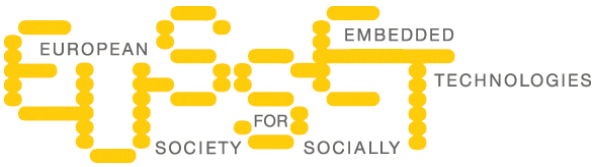
ISSN 2510-2591



Reports of the European Society for Socially Embedded Technologies

volume 2 issue 4
2018

Proceedings ERCIM-Blockchain 2018: Blockchain Engineering: Challenges and Opportunities for Computer Science Research

Guest Editors

Wolfgang Prinz

Series Editor

Michael Koch

Impressum

The **'Reports of the European Society for Socially Embedded Technologies'** are an online report series of the European Society for Socially Embedded Technologies (EUSSET). They aim to contribute to current research discourses in the fields of 'Computer-Supported Cooperative Work', 'Human-Computer-Interaction' and 'Computers and Society'.

The 'Reports of the European Society for Socially Embedded Technologies' appear at least one time per year and are exclusively published in the Digital Library of EUSSET (https://dl.eusset.eu/). The main language of publication is English.

ISSN 2510-2591

https://www.eusset.eu/report-series/

EUSSET is an institute of Social Computing e.V., a non-profit association according to the German legal system – founded on November 13th 2012 in Bonn, Germany (Nordrhein-Westfalen Amtsgericht Bonn VR 9675).

c/o Prof. Dr. Volker Wulf
Fakultät III
Universität Siegen
57068 Siegen
E-Mail: volker.wulf@uni-siegen.de

Table of Contents

Privacy-preserving KYC on Ethereum

Alex Biryukov
University of Luxembourg
alex.biryukov@uni.lu

Dmitry Khovratovich
University of Luxembourg
(now with ABDK Consulting and Evernym Inc.)
khovratovich@gmail.com

Sergei Tikhomirov
University of Luxembourg
sergey.s.tikhomirov@gmail.com

ABSTRACT

Identity is a fundamental concept for the financial industry. In order to comply with regulation, financial institutions must verify the identity of their customers. Identities are currently handled in a centralized way, which diminishes users' control over their personal information and threats their privacy. Blockchain systems, especially those with support for smart contracts (e.g., Ethereum), are expected to serve as a basis of more decentralized systems for digital identity management.

We propose a design of a privacy-preserving KYC scheme on top of Ethereum. It would let providers of financial services leverage the potential of blockchain technology to increase efficiency of customer onboarding while complying with regulation and protecting users' privacy.

Author Keywords

blockchain, smart contracts, Ethereum, know your customer, KYC

INTRODUCTION

Digital identity is information used by a computer system to represent a user. It serves two purposes:

- Authentication: to prove that the user is who they claim to be;
- Authorization: to ensure that the user has the right to perform the action they are trying to perform.

Modern financial system adheres to the centralized identity model and depends on government-issued identities. Regulation in most jurisdictions demand that banks obtain proof of identity from customers before doing business with them ("know your customer", or KYC). "Anti money laundering" (AML) and "counter terrorist financing" (CTF) are related regulations that require banks to stop and report suspicious transactions.

Modern KYC is not only cumbersome but also privacy violating. Users' sensitive information is stored in banks' databases, where it is difficult to update and can be stolen by corrupt employees or external hackers. Banks implement KYC/AML procedures independently, which leads to high compliance cost for the industry as a whole, as well as multiplies the risk of identity theft and privacy violations.

Open blockchains, the first one being Bitcoin, take a different approach to identity: users join the network without any identification. This technology enabled the creation of more sophisticated decentralized networks with rich programming capabilities, e.g., Ethereum. Banks and other financial services companies see the potential of blockchain technology and are collaborating on its applications in consortia such as Enterprise Ethereum Alliance [17], Hyperledger [1], and R3 [2]. Though to comply with regulation, they have to handle government-issued identities in a blockchain setting, which is a non-trivial task. Taking into account the users' demand for better privacy protection, this becomes even harder. The upcoming European privacy regulation (GDPR [18]) coming into force in May 2018 poses even more challenges for organizations that handle users' personal data.

We first explore the centralized and decentralized approaches to identity. We then propose KYCE – a privacy preserving Ethereum-based KYC implementation for smart contract based financial services. KYCE allows banks to implement KYC checks via an external smart contract – a KYC provider. Our scheme uses zero-knowledge proofs to check users' eligibility without disclosing their private information to anyone except the KYC provider. The whitelist is stored in the KYC smart contract in the form of a cryptographic accumulator. This construction allows users to be efficiently added to, removed from, and checked against a list without storing any plaintext data on the blockchain. We then discuss possible use cases, implementation challenges, and outline the direction for future work.

Centralized identity

We can re-formulate the notion of identity in terms of asymmetric cryptography. Identity I of user U is a public-private key pair $(pub_U, priv_U)$. The public key pub_U authenticates the user (or, equivalently, links the current action to some past actions). Public identifiers like username or address are derived from pub_U. The private key $priv_U$ allows U to sign messages on behalf of I. From the point of view of the system, U is whoever possesses $priv_U$.

Biryukov, Alex; Khovratovich, Dmitry; Tikhomirov, Sergei (2018): Privacy-preserving KYC on Ethereum. In: W. Prinz & P. Hoschka (Eds.), Proceedings of the 1st ERCIM Blockchain Workshop 2018, Reports of the European Society for Socially Embedded Technologies (ISSN 2510-2591), DOI: 10.18420/blockchain2018_09

In the centralized model of identity, which is prevalent on the internet today, users delegate managing their private keys to a trusted party and use a password to access them when necessary. This approach is sub-optimal in many regards. First of all, users do not control their identities. The trusted party always has the technical ability to sign messages without the user's consent or to prevent the user from signing the message they want. Moreover, users' personal data is stored by a centralized entity, which creates additional incentives for malicious actors to attack it. Finally, users have to create a new identity for each website they wish to register with. As a consequence, they adhere to a risky practice of reusing passwords. This problem is partially addressed with the "login with" feature, often implemented using protocols such as OAuth [25] and OpenID [28]. In this scheme, a third-party website queries the website that holds the user's existing identity (e.g., Google) and asks for permission to access a subset of the user's data (e.g., name and email). Upon approval, the access is granted. This approach alleviates the password management problem but increases the impact of a potential identity theft.

Even though users can revoke the access at any time, the "login with" scheme is still privacy violating. Imagine a user that reveals their date of birth to prove to a website that they are 18 years of age or older. Even if they later revoke the access, their date of birth will never change. Thus, they grant the third-party website an effectively unlimited access to a piece of private information.

Maintaining correspondence between "real world" identities and public keys has long been a challenge. Centralized solutions like PKI generally work, but suffer from risks associated with centralization: a fraudulent authority can issue rogue certificates [32].

Decentralized identity and open blockchains

A noteworthy approach to decentralized identity is the PGP "web of trust" [19]. It has not gained significant traction due in part to usability challenges [34] and concerns about the security of the long-term key model [42].

Bitcoin [24] is the first practical implementation of fully decentralized digital cash. It eliminates the problem of connecting public keys to identities in a radical manner: in Bitcoin, public keys *are* identities. Since its launch in 2009, hundreds of alternative open blockchains were developed, most of them adhering to this approach to identity management.

Ethereum [8] [45] is a decentralized blockchain-based smart contracts platform. Smart contracts were initially defined as "a set of promises, specified in digital form" [39]. In Ethereum, a smart contract is a piece of code in Ethereum virtual machine (EVM) bytecode, a Turing complete language. Programmers write contracts in high-level languages targeting EVM, most popular being Solidity, and deploy them onto the blockchain. Users interact with contracts by broadcasting transactions. Upon receiving a transaction, Ethereum nodes execute the corresponding function of the specified contract with given arguments. Nodes maintain a common view of the state using a proof-of-work consensus mechanism.

Contracts can call other contracts' functions and send them units of the Ethereum native cryptocurrency *ether*. Each EVM operation has a cost denominated in units of *gas* to prevent denial-of-service attacks. The user determines the maximum amount of resources their computation will consume and pays for it upfront when sending the transaction. If the computation executes normally, the user gets a refund for the remaining gas. In case of an exception, all allocated gas is consumed, but the transaction has no effect on the state of the blockchain[1].

Traditional financial institutions are becoming interested in blockchain technology, especially in networks enabling smart contracts [13]. However the way open blockchains handle identity may come at odds with financial regulation. We propose a design that will simultaneously leverage the power of blockchain-based smart contracts, enable banks to implement KYC to comply with the law, and preserve users' privacy.

KYCE: A DECENTRALIZED KYC-COMPLIANT EXCHANGE

Definitions and security properties

KYC requirements differ depending on jurisdiction [33] (see Appendix A for a brief overview of the regulatory landscape in the EU). A typical KYC procedure links users' real-world identities to their accounts and checks users against a whitelist or a blacklist. The details of the KYC procedure do not affect our design.

DEFINITION 1. *A **KYC procedure** is a process that determines if a given user is eligible for a given transaction.*

DEFINITION 2. *A **KYC provider** is an entity that performs a KYC procedure.*

DEFINITION 3. *A **financial service** is an information system that allows users to exchange units of value.*

DEFINITION 4. *A financial service is **KYC-compliant** w.r.t. the KYC procedure iff all users are eligible for all transactions they perform.*

DEFINITION 5. *A KYC-compliant financial service is **privacy-preserving** iff only the KYC provider has access to the users' private data.*

Tokens and exchanges

Our KYC solution can be applied for any type of service. For concreteness, consider a token exchange as an example of a financial service.

DEFINITION 6. *A **token** is a transferable fungible unit of value maintained by a smart contract.*

[1] After the Byzantium update in October 2017, certain types of exceptions no longer consume all gas.

ERC20 [44] is the de-facto standard API for implementing token contracts in Ethereum. A token contract keeps track of users' token balances and enables them to transfer tokens using the following functions:

- `transfer` sends a given amount of tokens to a given address.
- `approve` allows a given user to withdraw up to a given amount of tokens from the account of the user calling the function.
- `transferFrom` sends a given amount of tokens from one given address to another (the amount has to be `approve`d beforehand).

DEFINITION 7. *An **exchange** is a service that enables users to exchange tokens.*

The most prevalent type of exchanges is centralized ones, implemented as a regular web service. In this work, we are mostly interested in decentralized, or on-chain exchanges, implemented as smart contracts.

An exchange without KYC support may be used as follows.

1. Alice creates an order to sell X A-tokens for Y B-tokens.
2. Bob creates an order to sell Y B-tokens for X A-tokens.
3. The exchange matches the two orders and transfers (by calling `transferFrom`) X A-tokens from Alice to Bob and Y B-tokens from Bob to Alice.

The transaction succeeds if Alice and Bob `approve`d the exchange with sufficient amount of A- and B-tokens respectively before `transferFrom` is called. Users withdraw tokens from the exchange by calling `approve(exchangeAddress,0)`.

Privacy-preserving KYC

We propose KYCE – a privacy-preserving KYC design for Ethereum-based financial services.

A KYC contract provides an API to other contracts so that external services can determine if a given user is KYC-approved for using a given token. A KYC provider (a governmental entity or company in charge of customer onboarding) performs the necessary checks for a new customer and adds their address to the whitelist.

A naive approach to implementing KYC check with a separate contract would be the following. The KYC contract stores the whitelist of approved addresses. On every `transfer`, token contracts check if the address which is being used belongs to the whitelist. This design has a fundamental drawback from the privacy-preserving standpoint: all whitelisted addresses are stored on the blockchain in plaintext. Moreover, users must use the same addresses they registered with the KYC provider, which violates privacy: an adversary can link the user's transactions in the public blockchain.

Our approach

We use cryptographic techniques to design a privacy preserving KYC solution. In KYCE, the KYC contract stores a **cryptographic accumulator** of the whitelisted addresses.

A cryptographic accumulator A absorbs certain algebraic objects and provides an interface to generate and verify zero-knowledge proofs that a certain value was accumulated. In our construction, to generate a proof for value $x \in A$ one needs a *witness*, which depends on A and x and is provided by the accumulator owner to the user who submitted x. We suggest an accumulator based on bilinear maps due to Camenisch et al. [9].

Briefly, the KYC setup and workflow is as follows. The KYC provider creates and publishes a smart contract, which is initialized with an empty accumulator. The User interacts with the KYC provider physically or online and provides credentials needed to pass the KYC procedure. He also generates his own master secret m and during the authenticated session gives the provider a Pedersen commitment $g_1^m \cdot g_2^r$ to it, where g_1, g_2 are certain group generators[2] and r is random. If the checks are passed, the provider updates the accumulator with user-dependent data and provides the User with a witness, needed to prove the KYC property in the future. In every Ethereum transaction to KYCE, the User provides a proof that he has been registered in the accumulator, that his right has not been revoked, and that the proof owner and the transaction sender are the same person. The latter statement is verified by KYCE, whereas the rest is submitted to the KYC contract for verification against the current accumulator value. If the checks pass, the command is executed in KYCE.

Details on the accumulator construction

We follow the approach by Camenisch et al. [9], who construct an accumulator based on a pairing function $e(\cdot,\cdot)$ in some pairing setting [3]. The accumulator contains just serial numbers, possibly consecutive integers[4]. The accumulator is constructed as follows. We assume a bilinear pairing $e : G \times G \to G_T$ where G, G_T are groups of order q. The KYC provider selects generator g and the secret value $\gamma \xleftarrow{\$} \mathbb{Z}_q$. It also selects L as an upper bound of users enabled for KYC and computes $z = e(g,g)^{\gamma^{L+1}}$. The accumulator value A is initialized by 1.

Let us denote $g_i = g^{\gamma^i}$. The provider publishes $\mathrm{A}, \{g_i\}_{1 \le i \le L,\, L+2 \le i \le 2L}$, the set of registered KYC indices $V = \emptyset$, and the parameters g, z needed to perform a verification.

[2]Here and in the further text all multiplications take place in the pre-selected group of prime order q, typically an elliptic-curve group.

[3]The original paper [9] uses type-1 pairings, but type-3 pairings can be adopted as well.

[4]It is possible to store public keys but it would be less efficient.

Every User who passes the KYC check is issued a new serial number i, the witness $w_i = \prod_{j \in V, j \neq i} g_{L+1-j+i}$, where V is the set of all issued serial numbers, and a signature σ_i of $g_i||i$ on the provider's private signature key. The witness is used to generate a proof of accumulating[5]. The accumulator is updated by the KYC provider with i by

$$\mathrm{A}_{\mathrm{V} \cup \{\mathrm{i}\}} \leftarrow \mathrm{A}_{\mathrm{V}} \cdot g_{L+1-i}$$

multiplying it by $g_{L+1-i} = g^{\gamma^{L+1-i}}$, and i is published as a new valid serial number. To prove that i has been committed to A and has not been revoked without disclosing it, the holder of w_i must update it[6] so that the following equation holds:

$$\frac{e(g_i, A)}{e(g, w_i)} = z.$$

Note that revocation is also efficient: the KYC contract owner simply multiplies the accumulator value by the inverse of g_{L+1-i}. The witness value can not be updated anymore.

Presentation

When issuing a transaction to use the exchange (e.g., create an order), the user submits a **zero-knowledge proof** of the following statement:

- I know the private key of the current user address (`msg.sender`), and
- I know a signature σ_i and a witness w_i for some number i that has been accumulated in the accumulator A in the KYC contract.

It is crucial that this compound statement is *atomic*, i.e. the sub-statements can not be extracted as separate valid proofs, as this would make the transaction malleable.

The atomicity (and thus non-malleability) are ensured as follows. Let us denote the proof of knowledge for the witness and signature by PK_w, which is given in [9], Section 4.2. Then Prover submits

$$P = \{PK_w \wedge PK_s\},$$

where PK_s is the proof of knowledge of the private key of the `msg.sender`'s ECDSA public key, which can be taken from [11]. The technique to make a composite proof of knowledge is straightforward as both PoKs are non-interactive and is standard in complex PoK protocols:

1. Prover collects a set $\mathcal{C}$ of commitments asserted in subproofs PK_w and PK_s.
2. Prover makes necessary randomization of $\mathcal{C}$ to create t-values $\mathcal{T}$.
3. Prover computes $c \leftarrow H(\mathcal{C}, \mathcal{T})$.
4. Prover computes s-values $\mathcal{S}$ using $\mathcal{C}$, $\mathcal{T}$, and c.
5. The proof P is $(\mathcal{C}, \mathcal{S}, c)$. To verify it one computes asserted t-values $\widehat{\mathcal{T}}$ and verifies
$$c \stackrel{?}{=} H(\mathcal{C}, \widehat{\mathcal{T}}).$$

The resulting proof P is submitted as an Ethereum transaction argument. KYCE retrieves the most recent accumulator value and verifies P against it and the public key of the message sender, which is available in the transaction metadata. If the proof is correct the order is executed.

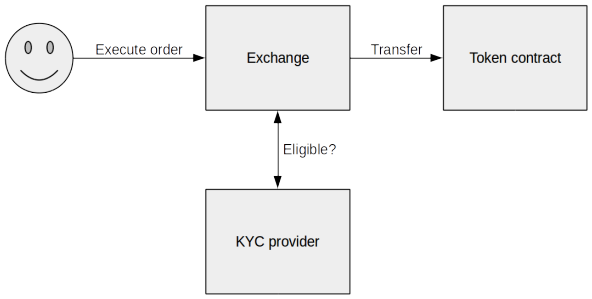
Use cases

Either the exchange contract or the token contract must be KYC-compliant – i.e., check eligibility of transacting parties using the implementation of the cryptographic scheme described above in the KYC contract.

KYC-compliant exchange

If the exchange is KYC-compliant, the tokens do not need to be aware of the KYC.

Figure 1. KYC-compliant exchange



Consider an established exchange that trades dozens of tokens. It applies for official approval in a jurisdiction that requires all customers to pass the KYC procedure. The governmental body acts as a KYC provider, deploys a KYC contract, and publishes its address. The exchange adds KYC checks to its codebase and continues operation. Users who do not want to apply for KYC can simply withdraw their tokens from the exchange and use them elsewhere.

KYC-compliant token

If the token is KYC-compliant, the exchange does not need to be aware of the KYC.

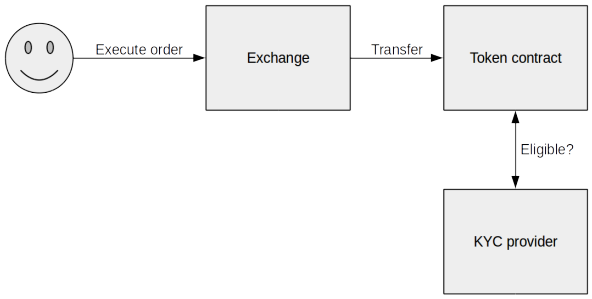
Consider a government that issues its own tokens[7]. Government tokens could be used by KYC-approved users for tax payments, fees, fines, etc. Such solution leverages the flexibility and auditability of smart contracts while limiting the userbase of the token to the approved entities only. The KYC-enabled government token can be also traded on exchanges. This allows citizens to hold their wealth in currency portfolios of their choice and

[5] We refer an interested reader to [9] for the details.

[6] We omit the details, but the update can be performed just before the presentation, not necessarily after every accumulator update.

[7] Bank of England [12] and the Monetary Authority of Singapore [4] already did research in this direction.

Figure 2. KYC-compliant token



only purchase government tokens to transact with the state.

Transaction-dependent checks

Many jurisdictions impose additional restrictions that depend on the value of the transaction. E.g., the EU regulation [30] states that "the obligation to check whether information on the payer or the payee is accurate should [...] be imposed only in respect of individual transfers of funds that exceed €1000". EU member states impose further restrictions for transactions of higher value, e.g., exceeding €10000 in Belgium, €15000 in Germany and in the Netherlands [33]. Either the exchange contract or the token contract can perform such checks by storing the following mappings:

- address => accumulated transaction volume in the current period (day, month, year);
- address => timestamp of the latest transaction.

IMPLEMENTATION DETAILS

We created a proof-of-concept implementation of the proposed design. Our project consists of two smart contracts written in Solidity: KycProvider and KyceToken.

Initial (not privacy-preserving) implementation

In the initial (not privacy-preserving) implementation, KycProvider maintains a 2-dimensional boolean array that stores the eligibility status across users and tokens. On initialization, the address that deploys the contract to the blockchain is made the *owner*, allowing it to add and remove users from the array. The ownership may be transferred (using the functionality inherited from the standard Ownable contract).

The KycProvider exposes the following API:

- `add(address _user, address _token)` – makes the user eligible for using the token (callable only by the owner)
- `remove(address _user, address _token)` – makes the user not eligible for using the token (callable only by the owner)
- `isEligible(address _user, address _token)` – checks if the user is eligible for using the token

KyceToken adheres to the de-facto standard token API in Ethereum – ERC20. To minimize the risk of security issues due to implementation subtleties, we inherit a widely used and tested ERC20 implementation by OpenZeppelin. We override the functions `approve`, `transfer`, and `transferFrom` to check if the given user (`msg.sender`) is eligible for using this token. Namely, the function `isEligible` is called. If the returned value is `false`, the execution stops; is it is `true`, the corresponding function of the super class is invoked.

The implementation of the proposed scheme requires cryptographic primitives partially already available in Ethereum as pre-compiled contracts (namely, elliptic curve addition and scalar multiplication, as well as pairing checks). For the proposed scheme to be fully implemented, pairing evaluation is also required. We are looking into the possibilities to add this functionality.

RELATED WORK

Parra-Moyano and Ross use distributed ledger technology to improve the KYC process [31]. Their proposal can be summarized as follows:

- the regulator maintains a database with all users' private data;
- the first bank a user signs a contract with (the "home bank") stores hashes of the user's documents in a smart contract in a permissioned blockchain;
- all subsequent banks the user wants to work with obtain the user's documents from the database and look the hash up to ensure that the user had been KYC-approved (without knowing which home bank had done it);
- a cost-sharing mechanism for banks allows to proportionally share the cost of the initial KYC approval among all banks that use it.

In this design, all banks store users' private data – contrary to our solution, where it is stored only with the KYC provider. A more decentralized design is also proposed, but the authors claim it to be of a lesser practical relevance.

Sullivan and Burger investigate possible implications of further development of the Estonian e-residency program using blockchain technology [38]. E-residency of Estonia is a governmental program that provides applicants with a digital identity, which can then be used, e.g., to register a company and open a bank account. Estonian e-residency disconnects a digital identity from citizenship or physical residence. Within the e-residency program, Estonia collaborates with a blockchain project Bitnation [6] [14]. Oraclize, a company that provides trusted external data to Ethereum smart contracts, implemented a connector that lets Ethereum contracts handle e-residency identities [29].

An existing project [27] implements a KYC scheme in an Ethereum smart contract, but stores the KYC status on the blockchain in plaintext.

