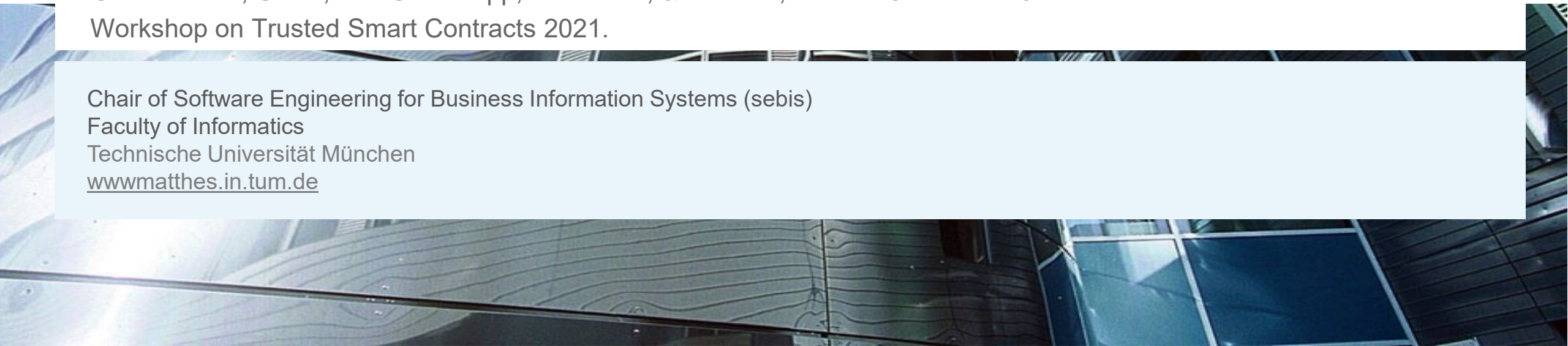
Mirroring Public Key Infrastructures to Blockchains for On-chain Authentication

Gallersdörfer, Ulrich, and Groschupp, Friederike, & Matthes, Florian. 5th March 2021.
Workshop on Trusted Smart Contracts 2021.

