Leveraging TLS/SSL-based Identity Assertion and Verification Systems for on-chain authentication and authorization of real-world entities

Jan-Niklas Strugala, 11.01.2021, Final Presentation Master's Thesis

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

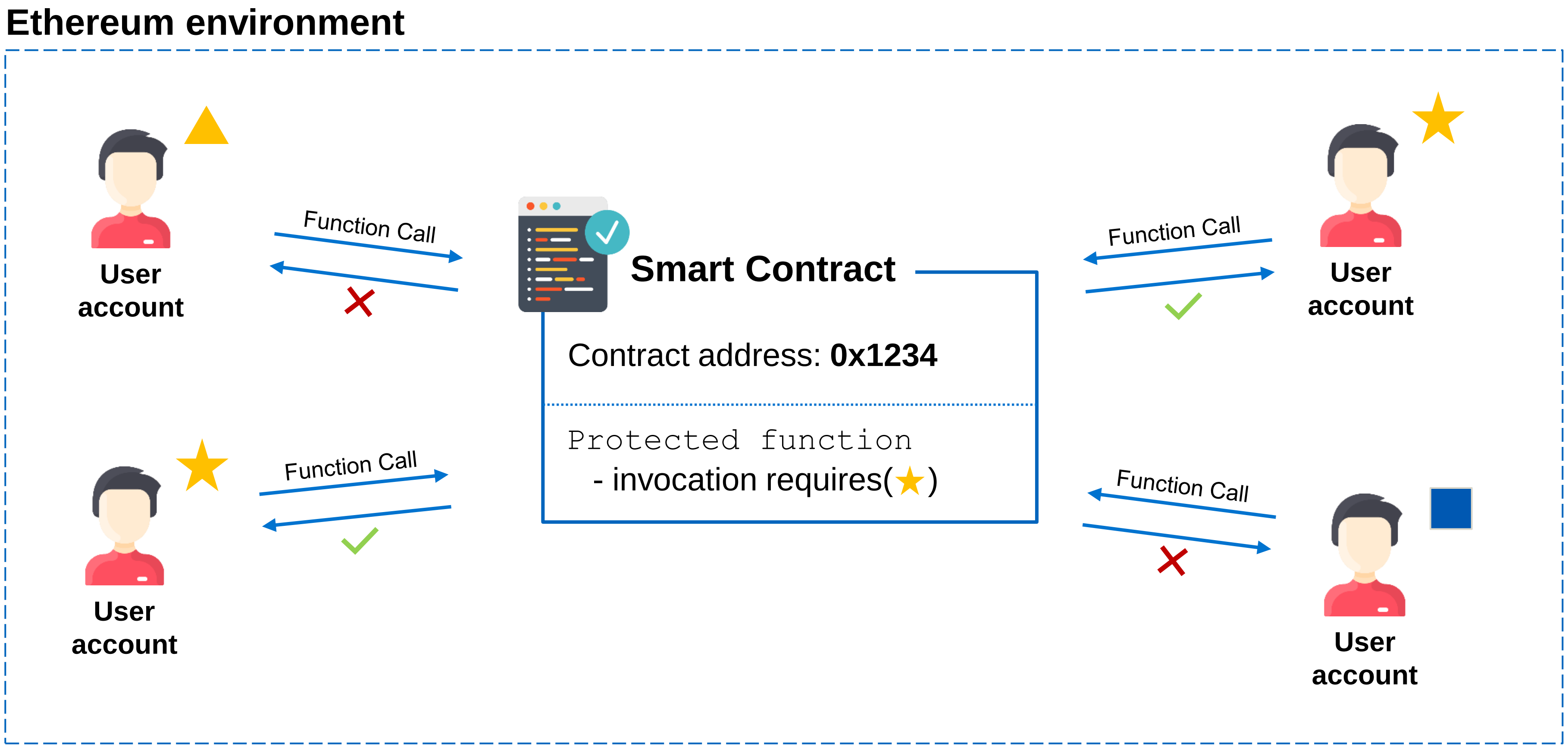
Outline

1. Introduction
   - Motivation
   - Conceptual Design & Problem Statement
   - Research Approach & Contribution
2. Research Questions
3. Evaluation
4. Discussion
   - Conclusion
   - Future Work

Motivation

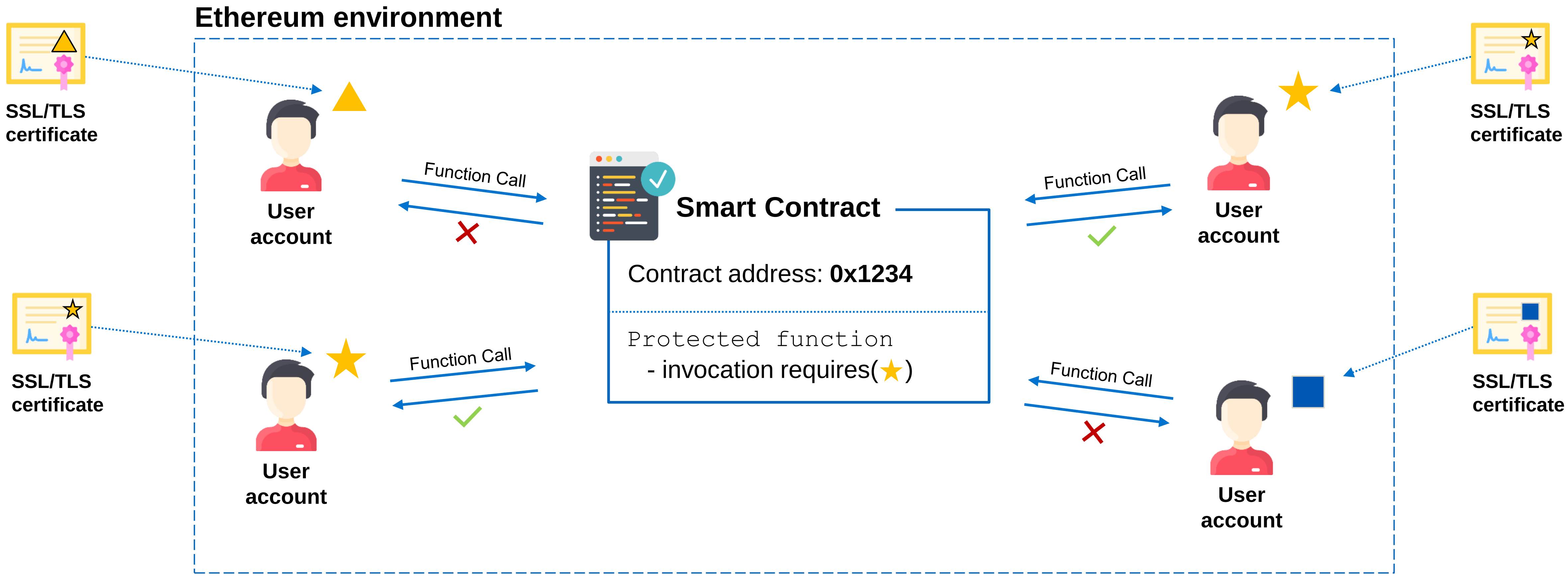
Enable authentication and access control at smart contracts

Protect access to SC functions such that they can only be invoked by accounts with certain characteristics



Motivation

Endow an Ethereum account of a real-world entity with trust from SSL/TLS certificates
Leverage the SSL/TLS certificate PKI to obtain and use trusted characteristics



Motivation

Enable **authentication** and **access control** of real-world identities at smart contracts, based on requester's characteristics

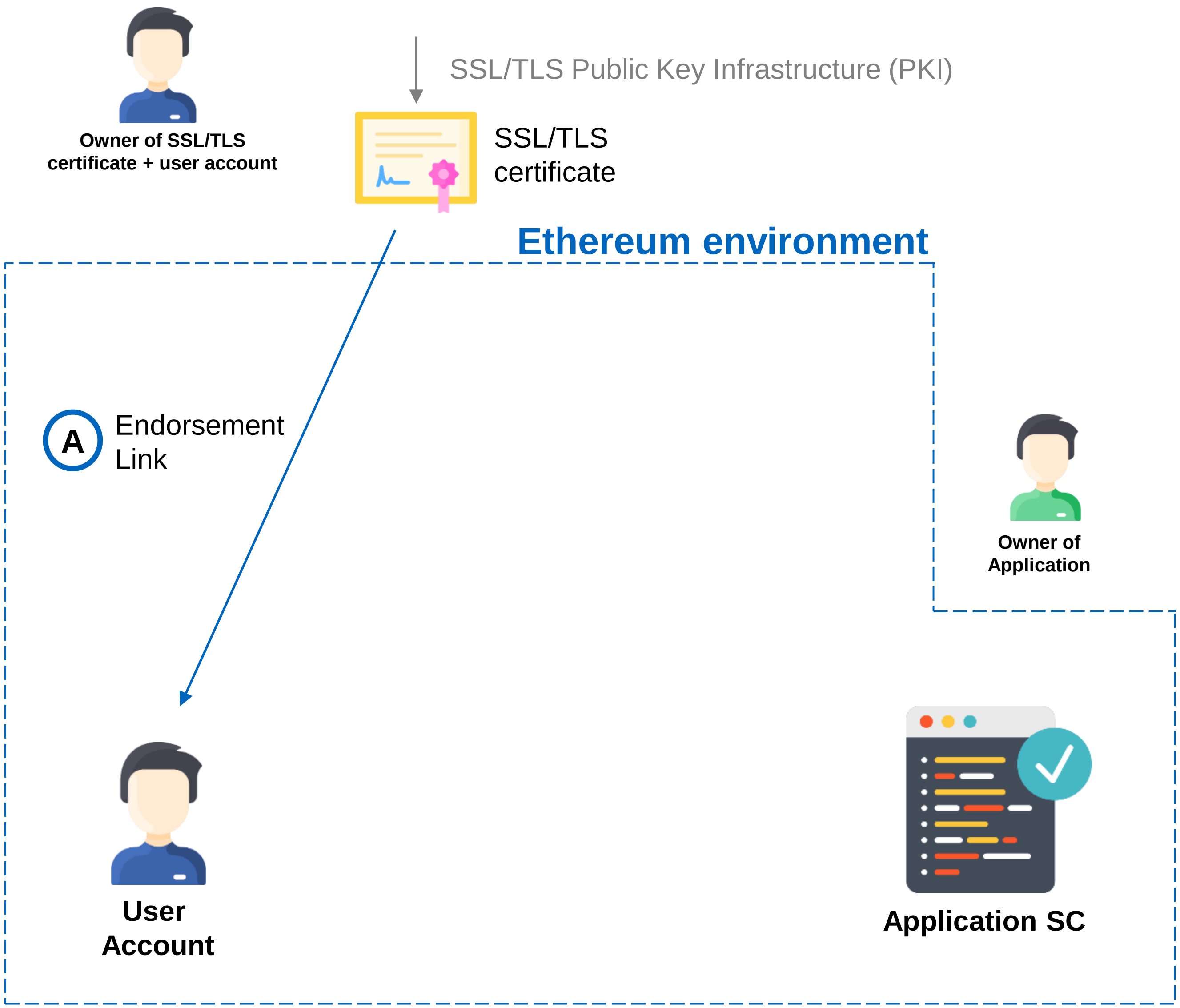
No standard for linking characteristics for access control to blockchain accounts

Leverage an established trust infrastructure **(SSL/TLS certificate PKI)** to link trusted attributes and endow accounts of real-world identities with trust

Conceptual Design & Problem Statement

Endorsement Framework:

A Creation of endorsement link with **TeSC-onchain**
- SSL/TLS-certificate endorsed Smart Contracts
- Allows to create endorsement from SSL/TLS certificates for accounts on the blockchain



Conceptual Design & Problem Statement

Endorsement Framework:

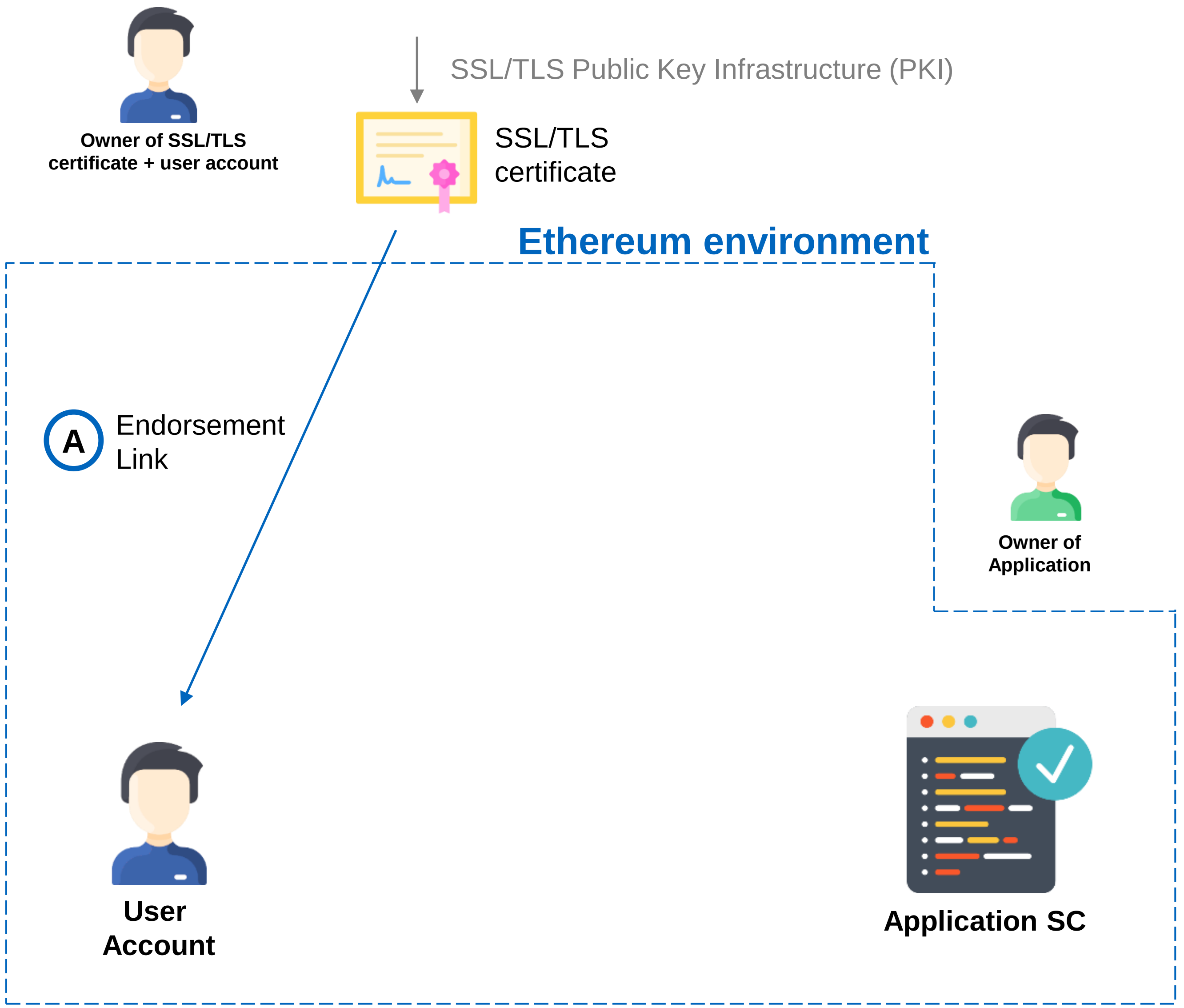
A Creation of endorsement link with **TeSC-onchain**

- SSL/TLS-certificate endorsed Smart Contracts
- Allows to create endorsement from SSL/TLS certificates for accounts on the blockchain

Signed with private key of cert

Endorsement =

Account address: **0x1234**
Domain name: **www.xyz.com**
…



Conceptual Design & Problem Statement

Endorsement Framework:

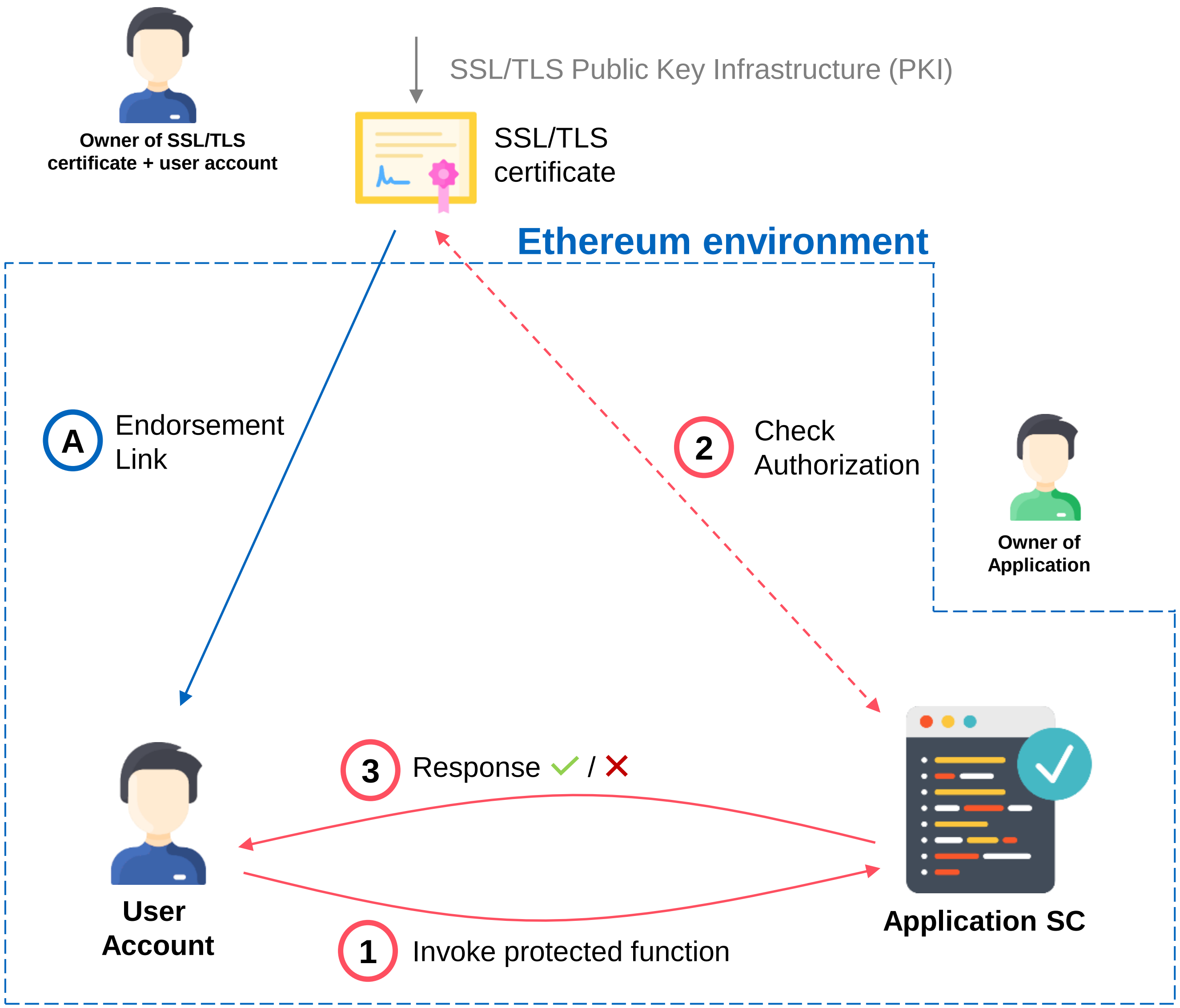
(A) Creation of endorsement link with **TeSC-onchain**
- SSL/TLS-certificate endorsed Smart Contracts
- Allows to create endorsement from SSL/TLS certificates for accounts on the blockchain

Access Control Framework:

(1) Invocation of protected function at an Application SC

(2) Application checks the authorization of requester
- Validate **endorsement** link
- Check **attributes** of SSL/TLS certificate

(3) Successful invocation of function or rejection

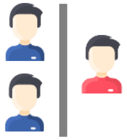


Problem Statement

No source of trust for user accounts of real-world entities on the blockchain

Attributes of **SSL/TLS certificates are not designed** for authentication and access control on the blockchain

Integration with **TeSC-onchain**

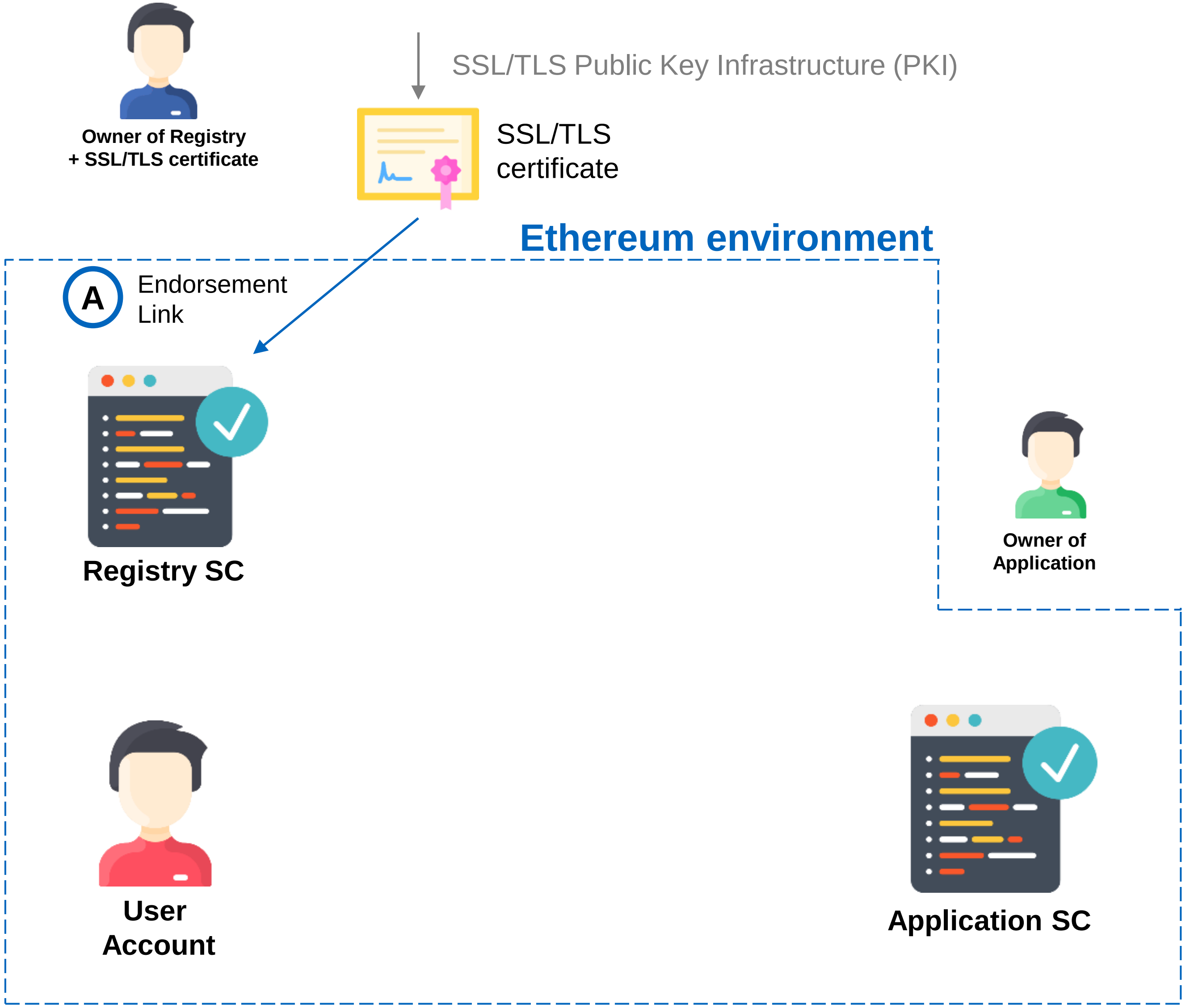


Exclusion of real-world entities without a SSL/TLS certificate

Conceptual Design & Problem Statement

Endorsement Framework:

A Creation of **endorsement link** with **TeSC-onchain**
- SSL/TLS-certificate endorsed Smart Contracts
- Allows to create endorsement from SSL/TLS certificates for accounts on the blockchain

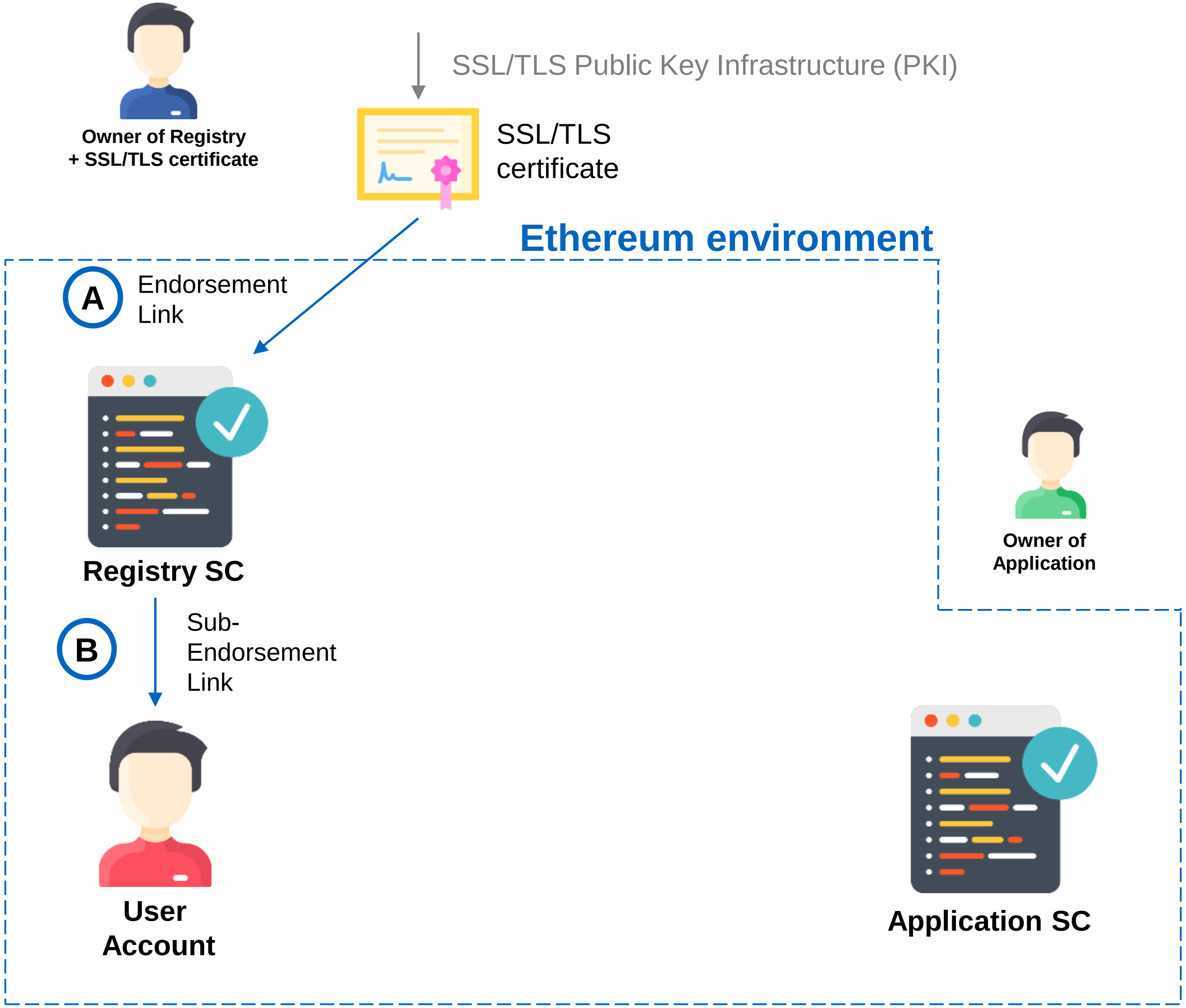


Conceptual Design & Problem Statement

Endorsement Framework:

A Creation of **endorsement link** with **TeSC-onchain**
- SSL/TLS-certificate endorsed Smart Contracts
- Allows to create endorsement from SSL/TLS certificates for accounts on the blockchain

B Creation of **sub-endorsement link** by adding user accounts to one or multiple Registries SCs



Conceptual Design & Problem Statement

Endorsement Framework:

A Creation of **endorsement link** with **TeSC-onchain**
- SSL/TLS-certificate endorsed Smart Contracts
- Allows to create endorsement from SSL/TLS certificates for accounts on the blockchain

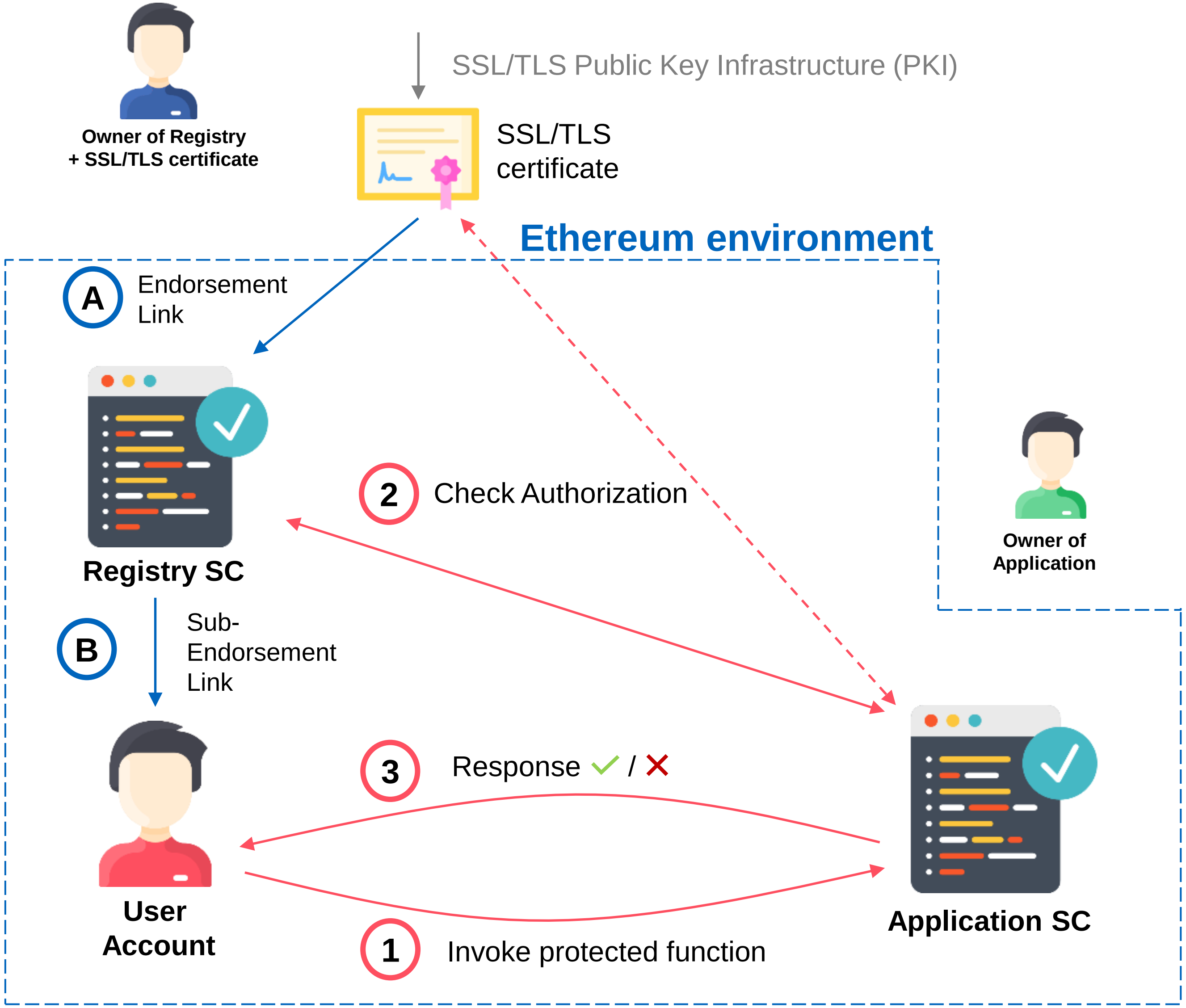
B Creation of **sub-endorsement link** by adding user accounts to one or multiple Registries SCs

Access Control Framework:

1 Invocation of protected function at an Application SC

2 Application checks the authorization of requester
- Validate **sub-endorsement** of user account
- Validate **endorsement** of Registry
- Check **attributes** of SSL/TLS certificate

3 Successful invocation of function or rejection



Conceptual Design & Problem Statement

Endorsement Framework:

A Creation of **endorsement link** with **TeSC-onchain**
- SSL/TLS-certificate endorsed Smart Contracts
- Allows to create endorsement from SSL/TLS certificates for accounts on the blockchain

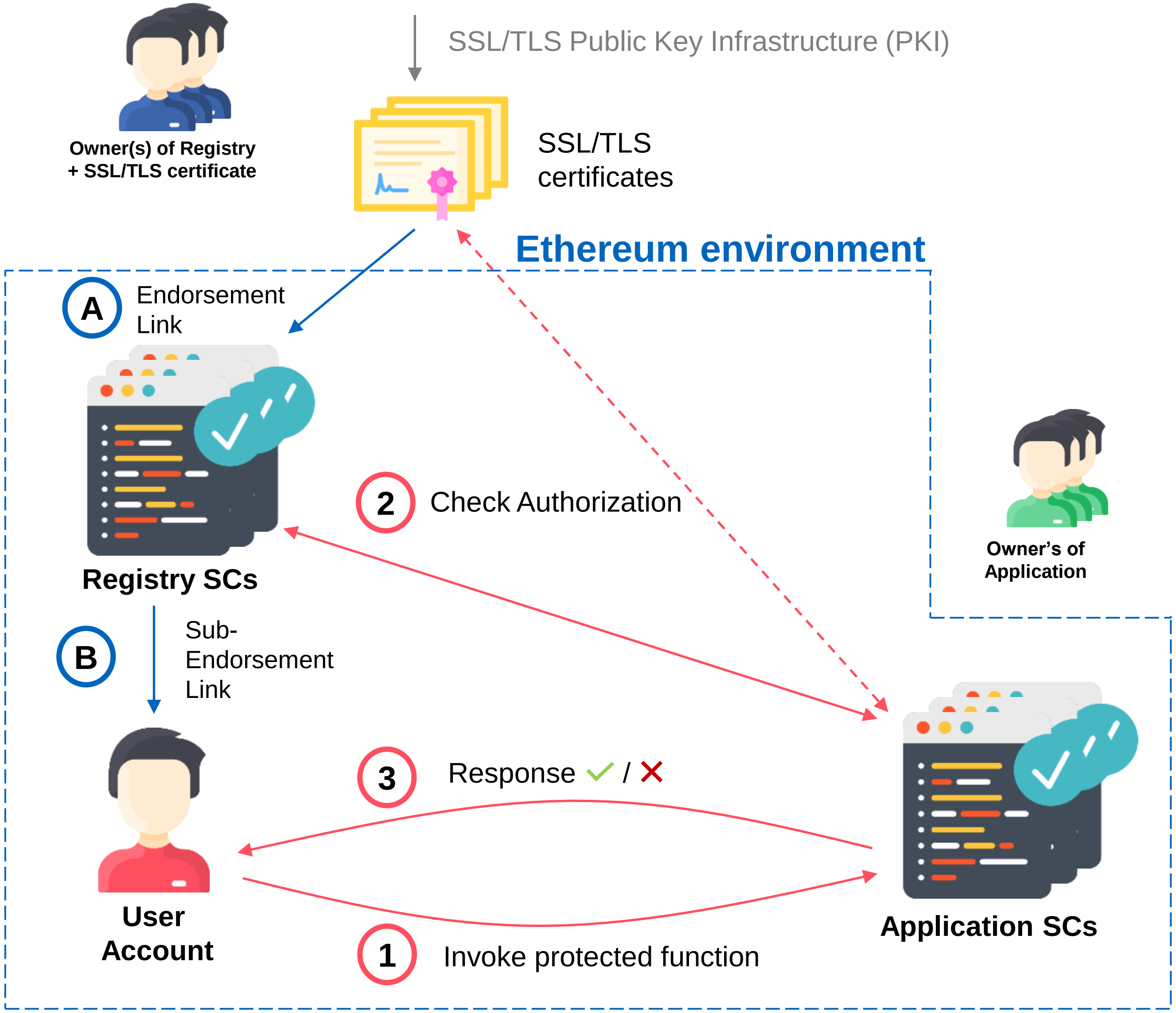
B Creation of **sub-endorsement link** by adding user accounts to one or multiple Registries SCs

Access Control Framework:

1 Invocation of protected function at an Application SC

2 Application checks the authorization of requester
- Validate **sub-endorsement** of user account
- Validate **endorsement** of Registry
- Check **attributes** of SSL/TLS certificate

3 Successful invocation of function or rejection



Research Approach & Contribution

Problem Identification

- Discussions with supervisor
- Literature review

Adopted from [53] Pfeffers et al., The Design Science Research Process: A Model for Producing and Presenting Information Systems Research

Research Approach & Contribution

Problem Identification

- Discussions with supervisor
- Literature review

Design & Development

1st
- Concept + Use-cases
- Requirements
- High level architecture

Adopted from [53] Pfeffers et al., The Design Science Research Process: A Model for Producing and Presenting Information Systems Research

Research Approach & Contribution

Re-Iterate

Problem Identification

- Discussions with supervisor
- Literature review

Design & Development

1st
- Concept + Use-cases
- Requirements
- High level architecture

Evaluation

1st
- Concept Design
  (8 expert interviews)

Adopted from [53] Pfeffers et al., The Design Science Research Process: A Model for Producing and Presenting Information Systems Research

Research Approach & Contribution

Re-Iterate

Problem Identification

- Discussions with supervisor
- Literature review

Design & Development

1st
- Concept + Use-cases
- Requirements
- High level architecture

2nd
- Refine + finalize design
- Prototype in Solidity with Truffle

Evaluation

1st
- Concept Design (8 expert interviews)

2nd
- Prototype Performance (Gas costs)

Adopted from [53] Pfeffers et al., The Design Science Research Process: A Model for Producing and Presenting Information Systems Research

Research Approach & Contribution

Re-Iterate

Problem Identification

- Discussions with supervisor
- Literature review

Design & Development

1st
- Concept + Use-cases
- Requirements
- High level architecture

2nd
- Refine + finalize design
- Prototype in Solidity with Truffle

Evaluation

1st
- Concept Design (8 expert interviews)

2nd
- Prototype Performance (Gas costs)

Presentation

- Master Thesis
- Final Presentation

Adopted from [53] Pfeffers et al., The Design Science Research Process: A Model for Producing and Presenting Information Systems Research

Research Approach & Contribution

Re-Iterate

Problem Identification

- Discussions with supervisor
- Literature review

Design & Development

1st
- Concept + Use-cases
- Requirements
- High level architecture

2nd
- Refine + finalize design
- Prototype in Solidity with Truffle

Evaluation

1st
- Concept Design (8 expert interviews)

2nd
- Prototype Performance (Gas costs)

Presentation

- Master Thesis
- Final Presentation

Main contributions

Literature review of access control (on the blockchain)

Design, development and evaluation of prototype that enables access control at smart contracts

Adopted from [53] Pfeffers et al., The Design Science Research Process: A Model for Producing and Presenting Information Systems Research

Outline

1. Introduction
   - Motivation
   - Conceptual Design & Problem Statement
   - Research Approach & Contribution

2. Research Questions

3. Evaluation

4. Discussion
   - Conclusion
   - Future Work

Research Questions

R1 Which are the major access control practices and technologies?

R2 How can a SSL/TLS-based identity assertion and verification system contribute trust to authentication and access control on the blockchain?

R3 How can we achieve on-chain authentication and access control of real-world identities considering the constraints of Blockchain?

R1 Which are the major access control practices and technologies?

Access Control

"[…] the decision to permit or deny a subject access to system objects (network, data, application, service, etc.)"[1]

Figure from V. C. Hu et al. NIST Special Publication 800-162 - Guide to Attribute Based Access Control (ABAC) Definition and Considerations. Tech. rep. 2014, pp. 1–37.
[1]*V. C. Hu et al. Nistir 7316: Assessment of access control systems. Tech. rep. 2006, pp. 1–51*

R1 Which are the major access control practices and technologies?

Access Control

"[…] the decision to permit or deny a subject access to system objects (network, data, application, service, etc.)"[1]

Dominant mechanisms

Mandatory Access Control (MAC)

Discretionary Access Control (DAC)

Role-based Access Control (RBAC)

Attribute-based Access Control (ABAC)

- Attributes assigned to subjects; access restricted by attributes
- Easy integration with SSL/TLS certificates attributes
- Works well for distributes Systems

Figure from V. C. Hu et al. NIST Special Publication 800-162 - Guide to Attribute Based Access Control (ABAC) Definition and Considerations. Tech. rep. 2014, pp. 1–37.

[1]V. C. Hu et al. Nistir 7316: Assessment of access control systems. Tech. rep. 2006, pp. 1–51

R1 Which are the major access control practices and technologies?

Access Control

"[…] the decision to permit or deny a subject access to system objects (network, data, application, service, etc.)"[1]

Dominant mechanisms

Mandatory Access Control (MAC)

Discretionary Access Control (DAC)

Role-based Access Control (RBAC)

Attribute-based Access Control (ABAC)

- Attributes assigned to subjects; access restricted by attributes
- Easy integration with SSL/TLS certificates attributes
- Works well for distributes Systems

Attribute-based Access Control

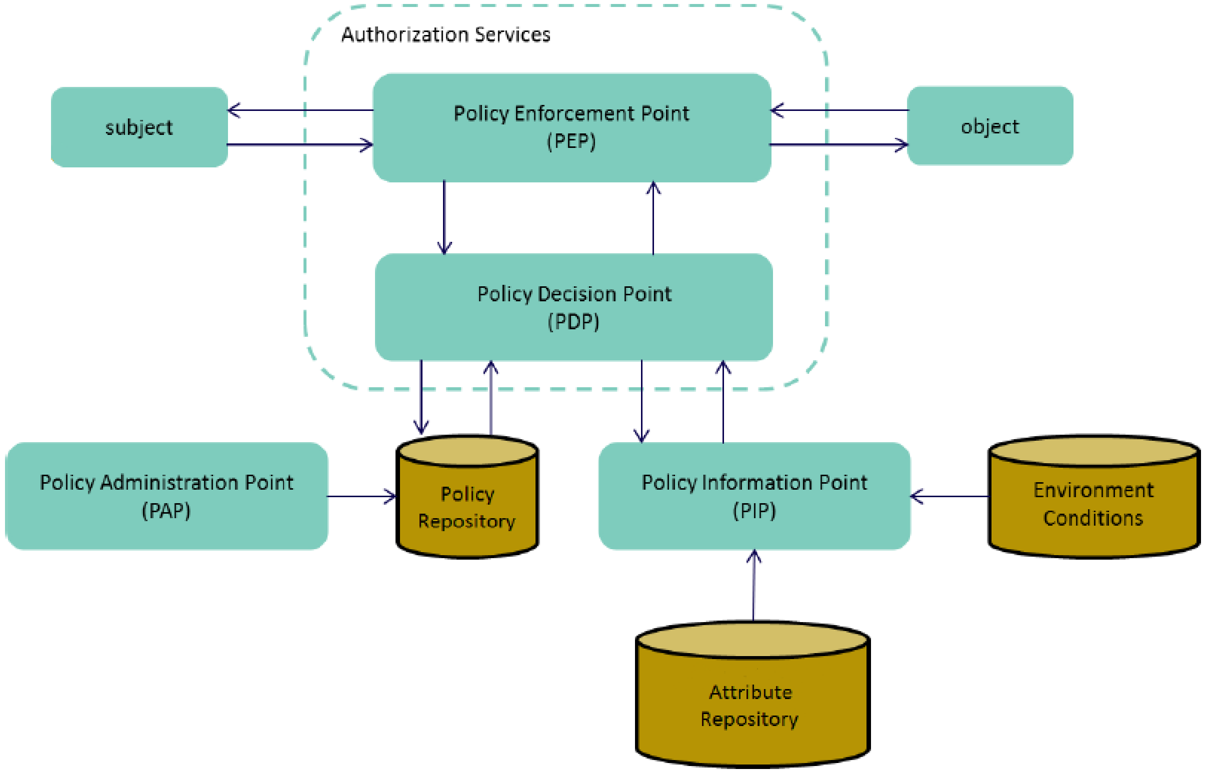


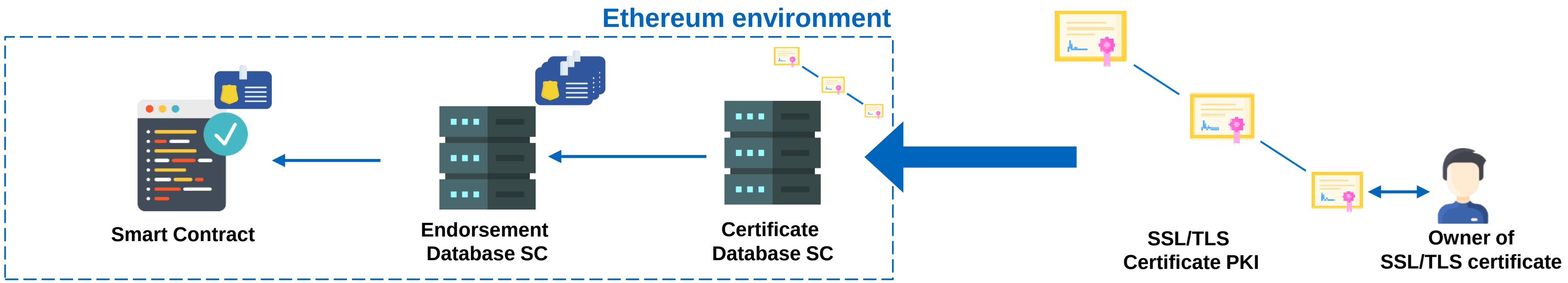
Figure from V. C. Hu et al. NIST Special Publication 800-162 - Guide to Attribute Based Access Control (ABAC) Definition and Considerations. Tech. rep. 2014, pp. 1–37.

[1]V. C. Hu et al. Nistir 7316: Assessment of access control systems. Tech. rep. 2006, pp. 1–51

R2 How can a SSL/TLS-based identity assertion and verification system contribute trust to authentication and access control on the blockchain?

TeSC-onchain = SSL/TLS-based identity assertion and verification system

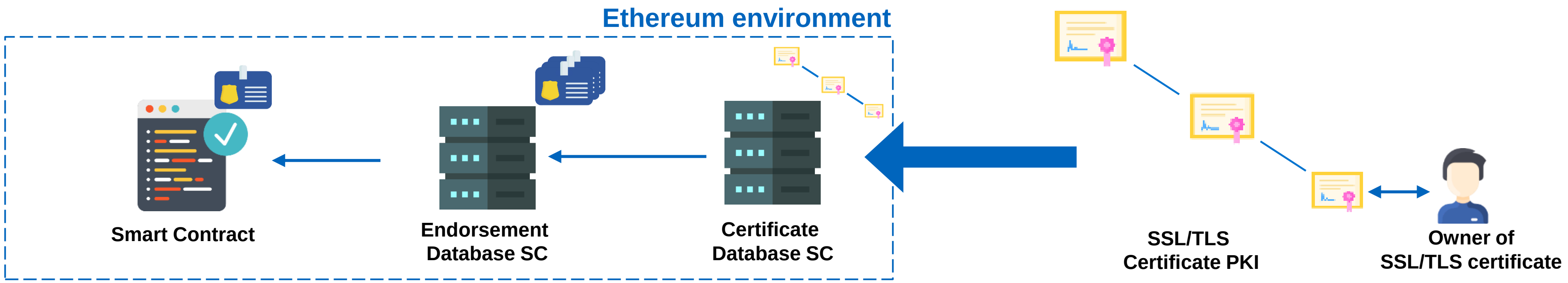
➔ Allows to authenticate who the owner of a smart contract is



R2 How can a SSL/TLS-based identity assertion and verification system contribute trust to authentication and access control on the blockchain?