There are multiple projects aimed at easing customer onboarding (creating an identity for a new user and ensuring KYC compliance) for banks. Some of the projects are: Cambridge Blockchain [7], Cetas [10], Fundchain [20][8], KYC-chain [22], KYCStart [15], SnapSwap [36], Tradle [40]. Blockchain consortium R3 developed a proof-of-concept implementation of a shared KYC between ten banks based on its blockchain platform Corda [3].

CONCLUSION AND FUTURE WORK

We proposed a modular design of an Ethereum-based financial service with an external KYC check, which brings benefits to all participants:

- **Users** obtain a unified identity which they can use to utilize multiple financial services. Users' personal data is stored only with the KYC provider and can be easily updated. Personal data is neither stored on the blockchain nor transmitted to third parties.

- **Financial services** greatly simplify the KYC process: it boils down to a single API call. Our design lets them cut KYC costs while at the same time diminishing risks of handling sensitive data.

- **Governments** get an opportunity to stimulate innovation in the financial sector by providing a unified and simple KYC API. This is especially important in the context of rapidly growing fintech and blockchain industries.

Our design is agnostic to the nature of the entity behind the KYC contract: it does not have to be a government body. The proposed solution can be used in any setting where a smart contract based service wants to limit the set of its users according to some criteria. For instance, many jurisdictions (e.g., the US [35]) only allow certain type of investment to be offered to "accredited investors" – typically, high-net-worth individuals and financial institutions. This logic can be replicated in a blockchain setting. Consider a blockchain-based financial service that only wants to deal with experienced cryptocurrency users (e.g., those who possess more than $10000 in ether and did their first transaction earlier than 2016). The "accrediting" functionality is delegated to a third party KYC provider. Proving net worth and previous activity on the blockchain is straightforward; additional checks can also be added. Once accredited, a blockchain investor uses multiple "restricted" services without revealing any personal details to their developers. Privacy-preserving KYC might be a good use case for Ethereum-based identity projects [23], e.g., Sovrin [37] and uPort [41].

[8] A blockchain-based asset management solution including KYC implementation.

ACKNOWLEDGEMENTS

A proof-of-concept implementation of the design described above was created in May 2017 during the Luxblock hackathon in Luxembourg by the CryptoLUX team, and was awarded a joint first prize. The team included Daniel Feher, Dmitry Khovratovich, Sergei Tikhomirov, Aleksei Udovenko, and Maciej Żurad.

REFERENCES

1. 2018. Hyperledger Business Blockchain Technologies. (2018). https://www.hyperledger.org/.
2. 2018. R3. (2018). https://www.r3.com/.
3. Ian Allison. 2016. R3 develops proof-of-concept for shared KYC service with 10 global banks. (2016). http://www.ibtimes.co.uk/r3-develops-proof-concept-shared-kyc-service-10-global-banks-1590908.
4. Monetary authority of Singapore. 2017. The future is here. Project Ubin: SGD on Distributed Ledger. (2017). http://www.mas.gov.sg/Singapore-Financial-Centre/Smart-Financial-Centre/Project-Ubin.aspx.
5. Matthias Berberich and Malgorzata Steiner. 2016. Blockchain Technology and the GDPR – How to Reconcile Privacy and Distributed Ledgers? *European Data Protection Law Review* 2 (2016), 422 – 426. Issue 3. http://edpl.lexxion.eu/article/EDPL/2016/3/21.
6. Bitnation. 2015. Estonia e-residency program & Bitnation DAO public notary partnership. (2015). https://bitnation.co/blog/pressrelease-estonia-bitnation-public-notary-partnership/.
7. Cambridge Blockchain. 2017. (2017). http://cambridge-blockchain.com/.
8. Vitalik Buterin. 2014. A Next-Generation Smart Contract and Decentralized Application Platform. (2014). https://github.com/ethereum/wiki/wiki/White-Paper.
9. Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. 2009. An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials. In *Public Key Cryptography (Lecture Notes in Computer Science)*, Vol. 5443. Springer, 481–500.
10. Cetas. 2017. (2017). https://cetas.systems/.
11. Melissa Chase, Chaya Ganesh, and Payman Mohassel. 2016. Efficient Zero-Knowledge Proof of Algebraic and Non-Algebraic Statements with Applications to Privacy Preserving Credentials. In *CRYPTO (3) (Lecture Notes in Computer Science)*, Vol. 9816. Springer, 499–530.

12. George Danezis and Sarah Meiklejohn. 2015. Centrally Banked Cryptocurrencies. *CoRR* abs/1505.06895 (2015). http://arxiv.org/abs/1505.06895.

13. Michael del Castillo. 2017. Enterprise Ethereum Alliance Adds 86 Members to Blockchain Consortium. (2017). http://www.coindesk.com/enterprise-ethereum-alliance-new-members-blockchain/.

14. e-Estonia. 2015. New Possibilities for e-residents. (2015). https://e-estonia.com/new-possibilities-for-e-residents/.

15. EconoTimes. 2017. Deloitte Luxembourg develops blockchain PoC 'KYCStart' to perform customer onboarding. (2017). http://www.econotimes.com/Deloitte-Luxembourg-develops-blockchain-PoC-KYCStart-to-perform-customer-onboarding-691965.

16. Meghan Elison. 2016. Christopher Kong: PSD2 Means Opportunity. (2016). https://ripple.com/insights/christopher-kong-psd2/.

17. Enterprise Ethereum Alliance. 2017. (2017). https://entethalliance.org/.

18. EUGDPR. 2016. EU General Data Protection Regulation. (2016). http://www.eugdpr.org/.

19. Patrick Feisthammel. 2017. Explanation of the web of trust of PGP. (2017). https://www.rubin.ch/pgp/weboftrust.en.html.

20. Fundchain. 2017. (2017). http://fundchain.lu/.

21. Viola Hellström. 2017. PSD2 – the directive that will change banking as we know it. (2017). https://www.evry.com/en/news/articles/psd2-the-directive-that-will-change-banking-as-we-know-it/.

22. KYC-Chain. 2017. (2017). http://kyc-chain.com/.

23. Elena Mesropyan. 2017. 21 Companies Leveraging Blockchain for Identity Management and Authentication. (2017). https://letstalkpayments.com/22-companies-leveraging-blockchain-for-identity-management-and-authentication/.

24. Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. (2008). https://bitcoin.org/bitcoin.pdf.

25. OAuth. 2017. (2017). https://oauth.net/.

26. Council of the EU. 2016. Money laundering and terrorist financing: Council agrees its negotiating stance. (2016). http://www.consilium.europa.eu/en/press/press-releases/2016/12/20-money-laundering-and-terrorist-financing/.

27. Mikko Ohtamaa. 2016. Know Your Customer partner integration. (2016). https://github.com/TokenMarketNet/ethereum-tokens/blob/master/KYC.rst.

28. OpenID. 2017. (2017). https://openid.net/.

29. Oraclize. 2017. Identity on the blockchain – chapter 3. (2017). https://blog.oraclize.it/identity-on-the-blockchain-chapter-3-585bc5c7e2c7.

30. European Parliament. 2015. Regulation (EU) 2015/847 of the European Parliament and of the Council of 20 May 2015 on information accompanying transfers of funds and repealing Regulation (EC) No 1781/2006 (Text with EEA relevance). (2015). http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=celex:32015R0847.

31. José Parra-Moyano and Omri Ross. 2017. KYC Optimization Using Distributed Ledger Technology. (2017). https://ssrn.com/abstract=2897788.

32. Business Unit Prins, JR and Cybercrime. 2011. DigiNotar Certificate Authority breach "Operation Black Tulip". *Fox-IT, November* (2011).

33. PWC. 2015. Know your customer: quick reference guide. (2015). https://www.pwc.lu/en/anti-money-laundering/docs/pwc-aml-know-your-customer-2015.pdf.

34. Scott Ruoti, Jeff Andersen, Daniel Zappala, and Kent E. Seamons. 2015. Why Johnny Still, Still Can't Encrypt: Evaluating the Usability of a Modern PGP Client. *CoRR* abs/1510.08555 (2015). http://arxiv.org/abs/1510.08555.

35. US securities and exchange comission. 2014. Accredited Investors. (2014). https://www.sec.gov/fast-answers/answers-accredhtm.html.

36. SnapSwap. 2017. (2017). https://snapswap.eu/.

37. Sovrin. 2017. (2017). https://www.sovrin.org/.

38. Clare Sullivan and Eric Burger. 2017. E-residency and blockchain. (2017). https://doi.org/10.1016/j.clsr.2017.03.016.

39. Nick Szabo. 1996. Smart Contracts: Building Blocks for Digital Markets. (1996).

40. Tradle. 2017. (2017). https://tradle.io/.

41. Uport. 2017. (2017). https://www.uport.me/.

42. Filippo Valsorda. 2016. I'm giving up on PGP. (2016). https://blog.filippo.io/giving-up-on-long-term-pgp/.

43. Niels Vandezande. 2017. Virtual currencies under EU anti-money laundering law. (2017). https://doi.org/10.1016/j.clsr.2017.03.011.

44. Fabian Vogelsteller. 2017. ERC: Token standard. (2017). https://github.com/ethereum/EIPs/issues/20.

45. Gavin Wood. 2014. Ethereum: A secure decentralised generalised transaction ledger. (2014). http://gavwood.com/paper.pdf.

APPENDIX

FINANCIAL AND PRIVACY REGULATION IN THE EU

The current EU legislation "on information accompanying transfers of funds" came into effect in 2015 [30]. In the wake of the rapid growth of cryptocurrencies, the EU is tightening its **anti-money laundering regulations**, stating that "virtual currency exchange platforms and custodian wallet providers will have to apply customer due diligence controls, ending the anonymity associated with such exchanges" [26]. Vandezande analyzes virtual currencies under the EU anti-money laundering law [43].

2018 is set to be a "game-changing" year for European financial industry, as two important regulations come into force.

The **Revised Payment Service Directive** (PSD2) obligates banks to provide third-party providers access to their customers' accounts through open APIs [21]. This is meant to foster competition and give rise to third-party financial service providers. For instance, unified banking API will likely make connecting banks' infrastructure to open blockchains simpler [16].

The **General Data Protection Regulation** (GDPR), coming into force on 25 May 2018, harmonizes data privacy laws across the EU [18] and introduces stricter rules for handling data of EU residents even for companies from outside the EU. Berberich and Steiner describe possible implications of blockchain adoption from the point of view of the EU data protection regulation [5].

Techruption Consortium Blockchain – what it takes to run a blockchain together

Oskar van Deventer
TNO
Den Haag, Netherlands
oskar.vandeventer@tno.nl

Frank Berkers
TNO

Mischa Vos
Rabobank

André Zandee
APG

Tom Vreuls
APG

Laurens van Piggelen
APG

Alexander Blom
Bloqzone

Bas Heeringa
BSSC

Saïd Akdim
Kamer van Koophandel

Paul van Helvoort
CZ

Leon van de Weem
Zuyderland Medisch Centrum

Douwe van de Ruit
KPN

ABSTRACT
This paper presents initial results of the Techruption Consortium Blockchain experiment. The purpose of the experiment is to learn what it takes to run a permissioned consortium blockchain infrastructure together, not only from a technical perspective, but also governance and business model. The experiment turned out to be surprisingly complex, running into buggy open-source software, extensive firewall and connectivity issues, a complex legal context, a plethora of governance issues, many business model alternatives, and an ever-present human resource limitation. Based on our experiences, we conclude that instead of developing dedicated technical infrastructure, governance and business models for each blockchain application individually, there is a need for a shared blockchain infrastructure with basic governance and business models to spur further innovation in blockchain applications and enabling technologies.

Author Keywords
Blockchain; consortium blockchain; permissioned blockchain, governance, operational/technical, business model, experiment

ACM Classification Keywords
H.4.m. Information systems applications: Miscellaneous

van Deventer, Oskar; Berkers, Frank; Vos, Mischa; Zandee, André; Vreuls, Tom; van Piggelen, Laurens; Blom, Alexander; Heeringa, Bas; Akdim, Saïd; van Helvoort, Paul; van de Weem, Leon; van de Ruit, Douwe (2018): Techruption Consortium Blockchain – what it takes to run a blockchain together. In: W. Prinz & P. Hoschka (Eds.), Proceedings of the 1st ERCIM Blockchain Workshop 2018, Reports of the European Society for Socially Embedded Technologies (ISSN 2510-2591),
DOI: 10.18420/blockchain2018_06

INTRODUCTION: BLOCKCHAIN

Blockchain has been receiving a lot of industry attention for the last few years. Inspired by the illustrious Bitcoin [1] system, initiated by the illusive Satoshi Nakamoto, many other blockchain technologies have been developed as well as numerous applications that rely on a blockchain. Some blockchain technologies are relatively general-purpose, like Ethereum [2] for enforcing smart contracts and Hyperledger Fabric [3] for running chaincode, whereas other blockchain technologies have more specific purposes, like Sovrin [4] for identity transactions, Ripple [5] for financial transactions, and BigchainDB [6] for long-term data storage.

Siegel [7] explains blockchain as "A blockchain is a shared ledger that everyone trusts to be accurate forever". A ledger is a record of transactions. Shared means that there is a single ledger that is the same for all participants. This combination is a unique selling point of blockchains, alleviating the efforts for synchronization between ledgers of individual participating organizations or individuals, and hence reducing transaction cost and bureaucracy. Trust is the keyword. Participants are no longer required to trust a single organisation for the contents of the shared ledger, but they trust a blockchain business ecosystem instead, where no single party has the power to make unauthorized changes to recorded transactions.

A blockchain infrastructure consists of nodes (servers) that are run by organizations and individuals that have an incentive to run a part of the infrastructure. Different terms are used for this business role like miner, validator, steward and server, depending of the technology that is used. We shall use the generic term "blockchain service provider". A blockchain service provider enables others to submit transactions to the blockchain and to read transactions from the blockchain. The group of blockchain service providers

validates and confirms transactions to maintain a single consistent ledger.

Mark Peplow et al [8] distinguish two dimensions to characterize blockchains: permissionless-permissioned and public-private. The first dimension is about who can act as blockchain service provider. In a permissionless ("unpermissioned") blockchain, anyone can validate and confirm transactions and consensus algorithms like proof-of-work and proof-of-stake are used to keep the blockchain consistent. In a permissioned blockchain, only identified participants can validate and confirm transactions, and some type of "byzantine-fault-tolerant" voting mechanism is used to keep the blockchain consistent [18]. The second dimension is about who can access and use the blockchain. In a public blockchain, anyone can perform blockchain transactions. In a private blockchain, only identified participants can perform blockchain transactions.

Based on these two dimensions, three types of blockchains may be distinguished [8], see Figure 1.

1) Cryptocurrency blockchain
2) Private blockchain
3) Public consortium blockchain

	Permissionless: **Anyone can validate and confirm transactions**	**Permissioned:** **Only participants can validate and confirm transactions**
Public: **Anyone can perform blockchain transactions**	1) "Cryptocurrency blockchain"	3) "Public consortium blockchain"
Private: **Only participants can perform blockchain transactions**	N/A	2) "Private blockchain"

Figure 1. Categorization of blockchain types

Cryptocurrency blockchains like Public Bitcoin and Public Ethereum have the benefit that they are up and running, and they have a proven past performance. They are public, i.e. visible and accessible to anyone. They are permissionless, so nobody can prevent one from participating. This is why start-ups typically use a cryptocurrency blockchain for their to-be-trusted core applications. However, cryptocurrencies are volatile and their blockchains are frequently forking, which is how these blockchains are governed. Escalating transactions fees and transaction confirmation times have become an issue for applications and exchanges [19]. Regular forking implies that an industry sector would continuously need to coordinate and resolve which fork to use.

Private blockchains, e.g. based on Hyperledger Fabric or Ethereum Enterprise technology, have the benefit that partners do not need to rely on others for blockchain access. Instead, the consortium partners that have agreed on an application-specific bureaucracy-reduction solution are the participants in their own joint private blockchain. Several industry sectors are already developing their own private blockchain solutions, in many cases supported by an American tech giant. Private blockchains also have disadvantages. As we are learning from Techruption Consortium Blockchain (see below), it is complex and costly to run a blockchain network together. Outsourced blockchain operations run the risks of technology and vendor lock-ins. Also, the appearance of a plethora of application-specific micro blockchains impedes trans-sectoral innovation, like combining blockchain solutions on finance and logistics, or energy and identity.

A public (permissioned) consortium blockchain facility, which may also be based on Hyperledger Fabric or Ethereum Enterprise technologies, is run by a consortium of participants for whom trust is their core business and who have the expertise to run such a facility as efficiently and reliably as possible, e.g. banks and telecom operators. A public consortium blockchain may have lower operational cost, and hence lower transaction fees, than (networks of) private blockchains. They may resolve several of the above-mentioned issues with private blockchains. Also they allow industry sectors to focus on their core business, which running blockchains is not. Whereas there are already some public consortium blockchains in existence (e.g. Sovrin for identity solutions, and Interplanetary Database for data storage), the concept is still far from being a proven solution.

TECHNOLOGICAL TURBULENCE

As illustrated above, blockchain technology is in practice a collection of technologies, both complementary and mutually exclusive ones. Moreover, this class of technologies is still heavily in development, with new alternatives and trials popping up virtually on a daily basis. This characterises the turbulence associated with the market adaptation phase [16] and brings forward high levels of uncertainty to those seeking to utilize the immense promises of these technologies. Two major strategies are available: wait until the market is stabilized or explore how the characteristics of the technology can be put to benefit. The infrastructural or platform character of the technology

implies that the blockchain technology can be used generically to enable applications. This puts forward the challenge to understand application requirements and implications for this infrastructure [15] and vice versa – in order to be able to recruit a critical mass of users of the platform. The huge difference between the number of ideas and the number of active blockchain applications suggests that also in the area of blockchain enabled applications a phase of exploration is pertinent. However, if we can manage the technical uncertainty and provide a reliable infrastructure to potential blockchain enabled application, this is likely to lead to a flux of innovation initiatives. For many, mostly incumbent organisations both these uncertainties as well as the open and fully decentralized character drive their need for experimentation, but in a more controlled environment. The inherent characteristic of blockchain technology to involve multiple parties drives the desire to do experimentation in a joint controlled environment.

TECHRUPTION CONSORTIUM BLOCKCHAIN

Techruption Consortium Blockchain is a project within the 15+ partner Techruption [9] program. The program aims to jointly develop use cases on a.o. blockchain. The project partner are a bank, a pension fund, a health insurer, a hospital, Dutch chamber of Commerce, a telco, a start-up and a research institute. The rationale of this project is that any blockchain initiative will sooner or later run into a make-or-buy decision with respect to the infrastructure on which the blockchain application runs. Will the blockchain application run on a blockchain infrastructure that is newly-created by the initiative itself? Will the blockchain application run on a blockchain that is run by third parties? What type of blockchain should the application run on? Etcetera.

When the project started spring 2017, the partners (see author list) realized that more research was needed to provide substantial insight in how to make the above-mentioned make-or-buy decision. How difficult is it to run a blockchain from the technical perspective? How much does it cost? What is the business model? What would the governance look like? What components can be outsourced/bought? How mature is the technology? What are the risks. Etcetera.

The partners are participating in the project for a multitude of reasons, which differ per partner.

- Getting experience in developing private blockchains, and using that experience for the own industry sector.
- Developing a blockchain platform infrastructure for research use and research projects.
- Developing blockchain applications, and using the developed infrastructure for technical and market testing.
- Understanding governance requirements.
- Executing business simulations.
- Working toward a professionally-run public Dutch blockchain facility, possibly including neighbouring countries.
- Networking and collaboration opportunity.

Our main research questions for this experiment was “What does it take to run a consortium blockchain together, in setting up and managing technology? What governance model and business model is suitable for managing the consortium blockchain?”. We found that an experiment – as a study for a potential exploitation - with either one of these three dimensions would fall short. Running a consortium blockchain requires a coordination with respect to e.g. versioning, sharing of node-IPs, testing and monitoring and consequently some form of governance would be required. Coordinated decisions on e.g. technology, number of nodes etc. affects performance and functionality of the consortium and consequently potential value of the infrastructure for intended users. This implies that technology and governance are relevant for the exploitation and vice versa. Therefore we chose to experiment with technology, governance and business modelling intertwined.

RESEARCH APPROACH

We followed the “Groeifabriek” approach as an innovation management approach [10] to guide us from ideation to our current stage. The chosen research methodology is action research, which is a structured form of learning by doing [11]. We approached our research questions with respect to technology, governance and business as a practical experiment in which we try to establish the required technology and define the governance and business approach for exploiting the consortium infrastructure. In this experiment we tried to apply the chosen governance principles to the project as if we were a consortium actually exploiting the infrastructure. In order to capture the consensus view on these aspects, we kept to a contribution driven blueprint document that requires contributions to be approved by decision meetings.

In the ideation stage, the general direction of the project was decided. In the exploration stage, we developed an initial business model and we set up an initial governance structure. In the experimentation stage, we experimented with the technology, governance and business models, documenting our joint consensus vision in a blueprint document [12].

The remainder of this paper discusses the learning experiences from the technical infrastructure experiments, governance design and experiences, business modelling and phasing, and envisioned next steps.

TECHNICAL INFRASTRUCTURE EXPERIMENTS

The technical infrastructure experiments were kicked off by inviting Accenture, technical partner of Techruption, to guide us through the technology selection and instantiation process.

Accenture used their Blockchain Vendor Assessment Framework to show the different options in a structured way. The process involved deciding about the layer in which the project is active (applications and solutions, platforms, base protocols & infrastructure), the functionality of the infrastructure (transaction processing and data storage, basic

distributed execution platforms, advanced distributed execution products), as well as practicalities like available technology expertise and architecture design.

Quorum [13] was chosen as blockchain technology. Quorum is a permissioned variant of Ethereum technology, and compatible with Ethereum at the application level. Quorum 1.2 was the latest version when the experiments started. Each of the Quorum node instances is running on a software stack with Docker and Linux to provide flexibility for instantiation, moving and future upgrading of nodes, see Figure 2. The software stack is maintained on a joint GitLab repository to enable proper joint version control.

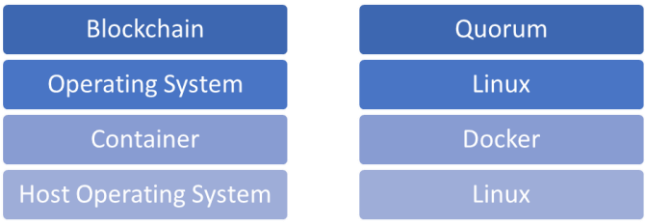


Figure 2: Software stack.

Different partners use different platforms to run the software stack: own cloud infrastructure, third-party cloud infrastructure and even a RaspberryPi. The blockchain was initiated during a one-day workshop. A genesis block was created, IP addresses and Quorum enode IDs were exchanged, firewalls were opened, connectivity was tested and the blockchain was started.