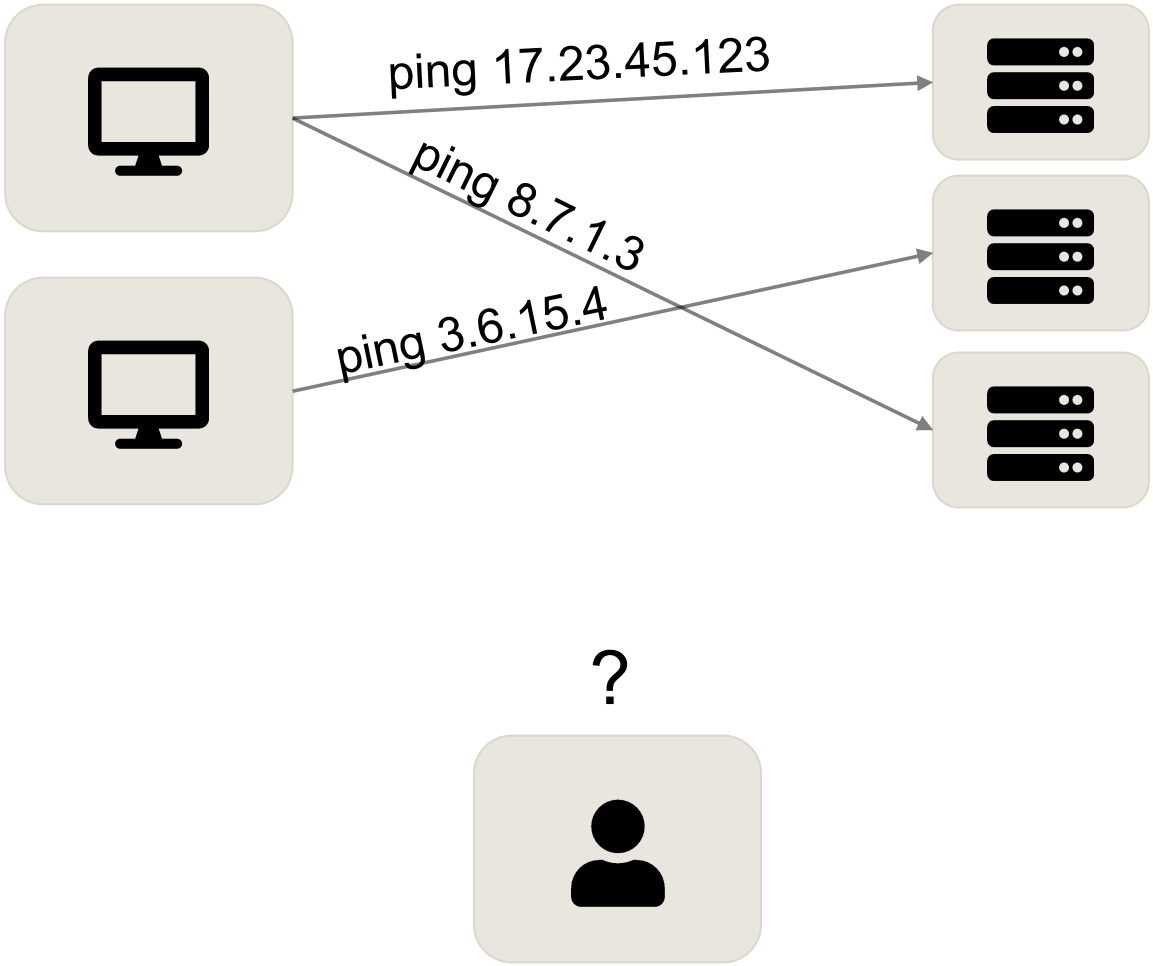


Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

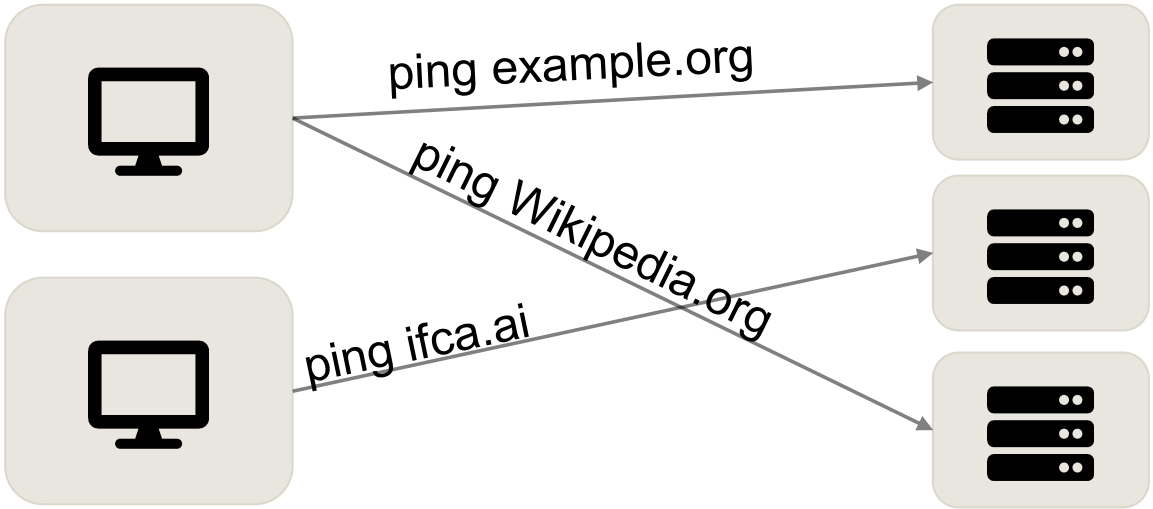
Outline

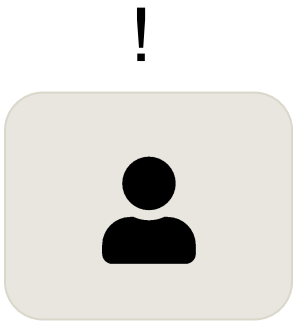
1. Motivation and Research Questions
2. Background: Transport Layer Security
3. System Design and Architecture
4. Evaluation of the System
5. Conclusion and Future Work