TeSC-onchain = SSL/TLS-based identity assertion and verification system

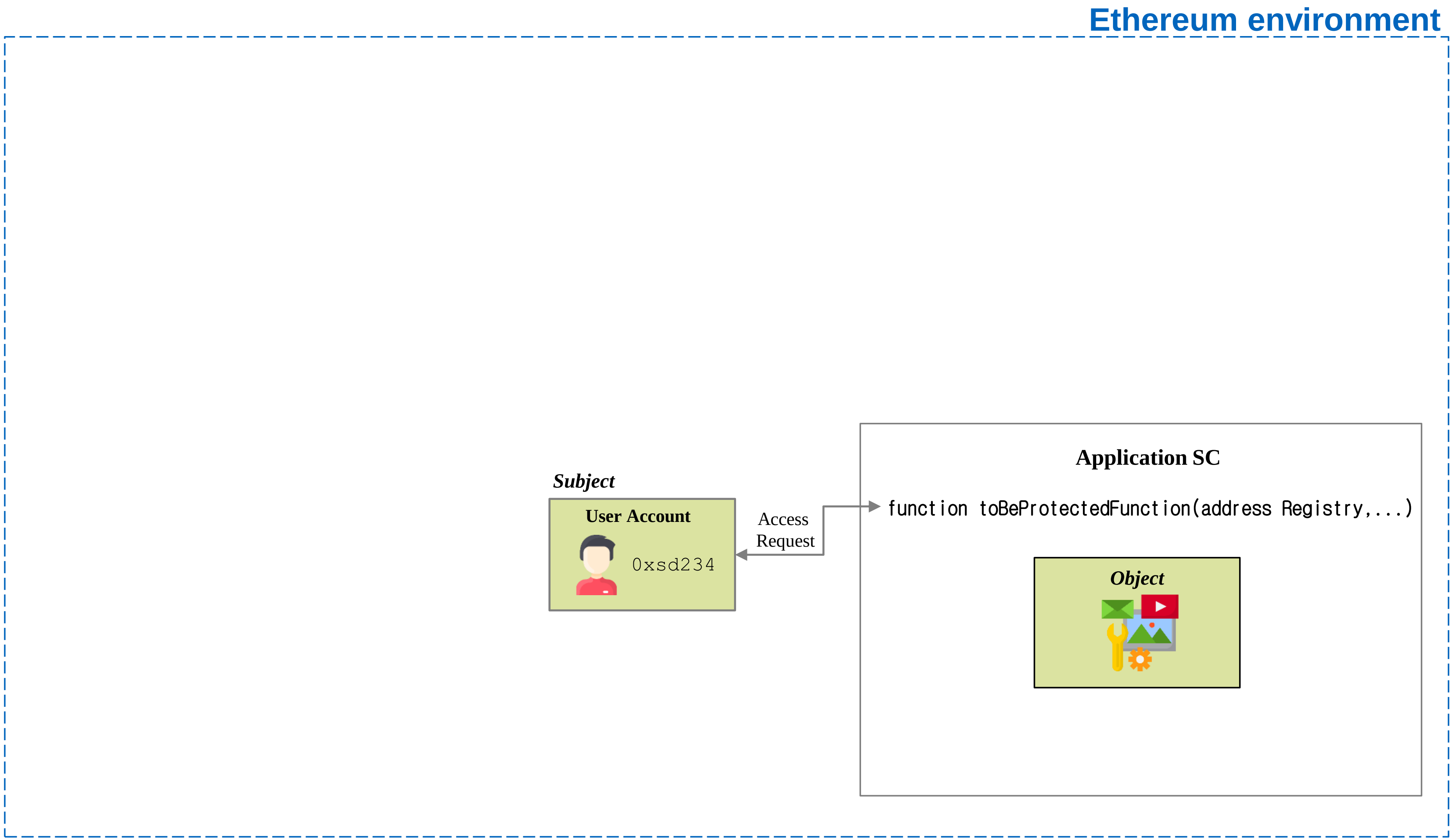
➔ Allows to authenticate who the owner of a smart contract is



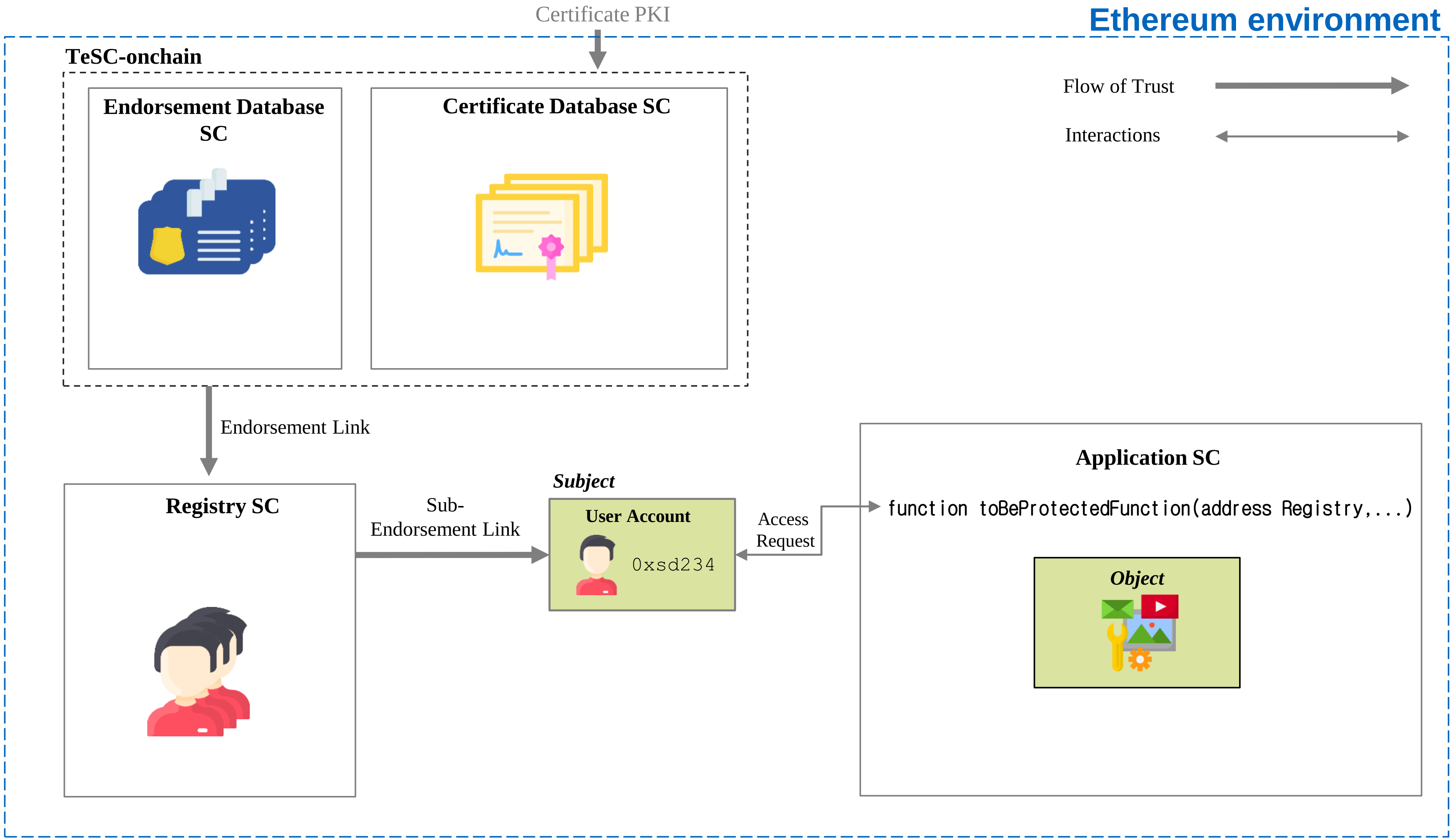
Contribution

- (Indirect) authentication of accounts of real-world entities on the blockchain
- Trusted attributes for accounts of real-world entities on the blockchain
- Infrastructure to store and access trusted attributes of SSL/TLS certificates for ABAC on the blockchain

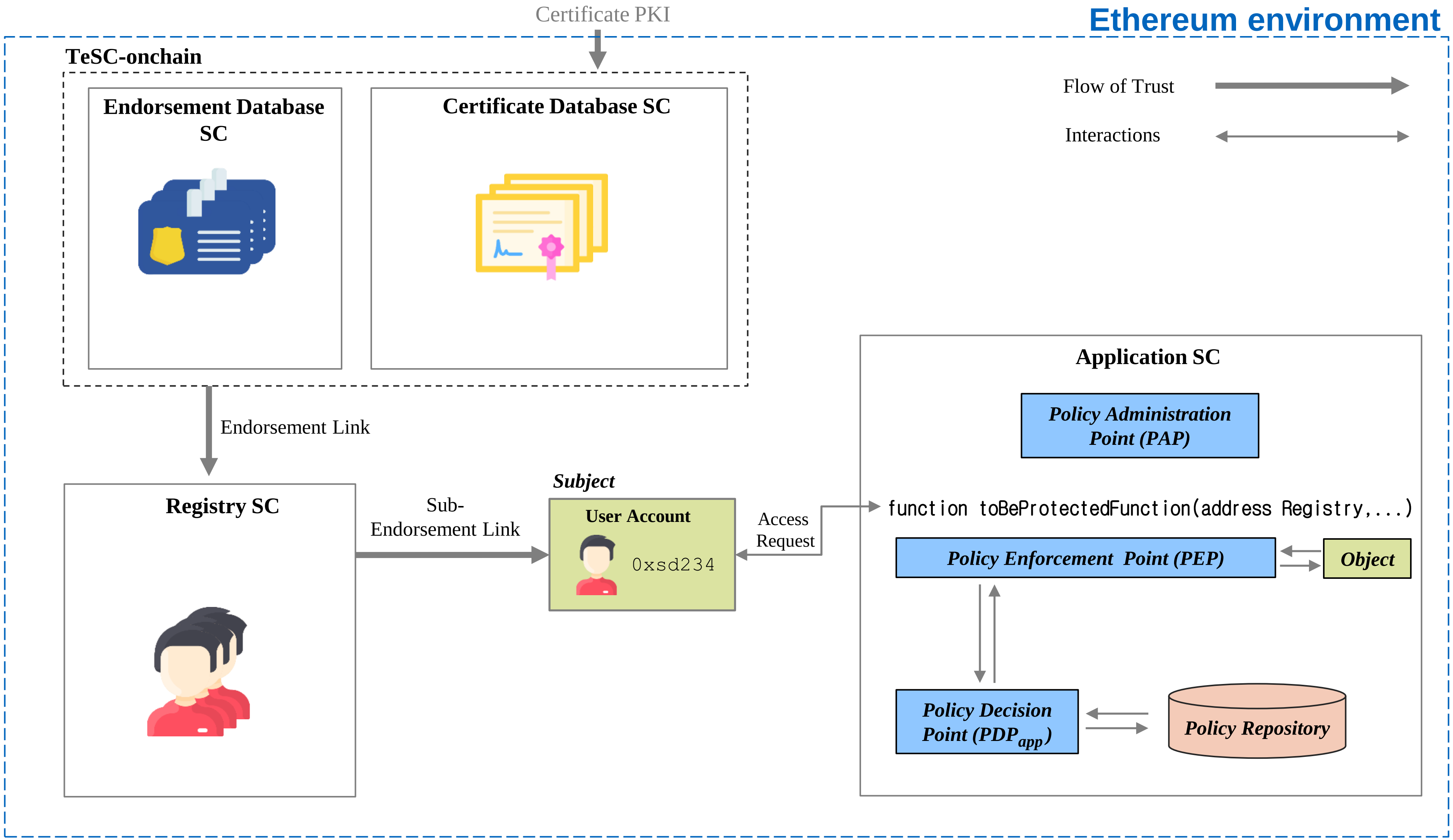
R3 How can we achieve on-chain authentication and access control of real-world identities considering the constraints of Blockchain?



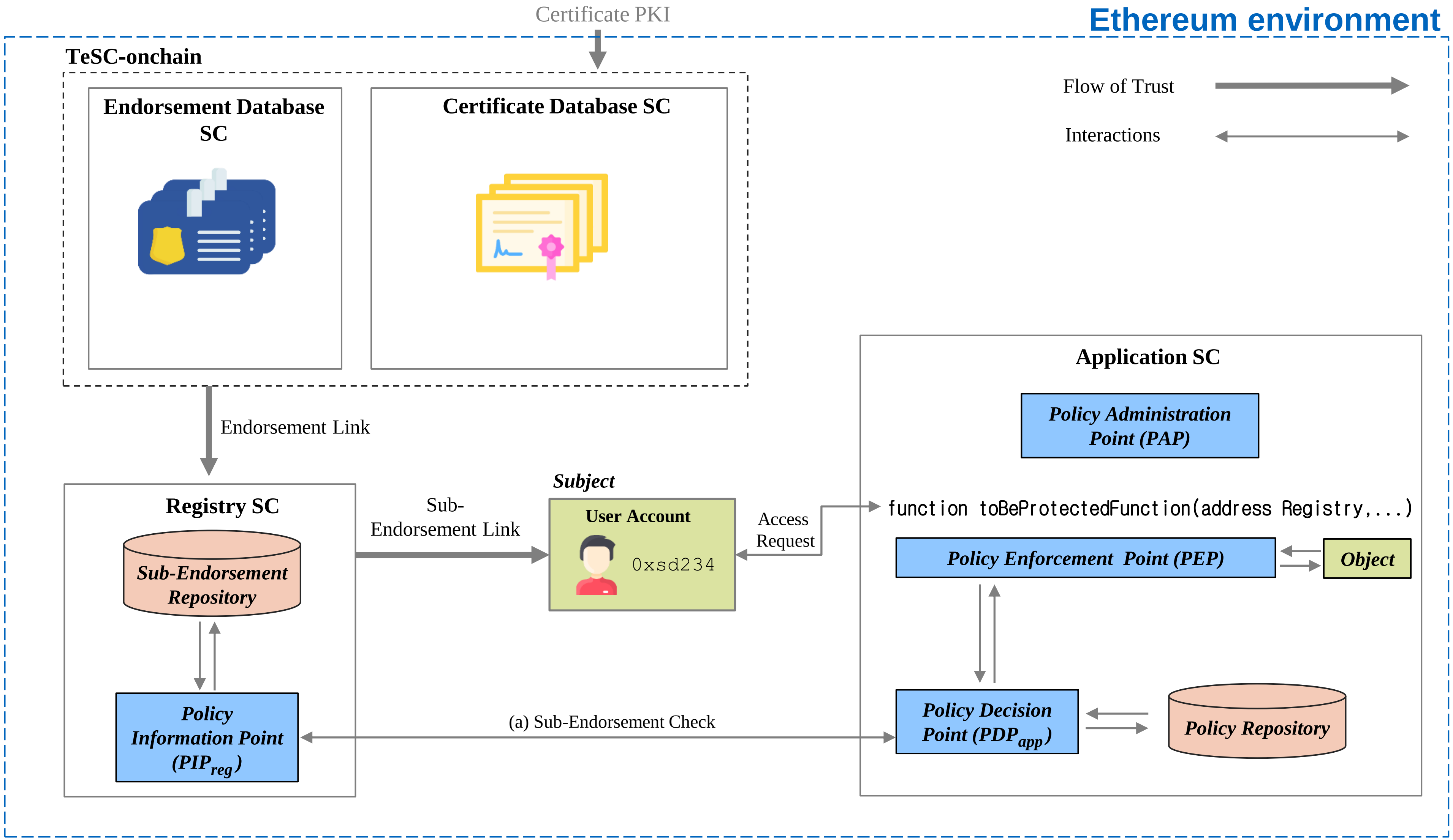
R3 How can we achieve on-chain authentication and access control of real-world identities considering the constraints of Blockchain?