Many problems arose during the six months that the technical infrastructure is running. In many cases, it was hard to make proper technical diagnoses or find robust solutions, e.g. relating to bugs in the Quorum releases or IT settings or our own organizations, as there is a lack of central monitoring tools (who will trigger actions when some nodes are not syncing well?), whereas system logs and port scans turned out little information. Other problems were more of an organizational nature, e.g. getting the right people to develop contributions.

It took a full five days to get all five initial nodes connected and synchronized. We still do not know why this took so long. Many parameters need to be configured in the software stack, and we are only gradually learning their impact. At many times, nodes went down for unexplainable reasons, leading to speculation about the robustness/bugginess of the Quorum software. At many times, the network was diagnosed to be less than a full mesh. Lots of work went into configuring and reconfiguring firewalls, as many nodes were restarted from scratch from a different IP address. One partner had a dedicated cloud infrastructure for this type of systems, but the administrator refused access for unclear/bureaucratic reasons. One partner has a system with a multitude of firewalls, where each minor firewall reconfiguration requires a call to a helpdesk and a one-day execution time. One partner has an "intrusion prevention system" that intercepts, decrypts and re-encrypts all traffic, leading to major SSL certificate issues. Two of the nodes permanently crashed when the blockchain outgrew the assigned memory allocation. During the experiment, a new Quorum 2.0 version was released. One partner was unable to make the upgrade, whereas some other partners were unable to maintain the deprecated version. We were unable to smoothly migrate/fork the state of the initial network to Quorum 2.0, so we decided launch an independent Quorum 2.0 network.

The good news is that there were no major issues at the application level. We have successfully deployed and interacted with a multitude of Ethereum/Solidity smart contracts, including a "hello-world" smart contract, a VoIP communication-management application by one of the partners, and a self-sovereign-identity application from a neighbouring project of the Techruption Blockchain program.

As can be derived from the above, many of the technical challenges we encountered were of a generic nature, e.g. firewall configuration, access to skills, buggy software and the coordination required to diagnose it. This suggests that setting up consortium blockchains could benefit from proper design of governance and allocation of personnel for setting up, testing, debugging and accepting the technology.

GOVERNANCE DESIGN AND EXPERIENCES

The governance design was kicked off using a governance framework developed by TOBLOCKCHAIN, a blockchain start-up and partner of Techruption. Three questions are central in this framework.

- What issues can the decisions be about?
- How are decisions made? (decision process)
- Who participates in the decision process?

The list of potential issues governed by the decision process is a long one, including technical choices for the software stack, version control of the used third-party open-source software, technical requirements on connectivity and firewalls, monitoring and maintenance of key performance parameters, division of cost and revenues, procedures for onboarding new partners and new customers, procedures for offboarding, business model and phasing, and of course the decision process itself.

The governance of the *project* was split in an informal process and a decision process, see Figure 3. During the informal process (workshops, etcetera) opinions are formed and consensus is sought. Volunteers make written proposals and change request based on this. The decision process is centred around a blueprint document, that is updated at every decision meeting (semi-weekly conference call), based on decisions on the provided inputs, see Figure 4.

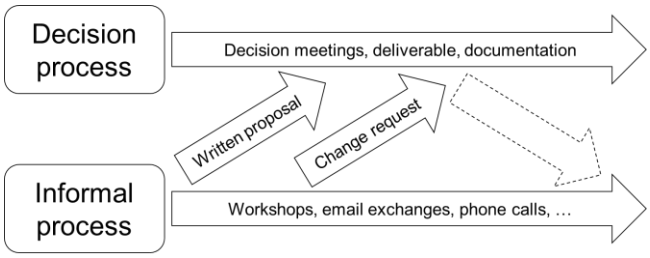


Figure 3: Governance of the project

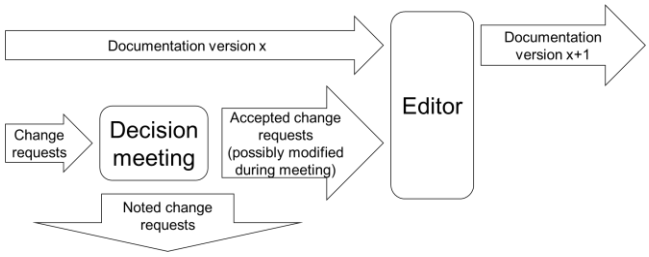


Figure 4: Handling change requests at a decision meeting

As one could expect, the practice was a bit more complicated. The initial version of the governance process, as we designed for the *future ecosystem*, requires full consensus between the founding fathers of the project. However, we never had a full set of representatives present at our decision meetings due to conflicting appointments, illness and so on. Also, one partner withdrew during the project, while two others joined. A more practical approach was to achieve consensus between those present at the meeting, and assuring that enough people are present. This shows that the governance framework is dynamic.

Also the volunteer-based contribution-driven approach has its limitations. Having multiple authors has its quality impacts, including variations in writing style and terminology. Moreover, not all partners could contribute equally, so some partners carried a larger contribution load than others. Still, a strong point of the chosen approach is that all partners have a stake in the resulting blueprint, as it was developed by the partners themselves and contributions were included by consensus.

The experience suggests that the governance model needs to be adaptable to the (increasing) complexity of the situation and yet be pragmatic. As can be observed in the governance of many foundations, working groups lead by champions that focus and take responsibility on certain aspects could have contributed to a more smooth advancing of topics – as opposed to the all-contribute all-decide approach.

BUSINESS MODELLING AND PHASING

Similar to the governance and technology tracks, this topic was initiated with an informal and interactive session based on the business model canvas [14] led by the Groeifabriek.

In order to further facilitate the scoping choice of the consortium, TNO prepared a so called strategic options model, see Figure 5. This model combines the phases of innovation with platform business model theory [15]. The latter essentially distinguishes a platform or infrastructure on which multiples sides interact (also referred to as multi-sided market). In our case these sides are represented by the applications that require a blockchain ("demand") and the providers of components that jointly make up the blockchain infrastructure ("supply"). These sides theoretically have network effects as blockchain applications only make sense if the underlying infrastructure is sufficiently large in number of nodes and has sufficient support. On the other hand, contributing to a blockchain infrastructure only makes sense if sufficient applications utilize this infrastructure. The innovation management perspective distinguishes exploration (or experimentation) from exploitation [16]

In the exploration phase, new technologies are tried and tested and eventually, if considered feasible applied for exploitation. This means that the technology is actually used in business. The combination of the two leads to distinguish exploration and exploitation in both applications as well as platform. Moreover, if considered more closely, it would make sense to exploit an infrastructure that is specifically suited to support the experimentation with blockchain applications in order to spur the innovation of blockchain applications (A2 in Figure 5). After all, value lies in the actual use of applications (A3 in Figure 5).

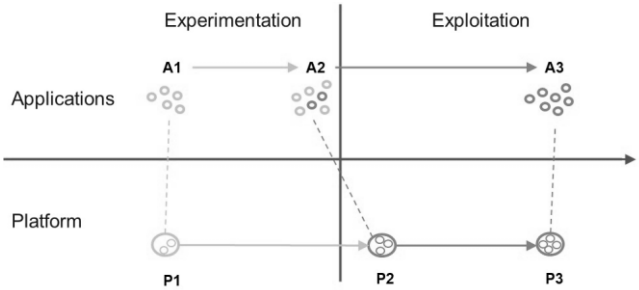


Figure 5: TCB Strategic Options Model

We named this platform "P2", a 'professionally ran platform for the experimentation with blockchain applications', to distinguish the phase of the platform in between "P1", where the platform itself is experimental (current phase) and "P3", where the platform is suited for real-life blockchain applications that rely on existence of a well-managed infrastructure. This distinction is similar to the testnets for public blockchains. The consortium unanimously opted for the P2 scope as a target for the consortium blockchain. This is all further documented in [12]. The P2 platform has value for its founding fathers who have plenty of blockchain application ideas as well as start-ups interested in developing blockchain or complementary applications.

Later on, based on the implied analogy with systems implementation procedures that typically distinguish development, test, acceptance and production [17] it was established that "P3" would be the 'acceptance test platform' which closely resembles the production environment and "P4" is the production environment. This distinction did not affect the choice for P2.

At that time, it was unclear whether the consortium was considering the platform to evolve from P1 to P2 etc. or whether P1, as an explorative infrastructure, could exist besides P2. Based on the current pace of development in blockchain technologies, the consortium still has a need to explore such technologies, hence chose to maintain a P1 instance besides P2.

In order to specify the value proposition [20] of the consortium blockchain in more detail another informal session was devoted to identify the service elements that TCB provides to its users. It was decided that TCB would not support the application developers' development phase. This is something that developers typically do "offline".

At this point TCB neither provides support for acceptance testing or production as the specs for such support are not clear yet. Thus TCB focuses on the 'test phase' for application developers. The support can be split into two categories: during test and prior to test. The services provided for testing are fairly basic and include the necessary APIs and monitoring and alerting. No 24/7 helpdesk will be operational, but support will be provided at best effort through the TCB community and an escalation/routing mechanism.

The services prior to testing are mainly informational services that the developer needs to prepare for the testing. These include: configuration information; platform status and performance information; "Service Level Agreement" (what a developer can expect); a roadmap of development plans for additional functionalities; a manual that describes procedures for deployment, testing and decommitting; deployment automation software; release management systems; optional generic functionality (e.g. identity management) that can be included in the applications (e.g. as libraries); procedures for proposals from the developers to the infrastructure; cost and performance information.

Currently these services are targeted at the parties and their ventures that are part of the Techruption community. This puts BSSC, as the host organization for the Techruption community, in a key-role to adopt and orchestrate the further development of the TCB. The Techruption community is open for participation by third parties. At this point, and based on a developed roadmap for technical, organizational and ecosystem development, TCB estimates to need around 6-9 months to fully achieve this "P2" stage.

NEXT STEPS

With the first project phase completed, the partners are now at the stage of committing to the next phase, which includes a next level of professionalization. More clarity needs to be obtained on the cost of running this blockchain ecosystem, the value of the ecosystems and transactions to its customers, the acquisition of such customers, tariffing models and further professionalizing the governance. There is also work to be done at the technical level, including improving (dockerised) template deployment to minimize the faults and standardise the configuration, network monitoring to identify failing nodes, strict rules and rigid mitigations on version management, handling of deprecations in smart contract languages, and implementing security measures.

At the time of writing this paper, there are still a significant number of open issues to be addressed for the exploitation phase. Should we engage in a utility service? How can we benefit as a business? Should this all be governed by an umbrella organisation? Should we go commercial or non-profit? How open should the platform service be? Would it run only private blockchains or also public blockchains? Should permissionless blockchains also be considered? Although fundamental for the organisation's future in actually using blockchain technology in their business, these issues do not disqualify the need for "P2". On the contrary, further development and exploration is needed to answer these questions.

REFLECTION AND CONCLUSIONS

The experiment was evaluated using brainwriting and face to face discussion and focused on identifying aspects to maintain and things to improve in the collaboration. Aspects that emerged (no prior structure was given) include strategy, scope and output as well as operational organisational aspects. Key insights beyond a unanimous agreement on willingness to continue were that the group was perceived as multidisciplinary, open minded, constructive and very knowledgeable. Remarkably so, since participants were not selected on a specific profile. The group agreed that this was definitely an aspect to cherish. On a more critical note the group concluded that more emphasis should be on actually iteratively developing and further scaling the infrastructure and other achievements valuable for users of the infrastructure. This, rather than emphasis on, but explicitly not replacement of, analytical discussion and documentation. A more formal project plan was deemed instrumental. The group considered this now to be a natural moment for the shift from an informal, open and explorative experimentation phase to more structure, although some of the participants clearly desired that some time ago.

After nine months of action research, we conclude that setting up and running a consortium blockchain together is much more complex than we had anticipated. The group deliberately chose not to hire (turn-key) solution providers for either of three areas technology, governance and business in order to learn in practice. Consequently learnings include new aspects and aspects known (elsewhere). Overall it became clear that setting-up and running a consortium blockchain can be considered a business in its own right. We learned how much the software is still developing, that there are lots of things to be configured in the software stack and that it is hard to technically maintain a stable-running blockchain. Also the governance turned out to be complex, so many issues for which procedures would need to be developed, and already running into scale issues with only a handful of partners. Moreover, the experiment clearly

illustrated how much technology, business and governance of a blockchain infrastructure at this phase of technological turbulence are intertwined. We believe that neither one of these aspects can be meaningfully developed in isolation. Group learning undeniably includes phases of getting to know each other and developing a common language. This feels unproductive in phases, however the different perspectives and backgrounds eventually added to the level of understanding. Thus, on top of the basic blockchain infrastructure and the governance and business principles documented in the blueprint, we consider the joint team as a fourth asset put forward by this experiment.

Together, we have developed a roadmap for which we have just completed the experimental "P1" phase, starting further professionalisation with the "P2" phase soon.

We recognize that many of the challenges that we have run into are independent of the number of blockchain instances that we are running. The governance processes are not much harder for running multiple blockchain instances, compared to just one. The same goes for technologies and business models. So it makes a lot of sense to run multiple blockchain instances and technologies in parallel, instead of developing dedicated technical infrastructure, governance and business models for each blockchain instance individually.

ACKNOWLEDGMENTS

This work was partly funded by TKI-toeslag.

REFERENCES

1. Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf
2. Vitalik Buterin. 2013. Ethereum White Paper: a next generation smart contract & decentralized application platform. http://www.the-blockchain.com/docs/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf
3. Hyperledger Fabric. https://www.hyperledger.org/projects/fabric
4. Sovrin. For self-sovereign identity and decentralized trust. https://sovrin.org/
5. Ripple. One Frictionless Experience To Send Money Globally. https://ripple.com/
6. BigchainDB. The blockchain database. https://www.bigchaindb.com/
7. David Siegel. 2017. What is this Blockchain Thing?, https://medium.com/startup-grind/what-is-this-blockchain-thing-a5d2abb99297
8. Mark Peplow, et al. 2016. Distributed Ledger Technology: beyond block chain. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf
9. Techruption. 2016-… https://www.techruption.org/
10. GroeiFabriek. https://groeifabriek.com
11. Mårtensson, Pär, and Allen S. Lee. "Dialogical action research at omega corporation." MIS Quarterly (2004): 507-536.
12. Oskar van Deventer et al. Techruption Consortium Blockchain – blueprint of a permissioned blockchain. March 2018. Available through the first author.
13. J.P.Morgan. Quorum, advancing blockchain technology. https://www.jpmorgan.com/global/Quorum
14. Osterwalder, Alexander, and Yves Pigneur. Business model generation: a handbook for visionaries, game changers, and challengers. John Wiley & Sons, 2010.
15. Eisenmann, Thomas, Geoffrey Parker, and Marshall W. Van Alstyne. "Strategies for two-sided markets." Harvard business review 84.10 (2006): 92.
16. Ortt, J. Roland, and Jan PL Schoormans. "The pattern of development and diffusion of breakthrough communication technologies." European Journal of Innovation Management 7.4 (2004): 292-302.
17. DTAP, https://en.wikipedia.org/wiki/Development,_testing,_acceptance_and_production
18. Shehar Bano et al. SoK: Consensus in the Age of Blockchains. https://arxiv.org/abs/1711.03936v2. 2017.
19. Joseph Young, "Ethereum's high fees have become an issue for applications and exchanges", https://btcmanager.com/ethereums-high-fees-issue-for-applications-and-exchanges/, January 2018
20. Radziwill, Nicole. "Value Proposition Design." The Quality Management Journal 22.1 (2015): 61.

Developing an Evaluation Framework for Blockchain in the Public Sector: The Example of the German Asylum Process

Gilbert Fridgen
University of Bayreuth
Fraunhofer FIT
Wittelsbacherring 10
95447 Bayreuth, Germany
gilbert.fridgen@fim-rc.de

Florian Guggenmos
University of Bayreuth
Fraunhofer FIT
Wittelsbacherring 10
95447 Bayreuth, Germany
florian.guggenmos@fim-rc.de

Jannik Lockl
University of Bayreuth
Fraunhofer FIT
Wittelsbacherring 10
95447 Bayreuth, Germany
jannik.lockl@fim-rc.de

Alexander Rieger
University of Bayreuth
Fraunhofer FIT
Wittelsbacherring 10
95447 Bayreuth, Germany
alexander.rieger@fim-rc.de

André Schweizer
University of Bayreuth
Fraunhofer FIT
Wittelsbacherring 10
95447 Bayreuth, Germany
andre.schweizer@fim-rc.de

Nils Urbach
University of Bayreuth
Fraunhofer FIT
Wittelsbacherring 10
95447 Bayreuth, Germany
nils.urbach@fim-rc.de

ABSTRACT
The public sector presents several promising applications for blockchain technology. Global organizations and innovative ministries in countries such as Dubai, Sweden, Finland, the Netherlands, and Germany have recognized these potentials and have initiated projects to evaluate the adoption of blockchain technology. As these projects can have a far-reaching impact on crucial government services and processes, they should involve a particularly thorough evaluation. In this paper, we provide insights into the development of a framework to support such an evaluation for the German asylum process. We built this framework evolutionarily together with the Federal Office for Migration and Refugees. Its final version consists of three levels and eighteen categories of evaluation criteria across the technical, functional and legal domains and allows specifying use-case specific key performance indicators or knockout criteria.

Author Keywords
Blockchain; Public Sector; Migration; Asylum; Evaluation Criteria; Use Case Evaluation

Fridgen, Gilbert; Guggenmos, Florian; Lockl, Jannik; Rieger, Alexander; Schweizer, André; Urbach, Nils (2018): Developing an Evaluation Framework for Blockchain in the Public Sector: The Example of the German Asylum Process. In: W. Prinz & P. Hoschka (Eds.), Proceedings of the 1st ERCIM Blockchain Workshop 2018, Reports of the European Society for Socially Embedded Technologies (ISSN 2510-2591), DOI: 10.18420/blockchain2018_10

INTRODUCTION
With digitalization rapidly advancing, organizations both public and private increasingly face emerging digital technologies with the potential to improve their processes, products, and services. At the same time, these technologies can also disrupt current business models and change external expectations [7, 21, 43, 45]. One of these emerging technologies currently dominating public perception is blockchain [4, 23]. It first appeared as the technological backbone behind bitcoin [36]. Since then, blockchain has evolved rapidly, and 2nd generation blockchains such as Ethereum provide smart contract functionalities which enable considerably broader applications [5, 51]. These smart contracts, or "chain-code", allow embedding of executable logics on a blockchain [48, 51]. Exemplary applications of these 2nd generation blockchains include crowdfunding [45], supply chain processes and mechanisms [28, 35], security as well as privacy in the internet of things [12, 47], and the energy sector [30, 33]. Initiatives such as decentralized autonomous organizations (DAOs) take an even further step and leverage smart contracts to automate the organization's processual logic entirely [15]. Based on this increasing number of options, both academia and practitioners increasingly argue that blockchain could have a groundbreaking impact on society [4, 29, 37, 45].

Opinions on the merits of blockchain differ, however. Whereas some organizations worry about its effects, others consider it a promising IT infrastructure [14, 23, 45]. While this ambiguity effectively calls for guidelines on how to assess the impact of blockchain [43], research is still predominantly invested in exploring its theoretical foundations [3, 5, 45] and technological details [6, 11, 44]. In contrast, evaluation guidelines and criteria are only

available for selected applications in the financial sector [19, 20], cryptocurrency security [13], social businesses (e.g., crowdlending) [45], logistics [35], or the evaluation of smart data projects [2].

For the public sector, however, such criteria and guidelines do not yet exist [17, 29, 41, 44]. Our research aims to fill this gap and support evaluation of potential use cases of blockchain technology in the public sector. We thus took an action design research (ADR) approach [46] to develop a blockchain use case (BUC) evaluation framework and validated it as part of a proof of concept project with the Federal Office for Migration and Refugees (Bundesamt für Migration und Flüchtlinge – BAMF). This project aims to evaluate the applicability of blockchain in the German asylum process.

We began our framework development by conducting a systematic literature review, following the methodology of Okoli and Schabram (2010), in the area of blockchain, emerging technologies, and evaluation criteria to derive valid ex-ante criteria [39]. Based on these criteria, we developed an ex-ante framework (i.e., the α-cycle of our ADR approach) which we validated in interviews and stakeholder workshops (i.e., the β-cycle of our ADR approach) to derive an ex-post framework of BUC evaluation criteria.

We acknowledge that these evaluation criteria present only a first step towards a general framework for the evaluation of blockchain technology in the public sector. Nevertheless, we are confident that they can support our BAMF use case and offer guidance for comparable use cases.

The remainder of this paper is structured as follows: First, we introduce blockchain technology, present selected examples of successful blockchain applications in asylum processes, and ultimately explain challenges in the German asylum process. After that, we explain our methodological approach. In the findings section, we describe the ex-ante framework, offer insights from the proof of concept project, and present the resulting ex-post framework. We also explain the identified criteria in detail. Finally, we discuss generalizability, rigor, and relevance of our findings, provide managerial implications, and offer an outline for further research.

THEORETICAL BACKGROUND

Blockchain

Satoshi Nakamoto introduced blockchain technology in 2008 to provide a distributed digital ledger for Bitcoin transactions [4, 5, 36]. Since 2008, global interest in blockchain has increased substantially, and many practitioners and researchers believe that it has the potential to change various industries radically [4]. As of 2018, blockchain has evolved into a multipurpose technology, and researchers and practitioners are exploring its applicability in many areas beyond cryptocurrencies [4].

A blockchain is a transparent, transactional, distributed database stored redundantly on the nodes of a peer-to-peer (P2P) network [22]. Research also describes it as an electronic registry for digital records, events, or transactions managed by the participants of a distributed computer network [45]. Blockchains store data in blocks with a chronological, structured order in which each block contains a reference to the previous block [18]. A so-called consensus algorithm run by selected or all participating nodes provides consistency and determines the correct order of the blocks (in the "chain") [22]. A large number of these consensus algorithms exist, and each of them provides slightly varying levels of security, latency, and energy consumption [9, 53]. Aside from their consensus mechanisms, blockchain systems also differ in their level of read/write permissions, centralization, and efficiency [9, 40, 53]. In general, blockchains emphasize data redundancy [42], use of cryptography [42] and consensus algorithms [18, 42], as well as decentralization [53] and auditability [53]. A more detailed description of these characteristics can be found, e.g., in [45]. Many blockchains also offer "smart contract" functionalities [17]. Smart contracts are "self-executing scripts" that incorporate exogenous effects or check exogenous conditions [9].

International Applications of Blockchain in Asylum Processes

Many ideas have emerged on how the public sector could capitalize on blockchain. The German Competence Center on Public IT ("Kompetenzzentrum Öffentliche IT") [50], for instance, expects promising potential in the context of:

- electronic parliamentary elections,
- cooperation between different administrations (i.e., digitization and acceleration of administrative processes),
- publicly managed registers and the administration of legal titles such as cadastral offices or land registers,
- integrity of data and documents (e.g., replacing the (digital) signature),
- origin of (pre-)products, and
- legally compliant inter-organizational collaboration.