Communication in Web 2.0



Communication in Web 2.0: Human Readable Domains

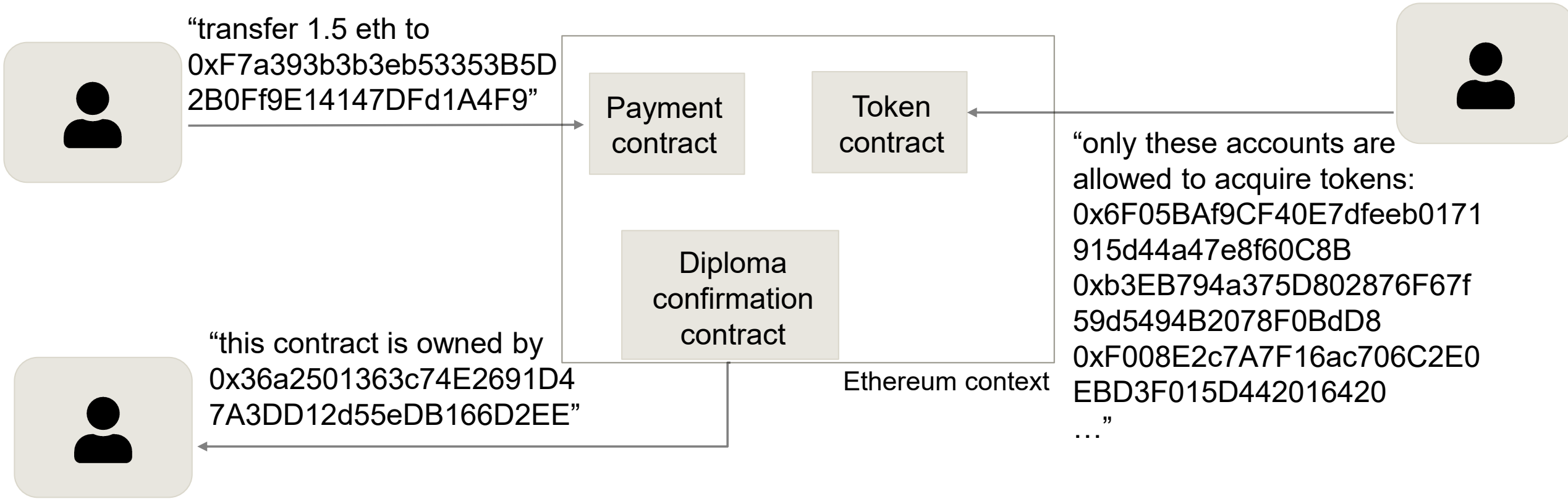




Technologies like **DNS**, **TLS**, **X.509** and others allow for human readable domains and an easy-to-use WWW and are **widely adopted**.

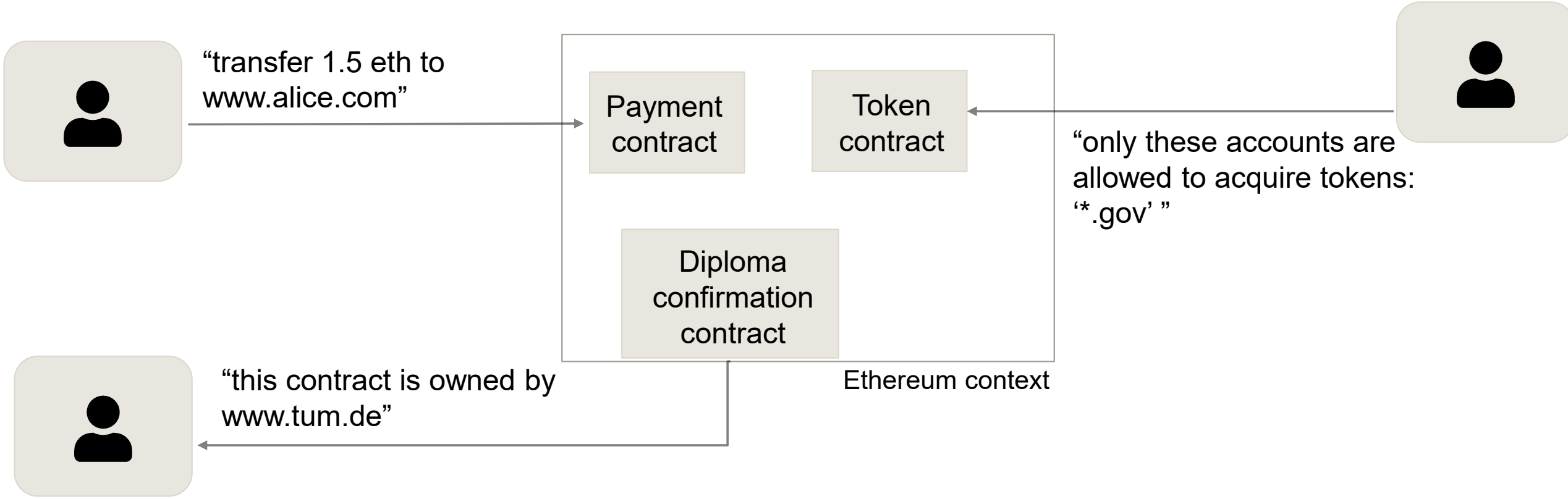
Communication in Web 3.0 / Blockchain environments

Blockchain systems face the same issues as the early WWW.