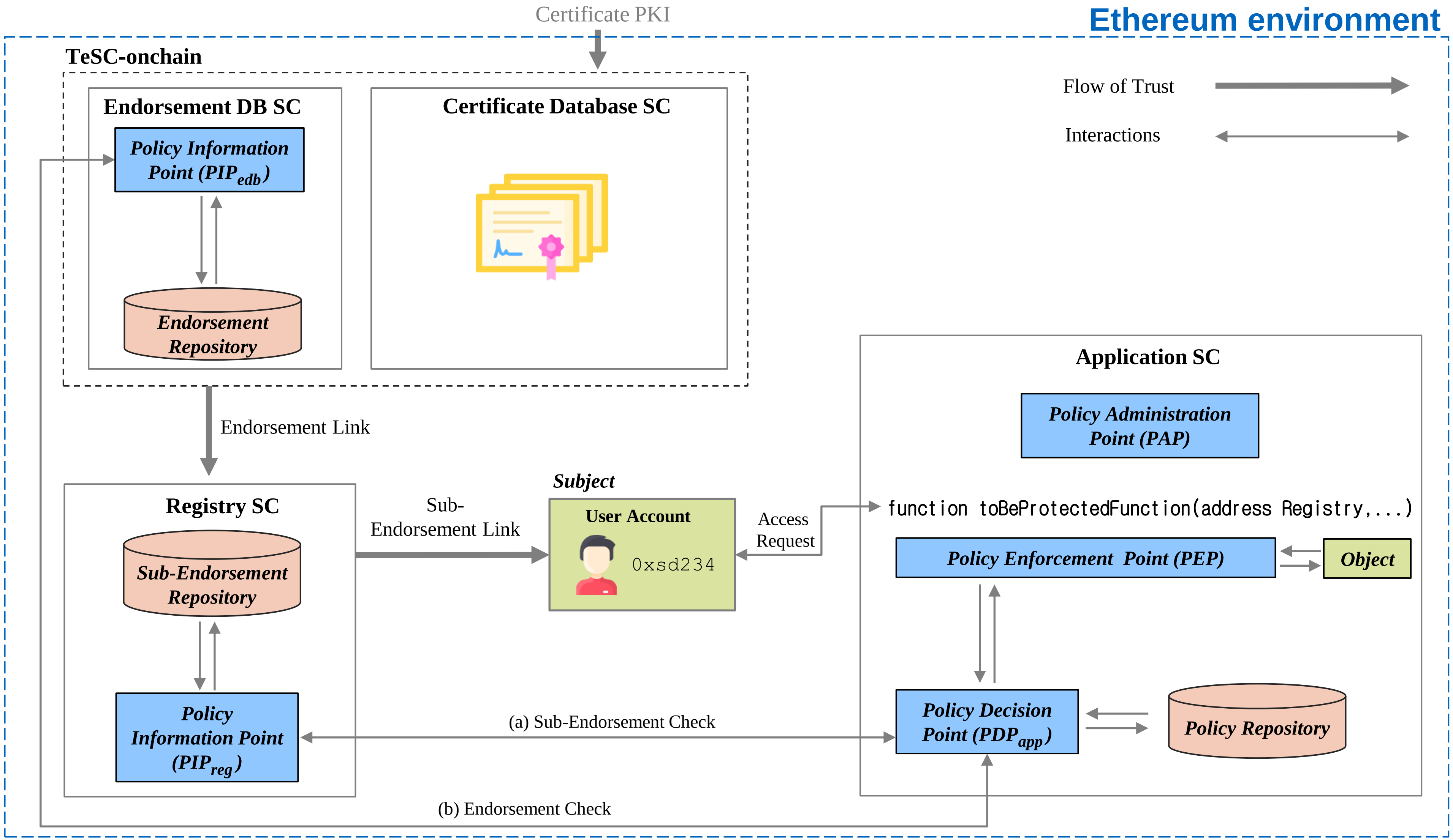
R3 How can we achieve on-chain authentication and access control of real-world identities considering the constraints of Blockchain?



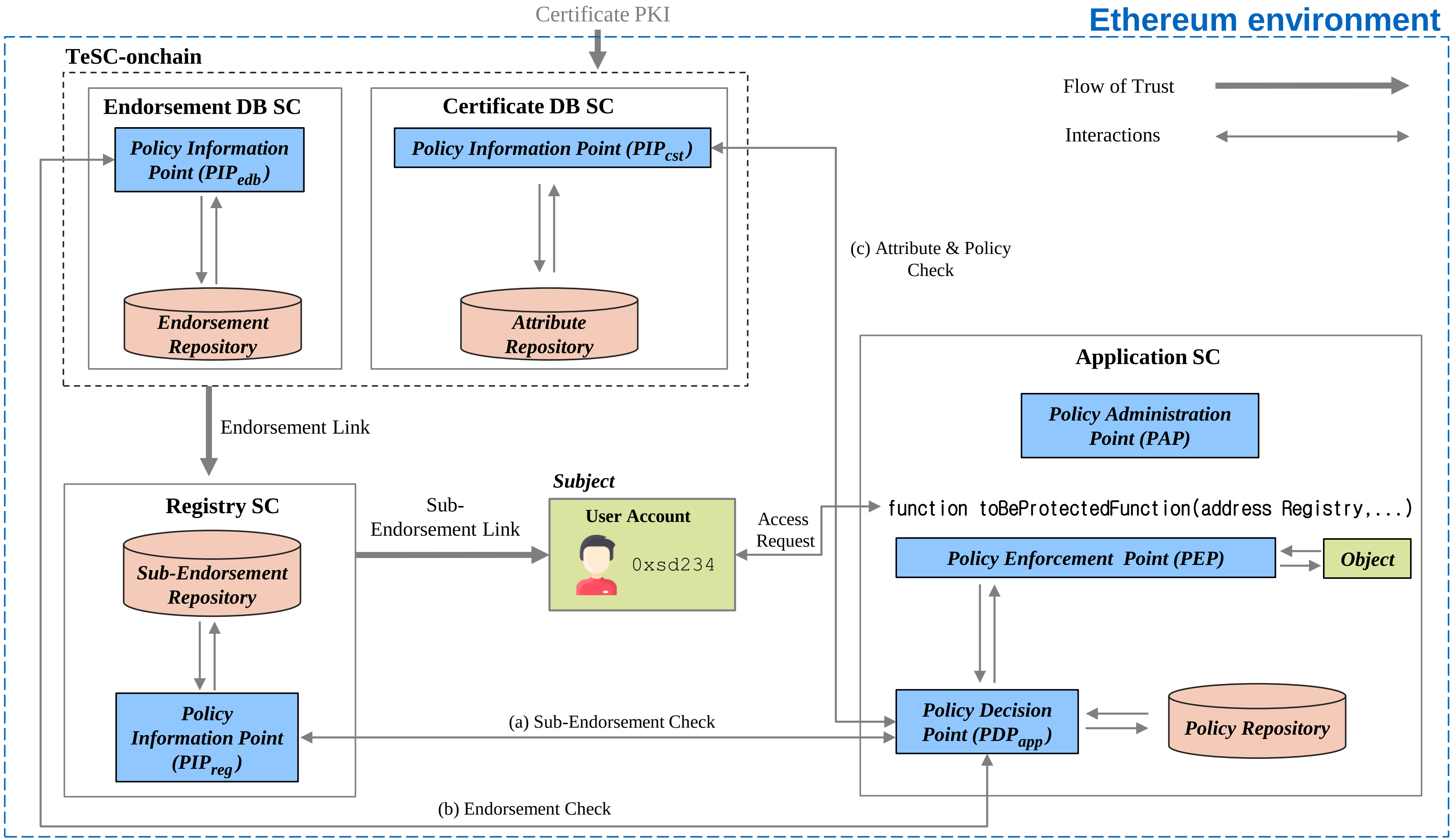
R3 How can we achieve on-chain authentication and access control of real-world identities considering the constraints of Blockchain?



R3 How can we achieve on-chain authentication and access control of real-world identities considering the constraints of Blockchain?



R3 How can we achieve on-chain authentication and access control of real-world identities considering the constraints of Blockchain?



Outline

1. Introduction
   - Motivation
   - Conceptual Design & Problem Statement
   - Research Approach & Contribution

2. Research Questions

3. Evaluation

4. Discussion
   - Conclusion
   - Future Work

Evaluation

Ex Ante Evaluation
(Expert Interviews)

Requirements

Use-Cases

System Design

Implementation

Ex Post Evaluation
(Truffle Prototype)

Performance Analysis: Cost

Evaluation

Ex Ante Evaluation
(Expert Interviews)

Requirements

Functional and Non-Functional requirements for development

Goal: Test if requirements are comprehensive and user-centric

→ Requirements confirmed and one functional added

Use-Cases

System Design

Implementation

Ex Post Evaluation
(Truffle Prototype)

Performance Analysis: Cost

Evaluation

Ex Ante Evaluation
(Expert Interviews)

Requirements
Functional and Non-Functional requirements for development

Goal: Test if requirements are comprehensive and user-centric

➔ Requirements confirmed and one functional added

Use-Cases
4 different use-cases in different context (e.g. research, IoT)

Goal: Determine relevance of research for application

➔ Strong positive feedback
➔ For any use-case with acc-ess control on blockchain

System Design

Implementation

Ex Post Evaluation
(Truffle Prototype)

Performance Analysis: Cost

Evaluation

Ex Ante Evaluation
(Expert Interviews)

Requirements
Functional and Non-Functional requirements for development

Goal: Test if requirements are comprehensive and user-centric

→ Requirements confirmed and one functional added

Use-Cases
4 different use-cases in different context (e.g. research, IoT)

Goal: Determine relevance of research for application

→ Strong positive feedback
→ For any use-case with access control on blockchain

System Design
Conceptual architecture with key components and interactions

Goal: Identification of potential flaws and improvements

→ Config for Registry SC
→ Add support for other trust sources

Implementation

Ex Post Evaluation
(Truffle Prototype)

Performance Analysis: Cost

Evaluation

Ex Ante Evaluation
(Expert Interviews)

Requirements
Functional and Non-Functional requirements for development

Goal: Test if requirements are comprehensive and user-centric

➔ Requirements confirmed and one functional added

Use-Cases
4 different use-cases in different context (e.g. research, IoT)

Goal: Determine relevance of research for application

➔ Strong positive feedback
➔ For any use-case with access control on blockchain

System Design
Conceptual architecture with key components and interactions

Goal: Identification of potential flaws and improvements

➔ Config for Registry SC
➔ Add support for other trust sources

Implementation

Ex Post Evaluation
(Truffle Prototype)

Performance Analysis: Cost

Deployment:
- Set-Up of **Registry SC**, its endorsement and certificate: 4,198,438Gas or **109.81$**
- Set-Up of **Application SC**: 1,578,342Gas or **43.51$**

➔ Reasonable costs, as they only must be paid once
➔ No significant barriers of entry for our system

Operation:
- Creation **sub-endorsement**: 65,566Gas or **1.71$**
- **Access request** evaluation: 115,633Gas or **3.01$**
  → 50 % of cost for attribute evaluation

➔ Access request evaluation requires major performance improvements
➔ Currently not competitive with non-blockchain solutions

Data from 26/11/2020: Gas price: 51Gwei[2] | 1 ETH = 1,000,000,000Gwei | 510.72 USD for 1 ETH[2]

[1]https://ethgasstation.info/index.php
[2]https://coinmarketcap.com/converter/eth/usd/

Outline

1. Introduction
   - Motivation
   - Conceptual Design & Problem Statement
   - Research Approach & Contribution

2. Research Questions

3. Evaluation

4. Discussion
   - Conclusion
   - Future Work

Conclusion

Any owner of a SSL/TLS certificate can deploy a Registry and **authorize the account** of **any** real-world entity to **access** functions of Application **smart contracts** – ⚠ **Given** the certificates attributes successfully evaluate under the policy of the Application

System is working **completely on-chain**, as relevant certificate and endorsement data is stored at the Registries, Endorsement Database and Certificate Database

SSL/TLS certificate PKI as central source of trust allows access control **without the need for a direct trust relationship** between Registry and Application smart contracts

ABAC System design **comprises all** relevant **components** for Attribute-based access control

⚠ $$$ **Challenge:** Cost for access control request evaluation (esp. Attribute & Policy Check) are still high compared to traditional access control systems

Future Work

Literature review and further testing

- Literature review of access control in **blockchain** systems **beyond ABAC**
- Further testing to evaluate **social** and **technical risks,** as well as **security**

Improve performance and functionality

- **Improve** performance with regards to **speed** and **cost** (esp. attribute check)
- **Add** support for complex access control **policies** and more **attributes**

Add support for other trust sources

- Decentralized Identifiers (**DIDs**)
- **Other** types of **certificates** frequently used by organizations

Prof. Dr.

Florian Matthes

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

Fulfillment of Requirements

#	Requirement	Status
FR1	Add sub-endorsement at Registry	✓
FR2	Revoke sub-endorsement at Registry	✓
FR3	Update sub-endorsement at Registry	✓
FR4	User account may check status of sub-endorsement at Registry	✓
FR5	Request access to Application	✓
FR6	Authenticate user account at Application	✓
FR7	Check authorization of user account at Application	✓
NFR1	Leverage attributes of SSL/TLS certificates	✓
NFR2	Use On-Chain AuthSC	✓
NFR3	On-chain access control decisions	✓
NFR4	Decentralized sub-endorsement allocation	✓
NFR5	Access control without a direct trust relationship with the Registry	✓
NFR6	Access control without pre-provisioning of the subject at the Application	✓
NFR7	Minimal costs of user management, authentication and authorization	X

Table 5.4: Compliance of the prototype with regards to our requirements

Research Questions

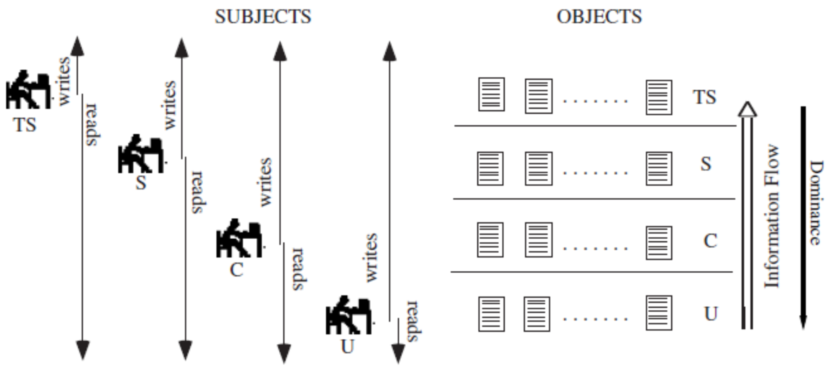
R1 Which are the major access control practices and technologies?

R1.1 Which access control practices and technologies are predominant in the literature?

R 1.2 Which access control practices and technologies are dominant in public blockchain?

Mandatory Access Control

- Users can not acquire the ownership of the object, only the security clearance to interact with it
- In order to evaluate access requests, MAC requires that **access classes**, which are in a partial order, are **assigned to all subjects and objects**.
- Each **access class is a tuple (SL, C)** of a **Security Level (SL)**, as Confidential, Secret or Top-Secret (hierarchically ordered) and a **category C** which describes the functionality as Development, Finance or Marketing.



Secrecy-Based Mandatory Policy

"**No-Read-Up**: A subject is allowed a read access to an object only if the access class of the subject dominates the access class of the object [1, p. 150].“

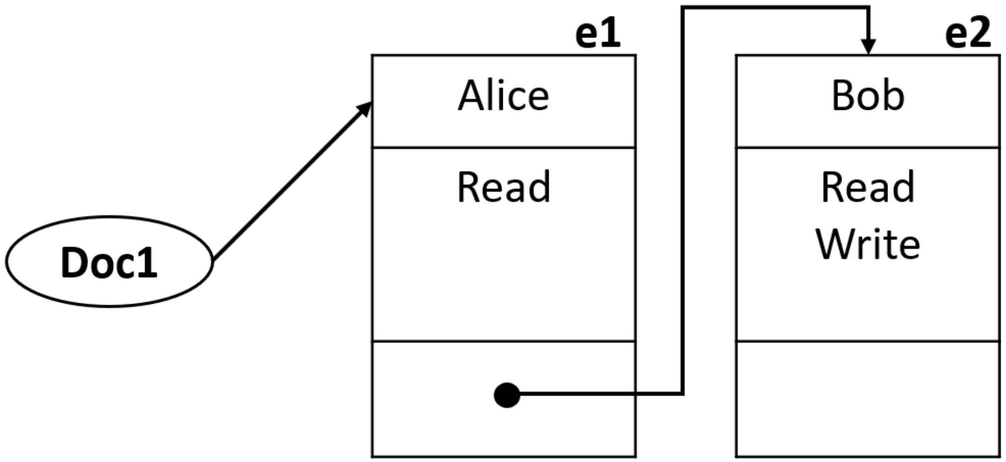
"**No-Write-Down**: A subject is allowed a write access to an object only if the access class of the subject is dominated by the access class of the object [1, p. 150]."

Figure from P. Samarati and S. d. C. di Vimercati. ‘Access control: Policies, models, and mechanisms’. In: Foundations of Security Analysis and Design. 2001, pp. 137–196.

R1 Which are the major access control practices and technologies?