Governments and international organizations have already begun to adopt blockchain technology, in particular, to support asylum processes. In Jordan, for example, the UN uses blockchain in a refugee camp to identify refugees unambiguously. Upon arrival, the camp's managing organization assigns and stores on a blockchain a unique refugee ID based on iris scans. The managing organization then couples the ID with a specific financial balance that allows refugees to purchase groceries in the camp's supermarket. The system has proven successful and has reduced identity fraud perceptibly [16, 24].

Finland similarly introduced a blockchain solution for refugees. As refugees often do not possess valid IDs, they cannot open bank accounts. The Finish blockchain solution provides such an ID to refugees and allows them to obtain

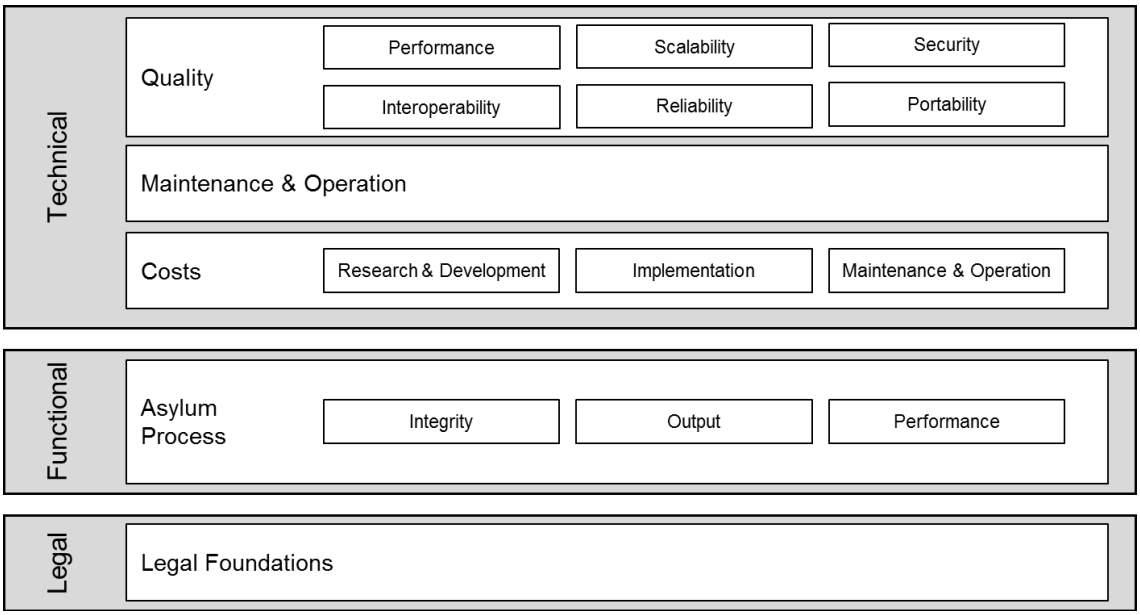


Figure 1. Evaluation Framework (Stage 1)

maestro cards linked to this ID. The card grants a certain degree of financial independence and serves both as a means of payment and as an identification instrument [31]. Moreover, Dubai considers a broad adoption for government services, including visa applications [10].

Challenges in the German Asylum Process

Ministries and organizations involved in the German asylum process face various challenges that present both opportunities and hurdles to the adoption of blockchain technology. Importantly, these organizations operate under a considerably stricter set of statutes and laws than private sector companies do. These laws effectively govern processes, responsibilities, and information exchange. They also change at frequent intervals and necessitate adjustments of processes and technologies supporting these processes. In federal systems, such as Germany, public sector organizations are also subject to different bodies of state and federal law. At the same time, proximity to lawmakers and frequent legal overhauls can present fertile opportunities to create a beneficial basis for the adoption of blockchain technology.

The involved organizations often operate different IT-systems with little mutual integration. They also partly rely on non-automated information exchange, even though considerable operational dependencies exist. This lack of integration can threaten process integrity and can lead to delays and errors. At the same time, it presents promising applications for technologies such as blockchain that can integrate various systems without requiring significant adjustments to legacy infrastructure. Process integration between these organizations is also often challenging due to separate jurisdictions. At the same time, the law requires that these organizations collaborate effectively. Hence, a technology that enables such cooperation offers essential benefits.

Table 1. Used literature for stage 1

Author(s)	Sector
Abramova and Böhme (2016) [1]	E-Commerce
Akoka and Comyn-Wattiau (2017) [2]	IT/IS
Brenig et al. (2016) [8]	IT/IS
Eskandari et al. (2015) [13]	Finance
Fridgen et al. (2018) [19]	Finance
Fridgen et al. (2018) [20]	Finance
Glaser (2017) [22]	IT/IS
Hyvärinen et al. (2017) [26]	Public Finance
Janze (2017) [27]	Publishing
Nærland et al. (2017) [35]	Logistics
Notheisen et al. (2017) [38]	Finance
Pilkington et al. (2017) [41]	Politics
Schweizer et al. (2017) [45]	Finance
Smith and Dhillon (2017) [47]	Law

METHODOLOGICAL APPROACH

Public sector organizations require suitable evaluation criteria to assess the benefits of different blockchain solutions for the asylum process. These criteria need to reflect all relevant technical aspects as well as functional (use case related) requirements. Moreover, the involved

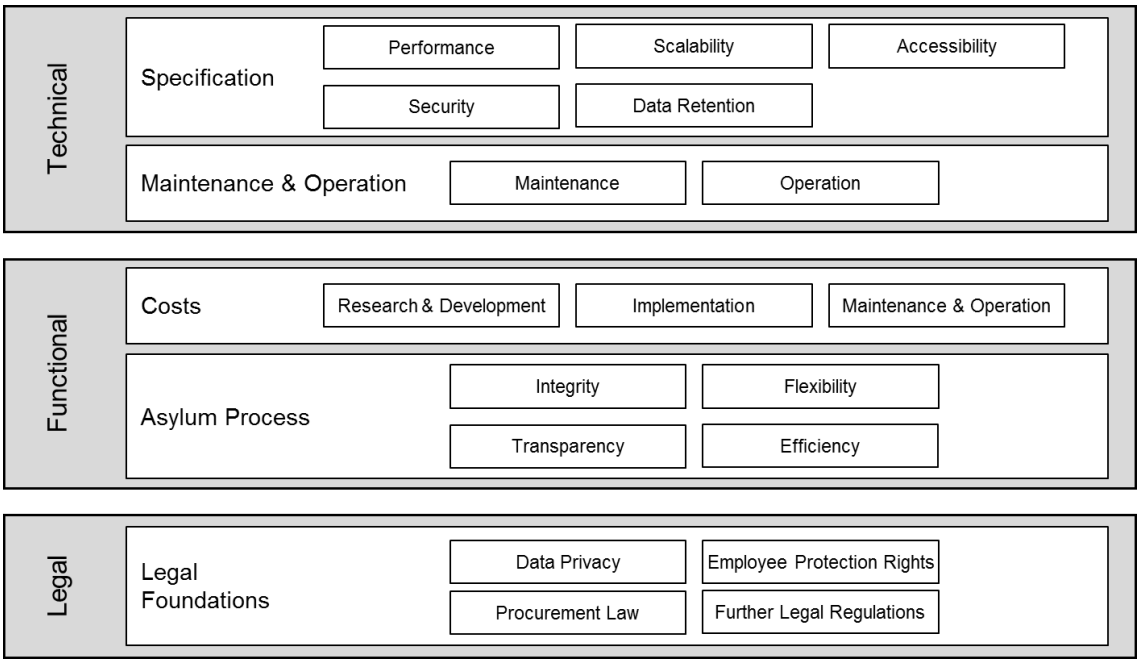


Figure 2. Evaluation Framework (Stage 3)

organizations must consider legal frameworks and statutes. To derive such evaluation criteria, we followed an ADR approach and a pragmatist paradigm, meaning that we co-developed our criteria with asylum process experts and stakeholders. To ground our evaluation framework, we followed the guidelines of Webster and Watson (2002) [49] and first conducted a systematic literature review [39]. To increase the reliability of this review, we did a structured database search. Our all fields search of the search terms blockchain AND (criteri* OR evaluat*) in the AIS Electronic Library (AISeL) yielded 51 hits. Further in-depth screening reduced this number to 14. Solely screening abstracts was not sufficient, however, as none of the papers in the AISeL embraced the aforementioned combination of search terms within their title, abstract, or keywords. With a forward search [49], we additionally identified five papers. Table 1 presents an overview of the papers we used to develop the first draft of the ex-ante framework.

As a parallel initial step, we followed the blockchain use case development (BUD) method of Fridgen et al. (2018) to derive a suitable BUC [18]. The BUD method stipulates that organizations follow six steps, from ideation methods to the conceptual phase before prototyping begins, to generate BUCs. Organizations should perform these steps within one-day or two-day workshops. After the first step, we developed an initial ex-ante framework. We frequently challenged our BUC evaluation criteria according to our ADR approach [46]. ADR consists of several iteration loops – mainly the α- and β-Cycle. The α-Cycle serves to develop a robust ex-ante framework while simultaneously integrating user feedback. In the α-Cycle of the ADR approach, we enhanced and validated our findings through semi-structured interviews [34]. The β-Cycle serves to validate the ex-ante framework. Hevner et al. (2004) recommend that researchers follow design science approaches to derive insights that allow generalization of their work [25]. Sein et al. (2011) extend this recommendation to the ADR approach by introducing a so-called β-Cycle that tests and improves the results of the α-Cycle using several novel sources of evidence [46]. Consequently, we added a β-Cycle consisting of two separate loops for which we conducted additional interviews, held further workshops and added participant observation [52]. The workshops helped us to understand the nature of BUCs in the asylum process better. We aligned those insights by pragmatically applying them within the project (i.e., we added participant observation). Thereby, we validated the ex-ante framework a first time. As we found new criteria in this first loop, we conducted a second β-Cycle consisting of additional workshops. These workshops verified the framework from the first loop as they confirmed all criteria and only suggested marginal adjustments.

FINDINGS: EX-ANTE CRITERIA, EVALUATION & EX-POST CRITERIA

As indicated in the previous sections, we developed our evaluation criteria in three stages.

Stage 1: In a first step, we selected a preliminary set of blockchain evaluation criteria from prior scientific (e.g., [13, 45]) and practical literature (e.g., [32]). This preliminary set already included three levels (domain, subdomain, and category – see Figure 1). At the highest level, we differentiated between the three domains "technical", "functional", and "legal". We divided the technical domain into three subdomains (quality, maintenance & operation, and costs). On the third level, the subdomain "quality" had

six categories (performance, interoperability, scalability, reliability, security, and portability). The quality subdomain included essential technical design aspects: IT security (reliability and security), transaction duration (performance), and the interaction of the blockchain solution with existing systems (interoperability and portability). Importantly, it also considered how a blockchain solution would perform if extended from a small prototype to a large-scale operational system (scalability). We did not divide the maintenance & operation subdomain into smaller categories. It considered whether 'non-specialized' employees could maintain and operate the blockchain system. We further divided the subdomain costs into three categories (research and development, implementation, maintenance & operation). We split the functional domain into the three categories "integrity", "output", and "performance". We did not subdivide the domain legal and only included a category "legal foundation(s)". It summarized all legal framework conditions that affect the feasibility of the blockchain.

Stage 2: After deriving our ex-ante set of evaluation criteria, we discussed our framework with experts and stakeholders in the Federal Office for Migration and Refugees. In particular, we used interviews and hosted interactive workshops to gather feedback from all relevant stakeholders (technical, functional, and legal). Stage 2 resulted in several changes to our framework (see figure 2). While the three domains (technical, functional, and legal) remained unchanged, we reduced the number of technical subdomains to two. Additionally, we shifted the subdomain costs to the functional domain. The costs of implementation strongly depend on the pre-existing infrastructure and therefore explicitly belong to the specific BUC. Also, the number and complexity of the blockchain applications that organizations need to develop strongly relate to the particular BUC. Given the changes in the subdomain "quality", we decided to rename it "specification". On the third level, the subdomain "specification" then included only four categories namely "performance", "scalability", "security", and "data retention" (new category). We also included "reliability" in security. Finally, we shifted "interoperability" and "portability" to the functional domain. Furthermore, we divided the subdomain "maintenance & operation" into two categories "maintenance" and "operation". As already mentioned, we shifted the subdomain "costs" to the functional domain. Therefore, the functional domain then included two subdomains (costs and asylum process). All cost categories remained unchanged, but we defined changes to the asylum process subdomain. The category "integrity" remained unchanged, but we renamed "performance" into "efficiency". Furthermore, we added two new categories ("flexibility" and "transparency"). These categories are important to evaluate whether a blockchain can serve different instances of the asylum process and whether it is possible to track the current process status. Finally, we divided the subdomain "legal foundation" respectively the legal domain into three categories (data privacy, employee protection rights, and further legal regulations).

Stage 3: After the first round of evaluations (stage 2), we held another interactive workshop with BAMF stakeholders from various departments. This workshop resulted only in minor adjustments and additions to the framework (see figure 3). We added the category "accessibility" to the subdomain specification. Accessibility is an essential feature in the public sector and guarantees that hearing and visually impaired persons can use information and IT system. Another essential requirement for software procurement in the public sector is the observance of competitive tenders. Therefore, we added the category "procurement law" to the legal domain. The functional domain remained unchanged.

DISCUSSION

Theoretical Contribution

This paper makes three theoretical contributions. First, we present insights from developing a framework to evaluate the applicability of blockchain along the German asylum process. Using semi-structured interviews, interactive workshops, and participant observation, we developed our framework in an evolutionary process. The final framework considers three primary domains, namely technical, functional, and legal. While the technical domain covers general technical aspects, the functional and legal domain relate to the investigated use case (i.e., asylum process). The final framework divides these domains into five subdomains that again group into 18 categories. Second, this paper provides a structured overview of BUC evaluation criteria. Although these criteria do not yet allow assessing BUCs in the true sense, they present a solid basis for the development of a key performance indicator system. Third, we enhance knowledge at the cutting edge of blockchain, prototype evaluation, and e-government (i.e., digitalization of the public sector) as well as refugee politics. Prior work provides helpful insights into how to define BUCs, into how to implement blockchain prototypes, or how to introduce and operate blockchain solutions in private and less in public sector. However, to the best of our knowledge, there is no work on how to evaluate the benefit of future blockchain solutions in a structured way. Therefore, this framework creates a new value in this field of research.

Limitations and Future Research

Naturally, our framework has its limitations. Importantly, we only identified categories of evaluation criteria. For a rating of these criteria, however, future research must specify these criteria in more detail. Alternatively, an extended framework would have to include defined key figures. We are currently working on this step and are identifying key figures, such as the number of (active) users or the bandwidth of the network, and their effects on the categories. The second limitation is that our present framework weighs each domain, subdomain, and category equally. In reality, however, some factors

outweigh others, and especially legal requirements present knockout criteria. Moreover, public sector organizations generally do not seek to maximize profit (e.g., by reducing staff) but to maintain jobs or create new ones. Therefore, public sector adopters must consider and weigh highly social aspects that we included in the employee protection rights category. For further research, we plan to extend our framework with weights for each category, subdomain, and domain as provided by experts. Ultimately, we also only investigated a single use case. To validate and generalize our framework, future research must examine additional BUCs. Exemplary, we recommend studying inter-organizational processes in integrating new citizens (i.e., the processes following a completed asylum and naturalization process).

Conclusion

From our evaluation, we conclude that technical, functional, and legal aspects play an equally important role. Overall, this paper is a first step in developing a general framework for the evaluation of blockchain uses cases. This preliminary version supports decision makers in the public sector and offers essential managerial implications.

Acknowledgment

We developed this work in the context of a joint project with the German Federal Office for Migration and Refugees (BAMF). The authors would like to thank everyone involved for their support.

Moreover, we developed this work (in part) in the context of the Project Group Business and Information Systems Engineering of the Fraunhofer Institute for Applied Information Technology FIT.

REFERENCES

[1] Svetlana Abramova and Rainer Böhme. 2016. Perceived benefit and risk as multidimensional determinants of bitcoin use. A quantitative exploratory study. *Thirty Seventh International Conference on Information Systems (ICIS)*.

[2] Jacky Akoka and Isabelle Comyn-Wattiau. 2017. A Method for Emerging Technology Evaluation. Application to Blockchain and Smart Data Discovery. In *Conceptual Modeling Perspectives*, Jordi Cabot, Cristina Gómez, Oscar Pastor, Maria R. Sancho and Ernest Teniente, Eds. Springer International Publishing, Cham, 247–258. DOI: https://doi.org/10.1007/978-3-319-67271-7_17.

[3] Marcella Atzori. 2015. *Blockchain Technology and Decentralized Governance: Is the State Still Necessary?* (2015). Retrieved September 1, 2017 from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2709713.

[4] Roman Beck and Christoph Müller-Bloch. 2017. Blockchain as Radical Innovation: A Framework for Engaging with Distributed Ledgers as Incumbent Organization. In *Proceedings of the 50th Hawaii International Conference on System Sciences*.

[5] Roman Beck, Jacob Stenum Czepluch, Nicolaj Lollike, and Simon Malone. 2016. Blockchain - The Gateway to Trust - free Cryptographic Transactions. In *Proceedings of the 24th European Conference on Information Systems*, 1–14.

[6] Jörg Becker, Dominic Breuker, Tobias Heide, Justus Holler, Hans P. Rauer, and Rainer Böhme. 2013. Can We Afford Integrity by Proof-of-Work? Scenarios Inspired by the Bitcoin Currency. In *The Economics of Information Security and Privacy*. Springer, Berlin Heidelberg, 135–156.

[7] Anandhi Bharadwaj, Omar A. El Sawy, Paul A. Pavlou, and N. Venkatraman. 2013. Digital Business Strategy. Toward a Next Generation of Insights. *MIS Quarterly* 37, 2, 471–482.

[8] Christian Brenig, Jonas Schwarz, and Nadine Rückeshäuser. 2016. Value of Decentralized consensus Systems-Evaluation Framework. *Twenty-Fourth European Conference on Information Systems (ECIS)*.

[9] Konstantinos Christidis and Michael Devetsikiotis. 2016. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access* 4, 2292–2303.

[10] Suparna D. D'Cunha. 2017. *Dubai Sets Its Sights On Becoming The World's First Blockchain-Powered Government* (2017). Retrieved March 16, 2018 from https://www.forbes.com/sites/suparnadutt/2017/12/18/dubai-sets-sights-on-becoming-the-worlds-first-blockchain-powered-government/.

[11] Christian Decker and Roger Wattenhofer. 2013. Information Propagation in the Bitcoin Network. In *Proceedings of the 13th IEEE International Conference on Peer-to-Peer Computing*, 1–10.

[12] Ali Dorri, S. Kanhere, Raja Jurdak, and Praveen Gauravaram, Eds. 2017. *Blockchain for IoT Security and Privacy: The Case Study of a Smart Home*.

[13] Shayan Eskandari, David Barrera, Elizabeth Stobert, and Jeremy Clark. 2015. A First Look at the Usability of Bitcoin Key Management. In *Proceedings 2015 Workshop on Usable Security*. Internet Society, Reston, VA. DOI: https://doi.org/10.14722/usec.2015.23015.

[14] Kurt Fanning and David P. Centers. 2016. Blockchain and Its Coming Impact on Financial Services. *J. Corp. Acct. Fin* 27, 5, 53–57. DOI: https://doi.org/10.1002/jcaf.22179.

[15] Pasquale Forte, Diego Romano, and Giovanni Schmid. 2015. Beyond Bitcoin – Part I: A critical look at blockchain-based systems. *Cryptology ePrint Archive*.

[16] Frankfurter Allgemeine Zeitung GmbH. *Supermarkt in Jordanien: Wo Flüchtlinge mit einem Augenblick bezahlen*. Retrieved March 14, 2018 from http://www.faz.net/aktuell/finanzen/digital-bezahlen/

jordanien-iris-scan-und-blockchain-bei-fluechtlingen-15306863.html.

[17] Gilbert Fridgen, Florian Guggenmos, Jannik Lockl, and Alexander Rieger. 2018. Challenges and Opportunities of Blockchain-based Platformization of Digital Identities in the Public Sector. Research in Progress. *ECIS2018 Workshop on Platformization*, 1–10.

[18] Gilbert Fridgen, Jannik Lockl, Sven Radszuwill, Alexander Rieger, André Schweizer, and Nils Urbach. 2018. A Solution in Search of a Problem: A Method for the Development of Blockchain Use Cases. *Working Paper*, 1–10.

[19] Gilbert Fridgen, Sven Radszuwill, André Schweizer, and Nils Urbach. 2018. Blockchain Won't Kill the Banks: Why Disintermediation doesn't Work in International Trade. *Working Paper*, 1–13.

[20] Gilbert Fridgen, Sven Radszuwill, Nils Urbach, and Lena Utz. 2018. Cross-Organizational Workflow Management Using Blockchain Technology - Towards Applicability, Auditability, and Automation. In *Proceedings of the 51th Hawaii International Conference on System Sciences*.

[21] H. Gimpel and M. Röglinger. 2015. *Digital Transformation: Changes and Chances*. *Insights Based on an Empirical Study* (2015). Retrieved May 5, 2016 from http://www.digital.fim-rc.de/.

[22] Florian Glaser. 2017. Pervasive Decentralisation of Digital Infrastructures: A Framework for Blockchain enabled System and Use Case Analysis. In *Proceedings of the 50th Hawaii International Conference on System Sciences*, 1543–1552.

[23] Florian Glaser and Luis Bezzenberger. 2015. Beyond Cryptocurrencies - A Taxonomy of Decentralized Consensus Systems. In *Proceedings of the 23rd European Conference on Information Systems*, Münster, Germany, 1–18.

[24] Yvonne Göpfert. 2018. *Flüchtlingshilfe via Blockchain* (2018). Retrieved March 14, 2018 from https://www.lead-digital.de/fluechtlingshilfe-via-blockchain/.

[25] Alan. R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. 2004. Design Science in Information Systems Research. *MIS Quarterly* 28, 1, 75–105.

[26] Hissu Hyvärinen, Marten Risius, and Gustav Friis. 2017. A Blockchain-Based Approach Towards Overcoming Financial Fraud in Public Sector Services. *Bus Inf Syst Eng* 59, 6, 441–456. DOI: https://doi.org/10.1007/s12599-017-0502-4.

[27] Janze. 2017. Design of a Dezentralized Peer-to-Peer Reviewing and Publishing Market. *Proceedings of the 25th European Conference on Information Systems (ICIS)*, 1713–1725.

[28] Kari Korpela, Jukka Hallikas, and Tomi Dahlberg. 2017. Digital Supply Chain Transformation toward Blockchain Integration. In *Proceedings of the 50th Hawaii International Conference on System Sciences*, 4182–4191.

