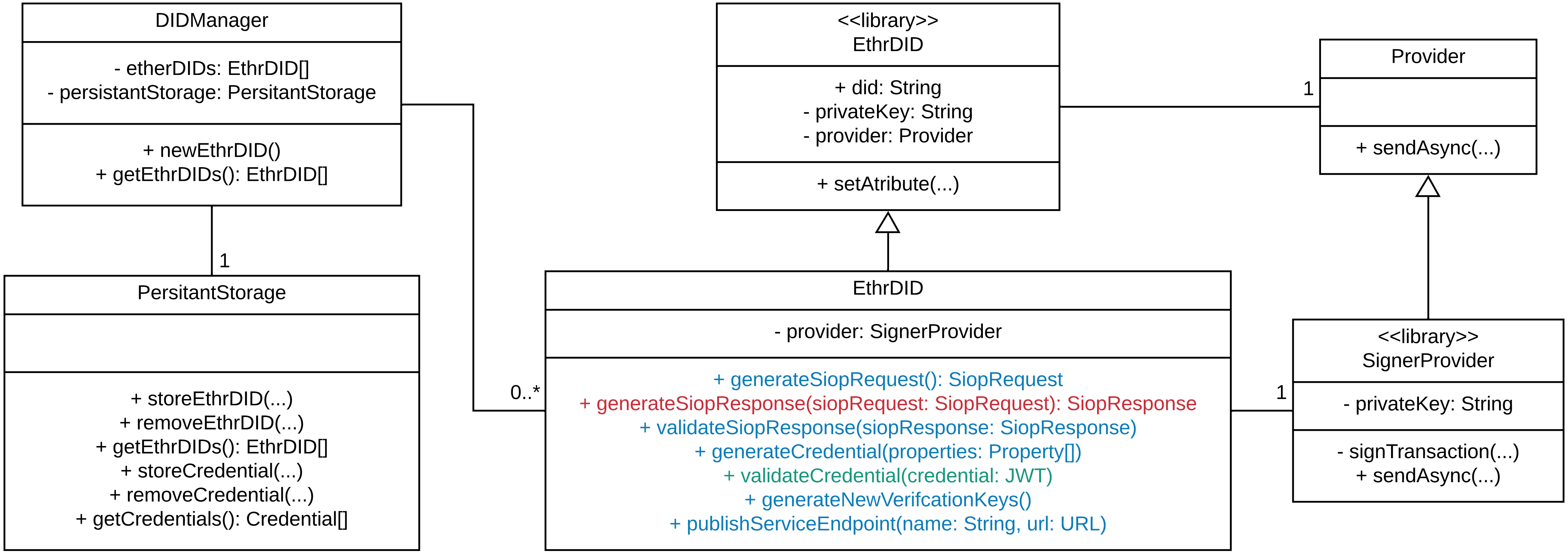
⇒ Inherent to most SSI use-cases to rely on centralized trust anchors

Fulfillment of Requirements

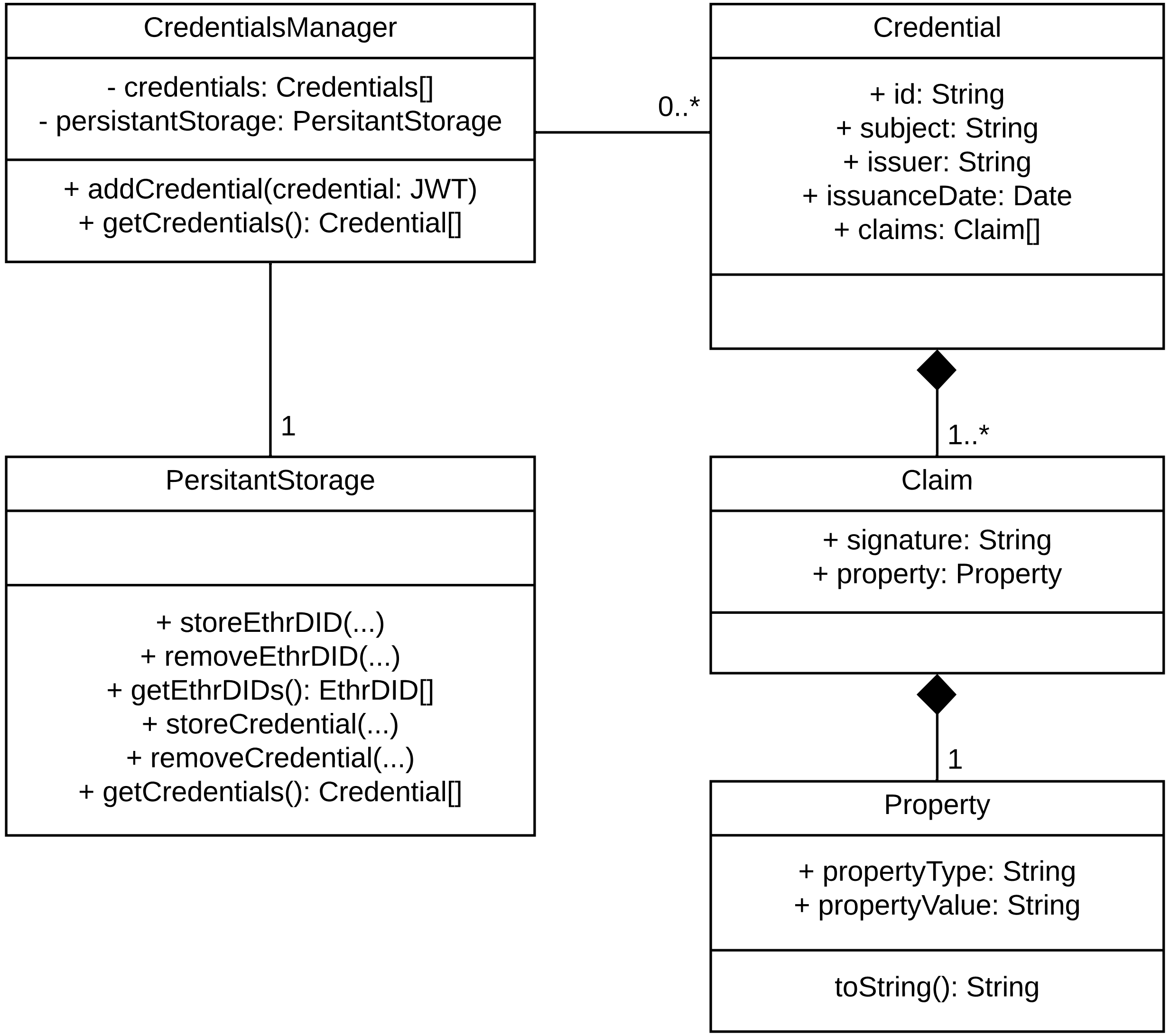
Functional Requirements	
Create DIDs	✓
Delete DIDs	✓
DID Login	✓
Request VCs	✓
Issue VCs	✓
Present VPs	✓
Out of wallet storage of VC	✗
Key Recovery	~
Key Rotation	~