Discretionary Access Control

- The decision of who can access the resource is at the discretion of the resource owner
- Possible that subject is endowed by the resource owner with the right to transfer access to other subjects



Access Control List

	Doc1	Doc2	Doc3
Alice	Read	Owner Read Write	
Bob	Read Write		Read

Access Control Matrix

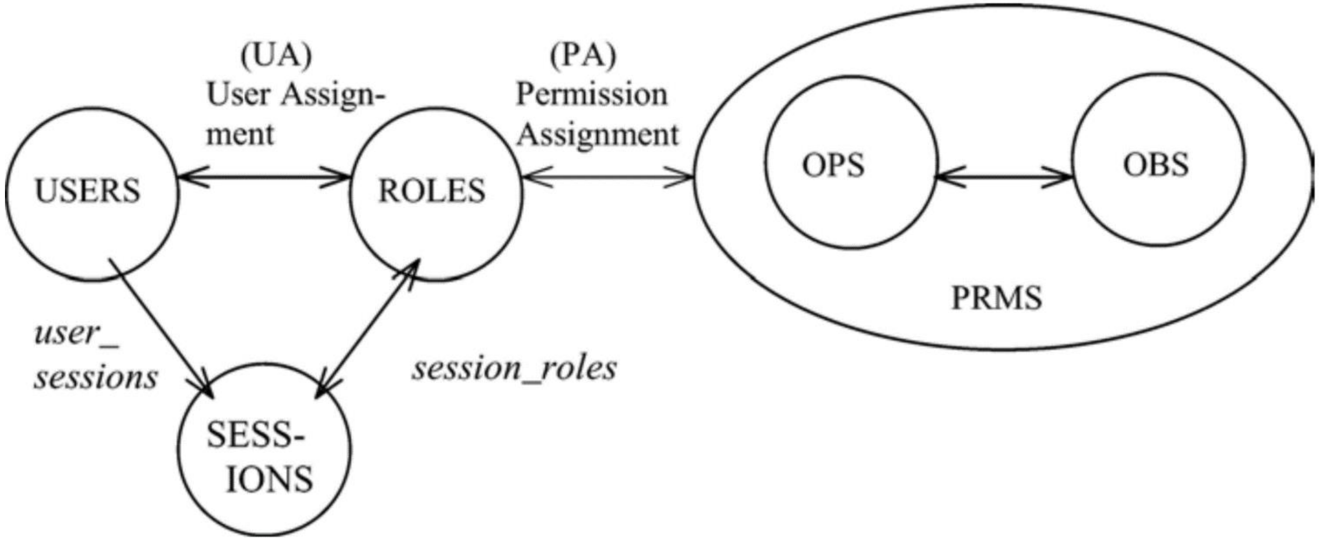
Problem: Can be performance intensive (especially if decentralized)

- If List elements are subjects, very intensive to determine all entities who have Write access
- Same for Access control matrix

R1 Which are the major access control practices and technologies?

Role-based Access Control

- Each role resembles a set of assigned permissions, which determine the operations that can be executed on the object by the subject who is a member of that certain role.
- The Permission Role Management System defines the operations that are executed on the objects



Problem:

- The system can be maintained and administered easily, but with too many roles get very complex
- Multiple layer structure with attributes, permissions and roles create a significant amount of overhead

Figure from International Committee for Information Technology Standards. ANSI INCITS 359-2004. Tech. rep. 2004, pp. 1–47.

R1 Which are the major access control practices and technologies?

R1.1 Which access control practices and technologies are predominant in the literature?

Access Control

"[…] the decision to permit or deny a subject access to system objects (network, data, application, service, etc.)"[1]

Dominant mechanisms

Mandatory Access Control (MAC)
- Only a central is entity allowed to authorize
- High security environment (e.g. military)

Discretionary Access Control (DAC)
- Access control lists and matrix
- Operating Systems (e.g. Windows and Unix)

Role-based Access Control (RBAC)
- Users assigned to roles; access restricted by roles
- Often used in Business Systems (e.g. SAP ERP)

Attribute-based Access Control (ABAC)
- Users assigned to attributes; access restricted by attributes
- Easy integration with SSL/TLS certificates attributes
- Works well for distributes Systems

Attribute-based Access Control

Subject attributes and ***object attributes***…
- …which are together with environment conditions, evaluated under access control ***policies***…
- …in order to determine the outcome of an access control decision.

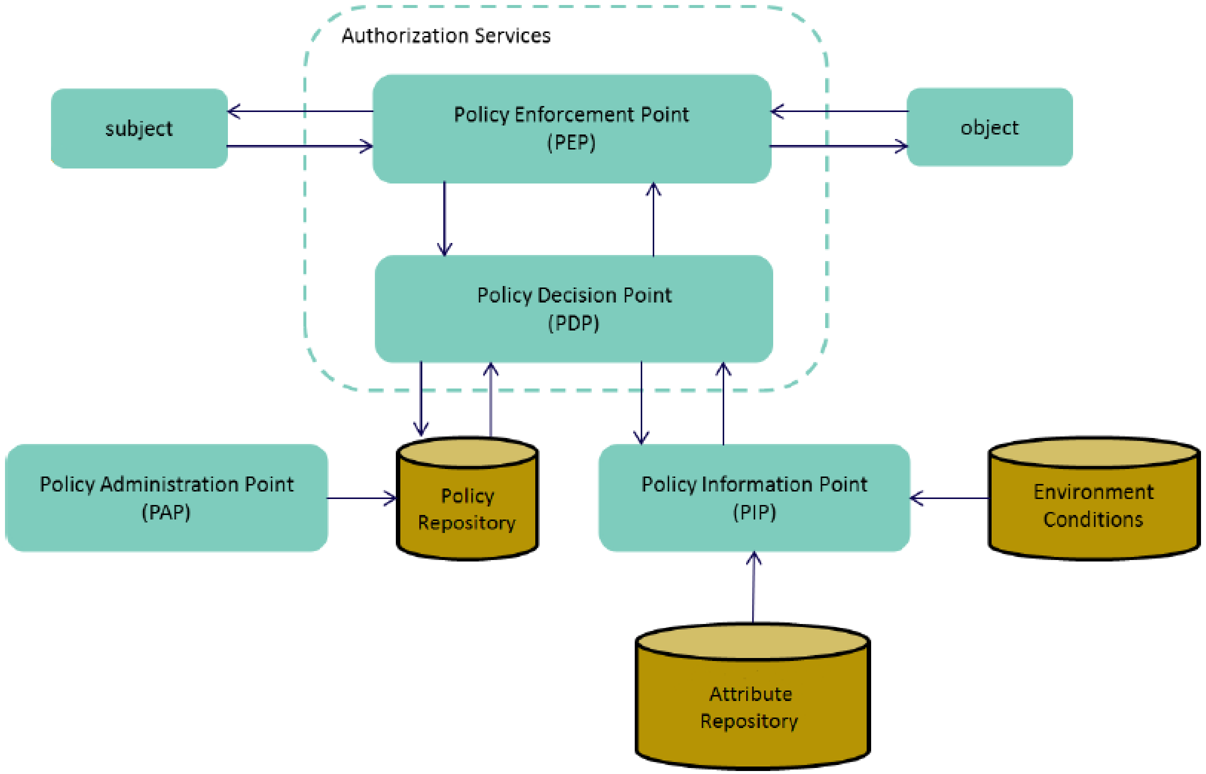


Figure from V. C. Hu et al. NIST Special Publication 800-162 - Guide to Attribute Based Access Control (ABAC) Definition and Considerations. Tech. rep. 2014, pp. 1–37.

[1]V. C. Hu et al. Nistir 7316: Assessment of access control systems. Tech. rep. 2006, pp. 1–51

R1 Which are the major access control practices and technologies?

R1.2 Which access control practices and technologies are dominant in public blockchain?

Blockchain **usually** is leveraged to decentralize **access control of external-resources**

IoT Devices

External Data

External Applications

Few research and implementations of blockchain **internal access control** (smart contracts) → the focus of our research

The few practices

Whitelisting of account addresses

- Mapping of account addresses
- Special case: OpenZeppelin “onlyOwner pattern”

Problem:
Extensive manual work

ABAC **Attribute**-based access control

- Attributes or tokens are stored in SC and mapped to accounts
- Attributes can be checked by other SCs

Problem in public blockchains:
Credibility of attributes

RBAC **Role**-based access control

- Roles are defined and assigned in SC where they are also used to restrict access to functions

Problem in public blockchains:
Credibility of roles

Research Questions

R2 How can a SSL/TLS-based identity assertion and verification system contribute trust to authentication and access control on the blockchain?

R2.1 Which of its properties endow a SSL/TLS certificate with an increased level of trust?

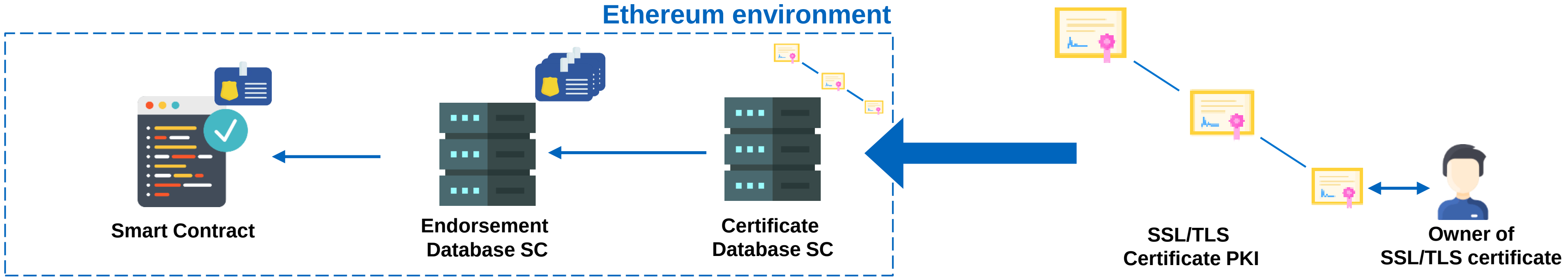
- Link to a chain of trust?

R2.2 What are challenges of bootstrapping a SSL/TLS-based identity assertion and verification system?

R2 How can a SSL/TLS-based identity assertion and verification system contribute trust to authentication and access control on the blockchain?

TeSC-onchain = SSL/TLS-based identity assertion and verification system

- ➔ Allows to authenticate who the owner of a smart contract is
- ➔ Endorsement is a practically unforgeable link between a SSL/TLS certificate and a smart contract
- ➔ **Mirrors** the **SSL/TLS certificate PKI** on the **blockchain**



Contribution

(Indirect) authentication of accounts of real-world entities on the blockchain

Trusted attributes for accounts of real-world entities on the blockchain

Infrastructure to store and access trusted attributes of SSL/TLS certificates for ABAC on the blockchain

Research Questions

R3 How can we achieve on-chain authentication and access control of real-world identities considering the constraints of Blockchain?

R3.1 What is the application life-cycle of a potential on-chain authentication and access control solution?

R3.2 Which are the constrains of Blockchain that affect the development of our solution?

R3.3 What are potential system designs for an on-chain authentication and access control solution?

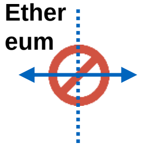
R3.4 What are the advantages and disadvantages of the different system designs?

R3 How can we achieve on-chain authentication and access control of real-world identities considering the constraints of Blockchain?

R3.2 Which are the constrains of Blockchain that affect the development of our solution?

No source of trust

- No sophisticated centralized or decentralized trust infrastructure resides on public Ethereum
  ➔ TeSC-onchain



Smart contract **communication is limited to the Ethereum**, as Ethereum requires **full determinism**

- limited access to blockchain external resources
  ➔ Only on-chain (supported by TeSC-onchain)

Immutability of smart contracts once they are deployed

- Still need to ensure updates
  ➔ Functions to update smart contract account addresses

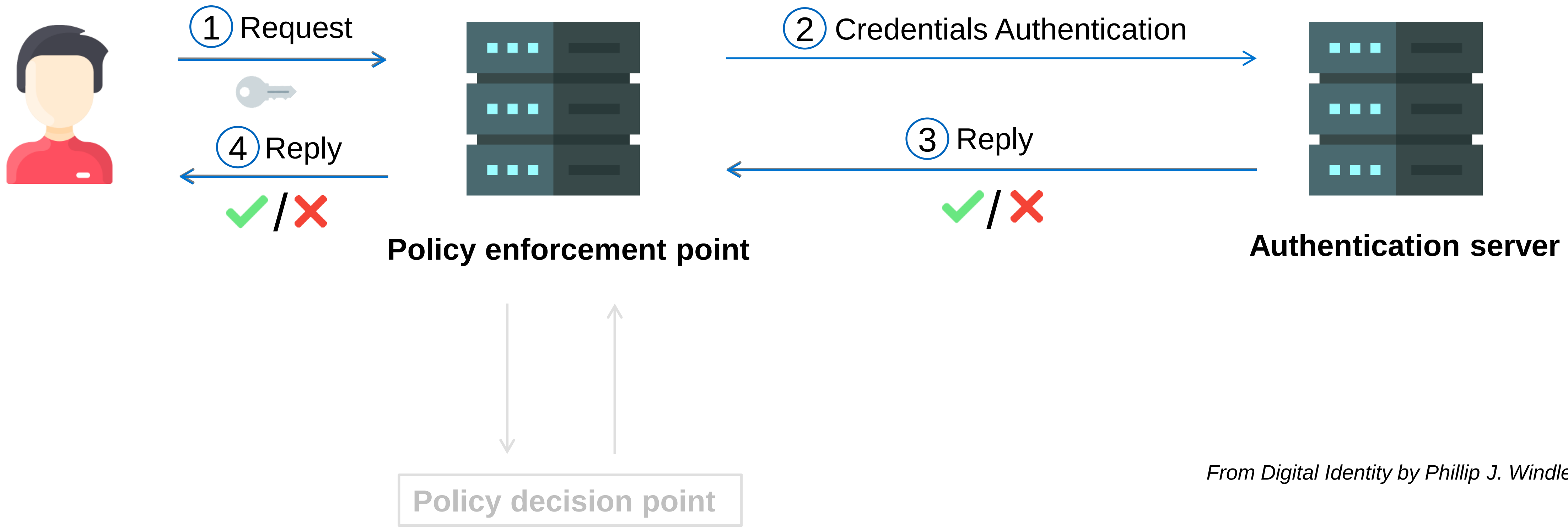
Limited Transaction Throughput

- Rather slow and expensive execution of access requests
  ➔ Minimize costs and interactions as much as possible

Authentication

Authentication:
"The act of proofing who you are"

Authorization:
"The act of granting someone access"

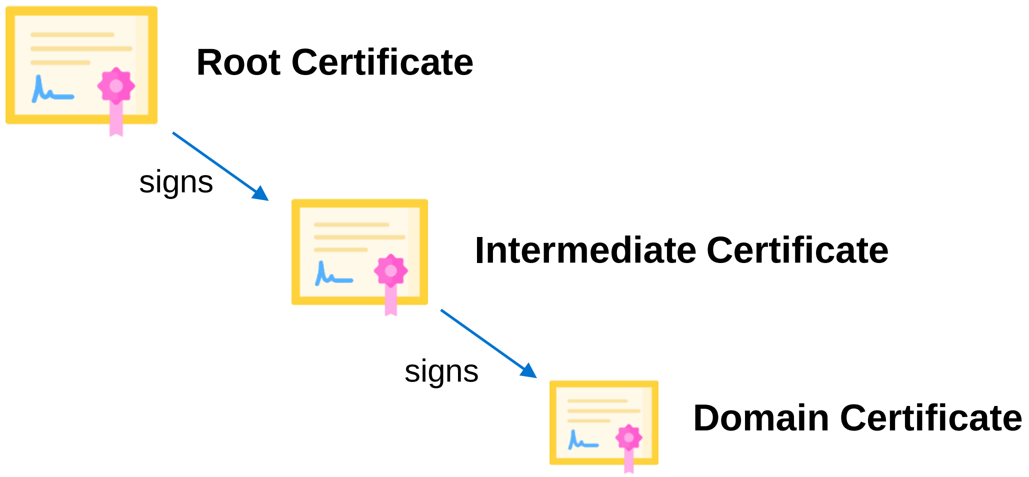


From Digital Identity by Phillip J. Windley

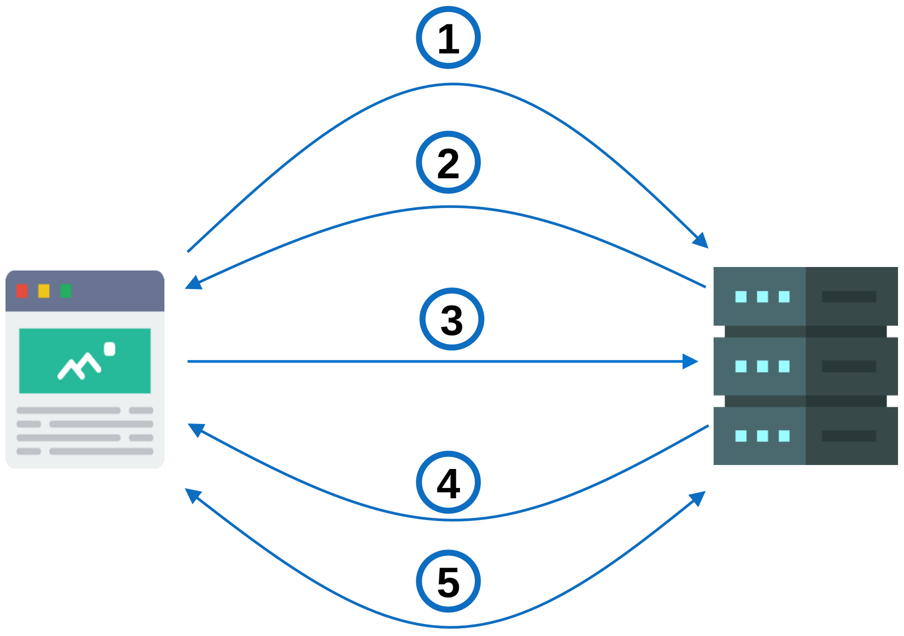
Secure connections with TLS, X.509 Certificates and PKI

1. Browser requests identification
2. Server provides SSL/TLS certificate + public key
3. Browser checks certificate
   - **Root Cert** on list of **trusted Root Certs?**
   - Matching domains?
   - Current date < Expiry date?
   - Certificate not revoked?