[29] Jannik Lockl, Alexander Rieger, Gilbert Fridgen, Maximilian Röglinger, and Nils Urbach. 2018. Towards a Theory of Decentral Digital Process Ecosystems - Evidence from the Case of Digital Identities. Research in Progress. *ECIS2018 Workshop on Platformization*, 1–2.

[30] Juri Mattila. 2016. *The Blockchain Phenomenon – The Disruptive Potential of Distributed Consensus Architectures*. The Research Institute of the Finnish Economy.

[31] Sascha Mattke. 2017. *Blockchain für Flüchtlinge: Digitale Identität mit Prepaid-Kreditkarte für Asylsuchende in Finnland* (September 2017). Retrieved March 14, 2018 from https://www.heise.de/newsticker/meldung/Blockchain-fuer-Fluechtlinge-Digitale-Identitaet-mit-Prepaid-Kreditkarte-fuer-Asylsuchende-in-3823031.html.

[32] Minestry of Economy, Trade and Industrie. 2017. *Evaluation Forms for Blockchain-Based System* (2017). Retrieved March 14, 2018 from http://www.meti.go.jp/english/press/2017/pdf/0329_004a.pdf.

[33] Eric Munsing, Jonathan Mather, and Scott Moura. 2017. *Blockchains for Decentralized Optimization of Energy Resources in Microgrid Networks* (2017). Retrieved September 1, 2017 from http://escholarship.org/uc/item/80g5s6df.

[34] Michael D. Myers and Michael Newman. 2007. The Qualitative Interview in IS Research. Examining the Craft. *Information and Organization* 17, 1, 2–26. DOI: https://doi.org/10.1016/j.infoandorg.2006.11.001.

[35] Kristoffer Nærland, Christoph Müller-Bloch, Roman Beck, and Søren Palmund. 2017. Blockchain to Rule the Waves - Nascent Design Principles for Reducing Risk and Uncertainty in Decentralized Environments. In *Proceedings of the 38th International Conference on Information Systems*.

[36] Satoshi Nakamoto. 2008. Bitcoin. A peer-to-peer electronic cash system.

[37] Fred Niederman, Roger Clarke, Lynda M. Applegate, John L. King, and Roman Beck. 2017. IS Research and Policy: Notes From the 2015 ICIS Senior Scholar's Forum. *Communications of the Association for Information* 40, 1, Article 5.

[38] Benedikt Notheisen, Jacob B. Cholewa, and Arun P. Shanmugam. 2017. Trading Real-World Assets on Blockchain. *Bus Inf Syst Eng* 59, 6, 425–440. DOI: https://doi.org/10.1007/s12599-017-0499-8.

[39] Chitu Okoli and Kira Schabram. 2010. A guide to conducting a systematic literature review of information systems research.

[40] Gareth W. Peters and Efstathios Panayi. 2016. Understanding modern banking ledgers through blockchain technologies. Future of transaction

processing and smart contracts on the internet of money. In *Banking Beyond Banks and Money*. Springer, 239–278.

[41] Marc Pilkington, Rodica Crudu, and Lee G. Grant. 2017. Blockchain and bitcoin as a way to lift a country out of poverty - tourism 2.0 and e-governance in the Republic of Moldova. *IJITST* 7, 2, 115. DOI: https://doi.org/10.1504/IJITST.2017.087132.

[42] Simone Porru, Andrea Pinna, Michele Marchesi, and Roberto Tonelli, Eds. 2017. *Blockchain-oriented software engineering*. *Challenges and new directions*. IEEE Press.

[43] Wolfgang Prinz, Wolfgang Graetner, and Sandra Klein. Use Case Identification Framework and Use Case Canvas for identifying and exploring relevant Blockchain opportunities. In *Proceedings of the 1st ERCIM Blockchain Workshop 2018*, Wolfgang Prinz and P. Hoschka, Eds. Reports of the European Society for Socially Embedded Technologies, Amsterdam. DOI: https://doi.org/10.18420/blockchain2018_02.

[44] Marten Risius and Kai Spohrer. 2017. A Blockchain Research Framework. *Business & Information Systems Engineering* 59, 6, 385–409.

[45] André Schweizer, Vincent Schlatt, Nils Urbach, and Gilbert Fridgen. 2017. Unchaining Social Businesses - Blockchain as the Basic Technology of a Crowdlending Platform. In *Proceedings of the 38th International Conference on Information Systems*.

[46] Maung K. Sein, Ola Henfridsson, Sandeep Purao, Matti Rossi, and Rikard Lindgren. 2011. Action Design Research. *MIS Quarterly* 35, 1, 37–56.

[47] Kane Smith and Gurpreet Dhillon. 2017. Blockchain for Digital Crime Prevention. The Case of Health Informatics.

[48] Nick Szabo. 1997. Formalizing and Securing Relationships on Public Networks. *First Monday* 2, 9.

[49] Jane Webster and Richard Watson. 2002. Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly* 26. DOI: https://doi.org/10.2307/4132319.

[50] Christian Welzel, Klaus-Peter Eckert, Fabian Kirstein, and Volker Jacumeit. 2017. *Mythos Blockchain*. *Herausforderung für den öffentlichen Sektor* (2017). Retrieved March 16, 2018 from https://www.oeffentliche-it.de/documents/10181/14412/Mythos+Blockchain+-+Herausforderung+f%C3%BCr+den+%C3%96ffentlichen+Sektor.

[51] Aaron Wright and Primavera de Filippi. 2015. *Decentralized Blockchain Technology and the Rise of Lex Cryptographia* (2015). Retrieved August 1, 2017 from http://ssrn.com/abstract=2580664.

[52] Robert K. Yin. 2017. *Case study research and applications*. *Design and methods*. Sage publications.

[53] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, and Huaimin Wang. 2016. *Blockchain Challenges and Opportunities: A Survey* (2016). Retrieved from.

Blockchain for Education: Lifelong Learning Passport

Wolfgang Gräther
Fraunhofer FIT
Sankt Augustin, Germany
graether@fit.fraunhofer.de

Sabine Kolvenbach
Fraunhofer FIT
Sankt Augustin, Germany
kolvenbach@fit.fraunhofer.de

Rudolf Ruland
Fraunhofer FIT
Sankt Augustin, Germany
rudolf.ruland@fit.fraunhofer.de

Julian Schütte
Fraunhofer AISEC
Garching, Germany
schuette@aisec.fraunhofer.de

Christof Ferreira Torres
University of Luxembourg
Luxembourg
christof.torres@uni.lu

Florian Wendland
Fraunhofer AISEC
Garching, Germany
wendland@aisec.fraunhofer.de

ABSTRACT
Certificates play an important role in education and in professional development in companies. Individual learning records become essential for people's professional careers. It is therefore important that these records are stored in long-term available and tamper-proof ledgers. A blockchain records transactions in a verifiable and permanent way, therefore it is very suitable to store fingerprints of certificates or other educational items. Blockchain reveals forgery of certificates and it supports learning histories. In this paper, we present the Blockchain for Education platform as a practical solution for issuing, validating and sharing of certificates. At first, we describe the conceptual system overview and then we present in detail the platform implementation including management of certification authorities and certificates, smart contracts as well as services for certifiers, learners and third parties such as employers. Finally, we describe use cases and first evaluation results that we gathered from end user tests with certifiers and conclude with a discussion.

Author Keywords
Certification, Certificate, Blockchain, Smart Contract, IPFS, BSCW.

ACM Classification Keywords
H.5 Information Systems

INTRODUCTION
In education, certificates confirm the achievement of certain learning outcomes and are until today mostly issued on paper or other physical formats. For example, a learner has participated in an enterprise-training course on usability engineering. After the successful completion of the course, the learner receives the certificate as a paper document that entitles him to use the protected title "Usability engineer with level A". Universities and educational institutions that award degrees to their students also issue many certificates.

Gräther, Wolfgang; Kolvenbach, Sabine; Ruland, Rudolf; Schütte, Julian; Torres, Christof; Wendland, Florian (2018): Blockchain for Education: Lifelong Learning Passport. In: W. Prinz & P. Hoschka (Eds.), Proceedings of the 1st ERCIM Blockchain Workshop 2018, Reports of the European Society for Socially Embedded Technologies (ISSN 2510-2591), DOI: 10.18420/blockchain2018_07

Certificates include several statements. The most important are: the kind of qualification or academic title that is attested, the name and address of the issuer organization, the name and signature of the certifier who has validated the facts and is certifying that the qualification is true, the name of the learner and a date of the examination. Depending on the type of certificate, there can be additional statements about the examination regulations, the period of validity or further information necessary to make use of the certificate.

Learners usually receive a paper document that presents the certificate. Using paper certificates has the advantage of being difficult to forge due to built-in security features. In addition, recipients can easily store paper certificates or can show them to any person and for any purpose. However, there are some disadvantages such as the mainly manual activity for third parties to verify the certificate or the need for certification authorities to maintain a registry or database for certificates for a long period of time [1].

An alternative to paper certificates are digital certificates that are cryptographically signed (in the following, we will use the simpler term certificate). Compared to paper certificates, management and use of digital certificates is simplified. However, more effort is needed to secure the registry for certificates and an open standard for digital signatures has to be used, otherwise the global verification of digital certificates is not possible.

In particular, in the area of educational certificates, there exists the severe problem of fake degrees. Ezell and Bear report about fraudulent practices and the billion-dollar industry behind it [2]. Hence, blockchain technology seems ideal to solve many of the above-mentioned problems of current paper or digital certificates and fake degrees.

In the context of education and certification, the blockchain technology supports counterfeit protection of certificates, easy verification of certificates even if the certification authority no longer exists and automation of monitoring processes for certificates with a time-limited validity. When we look at certification processes from a blockchain perspective, we identify three main tasks. Firstly, identities of certification authorities have to be created and maintained.

Secondly, these certification authorities have to issue certificates to learners and the third main task is the verification of certificates by employers, for example. These three tasks have to be supported adequately by a blockchain-based infrastructure including the sharing of certificates by learners.

The *Blockchain for Education* platform aims to support counterfeit protection as well as secure access and secure management of certificates according to the needs of learners, companies, education institutions and certification authorities. In the next section, we present related work. We then provide an overview of the system including a description of the minimal viable product and the conceptual system architecture. The section prototype implementation explains in detail the management of identities for certification authorities and certifiers as well as the management of certificates represented as extended Open Badges[1]. The description of application portfolios and the verification service complete this section. Use cases and first evaluation results are presented in the next section. Discussion and conclusion sections close this paper.

RELATED WORK

The University of Nicosia was the first higher education institution that stored academic certificates on the Bitcoin blockchain [3,4].

The MIT Media Lab Learning Initiative together with Learning Machine, an enterprise software vendor, has developed Blockcerts, an open-source ecosystem for creating, sharing, and verifying blockchain-based educational certificates. The educational certificates contain basic information such as the name of the recipient, the name of the issuer, an issue date, etc. Note that interoperability with Open Badges assertions is given. Educational certificates are registered on the Bitcoin blockchain, cryptographically signed, and tamper-proof. Blockcerts makes it possible to verify who a certificate was issued to, by whom, and to validate the content of the certificate itself [5,6].

Based on Blockcerts, a pilot for academic and professional certifications will be developed in Malta [7] and the Federation of State Medical Boards in the US is currently launching a pilot for the issuing of official documents with Blockcerts to the blockchain [8].

In July 2017 the company SAP introduced TrueRec a secure and trusted digital wallet for storing professional and academic credentials based on Ethereum. TrueRec was made available to people enrolled in the online course *Touch IoT course for SAP Leonardo*. Over 4500 students will receive and can manage their certificate through TrueRec [9].

TNO, Netherlands Organisation for Applied Scientific Research, started recently the blockchain project *self sovereign identity framework*. This framework is designed to help supply official information in digital form and only share a minimum amount of personal data that is managed and stored in encrypted form in a wallet on one's own cellphone. This information provides official confirmation about the identity of the person [10,11]. Sovrin is another infrastructure that aims to support digital identities on a global scale [12].

Similar to certification is notarization where ownership, existence and integrity of documents is important. The Apostille notarization service and use cases such as car ownership or digital media licenses are described by McDonald and Oliverio [13].

Work related to the design and development of smart contracts in the Blockchain for Education platform is concerned with the correctness of security-relevant Ethereum contracts. Blockchain for Education uses approved smart contract templates of the OpenZeppelin[2] project and extensions of existing code analysis tools like Oyente [14] and Mythril [15].

SYSTEM OVERVIEW

Our system mainly supports certification authorities, learners and employers. It ensures higher efficiency and improved security for certification authorities through digitization of current processes, issuing and registering of certificates in a blockchain as well as automatic monitoring of certificates. To follow the Industry 4.0 approach [16] the platform supports machine-readable certificates. Learners are enabled to manage their certificates and to give access to selected certificates to third parties, i.e. the protection of privacy for leaners is ensured. Trustworthy verification of certificates is offered for employers.

Minimal Viable Product

Several workshops and meetings with our application partners, educational institutions and two personnel certification authorities have been performed to elicit the requirements for our system and to derive the minimal viable product. Starting from the requirements, we conceptualized for each user group features for the minimal viable product.

Features for Certification Authorities

Currently, certification authorities manage data of learners, learning courses and other relevant regulations as well as examination results in their own databases or even MS Excel sheets. This data is used to issue paper certificates for learners. Therefore, the import of data and examination results from legacy systems is a first important feature for certification authorities. After importing the data, certification authorities can browse the generated certificates. In addition, certification authorities need means to search for learners or to gain an overview of learners and their examination results according to learning courses. The overview enables certification authorities to print all

[1] https://openbadges.org/

[2] https://github.com/OpenZeppelin/zeppelin-solidity

certificates of a learning course at once. The second main feature for certification authorities is signing of certificates and storing them into the blockchain. Both actions are easily carried out simultaneously for all the learners in the previously mentioned overview.

Acknowledgement of validity and authenticity of a certain certificate is a further feature for all user groups of the minimal viable product. Furthermore, certification authorities need a means to revoke certificates. This could be necessary when plagiarism has been detected or misconduct of the certified learner was proven. Usually revocation occurs mainly for certificates with time-limited validity when the necessary actions have not been taken.

Features for Learners

At present, learners mostly receive paper certificates with built-in security features. Learners send or email copies or certified copies, sometimes digitized (scanned), to prospective employers. Hence, the importing of certificates and creation of an application portfolio is a major feature for the minimal viable product. Furthermore, learners need means to manage application portfolios as well as means for sharing them.

After sharing application portfolios learners are supported with information about employer's activities on their certificates such as reading or verifying, i.e. notifications for learners are a further feature of the minimal viable product. Similarly, monitoring of certificates with a time-limited validity support learners and is an additional feature of the minimal viable product. Note, that monitoring is also relevant for certification authorities but implies different actions. Certification authorities could remind learners and revoke certificates, if the conditions for renewal are not met.

Features for Employers

Currently, employers only receive copies, sometimes notarized copies, of the learner's paper certificates. In the first case, employers can proof the validity of the copies only by asking the issuing organization for the authenticity and validity of the certificate. This is a time-consuming and expensive process. Therefore, reading and verifying certificates is an important feature of the minimal viable product.

The derived features for certification authorities, learners and employers have been exploited to develop the conceptual system architecture. This process was supported by the use case canvas for blockchain described in [17] and the engineering framework presented in [18].

Conceptual System Architecture

An overview of the prototype architecture is shown in Figure 1. It comprises the blockchain including smart contracts, a public storage holding profile information of certification authorities, a document management system managing the actual payload of certificates tracked by the blockchain and the parties involved in the system, namely accreditation and certification authorities, certifiers, learners and employers. In the prototype implementation, only the document management system is a centralized system component.

Bootstrapping the Platform

Initially, two smart contracts are submitted to the blockchain by the accreditation authority (1). The first smart contract (IdentityMgmt) supports management of identities in the Blockchain for Education platform and the second one (CertMgmt) manages the lifecycle of certificates issued over the blockchain. Once the contracts are deployed (2a), it is the accreditation authority's task to register the public keys of certification authorities as legitimate issuer of certifiers in the IdentityMgmt contract (3a) and to submit public and non-personal profile information to the public storage (2b). It is important to note that the profile information is read-only and publicly readable, i.e. it is not subject to the access control mechanisms of the IdentityMgmt contract. It merely holds long-time profiles of certification authorities, such as their name and country, but does not include any personal information of certifiers or even learners.

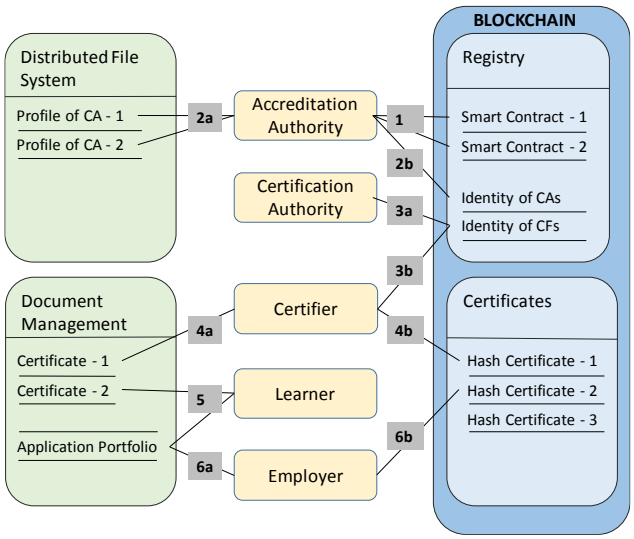


Figure 1: Conceptual Architecture

Registered certification authorities then add the public keys of certifiers to the registry of the IdentityMgmt contract (3b) and thereby delegate the right to issue certificates. That is, a holder of a private certifier key will typically be an employee of a certification authority who is entitled to issue certificates and signs them in the name of the certification authority.

Issuing Certificates
The certifier collects all information a certificate consists of. The dataset comprises qualification or title, name and address of the certification authority, name of the certifier, name of the learner, and the date. Then the certificate is signed by the certifier and stored on the document management system (4a) and its fingerprint is written to the blockchain (4b).

Creation and management of application portfolios
Learners are supported in the creation and management of application portfolios by a service of the document management system. Firstly, the learner has to register with the document management system. Then, a service for the flexible creation of application portfolios supports the learner (5). Completed application portfolios can be shared with potential employers who can verify the validity of these certificates.

Verifying Certificates
A service of the minimal viable product supports employers, for example, in verifying single certificates or all certificates of an application portfolio (6a, 6b).

PROTOTYPE IMPLEMENTATION

We implemented a prototype of the Blockchain for Education platform based on the Ethereum blockchain[3]. Two smart contracts written in Solidity[4] codify access control mechanisms (IdentityMgmt) and manage certificate records (CertMgmt) stored in the blockchain. The Interplanetary Filesystem[5] (IPFS) is used as a public distributed read-only storage for profile information of certification authorities. Finally, the BSCW document management system stores and validate certificates.

Identity Hierarchy and Rights Delegation

Identities in Blockchain for Education are managed in a hierarchy. On top is a set of accreditation authorities who are entitled to approve certification authorities. For instance, members of the European Co-operation for Accreditation could build the set of accreditation authorities in the Blockchain for Education platform. In Figure 2, we summarize the whole set of accreditation authorities to a single authority for the sake of simplicity and to reflect the current prototype setup. An accreditation authority is the owner of the smart contracts of an instance of the Blockchain for Education platform. It creates the initial smart contracts on the Ethereum blockchain. The IdentityMgmt contract allows accreditation authorities to create, update and remove certification authorities.

Certification authorities reside one level below accreditation authorities in the identity hierarchy. They are identified by their Ethereum address, which is derived from a cryptographic hash of their public keys. The address of a certification authority is mapped to its profile information stored on IPFS. Certification authorities cannot issue certificates themselves. They can only entitle employees by delegating the respective right to them. To do so, a certification authority calls the respective function of the IdentityMgmt contract and passes in the Ethereum address of its certifier. The smart contract ensures that only accredited certification authorities may delegate the right and automatically assigns the certifier to the delegating certification authority. Just as the right to issue certificates can be issued at any time to any certifier, it can also be revoked by the certification authority. This deauthorization could for example occur if a certifier leaves a certification authority or should otherwise loose the right to issue further certificates.

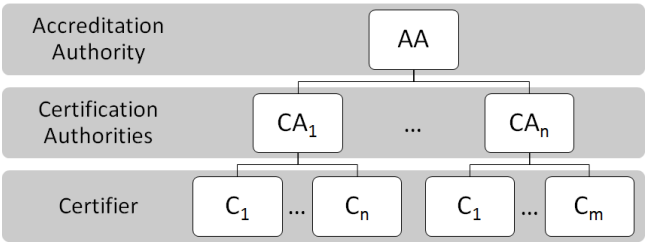


Figure 2: Identity hierarchy

Certifiers cannot delegate their rights further and cannot manipulate the access permissions. The role of certifiers is limited to the management of certificate records on the blockchain.

Certificate Management

Certifiers can create, revoke and delete references to certificates stored in the Blockchain for Education platform. This is implemented in the smart contract CertMgmt.

The accreditation authority instantiates the CertMgmt contract together with the IdentityMgmt contract. The CertMgmt contract requires the address of the IdentityMgmt contract to enforce access control. Any manipulative operation on the CertMgmt contract, such as adding a certificate, requires that the caller is a registered certifier of an accredited certification authority. Everyone can retrieve certificate records given the address of the CertMgmt contract and a hash of the certificate.

The CertMgmt contract uses certificate records to store certificate information in the blockchain. Currently, this information consists of the SHA256 hash of the certificate, the starting and expiration date and a status field (onHold) to indicate if a certificate is on hold. Dates are represented as UNIX timestamps and for future proofing, are stored as 256-bit unsigned integers. Similarly, the onHold status field stores a UNIX timestamp if a certificate is on hold. Thus, one can check when the onHold status was set for a certificate.

[3] https://ethereum.org/

[4] https://solidity.readthedocs.io/en/latest/index.html

[5] https://ipfs.io/

IPFS as a Public Tamper-Proof Read-Only Profile Storage

On the Ethereum blockchain, entities such as accreditation authority, certification authorities and certifiers are identified by their Ethereum addresses, i.e. a hash of their public keys. This provides anonymity and protects personal information, especially of the certifiers, as it is not easily possible to correlate an Ethereum address to a real person.

Certification authorities, however, must provide identifiable profile information to allow anybody who is verifying a certificate to verify the certification authority as well. Without this profile information, certifications would remain completely anonymous and consequently not suited to build a well-reputed track record for a learner. Therefore, every certification authority must provide an IPFS address where interested parties can look up the profile.

This is not only a requirement resulting from the European General Data Protection Regulation which objects any undeletable storage of personal information in a blockchain, but also an important feature for certification authorities who do not want to reveal personal information of their employees to competing authorities. In addition, actual storage on the blockchain is comparatively expensive. Therefore, profiles of certificate authorities are stored on the IPFS.