Non-Functional Requirements	
Simple UI	✓
Explanatory UI	✗
Use the ETHR DID Method	✓
Extend web services	✓
DID Login	✓
Paired DIDs	✓
DID communication	✗

Implementation – Distributed Identifiers


DIDManager
- etherDIDs: EthrDID[]
- persistantStorage: PersitantStorage
+ newEthrDID()
+ getEthrDIDs(): EthrDID[]
1
PersitantStorage
+ storeEthrDID(...)
+ removeEthrDID(...)
+ getEthrDIDs(): EthrDID[]
+ storeCredential(...)
+ removeCredential(...)
+ getCredentials(): Credential[]
0..*
<<library>>
EthrDID
+ did: String
- privateKey: String
- provider: Provider
+ setAtribute(...)
1
Provider
+ sendAsync(...)
EthrDID
- provider: SignerProvider
+ generateSiopRequest(): SiopRequest
+ generateSiopResponse(siopRequest: SiopRequest): SiopResponse
+ validateSiopResponse(siopResponse: SiopResponse)
+ generateCredential(properties: Property[])
+ validateCredential(credential: JWT)
+ generateNewVerifcationKeys()
+ publishServiceEndpoint(name: String, url: URL)
1
<<library>>
SignerProvider
- privateKey: String
- signTransaction(...)
+ sendAsync(...)

Implementation - Credentials



CredentialsManager

- credentials: Credentials[]
- persistantStorage: PersitantStorage

+ addCredential(credential: JWT)
+ getCredentials(): Credential[]

0..*

Credential

+ id: String
+ subject: String
+ issuer: String
+ issuanceDate: Date
+ claims: Claim[]

1

PersitantStorage

+ storeEthrDID(...)
+ removeEthrDID(...)
+ getEthrDIDs(): EthrDID[]
+ storeCredential(...)
+ removeCredential(...)
+ getCredentials(): Credential[]

1..*

Claim

+ signature: String
+ property: Property

1

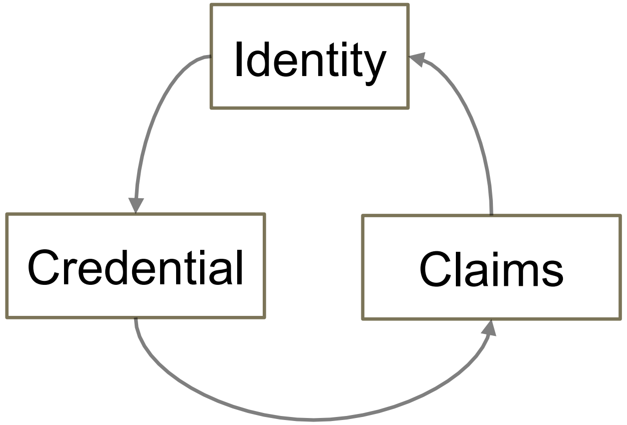
Property

+ propertyType: String
+ propertyValue: String

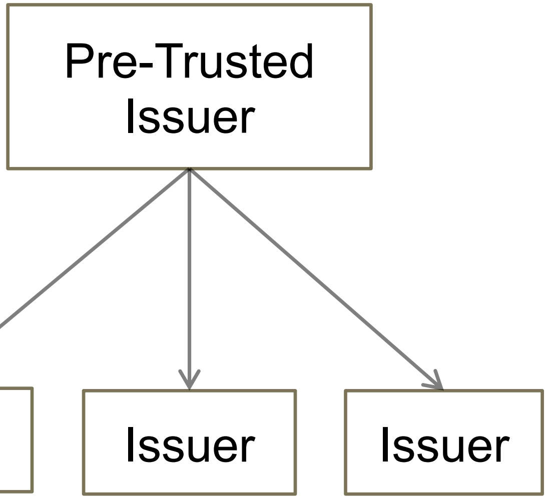
toString(): String

Trust Model

Circular Trust Model:



Pre-trusted issuers induce trust to new issuers:



A. Gruner, A. Muhle, T. Gayvoronskaya, and C. Meinel. “A quantifiable trust model for blockchain-based identity management.” In: Institute of Electrical and Electronics Engineers Inc., July 2018, pp. 1475–1482.