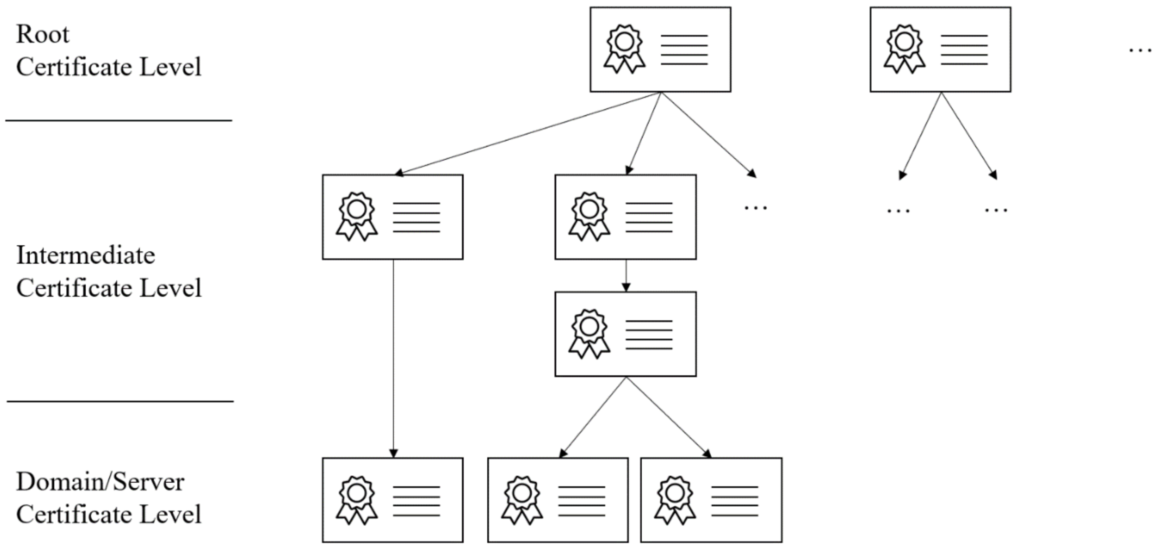
   Responds with encrypted session key by public key



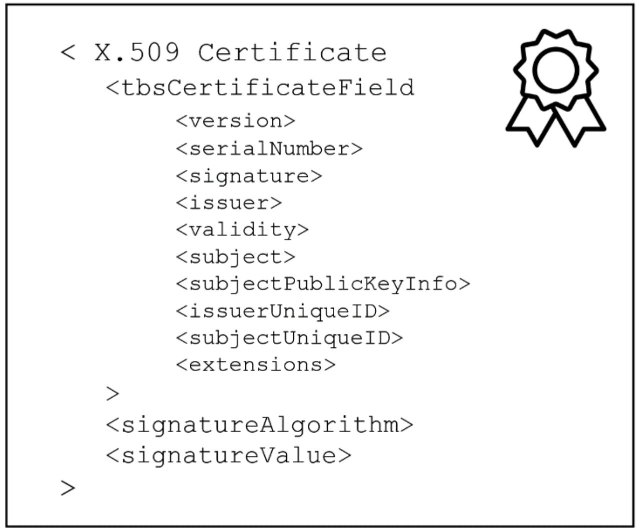
4. Server decrypts message + responds with message encrypted by the session key
5. Encrypted session



SSL/TLS, X.509 Certificates



SSL/TLS Certificate PKI

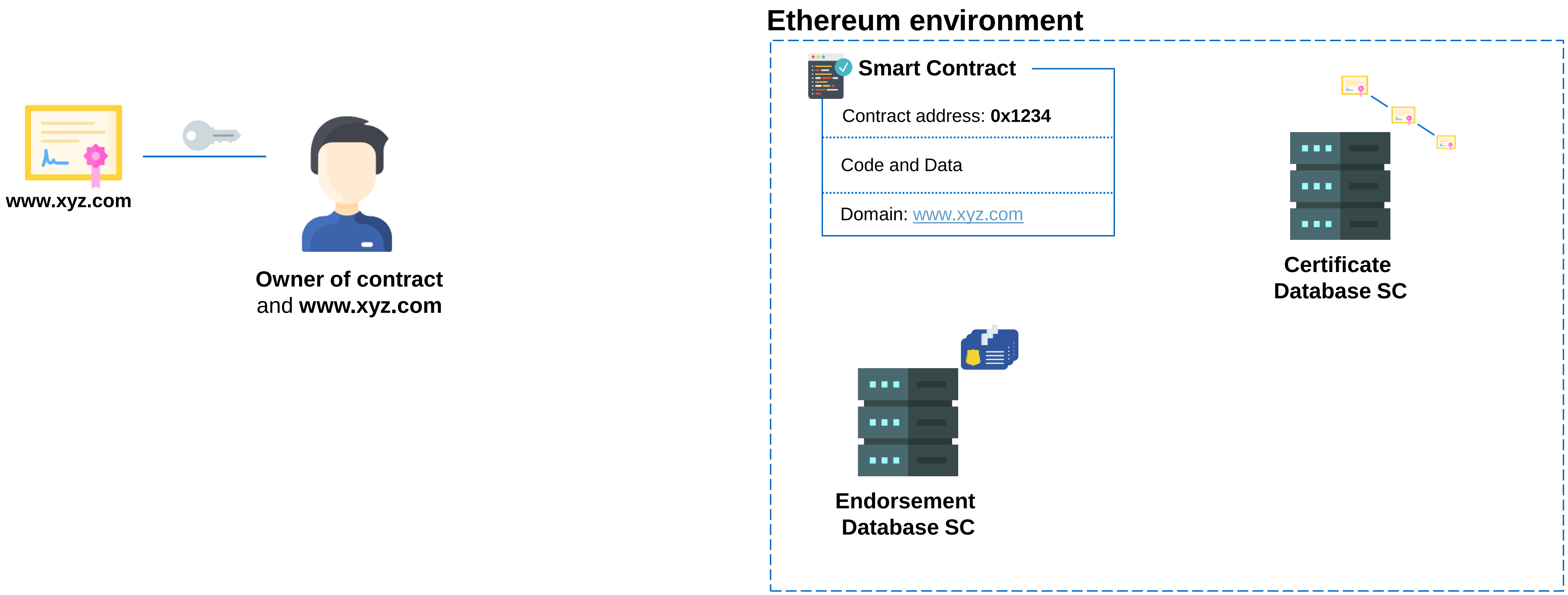


Structure of a X.509 Certificate

OID	Attribute Type	DV	OV	EV
2.5.4.3	commonName	X	X	X
2.5.4.6	countryName	-	X	X
2.5.4.7	localityName	-	X	X
2.5.4.8	stateOrProvinceName	-	X	X
2.5.4.10	organizationName	-	X	X
2.5.4.11	organizationUnitName	-	X	X

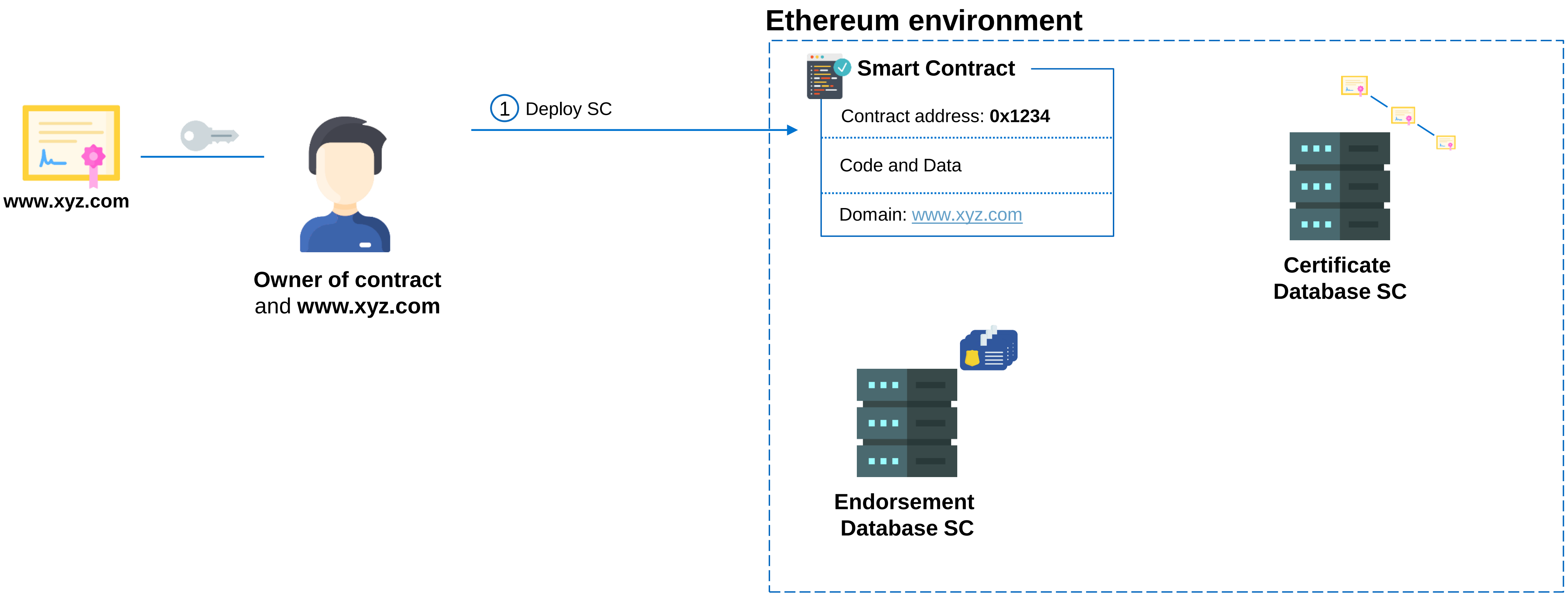
Table 3.1: Attribute types [17] supported by our ABAC system with respective OIDs [37] and certificate types