IPFS provides temper-proof, secure and distributed storage. The massively distributed block storage addresses entries by their hashes stored as a Merkle tree. The specifics of IPFS are abstracted away by numerous clients for different programing languages that let client programs access IPFS as any other block storage. Whenever an accreditation authority registers a new certification authority, it will first write the certification authority's profile information into IPFS and then submit the certification authority's public key and the IPFS address to the IdentityMgmt's registry. Afterwards, both the Ethereum transaction and the IPFS block with the profile information synchronized across all nodes in the network. It is thus the accreditation authority's responsibility to ensure that it does not register fake profiles and must validate profile information of certificate authorities before they are added to the blockchain.

The use of IPFS in the Blockchain for Education platform provides two advantages. First, no personal data is stored on the blockchain while providing proof of authenticity resulting from the immutable IPFS addresses. This allows the use of Blockchain for Education in fulfillment of data protection laws. For example, the European General Data Protection Regulation (GDPR) would in general object to any undeletable storage of personal information in a blockchain. Second, storing the profile information of certification authorities externally in an immutable way saves storage on the blockchain.

Certificates as Extended Open Badges

To digitize certificates we decided to represent certificates in JSON data format, compatible to Open Badges. According to the requirements of our application partners and our personnel certification authority, we extended the standard Open Badges schema by six additions. These are: unique id of the certificate, examination date and place, examination regulations in force, data about the certifier, data about the certificate recipient and the address of the trusted service that is offered to verify the certificate.

Unique Certification ID

This schema extension adds the property *assertionreference* of type *string* to our schema. Our personnel certification authority required this unique ID for legal reasons.

Examination Date and Place

Figure 3 shows the schema extension. The properties *startdate*, *enddate*, and *place* all of type *string* have been defined. The dates are formatted according to the ISO 8601 date definition. This schema extension is a prime example for all our other extensions.

```
{
"$schema": "http://json-schema.org/draft-06/schema#",
  "title": "Information on the examination date and place",
  "description": "This extension provides additional information on the
examination date and place.",
  "type": "object",
  "definitions": {
    "ISO8601Date": {
      "description": "ISO 8601 date format string. For example, 2016-12-
31T23:59:59+00:00 is a valid ISO 8601 timestamp.",
      "type": "string",
      "format": "date-time"
    }
  },
  "properties": {
    "startdate": {"$ref": "#/definitions/ISO8601Date"},
    "enddate": {"$ref": "#/definitions/ISO8601Date"},
    "place": {"type": "string"}
  },
  "required": ["startdate"]
}
```

Figure 3: Schema Extension for Examination

Examination Regulations in Force

This schema extension mainly adds the properties *title*, *url*, *regulationsid*, and *date* of the regulation to our schema.

Certifier

The properties *givenname*, *surname*, *certificationdate*, *certificationplace* and blockchain *address* are defined and added to our schema.

Certificate Holder

This schema extension adds the properties *givenname*, *surname*, *birthdate*, *birthplace*, and *email* to our schema.

Verify

Figure **4** shows the schema extension in detail. The properties *verifyaddress* and *assertionhash* have been defined. These properties allow third parties to implement their own verification service.

We used the validator service of the IMS Global Learning Consortium to verify our extended Open Badges certificates: Our certificates are valid in compliance with Open Badges 2.0.

```
{
"$schema": "http://json-schema.org/draft-06/schema#",
  "title": "Verify Certificate in           BlockchainForEducation",
  "description": "This extension provides the URL to the
BlockChainForEducation Verify SmartContract to verify a certificate.",
  "type": "object",
  "definitions": {
    "HashString": {
      "type": "string",
      "description": "Open Badges SHA-256 Hash",
      "pattern": "^sha256\\$[a-fA-F0-9]{64}$"
    }
  },
  "properties": {
    "verifyaddress": {
      "type": "string",
      "format": "uri"
    },
    "assertionhash": {"$ref": "#/definitions/HashString"}
  },
  "required": ["verifyaddress","assertionhash"]
}
```

Figure 4: Schema Extension for Verification

Managing Certificates in BSCW

BSCW is a Web-based groupware system [9] that is used in the context of the blockchain for Education project to store learning courses, data about participants and examination results. The import of data from legacy systems is supported by a service specially implemented for the blockchain for education project. Certification authorities import their data for a specific learning course, which results in a folder that contains generated certificates. A screenshot is shown in Figure 5.

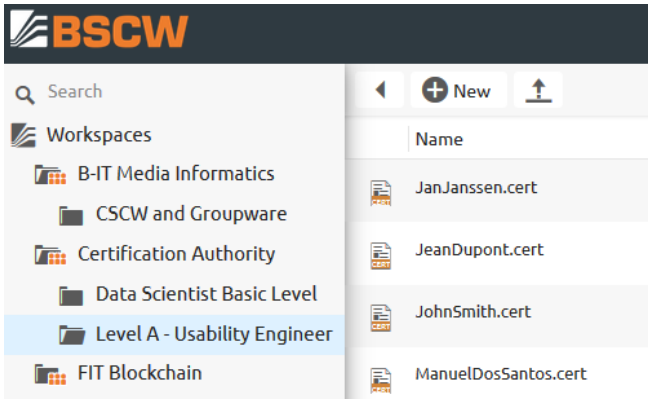


Figure 5: Personnel Certification Authority and List of Certificates

On the left hand side, Figure 5 shows a folder hierarchy. The folder named *Certification Authority* contains two folders for learning courses that contain for each learner, who has successfully finished the examination, the certificate in our extended Open Badge format. The folder *Level A - Usability Engineer* has been selected. Therefore, the respective certificates are presented on the right hand side of Figure 5. If certifiers click on a certificate, then a preview of the certificate is shown.

Send to Blockchain

After importing and previewing the certificates, the certifier can sign the certificates and write the certificates to the blockchain. This activity is carried out by the operation *send to blockchain*. Figure 6 shows this operation for the certificate *JanJanssen.cert*. If the operation could be executed, then the certifier is notified by a popup note that the certificate has been successfully written to the blockchain. Note, that only the fingerprint of the certificate and a few additional attributes are stored in the blockchain. The attribute *status* has by default the value valid, but it could be set to *on hold* or *invalid*. The attribute issuer contains the identity of the issuer of the certificate. A further attribute is the *issue date*. By default, the validity of certificates is not limited. In case of time-limited certificates, the attribute *expiration date* is set accordingly.

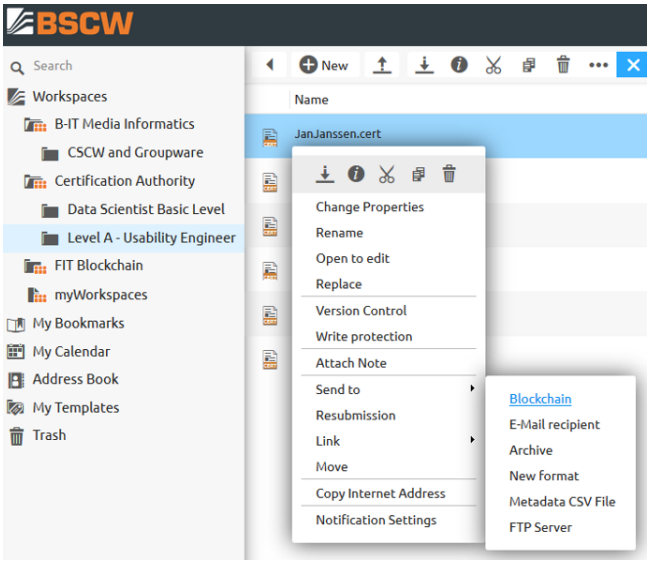


Figure 6: Write Fingerprint of Certificate to Blockchain

Sharing Certificates with Learners

After sending to the blockchain, the certifiers send learners their certificates in two formats: firstly, as encoded JSON file and secondly, as PDF document. Learners should then store the files safely in their personal archives. Note, that the PDF document contains as meta data the serialised JSON string of the certificate.

Application Portfolios in BSCW

Learners can self-register with BSCW, import their certificates into their personal folder and create different application portfolios adjusted to the respective employers. Application portfolios are mapped to folders in BSCW and structured in a two level hierarchy. The single folders contain the certificates in PDF format. The learner can share the application folder with a potential employer and the employer can verify the received certificates.

Verification of Certificates

For the trusted verification of certificates at hand, we have realized our own verification service. It is a free service offered on the landing page of the Blockchain for Education platform.

Users just drag and drop certificates, JSON or PDF documents are accepted, onto the service, which verifies the existence of the fingerprint of the certificate in the blockchain. As result not only true or false is presented, but

also information about the registered issuer (if it is a registered certification authority) and for the certificate the values of the attributes status, issuer, issue date and if set validity. Figure 7 presents the user interface of the verification service.

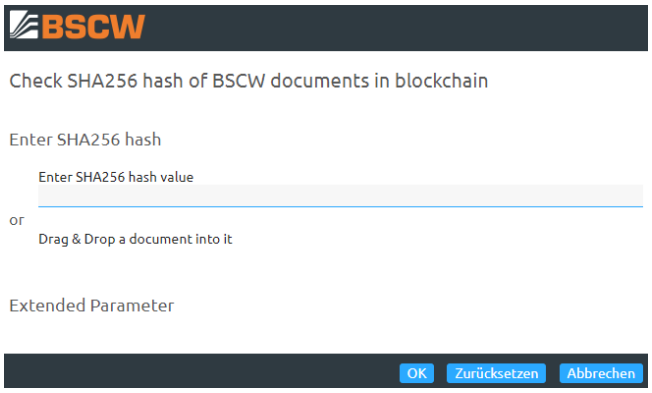


Figure 7: User Interface of Verification Service

USE CASES

The blockchain for Education platform enables tamper-proof archiving of certificates and their correct and permanent allocation to learners, as well as verification of certificates. In addition, three different scenarios are mainly supported. In the first scenario, a learner is interested in creating an application portfolio that contains selected certificates. The underlying groupware BSCW allows the creation of application portfolios. The learner adds the documents necessary for the application to the respective portfolio and share it with a potential employer. The employer can then verify the contained certificates by using the platform's verification service or other verification services that could cope with our extended Open Badges and that could call the specified smart contract.

In a second scenario, a learner has successfully passed an examination for a basic course on usability engineering. Later the learner took an additional qualification course on interaction and information design. After successfully passing this course, the learner automatically receives the qualification *senior usability engineer*. A smart contract is used to determine this new qualification. In a third scenario, a self-employed person presents *master craftsman in the area of high quality fitting* as professional qualification on the Web. Potential customers can verify the validity of the qualification as well as the issuing certification authority using a verification service.

Evaluation

The Blockchain for Education platform was developed in an iterative way with the participation of potential end users. A first version of the minimal viable product was intensively discussed with our personnel certification authority. This version contained already features for issuing and managing certificates. However, revocation of certificates was not foreseen and therefore introduced as an additional feature of the minimal viable product. In addition, our discussions with the personnel certification authority led to further smaller revisions and redesigns.

After internal testing, a workshop with a large German technical inspection association was organized to evaluate appropriateness of the Blockchain for Education platform for their certification authority. The workshop participants received a comprehensive presentation of the minimal viable product including the technical concepts and a demonstration of the prototypical platform was conducted. In the discussion, the participating certifiers confirmed our approach and were interested to use our platform for their certification processes. However, a few platform extensions will be necessary that primarily target import of examination results and specific extensions to our Open Badges schema. Issuing, validation and sharing of certificates remain almost unchanged.

DISCUSSION

Although certificates are currently issued as paper documents, we believe that there will be more digital certificates issued in the future. The usage of blockchain technology as presented in our paper has main advantages for digital certificates. Firstly, there is the decentralized immutable storage of digital certificates. Secondly, there is a verification service that allows third parties to verify easily the authenticity of certificates. Lastly, there are the identities of certification authorities and certifiers immutably stored in the blockchain. A non-blockchain platform that wanted to achieve counterfeit protection would have to implement appropriate services, especially services for digital signatures [20].

Certificates in the Blockchain for Education platform are represented according to the Mozilla Open Badges specification that became a quasi-standard. It is widely used and has the advantage, that its schema could be extended. There are a vast amount of APIs and tools available to create, manage or verify Open Badges.

A comparison of the Blockchain for Education platform with Blockcerts is especially interesting since it also supports certification processes based on blockchain technology. Blockcerts uses the Bitcoin platform and therefore cannot specify complex smart contracts. The Blockchain for Education platform employs smart contracts for the management of identities such as certification authorities or certifiers and for managing the lifecycle of certificates. In contrast to Blockcerts, our revocation model does not allow to show or validate revoked certificates. Other differences are mentioned in the next subsection on security and privacy.

Security and Privacy challenges

The Blockchain for Education platform tackles security and privacy challenges that have not been solved before. For instance, in contrast to the Blockcerts system, the hierarchical organization of identities, in the Blockchain for Education platform allows the actual certifiers to remain anonymous while still proving that they belong to an

accredited certification authority. The security of the Blockchain for Education smart contracts is based on approved templates from the OpenZeppelin collection and undergoes verification with *Osiris*, an extension of the Oyente symbolic execution tool we developed to discover integer over- and underflows. We implement safeguards to suspend the smart contracts of an Education for Blockchain instance in case of discovered vulnerabilities. This prevents future manipulation of the stored records while maintaining read-only access. For future iterations, we consider implementing an update mechanism for our smart contracts. This would allow us to patch vulnerabilities. Moreover, we are developing a privacy-preserving storage of personal information in an append-only public ledger with the help of advanced cryptographic protocols.

Limitations and Future Work

The Blockchain for Education platform is currently in a prototype state and can be extended and optimized in different aspects. First, the identity scheme is strictly hierarchical with the accreditation authority as a single powerful root node. In case the accreditation authority's private key is compromised or lost, the whole system is affected. In our future work, we will introduce a multisignature scheme for the accreditation authority where the power of a single private key is distributed to *k* out of *n* members which can act together as the accreditation authority – for example a number of national members of European Co-operation for Accreditation.

Further, as the system runs on the Ethereum blockchain, it introduces monetary overhead. For instance, adding a certificate to the blockchain implies transaction costs, which must be paid by the certifier. In a future version, a certifier might issue a pre-signed raw transaction to a proxy of the certification authority, which will refund the certifier and submit the transaction to the blockchain.

ACKNOWLEDGEMENT

We would like to thank all members of the Blockchain for Education project team for their support. We also gratefully acknowledge the insights from external application partners, educational institutions and certification authorities.

REFERENCES

1. Alexander Grech and Anthony F. Camilleri. 2017. *Blockchain in Education*. No. JRC108255. Joint Research Centre (Seville site).
2. Allen Ezell and John Bear. 2005. *Degree mills: The billion-dollar industry that has sold over a million fake diplomas*. Pyr Books.
3. BlockCerts to be developed in Malta. Retrieved March 12, 2018 from http://www.educationmalta.org/blockcerts -to-be-developed-in-malta/
4. Mike Sharples et al. 2016. Innovating pedagogy 2016: Open University innovation report 5.
5. Digital Certificates Project. Retrieved October 10, 2017 from http://certificates.media.mit.edu/
6. Certificates, Reputation, and the Blockchain – MIT MEDIA LAB. Retrieved October 10, 2017 from http://certificates.media.mit.edu/
7. Case Study Malta|Learning Machine. Retrieved March 12, 2018 from https://www.learningmachine.com/case-studies-malta
8. Case Study FSMB|Learning Machine. Retrieved March 12, 2018 from https://www.learningmachine.com/case-studies-fsmb
9. Meet TrueRec by SAP: Trusted Digital Credentials Powered by Blockchain. Retrieved March 22, 2018 from https://news.sap.com/meet-truerec-by-sap-trusted-digital-credentials-powered-by-blockchain/
10. Self-sovereign identity framework. Retrieved March 22, 2018 from https://www.techruption.org/usecase/xxcvxcvxcv/
11. Saving millions and increase privacy with blockchain. Retrieved March 22, 2018 from https://www.techruption.org/savings-millions-privacy-blockchain/
12. Sovrin-Protocol-and-Token-White-Paper.pdf. Retrieved March 22, 2018 from https://sovrin.org/wp-content/uploads/Sovrin-Protocol-and-Token-White-Paper.pdf
13. Apostille White Paper. Retrieved March 22, 2018 from https://nem.io/wp-content/themes/nem/files/ApostilleWhitePaper.pdf
14. Luu, Loi, et al. 2016. "Making smart contracts smarter." *Proceedings of the 2016 ACM SIGSAC Conference*. ACM.
15. Security analysis tool for Ethereum smart contracts. https://github.com/ConsenSys/mythril
16. Industry 4.0. Retrieved March 31, 2018 from https://en.wikipedia.org/wiki/Industry_4.0
17. Sandra Klein, Wolfgang Prinz, and Wolfgang Gräther. 2018. A Use Case Identification Framework and Use Case Canvas for identifying and exploring relevant Blockchain opportunities. DOI: 10.18420/blockchain2018_02
18. Thomas Osterland and Thomas Rose. 2018. Engineering Sustainable Blockchain Applications. DOI: 10.18420/blockchain2018_05
19. Wolfgang Appelt. 2001. What groupware functionality do users really use? Analysis of the usage of the BSCW system. IEEE.
20. Clemens Brummer. 2017. Edu thereum – A system for storing educational certificates in a public blockchain. Unpublished master thesis, University of Innsbruck.

Engineering Sustainable Blockchain Applications

Thomas Osterland
Fraunhofer FIT
Sankt Augustin, Germany
Thomas.Osterland@fit.fraunhofer.de

Thomas Rose
Fraunhofer FIT & RWTH Aachen
Sankt Augustin, Germany
Thomas.Rose@fit.fraunhofer.de

ABSTRACT

Blockchain technology has attracted attention as emerging paradigm for business collaboration. Blockchain's consensus mechanisms allow partners to cooperate in a business network. However, many applications reported in literature present merely a proof of concept from an engineering perspective. An industrialization of blockchain requires an engineering framework, which assures the sustainability of the application and in particular its network partnerships, i.e. each participant has to act as an active peer in the network rather than being a mere consumer with a wallet for participation in the blockchain. This paper presents the skeleton of such an engineering framework starting with an ideation of partnerships and collaboration patterns to clarify the incentives for participation via business model design for sustainable network operations towards the selection of an implementation platform for the business processes re-engineered. Moreover, an initial version of an interactive tool for community-oriented capturing of know-how about characteristics of blockchain platforms is presented.

Author Keywords

Blockchain Engineering, Incentives for Sustainable Operations, Technology Platforms, Correctness of Code, Modell Checking

INTRODUCTION

The digital currency Bitcoin has originally been the starting point of blockchain technologies, i.e. the distribution of transaction management across a network of computing peers combined with methods for consensus finding. The management of transactions is spread across a network of business collaborators replacing traditional intermediaries. Hence, establishing collaboration protocol agreements such as for conventional business-to-business cooperation with ebXML [5] is replaced by consensus finding.

New governance structures emerge due to the substitution of intermediaries. This change in structure directly calls for new business models and allow for a radical re-engineering of process landscapes [4]. Such a (re-) distribution of concerns combined with methods for consensus finding makes blockchain attractive for many application domains that require a consolidation of inputs from different parties, e.g.,imagine the potential of blockchain for an open business-to-business collaboration [7].

However, sustainability of the partner network is decisive, i.e. incentives for the partners to participate actively in order to maintain network viability. Otherwise, once partners only participate in a consumer-oriented fashion just with a wallet for information exchanges, network diversity becomes deserted finally yielding to umpire control. Hence, incentives for network participation become vital as an incentive of equal importance compared to any foreground advantages. Moreover, the business model in place and the operational processes have an impact on the implementation options for the blockchain platform, e.g., visibility of transactions or provision of smart contracts.

Hence, our engineering framework progresses in steps:

- *Incentive assurance* – ideation of the application for assessing its blockchain potential while identifying particular advantages and incentives for sustainable participation in the network;
- *Partnership network* – draft a new governance structure by identifying (new) stakeholders and their roles in the business network;
- *Network experience and business model* – business model for operating new services amid the network partnerships;
- *Platform properties for process implementation* – specific blockchain characteristics that are determined by re-engineered business processes, i.e. the functional and non-functional requirements of the processes guide the selection of the platform.

Thomas Osterland, Thomas Rose (2018): Engineering Sustainable Blockchain Applications. In: W. Prinz & P. Hoschka (Eds.), Proceedings of the 1st ERCIM Blockchain Workshop 2018, Reports of the European Society for Socially Embedded Technologies (ISSN 2510-2591), DOI: 10.18420/blockchain2018_05

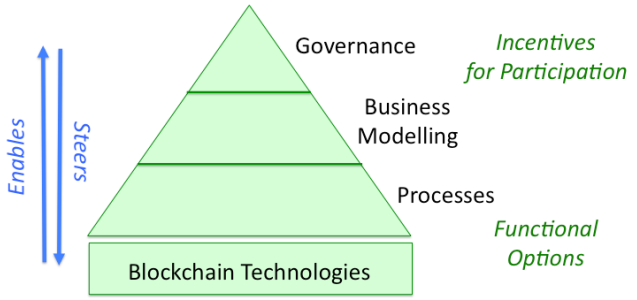


Figure 1: Blockchain Engineering Layers

SUSTAINABLE GOVERNANCE STRUCTURES – INCENTIVES FOR NETWORK PARTNERSHIPS

Trust in the transaction history of a blockchain is only as strong as the plurality of the community behind the blockchain network. A healthy incentive system for running the blockchain nodes is of essential importance for the sustainable operation of any blockchain network.

One important aspect for the sustainable operation of blockchain networks refers to the sustainability of network governance. The major advantage of blockchain is often described as the elimination of intermediaries. Strictly speaking, an intermediary is not eliminated, but replaced by consensus finding in a distributed network. Blockchain enables a fair collaboration between partners of different size and power independently from their available resources or institutional influence. Yet, once incentives to run nodes in blockchain network are not equally distributed, then there is no reason for every partner to run a blockchain node. As a consequence, the network will implode and finally a small number of members gains network sovereignty. Hence, it is a decisive engineering issue, whether a case constitutes a valid application for blockchains? Why not just create a conventional database between those partners?

The identification of incentives is not always apparent. Aspects such as increased freedom, improved security and potential fraud prevention are hard to assess economically. The slimming of business processes might be a measurable improvement in costs, as well as the replacement of several partners providing comparable functionality as an equally functional single blockchain application, but many effects are hard to assess in advance.

Blockchain is often advertised as means for securing the exchange of information in a way that no party can tamper with data that is maintained in the network. Moreover, smart contracts enabl a blockchain to introduce a new level of fairness into processes. Going down this avenue, business processes cannot only be automated on a blockchain, but also connect automatically to imposed enforcement fees that can make blockchain networks particularly more attractive for small partners. For instance, a bottler that depends on a certain type of bottles from a supplier can punish the bottle supplier, when she is not able to deliver the necessary amount of bottles. Although the bottle supplier is actually powerful enough to not care about a single bottler, the blockchain network will automatically punish the supplier. An active network node secures the tracking of situations of under-supply. Because the blockchain enforces this punishment against the unequal partner it becomes an incentive for the bottler to operate a blockchain node. Hence, there is a natural motivation to participate in the blockchain network, but not only as a consumer of transactions with no actual relevance regarding the process fairness from a personal perspective.