Rethinking Communication in Web 3.0 / Blockchain environments

Authenticating Ethereum addresses and their owners **enables new applications** and **promotes trust** in information and services provided.



Motivation

Authenticating Ethereum address owners **enables new applications** and on **promotes trust** in information and services provided

Slow adoption of identity solutions for Ethereum due to **lack of trusted information (e.g., ENS)**

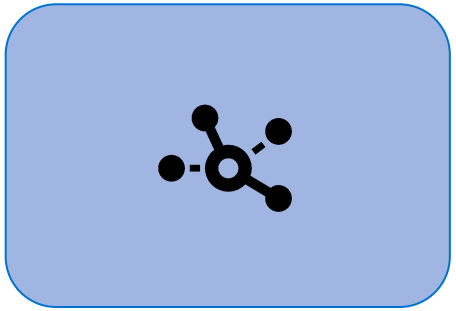
Leverage **established TLS certificates and public key infrastructure** (PKI)

Related Work

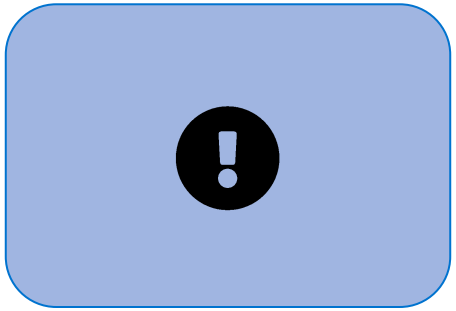
- **Blockchain-based PKI Solutions**
  - Major focus: Improve PKI solutions by leveraging blockchain technology
  - Other focus: PKI combined with Web-of-Trust solutions
  - Further approaches: certificate auditing, game-theoretic modeling, …
  - No migration of existing structures to the chain

- **Ethereum Name Service**
  - Similar idea, enable human-readable names
  - However, face large bootstrapping issues
  - More recently, work on integration of DNSSEC into ENS has shown up

Research Questions



How can naming attributes of existing PKIs in a on-chain blockchain context be leveraged?

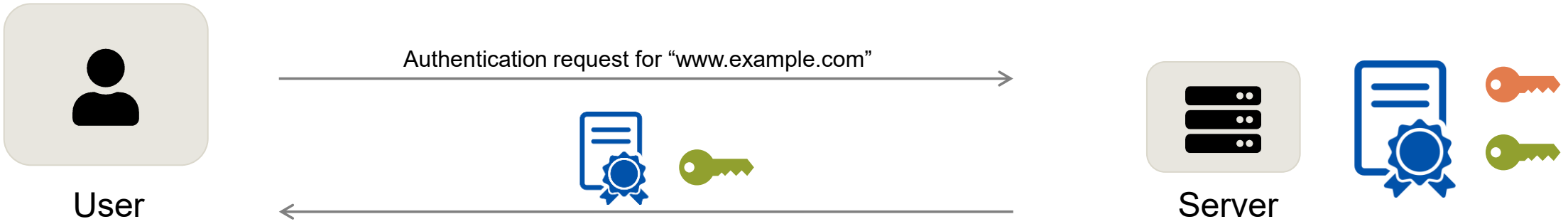


What are the constraints of leveraging existing PKIs in a blockchain environment?

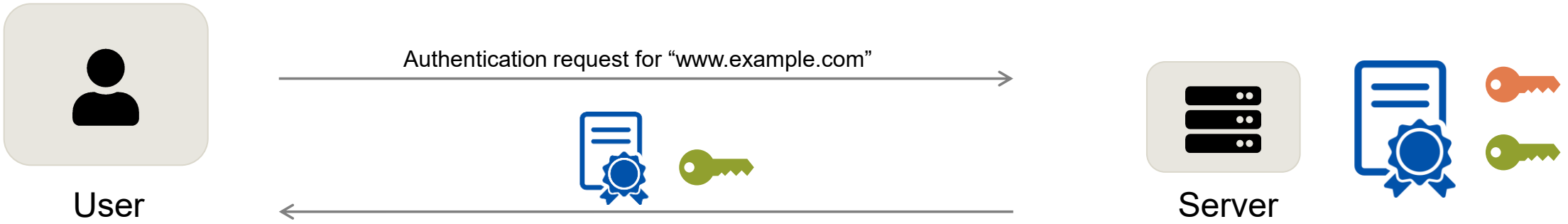
Outline

1. Motivation and Research Questions
2. Background: Transport Layer Security
3. System Design and Architecture
4. Evaluation of the System
5. Conclusion and Future Work