Current System: SSL/TLS-based Identity Assertion and Verification System



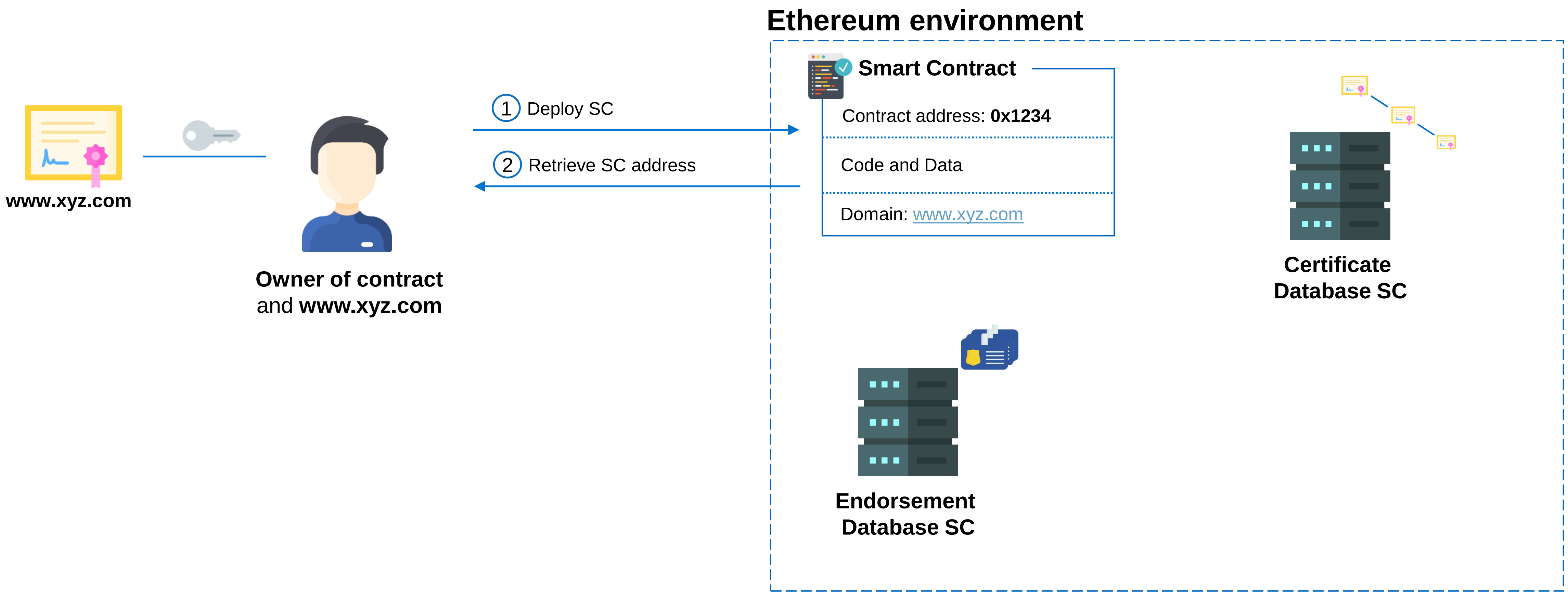
From MT Friederike Groschupp

Current System: SSL/TLS-based Identity Assertion and Verification System



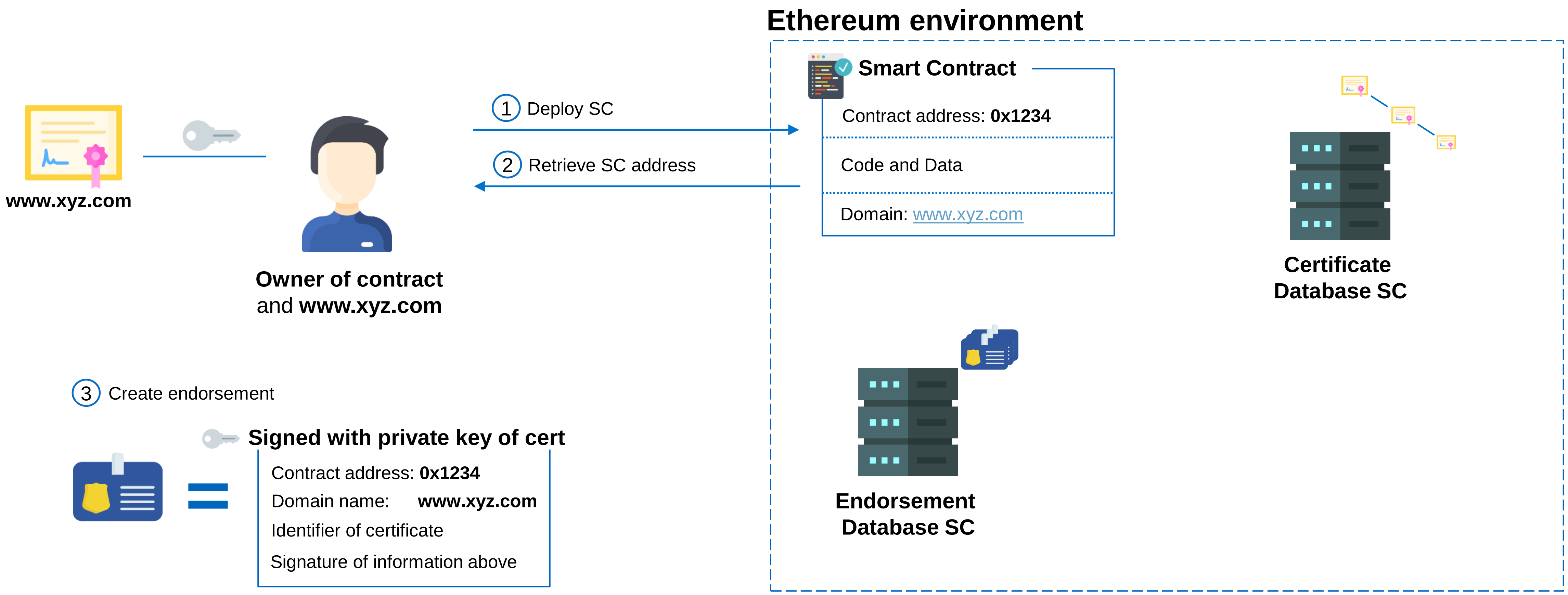
From MT Friederike Groschupp

Current System: SSL/TLS-based Identity Assertion and Verification System



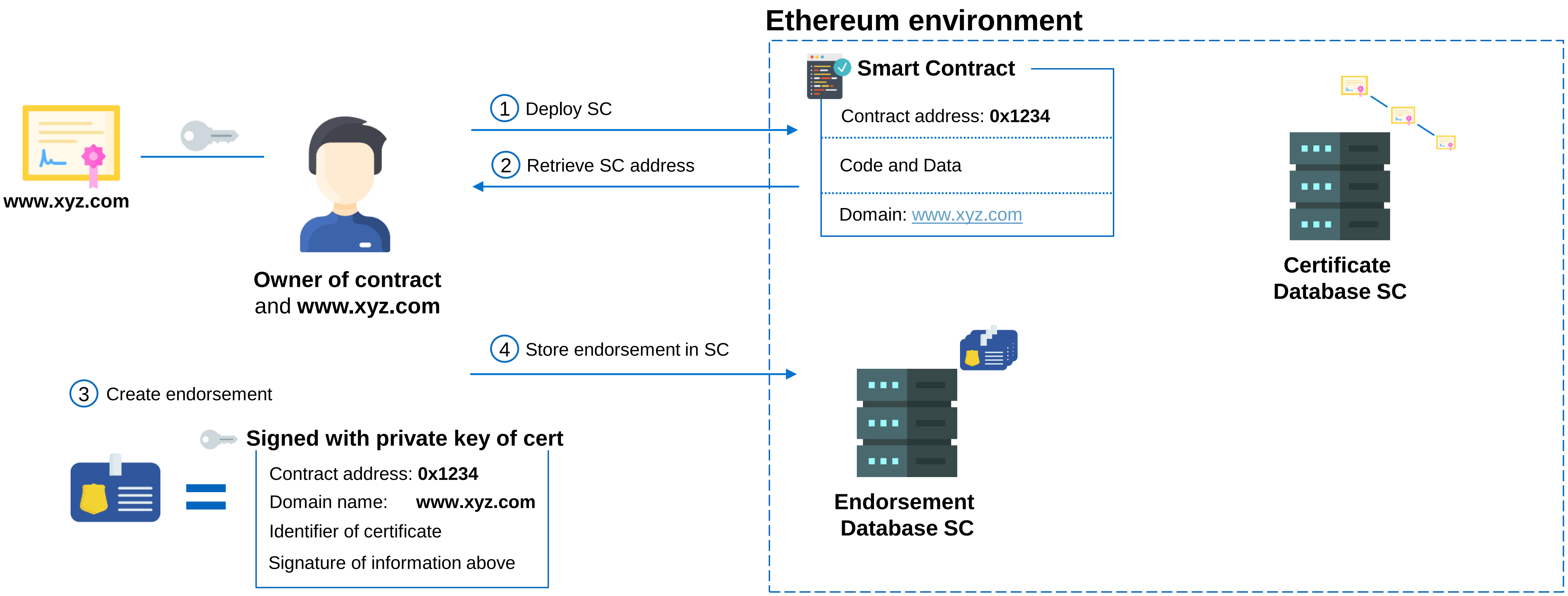
From MT Friederike Groschupp

Current System: SSL/TLS-based Identity Assertion and Verification System



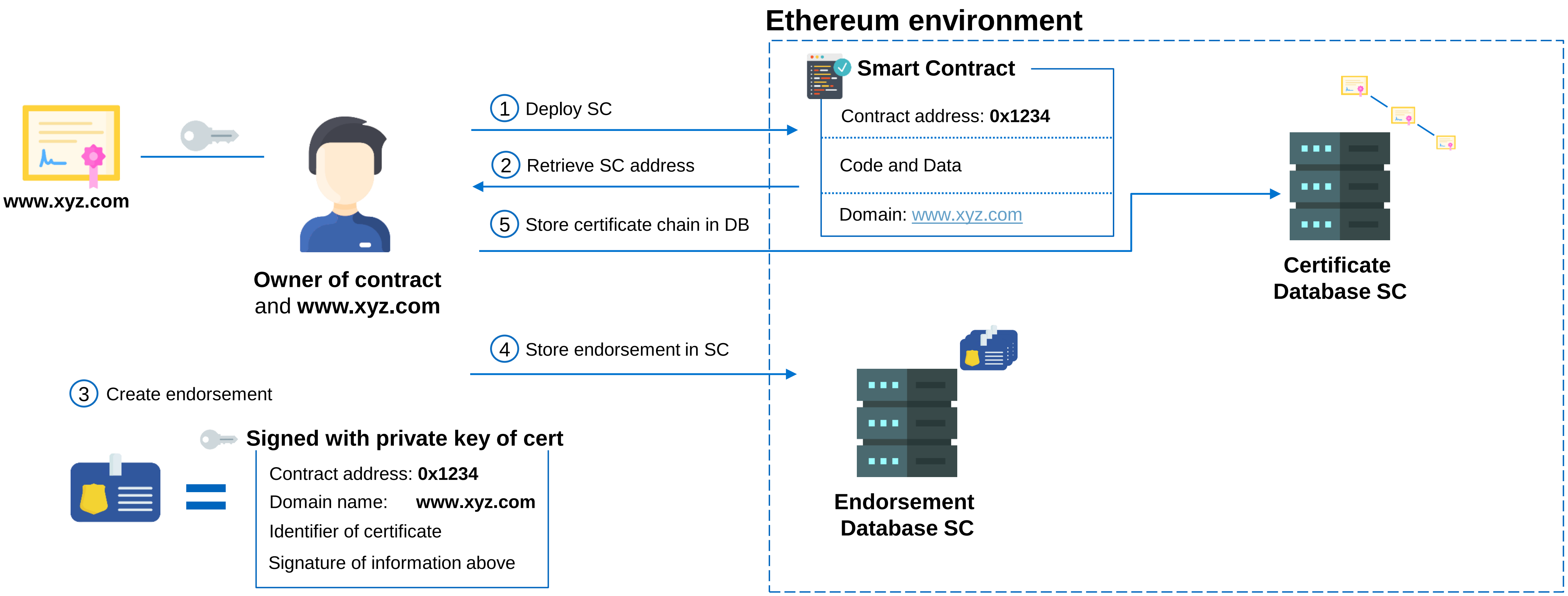
From MT Friederike Groschupp

Current System: SSL/TLS-based Identity Assertion and Verification System



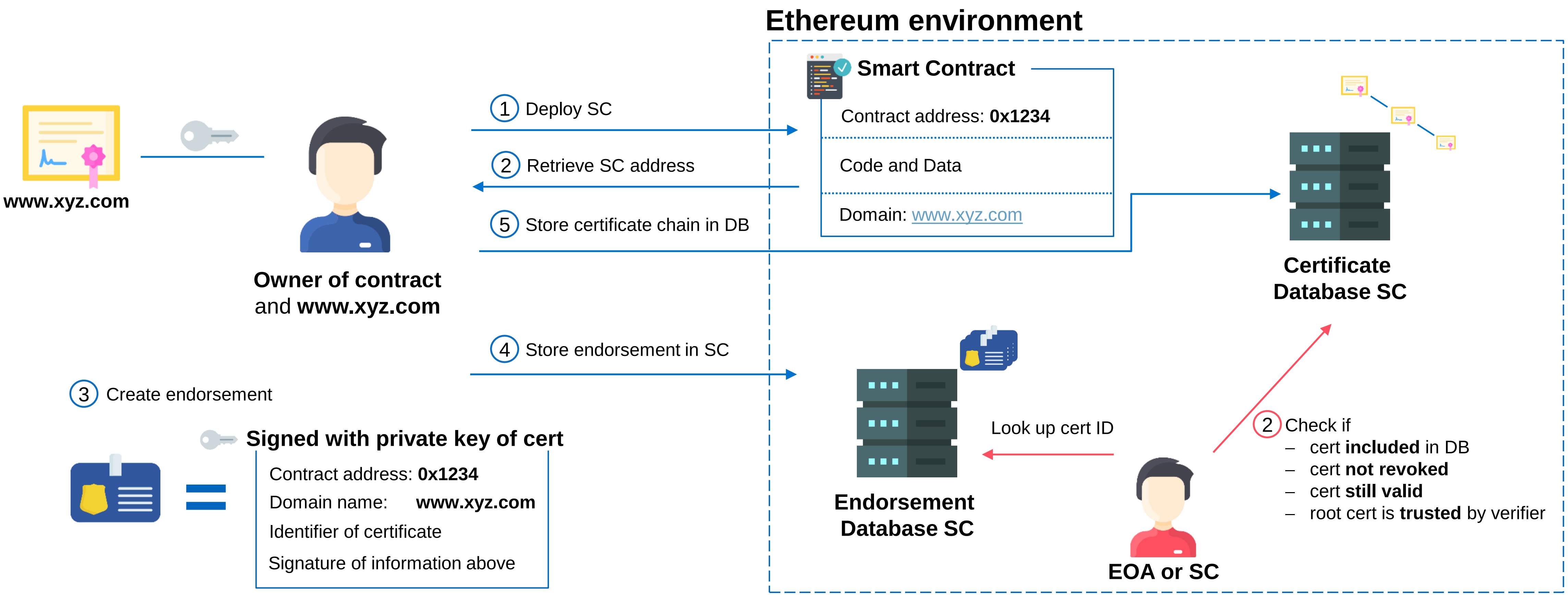
From MT Friederike Groschupp

Current System: SSL/TLS-based Identity Assertion and Verification System



From MT Friederike Groschupp

Current System: SSL/TLS-based Identity Assertion and Verification System



From MT Friederike Groschupp

Motivation

Enable authentication and access control at smart contracts

Protect access to SC functions such that they can only be invoked by authorized accounts

Authentication

"Act of proofing who the requester is"[1]

- ✓ Authentication of accounts is inherent in blockchain systems
- ✗ BUT not the authentication of the real-world identity controlling the account

Access Control

Check of requester's authorization

- ✗ No sophisticated standard for access control at SC
- ✗ No source of trust for trust endowment of user account's characteristics

User with account

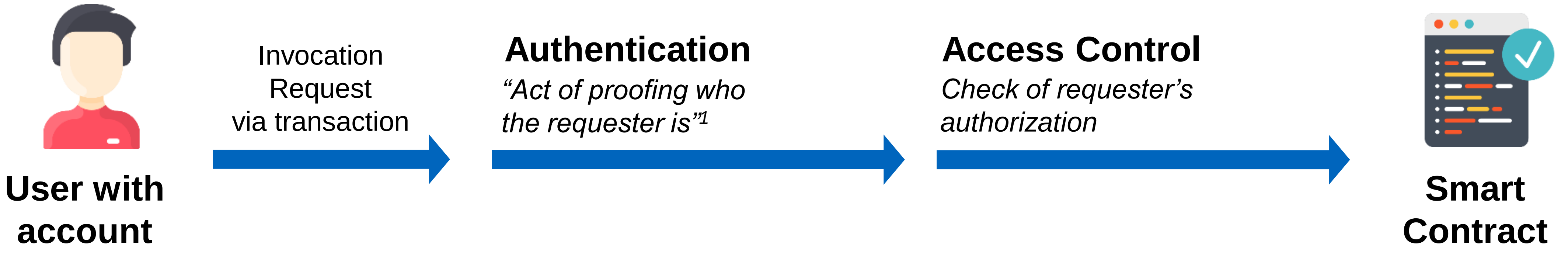
Invocation Request via transaction →

Smart Contract

[1]*Digital Identity by Phillip J. Windley*

Motivation

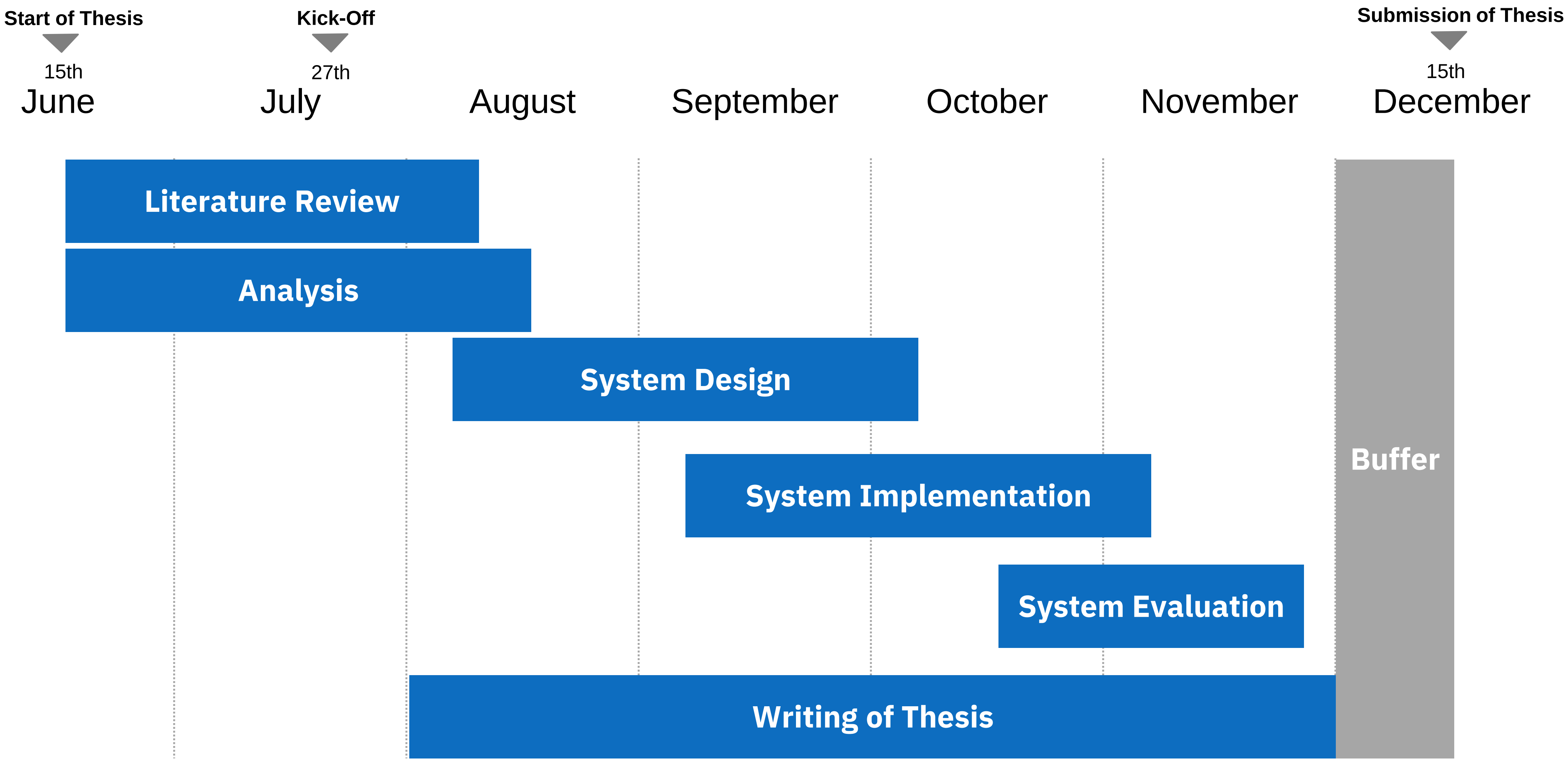
Enable authentication and access control at smart contracts
Protect access to SC functions such that they can only be invoked by authorized accounts



Leverage SSL/TLS certificates as source of trust for authorization of accounts
Protect access to SC functions such that they can only be invoked by authorized accounts

[1]*Digital Identity by Phillip J. Windley*

Timeline


Start of Thesis
15th
June
Kick-Off
27th
July
August
September
October
November
Submission of Thesis
15th
December
Literature Review
Analysis
System Design
System Implementation
System Evaluation
Writing of Thesis
Buffer