In a methodical stance, we propose a blockchain sustainability canvas for identifying incentives for active participation in a blockchain network. The sustainability canvas (displayed below) provides assistance in:

- Identifying incentives for different parties in a blockchain networks;
- Rating the incentives against each other and identify weak network partners;
- Rating the value and the quality of the network with respect to sustainability;
- Matching existing incentives with smart contract enabled business processes to enable the engineering of new incentives.

The canvas is organised into four areas. The upper left area covers the identification and assessment of incentives. In a first step the expected network participants can be collected in the box "*Network Participants*". For every network participant it is important to identify advantageous incentives that come with the network participation. These can be documented in the "*Participation Incentives*" box. Disadvantages for a certain party that origin from joining a blockchain network on the other hand, can be collected in the "*Participation Disadvantages*" box. The resulting aspects of these two sides can be invaluable assets in identifying weaknesses of a network or potential points of entry to start re-engineering an existing process.

After focusing on the sustainable operation of the blockchain network, the area in the upper center helps in analyzing the application of existing processes to a blockchain network and the identification of new potentials regarding collaboration and process optimization. In the box "*Trust Enabler*" we identify where a blockchain can provide trust in existing processes and how those processes profit from it. In general we observed two cases: In the first case, the existing process was defective and the blockchain can be used to provide trust in a way that previously was not feasible. In the second the blockchain can replace an existing source of trust. This often leads to the elimination of intermediaries.

"*Change of Governance*" covers the potential change in process governance by introducing a distributed blockchain network. For instance, who holds the sovereignty of a process when it is executed independently as a smart contract in the blockchain? But also who controls the access to the blockchain network? In case of a permissioned blockchain the requirements to join the network or to open it to

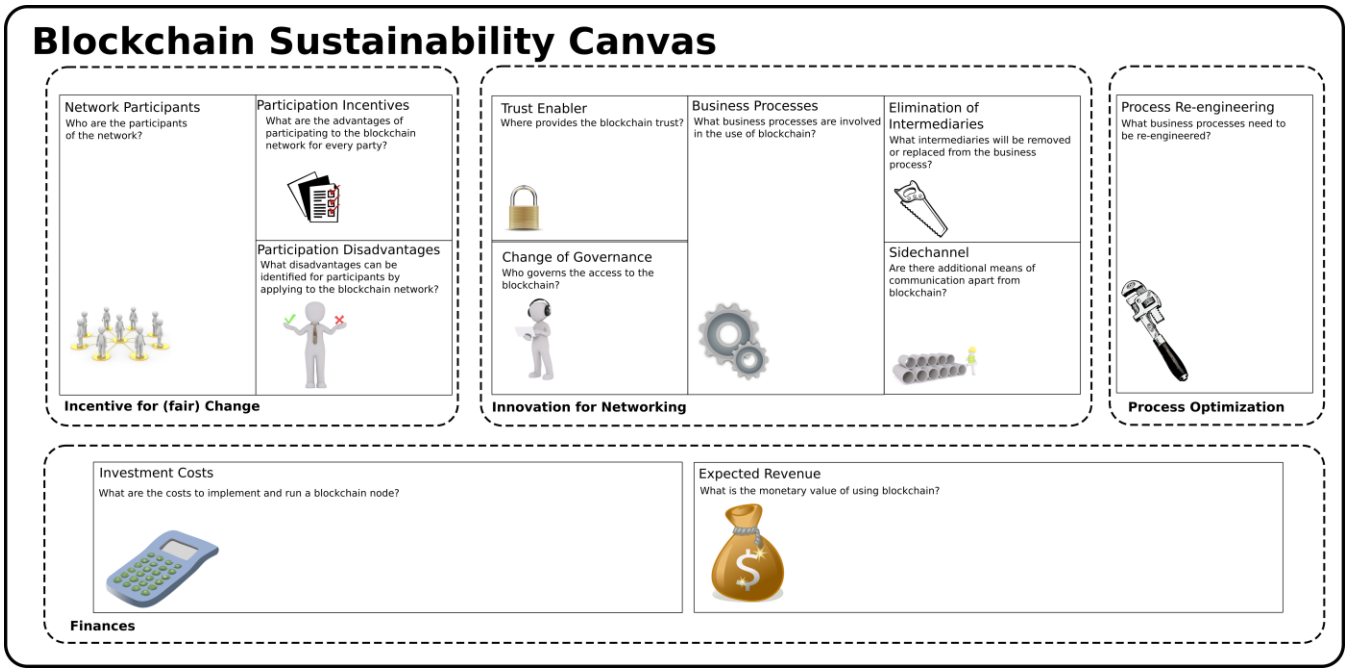


Figure 2: Blockchain Sustainability Canvas

additional parties must be specified. Does every member has the right to add a party or is it a voting based decision? Existing processes that might be supported, or that can be adapted to be used in the blockchain network are collected in the box "*Business Processes*". The potential loss of existing parties leads to immanent changes in the structure of a process. Those can be documented in the box "*Elimination of Intermediaries*". Complex processes often require the communication via multiple channels, e.g., it is still necessary to send paper documents due to legal conditions. Although not directly part of the blockchain these interactions are still part of the process. It must be evaluated how that affects the security of the blockchain and if it can be improved by re-engineering the process.

The upper right area covers the question whether processes can be re-engineered to be more suitable for the blockchain context after benchmarking existing business processes with respect to their applicability and performance regarding their application in a blockchain context. This might comprise the purposeful replacement of a party by a smart contract and thus the elimination of an intermediary or the slimming of exchange processes by utilizing the secured append-only ledger property of the blockchain.

Finally the lower area covers the aspects of costs regarding the operation of a blockchain network. In the left field operational costs can be documented. This can be energy costs, administration and maintenance costs, but also in case of public blockchains the costs of a transaction and an estimation about the volume of transactions.

The right field allows the documentation of expected revenues from the application of the blockchain. The slimming of processes decreases the number of involved parties and can crucially accelerate the processing time. But also costs for the audit of accounting processes can be decreased by allowing the auditing entity access to the blockchain network.

CRITERIA FOR TECHNOLOGY SELECTION

There exists a multitude of different blockchain platforms that are engineered with respect to different functional objectives. Some platforms focus on high transaction throughput that means the number of transactions that can be processed by a blockchain per second, e.g., Fabric, while others identified the visibility of data as major problem of existing blockchain platforms, e.g., Quorum. The development of new blockchain technologies is proceeding apace. The sheer volume of available technologies is often overwhelming when trying to get an overview. Similar to the engineering of software systems, it is important to choose the technology that solves a problem and not to find a problem that can be solved by a given technology the decision should be strongly based on dispassionate facts instead of personal preferences.

Starting with a large set of potential blockchain platforms, a subset of blockchain technologies being suitable for a given problem can be derived by considering the following aspects:

- *Access policy* – Permissioned vs. public blockchain;
- *Process integration* – Availability of smart contracts or chain code;
- *Scalability and transaction performance* – Transaction throughput;
- *Restricting data access* – Data privacy and visibility;

- *Network governance* – Ease of adding/removing nodes to the network;
- *Technology governance* – Open source, project management, development kits.

Figure 3: Technical Aspects

Access policy

A permissioned blockchain restricts the access to the blockchain network to only a selected number of people. Such a network is useful in case that known parties wish to cooperate and exchange data or participate on processes in a secured way, such that no party can tamper with it and the origin of transactions can be unambiguously dereferenced. The assumption is that data and processes that are stored in the blockchain are only of relevance for parties in the network.

In contrast, access to public blockchains is not restricted and can be easily acquired by creating a public/private key pair. In this case a large group of people can be addressed, but data is freely exchanged throughout the network. The decision for a public or a permissioned blockchain also depends on the targeted accessibility. A blockchain as a distributed ledger is not only a technology to securely store and enact transactions, but also a platform to enable the standardized communication of different parties.

Process integration

The next major criteria depend on the potential requirement for using smart contracts. Smart contracts allow the untampered execution of program code within the blockchain. Complex processes with a variety of execution options can be modeled by a single smart contract or a multitude of interacting smart contracts. However, if there is only the requirement to securely store and exchange data, then more lightweight technologies can be considered.

Scalability and transaction performance

The throughput of transactions determines the throughput of data that can be handled by a blockchain and this affects the potential number of people and interactions with the blockchain. For comparison: In Bitcoin the number of transactions per second (tps) is around 7 tps, in Ethereum it is around 15 tps and the permissioned blockchain Hyperledger Fabric claims a transaction throughput of 3,500 transactions per second [1]. Estimating the number of expected transactions that are applied on a blockchain can reduce the set of potential technologies drastically. Recently, many popular blockchain platforms are confronted with the problem of general scalability. Different solutions and approaches are proposed,, e.g., sharding [3] and it is a question of the future, which concept will prevail.

Restricting data access

Although these three aspects are of major importance by deciding for a blockchain technology there are additional aspects that must be considered: Data privacy and data visibility is important for many application scenarios. A user of a financial application does not want to share her income publically to every member of the blockchain and for transactions between companies even cartel considerations can influence the data visibility allowance. Depending on the project it must be ensured that only a subset of participating network members can access certain information. A fine-grained permission control systems must be supported.

Network governance

The ease of adding and removing nodes to the network influences the decision for a certain technology. Is there a high fluctuation of network members or is the network rather steady. In the first case the addition or removal of members should not lead to a necessary shutdown of the whole network to start the nodes with new configurations. Lightweight administration processes will increase the maintainability of the network and reduces configuration errors.

Technology governance

As a final aspect the governance of the technology is of major importance. Although in case that the selected technology is open source there is no secured guarantee about the future of the project, i.e. whether the selected technology will be periodically updated and in particular supplied with security fixes. High costs for exchanging the underlying blockchain technology or to continue the development with in-house resources must be considered in case of a non-continuation.

To support the methodical selection of suitable blockchain platforms we created a web based assistant that provides a structured questionnaire to assist in the selection of an appropriate platform. Platforms can be related, compared and analyzed with respect to different aspects. Considered platforms include smart contract enabled platforms as well

as merely transaction based technologies. Digital currencies are not part of it.

Currently we support only a relatively small number of 10 blockchain technologies. However, we provide interfaces so that users can propose or directly add further technologies. References to articles and sources are attached to the presented information to increase transparency. Hence, a community platform for knowledge exchange on blockchain technology elements and application-specific constraints is maturing.

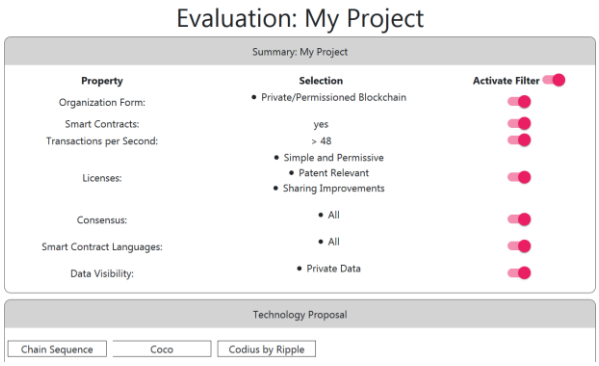


Figure 4: Blockchain Technology Assistant

Selecting the suitable technology for a blockchain project according to its requirements and based on dispassionate facts will ensure a strong technical foundation for a long living and sustainable operating blockchain network. By considering future usage behavior scalability problems can be identified or completely avoided.

ASSURING VALIDITY OF OPERATIONS BY SMART CONTRACTS

The security and strength of a blockchain strongly depends on the strength of the underlying network and thus on the network members that operate blockchain nodes. On the other hand the secure operation of blockchain nodes depends on the trust of network participants. There exists a mutual relationship. If a party does not trust a network she will not expend the work and costs to participate at the network. One might argue that the correctness of the blockchain is cryptographically ensured and that there is no reason for a party to not trust the blockchain. However, in case of smart contracts this is not always the case. Although the blockchain ensures that the program code is exactly executed as it is stored in the blockchain it is not always trivial to decide that a program code acts as expected. A popular example for a wrongly programmed smart contract is the splitting function of the DAO contract [6]. Although the program code is exactly executed the execution result does not match the expectations of the process participants. Such errors in the software can lead to advantages for one group and disadvantages for another. Thereby a simple replacement or update of a smart contract is not possible, since all parties need to accept this new version. If a party gains an advantage from a faulty version she might not be interested in switching to a new correct contract version.

As a consequence the correctness of smart contracts can affect the trust of parties into a blockchain and consequently affect the strength of the whole blockchain network.

There exists a multitude of methods for formally verifying program code. We suggest the use of model checking as a simple, potential one-click solution [2] for verifying the correctness of a smart contract. The idea is to model the expected behavior of a program with logical formulas as linear temporal logics (LTL) or computational tree logics (CTL). For model checking we analyze the state space of a program. The state space is the set of program states, where a program state represents the state, which is the valuation of every variable, of a program for a certain execution step. The state space of a program is examined whether one of the logical formulas is violated. In case of a violation a counter example is derived that supports the identification and correction of flaws in the program code.

Although the concept of model checking is easy to grasp the application of the technology imposes a major challenge known as the state space explosion problem. Even a small program can have a very large number of different program states and thus leads to an extremely large state space. A large state space can be unfeasible to analyze and by applying model checking it is one of the major challenges to reduce the state space. However, certain characteristics of smart contracts support the model checking process. So a smart contract is limited in the number of executional steps to prevent blocking the whole blockchain by for instance running into an infinite loop and secondly, the blockchain ensures the atomicity of transactions. The code called by a certain transaction is ensured to be executed without interruption, only the order of transaction execution cannot be predicted.

An additional idea for improving the quality of smart contracts and to simplify the formal verification during the development process is to provide repositories of formally verified libraries. Such a library can be integrated into a smart contract. During the model checking it can be assumed that this code is already correct (with respect to a certain set of logical formulas). This will crucially decrease the number of states that must be analyzed.

Although, formal verification will increase the quality of a smart contract and will ensure that a smart contract works correct with respect to the formally specified expectations, the formal verification is only as good as the specification of the expected behavior. Missing or incorrect constraints can lead to verified program code that still reacts unexpectedly in certain conditions.

To ensure a sustainable trust into the blockchain network and thereby a sustainable operation of the blockchain network, the correct functionality of smart contracts is crucially

important and using tools that are able to increase the quality of smart contract becomes a major requirement.

USE CASE EXAMPLE

In the following we apply the blockchain sustainability canvas to the use case of pool boxes. These are for instance boxes used to transport vegetables from a farmer to a retailer over a number of intermediate stations. There are four parties involved in the use case: *poolbox operator, filler*, *distributor* and *retailer*. The *filler* produces some sort of product and sells it to the distributor. The *distributor* wants the goods delivered in a certain type of box and makes a contract with the *poolbox operator* to provide these boxes to the filler. The filler pays a "refuel fee" to the poolbox operator and sends the filled boxes to the distributor. The distributor takes the boxes and pays a pledge to the filler. From the distributor the boxes will be delivered to the *retailer,* who again pays a pledge for receiving the boxes. The retailer empties the boxes and returns them back to the poolbox operator, who cleans and repairs them and then reintroduces them into the cycle.

This explanation represents the existing process and there are problems that motivate participants of the process to apply blockchain. Boxes often change ownership in an uncontrolled fashion, e.g. stolen from premises, because the ownership is not tracked and by returning a box to the box supplier the pledge is refunded. Also counterfeiting of boxes is a problem. Then pledge is refunded, although the boxes are potentially of bad quality or miss certain certificates allowing their use with food or dangerous goods. Another critical point is that fillers are hoarding boxes to prevent the lack of boxes for their (individual) peak times of production and hence delivery, while poolbox operators are interested in a continuous flow of the boxes to reduce the necessary number of boxes. On the other hand is it possible that fillers cannot bring goods to the market, since the poolbox operator is not able to deliver the necessary amount of boxes. In contrast to the poolbox operator there are multiple filler with relatively small influence in the process. So it is hard for filler to enforce potential claims.

The potential blockchain participants are the four parties, although in practice a role can be inherited by multiple entities. As a next step we need to identify the incentives and disadvantages for the parties to participate at the blockchain network. The governance of the unaltered process lies at the distributor and the poolbox operator. These two entities negotiate a contract and make the major decisions. When a poolbox operator does not deliver enough boxes to a filler the position of the filler is relatively weak. An incentive for a filler could be the smart contract based punishment of missing boxes. For every box that cannot be delivered the poolbox operator must pay a certain punishment fee to the filler. On the other hand the filler pays punishment fees back to the poolbox operator in case she is hoarding boxes. Of course introducing smart contracts in this way will reduce the supremacy of the poolbox operator what can be considered as a disadvantage.

In general by tracking and controlling the exchange of boxes with a blockchain every party gains the ability to take measurements of box distribution. This can be interesting for predicting the necessary amount of boxes for a given point in time. The tracking of box ownership will prevent stealing. Then a box can be only refunded, when the persons owns the box. Although a thief can still steal the box she is not able to refund it, since no clearing house will accept them when the person cannot proof the ownership with the blockchain.

The blockchain serves as trusted entity and partially transfers the process governance to the community. In this first approach only partially, since there is still the distributor that makes the decision about the box types and the poolbox operator. In this case we do not eliminate an existing intermediary, but enrich the process with box tracking abilities to eradicate flaws in the existing process. A consequent process re-engineering could additionally improve the fairness between different parties. By opening the process to arbitrary poolbox operators that are bound to deliver only a certain type of boxes and multiple distributors the supremacy of the two parties can be restricted and smaller entities as the filler and retailer can replace those in case of problems. That might be an additional incentive for the retailer to join a blockchain network.

Another potential extension is to completely replace the poolbox operator by a decentralized autonomous organization (DAO) where every party is a shareholder. The DAO works on a cost covering basis with the only target to provide the necessary amount of boxes to the different participants in the process. In this case the position of the poolbox operator is eliminated and replaced by a smart contract. Certain problems of the conventional process, as the hoarding of boxes, are naturally eliminated, since a high volume of circulating boxes increases the operation costs and thus provides a disadvantage for every stakeholder including the filler that hoards the boxes.

LESSONS LEARNT FOR INDUSTRIAL SCALE-UP

So far, reports on engineering paradigms are sparse. Papers are dominated by proof of concepts to propose blockchain as implementation vehicle for various domain applications. Any industrial scale-up will require the development of a methodological founded engineering paradigm that accompanies a blockchain' lifecycle from *the cradle to the grave*. To start with, the eligibility of business models for the application to blockchain is of major importance. The evaluation of existing business processes regarding its suitability for deploying a blockchain enables the re-engineering or the creation of new processes. Outgoing from these specifications it is important to engineer the sustainability of the emerging blockchain network. A mutable dialog allows updating a business process while increasing process fairness and thus optimizing the different party's incentives to participate at the blockchain.

The determination of the business processes will strongly affect the technology selection. Several aspects, such as the

scalability, visibility of data for different stake holders as well as interoperability of the blockchain network as a technology platform needs to be evaluated with respect to the process requirements. The great variety of existing technologies and the speed of development complicates the selection process and must be simplified by introducing standards for the blockchain technology. Besides the ease of comparing blockchain technologies this will also simplify the exchange of underlying platforms.

During the implementation phase of a blockchain in particular the quality of smart contracts is of major importance. A poorly engineered smart contract can ruin the user experience and more important, can sustainably affect the trust of users into the blockchain network. In case of a crucial failure most users will not blame the smart contract as an independent element, but the complete blockchain network or even the blockchain technology. A potential image loss is a big risk. Hence, the broad application of standardized smart contract libraries and paradigms for testing and verification will improve the overall quality of smart contracts, as well as, speedup the development process from an economical perspective.

Although stressing its distributed nature for transaction management, a blockchain is certainly not a highly performant repository technology for the management of mass data as in production processes for instance. However, the information sharing among manufactures and suppliers can be certified by a blockchain. Hence, a separation of concerns has to be decided: managing operational data of production processes by database management technology versus maintaining audit information between business partners by blockchain technology.

As a final step it is necessary to determine the estate administration of the blockchain. What happens to the data stored in the blockchain, when the blockchain lifecycle is at its end? There exist legal requirements like the "right to forget" that demands the loss of data. How can that be achieved by blockchain technologies.

Hence, there are several engineering issues to be researched in order to bring blockchain technology to industrial use and allow for an industrial scaling. One of the upmost challenges is certainly research on methods for consensus finding beyond a sole proof of work as well as proof of stake. Any wide-spread deployment of blockchain technologies will be tampered without any scaling of computing complexities for consensus finding.

CONCLUSIONS

This paper presents our approach to the engineering of blockchain applications with a particular emphasis on the sustainability of the partner network. Any public blockchain requires a lively and vivid network of partners actively supporting the network. Otherwise, the character of the collaboration has to be switched to a permissioned blockchain, which re-introduces the concept of umpires.

Moreover, the characteristics of the application have to be assessed carefully in order to select a platform from the array of available platforms. The structured questionnaire presented is only a first step to guide through this complex decision process. In addition, the questionnaire is connected with references for evidence-based decision making. An interactive editing component allows for a community-based capture of platform characteristics and implementation experiences.

REFERENCES

1. Elli Androulaki, Christian Cachin, Christopher Ferris et Al (2017). Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains, IBM.
2. Christel Baier, Joost-Pieter Katoen (2008). Principles of Model Checking, MIT Press.
3. Vitalik Buterin (2018). On sharding blockchains, https://github.com/ethereum/wiki/wiki/Sharding-FAQ.
4. Michael Hammer, James Champy (1993). *Reengineering the Corporation – A Manifesto for Business Revolution*, Harper Business.
5. Martin Sachs (2001). ebXML – Collaboration Protocol Profile and Agreement Specification, IBM T.J. Watson Research Center, Yorktown Hts, NY, https://www.oasis-open.org/committees/download.php/214/ebxml-cpp-cpa.pdf
6. David Siegel (2016). Understanding the DAO Hack for Journalists, https://medium.com/@pullnews/understanding-the-dao-hack-for-journalists-2312dd43e993.
7. Hemang Subramian (2018). Decentralized Blockchain-based Electronic Marketplaces, *Communications of the ACM 61*, 1, 78-84.
8. Sandra Klein, Wolfgang Prinz, and Wolfgang Gräther. 2018. A Use Case Identification Framework and Use Case Canvas for identifying and exploring relevant Blockchain opportunities. In: W. Prinz & P. Hoschka (Eds.), Proceedings of the 1st ERCIM Blockchain Workshop 2018, DOI: 10.18420/blockchain2018_02
9. Wolfgang Gräther et al. 2018. Blockchain for Education: Lifelong Learning Passport. In: W. Prinz & P. Hoschka (Eds.), Proceedings of the 1st ERCIM Blockchain Workshop 2018, Reports of the European Society for Socially Embedded Technologies (ISSN 2510-2591), DOI: 10.18420/blockchain2018_07

ACKNOWLEDGEMENTS

This work has been supported in part by the b-it foundation (http://www.b-it-center.de). Parts of this work are also based on joint work with Media Informatics students of b-it (Bonn-

Aachen International Center for Information Technology) in the context of Lab Courses and Thesis work.