Background Information: TLS, Certificates, and PKI



Background Information: TLS, Certificates, and PKI



- TLS certificates bind a public key to a domain name
- TLS certificates are issued by certification authorities (CAs)
- Server proves that it “is” the domain by producing a valid signature with the private key
- User decides whether the certificate is valid based on
  - the time of validation
  - the integrity of the certificate and its certificate chain
  - whether they trust the root certificate

Outline

1. Motivation and Research Questions
2. Background: Transport Layer Security
3. System Design and Architecture
4. Evaluation of the System
5. Conclusion and Future Work

System Design and Architecture: Claim and Endorsement

Claim
- Describes a link between an Ethereum address to a Fully Qualified Domain Name (FQDN)

Claim Content
- *addr*: **Address** of the endorsed account
- ID_{domain}: FQDN of the endorsed **domain name**
- ID_{cert}:Identifier of the **certificate** used to create the endorsement
- $date_{exp}$:(optional) **expiration** date

Endorsement
- Contains the claim and proof that a claim is correct

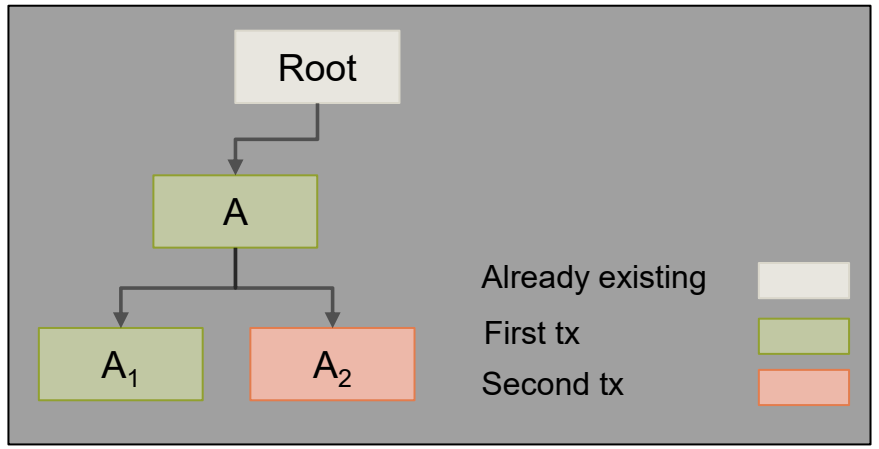
Endorsement content
- Claim itself
- **Signature** of the information above created with the private key

Endorsements are stored in a smart contract and can be retrieved by verifiers

$$C = \{addr|ID_{domain}|ID_{cert}|date_{exp}\}$$
$$E = \{C, sign(hash(C), key_{priv})\}$$

On-chain X.509 Certificate Storage and Validation

To enable **on-chain certificate validation**, we need to store the **complete certificate chain** on the Blockchain.



Exemplary certificate storage and validation

Structure:

- Stores the complete certificate chain for a new certificate
- Each certificate must be stored only once
- Root certificates can be added by anyone*

CRUD:

Create:
Submitted certificates are verified in accordance with RFC 5280.

Read:
Certificates are read by providing a unique certificate identifier.

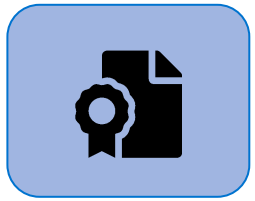
Update:
Only revocation status updates are allowed (e.g., CRL or OSCP).

Delete:
The deletion of certificates is not allowed

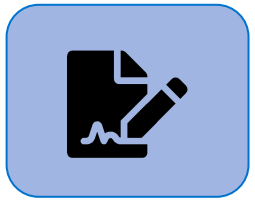
* This does not mean that the security of the system can be compromised. Every verifier has to state the root certificates it wants to trust.

On-chain Endorsement Validation

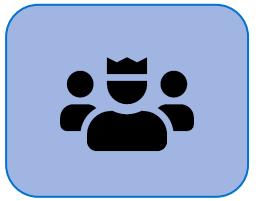
Endorsement Validation requires…



Validity of signer certificate



Validity of signature



Trusted root authority

➔ **Endorsement Database**

CRUD:

Create:
Endorsement verification using certificate storage

Read:
Either account address or FQDN is used

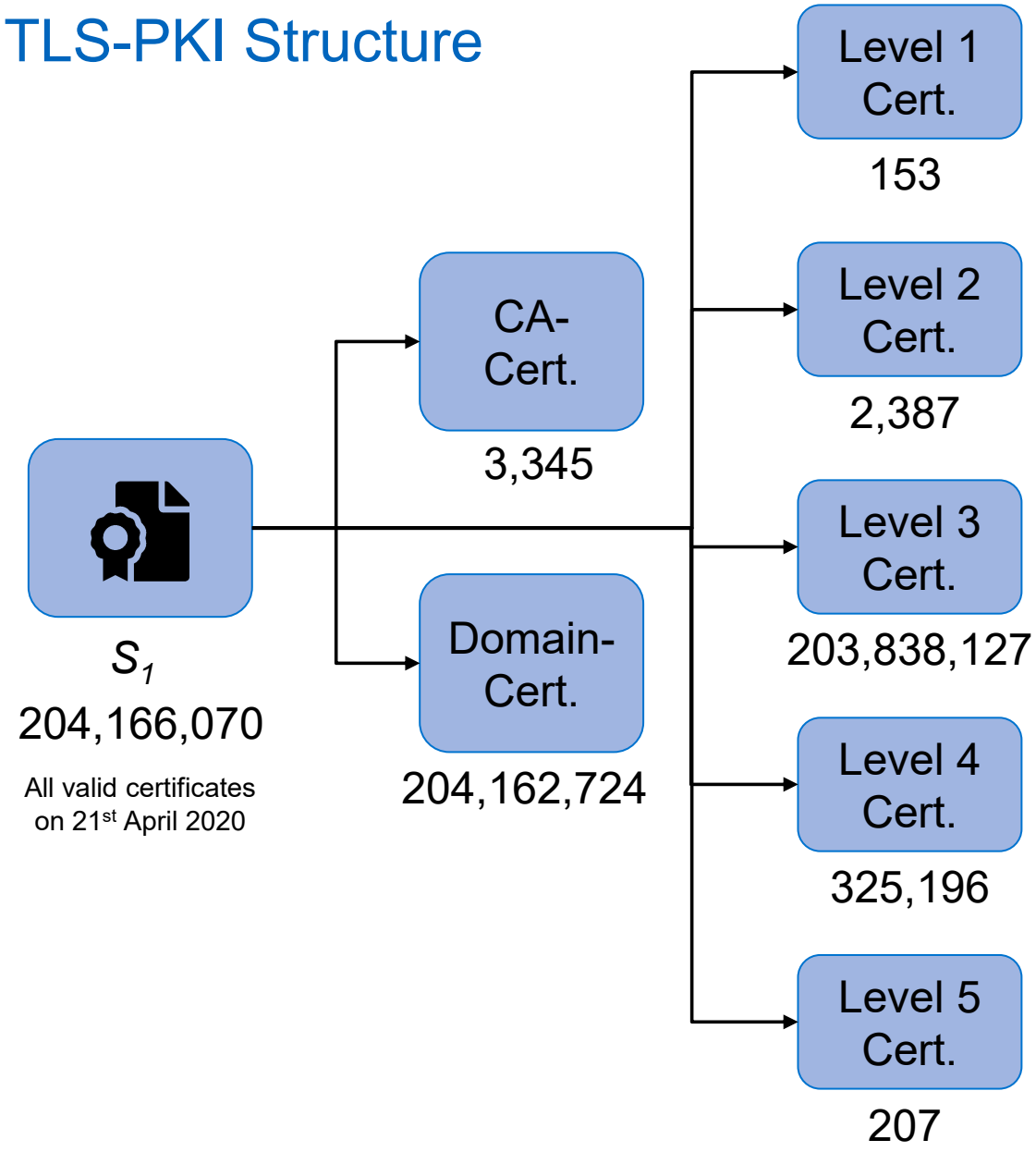
Update:
Endorsements are immutable, only revocation information

Delete:
Endorsements should not be deleted, as apps can still use them

Outline

1. Motivation and Research Questions
2. Background: Transport Layer Security
3. System Design and Architecture
4. Evaluation of the System
5. Conclusion and Future Work

TLS-PKI Structure

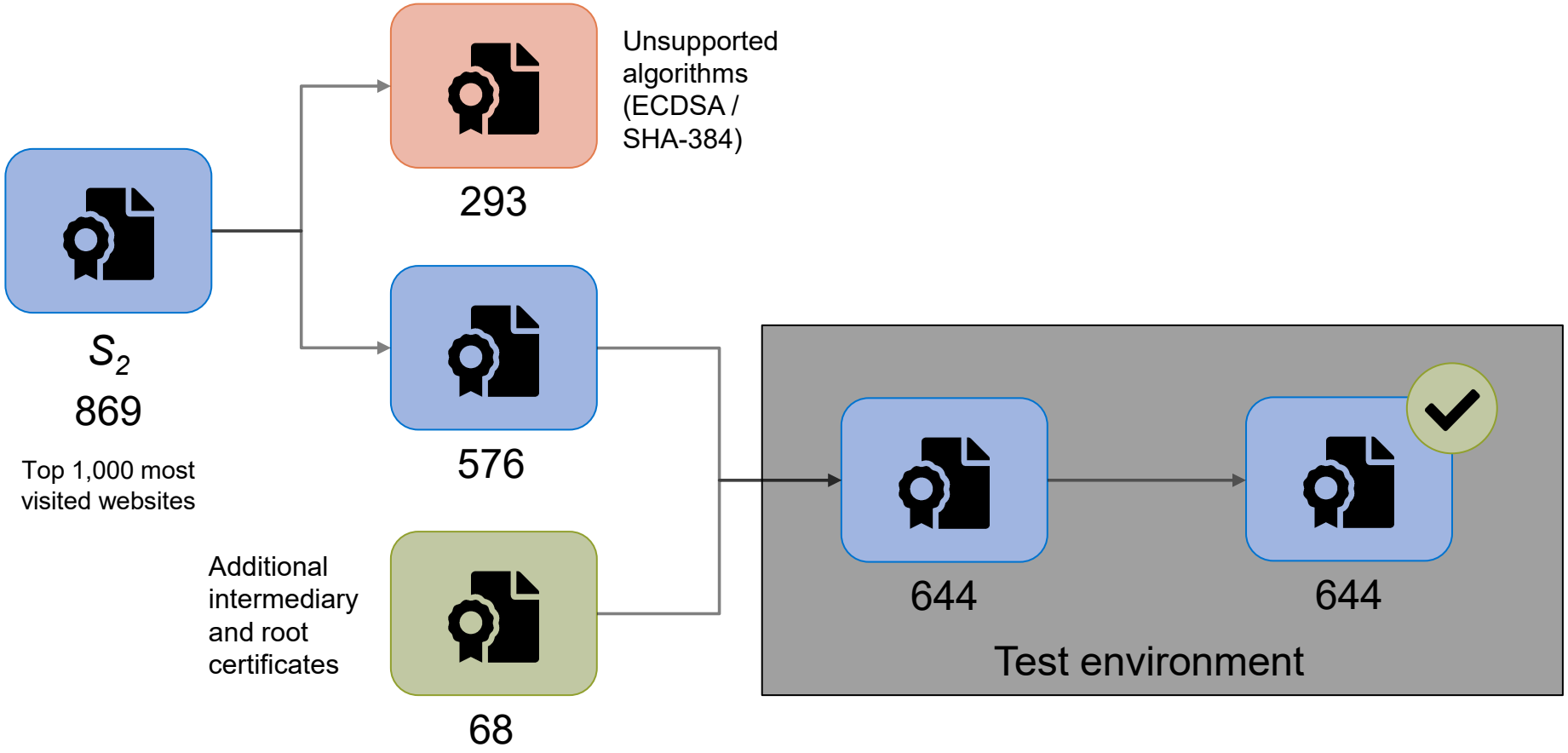


Verification becomes **cheaper** over time, as

- **Top five** intermediary certificates **cover 91%**, eight cover 95% of domain certs
- **Top six** certificate authority certificates **cover 93% of domain certificates**

➔ Trustless centralization of certificate verification has significant cost savings.

Compatibility



We were able to verify all certificates, as none of them used unusual critical X.509 extensions (which we do not support).

Prototype Performance

Submission of 576 certificates from the Top 1000 domains by daily visits

Average costs per certificate:

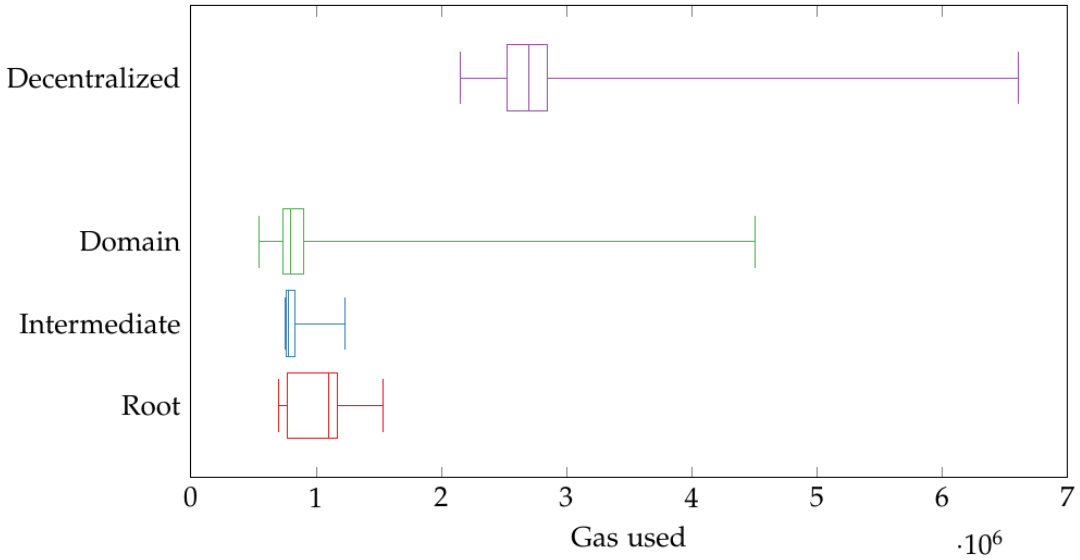
- 793,954 gas / 1,81 $ average cost per domain certificate, average **cost sinks for a higher number** of submitted domain certificates
- Decentralized approach: 2,762,042 gas / 6.25 $ average cost, does **not decrease** with more domain certificates

Cost of submitting endorsement:

- 577,219 gas / 1.32 $

Retrieving (and validating) an endorsement:

- ~ 35,000 gas / 0.08 $

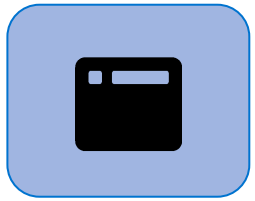


Amount of gas used for submission of root, intermediate, and domain certificates. Decentralized approach for comparison.

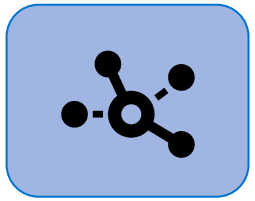
Gasprice: 11.1 Gwei
Ether price: 206 USD
April 2020

Security Considerations

Security relies on three pillars:



Security of the system itself



Security of TLS Ecosystem

Mapping domain names to entities

Outline

1. Motivation and Research Questions
2. Background: Transport Layer Security
3. System Design and Architecture
4. Evaluation of the System
5. Conclusion and Future Work

Conclusion

- Great advantage of using TLS certificates for identity assertion and verification on Ethereum: massive amount of trusted data available
- On-chain decisions are possible by migrating the necessary of the PKI on-chain
- Centralizing the validation and storage of certificates as well as of endorsements exploits characteristics of the TLS ecosystem, avoids redundancy, and reduces the total costs for all stakeholders
- Drawbacks: not a fully-fledged identity management system as only certificate owners can be authenticated, Ethereum is not cost-optimized for TLS certificates, some deem the TLS PKI unreliable

Future Work

- Extension and detailed testing of the current proof-of-concept implementation, restructure the internal endorsement scheme to make endorsement retrieval cheaper

- Investigate ways to combine TLS-based authentication framework with identity systems specifically designed for Ethereum: Bootstrap the system with TLS certificate information, but profit from the potential of an identity system designed for Ethereum

- Develop a more elaborate endorsement framework, could for example support “chains of endorsements” or different types of endorsements for different purposes

Ulrich Gallersdörfer
Friederike Groschupp
Florian Matthes

Ulrich.gallersdoerfer@tum.de
friederike.groschupp@tum.de
matthes@tum.de

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

Boltzmannstraße 3
85748 Garching bei München