A Use Case Identification Framework and Use Case Canvas for identifying and exploring relevant Blockchain opportunities

Sandra Klein
Fraunhofer FIT
Sankt Augustin, Germany
sandra.klein@fit.fhg.de

Wolfgang Prinz
Fraunhofer FIT/RWTH Aachen
Sankt Augustin, Germany
wolfgang.prinz@fit.fhg.de

Wolfgang Gräther
Fraunhofer FIT
Sankt Augustin, Germany
wolfgang.graether@fit.fhg.de

ABSTRACT
Blockchain is a new, foundational technology with a vast amount of application possibilities. However, practitioners might not be aware of which use cases in their own business model might benefit from blockchain technology. To aid them in analyzing their business regarding blockchain suitability, this paper introduces a use case identification framework for blockchain and a use case canvas. In the development process they have been evaluated with internal and external reviews in order to offer the best possible guidance. In combination they offer an analysis framework to help practitioners decide which use cases they should take into account for blockchain technology, which characteristics these blockchain implementations would have, and which specific advantages they would offer.

Author Keywords
Blockchain; Identification Framework; Use Case Canvas

MOTIVATION

About blockchain

Contracts, transactions and related data sets are concepts which are indispensable for everyday life. Ownership of assets or agreements between several parties need to be documented and made transparent. The digital transformation enables new opportunities to realize these documentations, but also poses new challenges which need to be acknowledged. Blockchain is a foundational technology which can hold up to these challenges, enabling data security and transparency while documenting transactions in a decentralized, secure, transparent and irreversible way [4].

Blockchain is defined by a few unique characteristics: Firstly, the technology uses distributed consensus-building between the nodes in the blockchain network instead of having an intermediary approving all transactions [9]. Rather than having to trust this third-party intermediary, trust is placed in the technology itself. Additionally, transactions can be processed almost immediately, instead of having to wait for the third party to process them [13]. Secondly, the blockchain enables the transfer of rights to real-world objects and values explicitly and permanently, making the verifiability of ownership and rights straightforward [9]. Thirdly, all transactions which are stored in the blockchain are irreversible and transparent, therefore the data cannot be tampered with.

Klein, Sandra; Prinz, Wolfgang, Gräther, Wolfgang (2018): A Use Case Identification Framework and Use Case Canvas for identifying and exploring relevant Blockchain opportunities. In: W. Prinz & P. Hoschka (Eds.), Proceedings of the 1st ERCIM Blockchain Workshop 2018, Reports of the European Society for Socially Embedded Technologies (ISSN 2510-2591), DOI: 10.18420/blockchain2018 02"

We define blockchain as a technology which can offer increased value for partners cooperating in a decentral network by providing data and process integrity, automation potential and enabling the transparent transfer of values and rights.

Using a blockchain to document transactions has the advantage of having a digital record and a digital signature for every transfer of rights to objects or values, but also for agreements, processes and tasks [4]. Using blockchain for agreements or process automation can be achieved by taking advantage of smart contracts, which enable the automatic processing of transactions if certain conditions are met: For example, if the arrival of a certain good is documented in the blockchain, e.g. by a sensor which documents the good's location, a smart contract can automatically trigger the payment process of this good based on the sensor's transaction. Smart contracts are saved in the blockchain and transparent to any member of the network.

While public blockchains like Bitcoin are permissionless, meaning that everyone can participate and issue transactions, private blockchains consist of a chosen consortium of users, making the blockchain inaccessible by the general public [9].

Relevance for practitioners

In the process of applying blockchain technology to different ecosystems, some business models are replaced as blockchain makes the processes more efficient and secure, e.g. auditors who get replaced by automated process audits [9]. On the other hand, new business models might be created, where the previous lack of trust between participants or economic inefficiency to build a partnership posed a challenge for a successful cooperation. For practitioners, it is important to be aware of use cases and application areas which benefit from blockchain technology.

Possible areas of application are vast, and new ideas how to apply blockchain to make business models better constantly evolve. The most known application area is Finance, with

Bitcoin being the most famous application. Cryptocurrencies and blockchain could potentially make banks obsolete, or at least improve processes such as the trading of foreign currencies, which is at the moment a time-consuming process but could be simplified by using blockchain [1,3]. Another application area is the Internet of Things (IoT): Transactions between smart objects, which happen without human interference, can be documented in the blockchain or even automatically triggered by using smart contracts [9]. An example for using blockchain for IoT use cases are smart locks, which enable an automated and safe way to rent objects and transfer the respective payment [2]. Smart grids, where private energy generators and energy consumers freely trade energy without going through an energy provider are an example which is already being implemented [6]. Blockchain might also revolutionize the way how the proof of origin for important documents or objects can be determined. Each document or object which is uniquely identifiable can be stored as a transaction in the blockchain which documents its owner [9]. This is for example being realized to track diamonds on the platform Everledger, and a platform to document digital certificates to provide a life-long learning documentation is currently being developed [5]. In the context of supply chain management, blockchain enables the tamper-proof documentation of changes of ownership and provides the possibility to automatically transfer funds between supply chain participants as the ownership changes [9]. Computer Supported Cooperative Work is not yet a widespread topic for blockchain research, but could be a promising application area [10]. Other application areas include, but are not limited to, medicine, media, the public sector or law.

Objective and Structure

Blockchain technology offers great potential for cost, time and efficiency improvements of existing business models. Therefore, practitioners need a structured guideline to analyze their business, find the right processes that are suitable for and can benefit from blockchain technology, and understand how blockchain can support these processes. To meet these needs, this paper proposes an analysis framework to guide practitioners in their blockchain implementation journey. This analysis framework consists of a two-step approach: At first, the user is supported in exploring which use cases are the most suitable for blockchain technology and afterwards he receives guidance on how exactly the new technology offers advantages for the specific use case. The goal of this analysis framework is not to give a simple yes/no answer to the question if blockchain is suitable, but it rather aims at helping practitioners develop a deeper understanding of how the blockchain could support their use cases.

To give this analysis framework some more context, a second section of this paper will explore related work, referring to other, popular frameworks and canvases. Afterwards, both the use case identification framework and the use case canvas will be introduced and described in detail. In the third section both the framework and the canvas are applied to some exemplary use cases to present how they can help practitioners in their blockchain decisions. The fourth section then explains how the analysis framework was developed and evaluated, before the last sections summarizes the advantages the framework offers, explains some limitations and proposes directions for further research.

RELATED WORK

Frameworks and canvases are often used to facilitate the application of complex theories to real-world use cases. They provide the user with a guideline on how to apply theoretical concepts to a specific application. One example is the Business Model Canvas proposed by Osterwalder and Pigneur [7], which helps to design and understand a business model by investigating its customer segments, customer channels, customer relationships, value proposition, key resources, key activities, key partnerships, revenue streams and cost structures. The interactions between those nine aspects summarize the overall business model. The Business Model Canvas provides the user with an open template that can be used to describe all nine aspects of his business model.

As a second example, design thinking is an approach describing creative design processes focusing on user needs. To facilitate this processes, frameworks have been developed to guide users through the process of design thinking. For example, the Stanford DSchool developed a playbook providing guidance on the five steps of design thinking, namely empathize, define, ideate, prototype and test [12]. For each iterative step, the Playbook provides the user with questions and instructions on how to apply design thinking.

Regarding blockchain, a framework to determine whether or not a blockchain is useful to solve a problem was proposed by Wüst and Gervais [14]. They present a flow chart which guides the user through several yes/no questions about their problem or use case, including the amount of partners in the network, the availability of a trusted third party and the level of trust with which the partners can be met. Depending on which path the user takes through the questions, the framework gives a recommendation of whether blockchain should be used, and if so, which type of blockchain (permissionless, public permissioned or private permissioned) would be the most appropriate.

USE CASE IDENTIFICATION FRAMEWORK

With blockchain technology being a growing topic of interest and an increasing amount of blockchain applications being developed, many businesses need to ask themselves how their business models are going to be affected by blockchain, and how they could use this new technology to exploit the advantages it offers for business model improvement. The proposed use case identification framework for blockchain displayed in Figure 1 addresses this challenge by providing guidance on the assessment of the suitability of specific use cases for blockchain technology.

To achieve this, the framework consist of the three categories *intermediary*, *data* and *process*. For a specific use case, these

three categories are evaluated separately regarding the blockchain suitability of the use case.

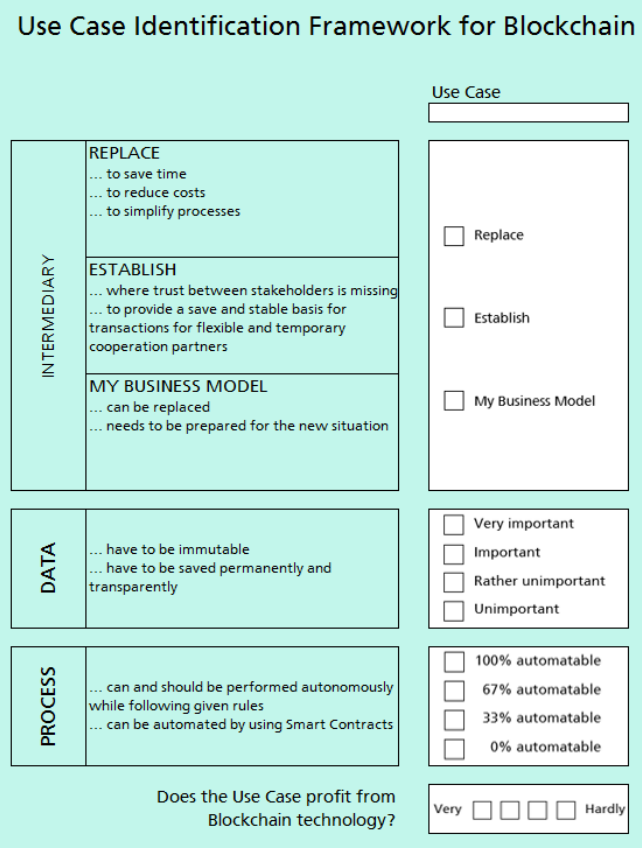


Figure 1: Use Case Identification Framework for Blockchain

The first category, *intermediary*, explores the existence and the role of intermediaries in the use case, since a blockchain functions as an independent and incorruptible intermediary. There are three scenarios in this category, *replace*, *establish*, and *my business model*. While the first two scenarios are usually relevant for anyone needing an intermediary for a use case, the third scenario is relevant for anyone functioning as an intermediary in a use case. Each scenario describes a specific situation and the user of the framework can decide which one, if any, is applicable in the use case. The first scenario, *replace*, describes a situation where an intermediary is currently existent and acting as a third party between stakeholders. However, the use of this intermediary might be time- or resource-consuming, or the process of interacting with other stakeholders through the intermediary could be tedious or complicated. In this case, blockchain technology could be used to save time, reduce costs or simplify the process.

The situation described in the second scenario, *establish*, is applicable for potential use cases where currently no intermediary is in place because there is a lack of trust between stakeholders and towards any possible intermediary. In this case, blockchain could provide a safe and stable basis for transactions without needing the partners to trust a third party, instead they can trust the technology. This blockchain scenario could be especially useful for flexible and temporary collaborations.

The third scenario, *my business model*, describes the situation from the view of an intermediary that could potentially be replaced by blockchain. In this case, possibilities to keep the business model useful for the partners in the network and provide added value compared to a blockchain need to be explored. To this end, blockchain technology might be used to provide the partners with a new solution and a better use case and thus preventing them to replace the old intermediary.

Since those three scenarios describe different situations from different viewpoints, usually only one scenario will be rated as true. If no situation is applicable to the use case, it might still benefit from blockchain technology, but it is probably not crucial to the use case.

The second category in the framework, *data*, assesses the use of data. Blockchain technology offers the possibility to save data permanently and transparently as well as preventing anyone from modifying the data after it has been entered into the blockchain. In this category, the user should evaluate how important those characteristics are to the data used in the use case. Depending on how necessary the protection of data from attacks and the permanent accessibility of the data are to the use case, the more important should this category be rated. The importance is measured on a four-point scale ranging from unimportant to very important.

In the third category, *process*, the potential for automation in the use case can be assessed. The user should evaluate if the processes contained in the use case can be (further) automated by designing rules to perform process steps autonomously. Since blockchain enables the use of smart contracts to automatically trigger transactions, those contracts can be used for automation purposes and thus for making the process more efficient. Therefore, if the use case would benefit from automation, blockchain technology could provide this automation. The user can evaluate the automation potential on a four point scale ranging from 0% automatable to 100% automatable.

After the assessment of each category for the use case, it needs to be evaluated whether the use case would overall benefit from blockchain technology. This is usually the case if one scenario in the category *intermediary* is rated as true and the other two categories, *data* and *process*, are rated as important/ automatable. The more positively the last two categories are being rated, the more suitable blockchain technology is for the use case. The evaluation of how much the use case would profit from blockchain can be assessed on a four point scale ranging from very to hardly. The user can then utilize this summarizing assessment to see if a single use case is suitable for blockchain, or, if several use cases were evaluated, which one would profit the most and should be implemented first.

USE CASE CANVAS

While the use case identification framework helps practitioners to identify which use cases are suitable for blockchain technology, there is still a need for understanding how exactly the blockchain would impact the use case. The proposed use case canvas for blockchain, partially displayed in Figure 2, enables the user to develop deeper insights into how a suitable blockchain would be structured. Additionally, it helps to identify the potentials that could be unlocked by using blockchain technology compared to the current use case without blockchain.

There are five categories presented in the canvas, which collectively describe relevant characteristics of a blockchain that would be suitable for a specific use case. The categories are *added value*, *data and process integrity*, *decentral network*, *values and rights*, and *automation* and for each category, the user can list all relevant aspects concerning this category in the canvas. Each aspect is then rated in the rating column with the rating high, medium or low, depending on how important this aspect is for the use case.

The first category, *added value*, is concerned with the difference the blockchain makes in the use case compared to the use case implementation without blockchain technology. Relevant aspects in this category are related to the tasks that are being supported by the blockchain, the processes that are being improved and how this improvement works, as well as which unique characteristics the use case gains by using blockchain as opposed to not using blockchain technology. Overall, this category assesses how the blockchain improves specific aspects of the use case implementation.

Data and process integrity, the second category, identifies which data needs to be managed securely. After having established in the use case identification framework that there is data being used in the use case that needs to be saved permanently and needs to be protected from data manipulation, it is now necessary to document which data exactly needs to be stored in the blockchain and which data can be stored in an external database or other legacy data warehouses, with references from the blockchain to the externally stored data. This category can thus be used to identify for which data it is crucial to be stored in the blockchain.

The third category explores the characteristic of a *decentral network*, which is one of the most prominent characteristics of a blockchain. Aspects in this category should document who, besides the user himself, are partners in the network.

Values and rights are the topic of the fourth category. Here, relevant aspects specify transactions that are being made on the blockchain for the use case. Blockchain in general is concerned with transferring values and rights between partners in the network, so practitioners should define which values or rights are being transferred in this use case.

The last category covers *automation* and describes which parts of the use case can be automated. If in the use case identification framework a certain potential for automation has already been identified, the canvas can then be used to specify which processes in the use case or which specific tasks in a process can be automated by using smart contracts specified in the blockchain.

After having collected all relevant aspects in these five categories, a better understanding of how a blockchain application would be structured for the specific use case should have been achieved. The canvas enables practitioners to clearly see the benefits of blockchain technology in combination with the specific use case as well as to understand the different components of the blockchain: Which data is stored on the blockchain each time a transaction is being made, who are the partners who exchange transactions with each other, which values and rights are transferred by the transactions and how the creation of transactions or whole processes can be automated.

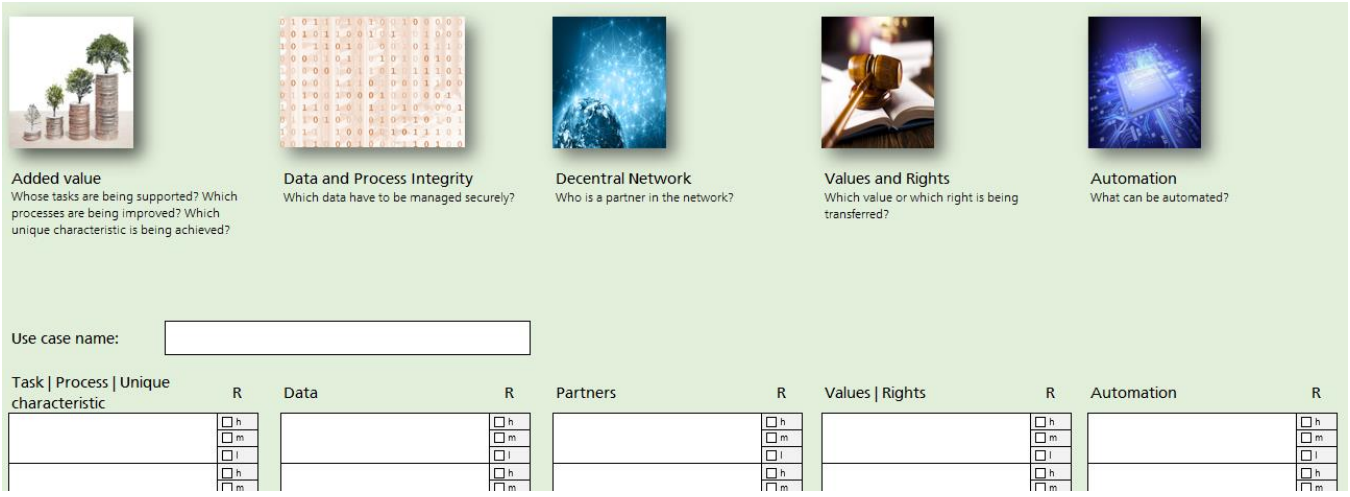


Figure 2: Use case canvas for blockchain

APPLYING FRAMEWORK AND CANVAS TO IDENTIFY AND STRUCTURE A SUITABLE BLOCKCHAIN USE CASE

Four use cases

In order to create a better understanding of how the framework and the canvas work in analyzing use cases, four use cases are going to be described and used as examples. The first use case describes the process using smart locks. Smart locks are a typical example for an IoT use case, since they enable a blockchain to manipulate physical objects [2]. If the owner of a smart lock decides that he wants to rent an object of his, like a house, a car or a bike, he can use the lock to secure his property and document the price and deposit amount for a specific rental period in a smart contract on a blockchain [8]. If someone wants to rent this object, this person can transfer the required price and deposit amount as a transaction on the blockchain, enforcing a smart contract which automatically sends a virtual key to the phone of the renter, enabling him to open the smart lock with his phone. The rental period can then be terminated again by another transaction in the blockchain, or if the rental period is over, which again enforces a smart contract which automatically returns the virtual key and re-transfers the deposit amount back to the renter. Smart locks are currently being developed as "Slocks" by the German company Slock.it.

The second use case involves smart grids, an opportunity for energy consumers to achieve independence from large energy providers. Inside a closed community, private, small-scale energy producers, for example households with photovoltaic systems, are enabled to sell their superfluous energy to neighbors who need energy [6]. This is realized by using a physical power distribution microgrid, connecting the different houses, and a virtual microgrid, connecting the smart meters that measure and monitor the energy generation and demand of every household. In a blockchain, smart contracts perform auctions by matching all buy (people who need energy) and sell (people who have energy to sell) offers and documenting every energy transaction on the blockchain. This way of trading energy can lower taxes and surcharges as well as providing nearly real-time access to energy whenever it is needed. This use case is currently realized as a prototype in Brooklyn, New York.

Foreign currency payments present the third use case. With regular bank transactions, transferring funds between currencies usually takes a long time and results in high fees [3]. Cryptocurrencies already enable people to make world-wide payments without a bank as an intermediary, endangering the business models of banks, as blockchain threatens to replace them. To prevent this from happing, banks can establish their own blockchain with banks from all over the world as partners on the blockchain, allowing them to transfer funds on the blockchain without any currency barriers [11]. With this use case, customers keep their level of comfort as they can still use their regular bank for international payments, but receive the benefits of fast transaction speed and lower fees.

As a last use case, the exchange of internal invoices will be investigated. In large companies with several subsidiaries, documentations of all transactions between the subsidiaries and the parent company are crucial. However, instead of using separate databases for this documentation, one central system should be used to make the transactions and the amount of money that is being accounted transparent. By using a blockchain, these transactions can be documented without the possibility of later manipulation while providing the necessary transparency. Any discrepancies between the accounts of two transaction partners are therefore being prevented.

Applying the framework

To demonstrate the application of the use case identification framework, the four use cases presented will be evaluated separately to analyze their suitability for blockchain technology. The completed framework for all use cases is displayed in Figure 3. Firstly, the use case of smart locks, which enable the automatic administration of virtual keys to objects, is going to be evaluated. Concerning the first category, *intermediary*, smart locks can be used to make it easier for landlords to rent out their properties, since they don't have to hand over a physical key. This "contactless" key transfer was previously not possible. Previous intermediaries could have been neighbors or friends of the object owner, however, those kinds of intermediaries would not necessarily have been trustworthy or always available. Therefore, since there hasn't been an intermediary comparable to a blockchain before, no intermediary is being replaced. However, the blockchain establishes a new intermediary and solves the problem of the lack of a trustworthy and available intermediary for flexible and temporary transactions. Since there was no previous intermediary, no business model is being replaced. Therefore, in the first category, the second scenario can be rated as being true. Regarding the second category, *data*, the immutability and transparency of the rental price and duration for each rental agreement is necessary, since it involves financial data and possibly impacts valuable properties of the owner. Financial data is always very sensitive, and the save handling of the virtual key is of great importance to the owner, thus the secure documentation of transactions can be described as being very important. Finally, evaluating the third category, *process*, the owner only needs to set a price and deposit amount once, and then automatically receives the payment every time his property gets rented. The tenant needs to trigger the renting process by transferring the rent and deposit amount on the blockchain, and then automatically receives the virtual key. Therefore, the process is not completely automatable since it requires input from the owner and the tenant, however, important aspects of the process can be automated. In conclusion, the use case does benefit from blockchain technology since one scenario of the first category is true and both the data and process categories are rated highly. However, since the process category does not receive the

highest ranking, and thus the use case cannot fully exploit the automation potential, it is not rated as benefitting very much from blockchain technology.

Smart grids as alternatives to the traditional system of large energy providers are the second use case being evaluated. Regarding the category *intermediary*, the blockchain-based microgrid replaces the traditional energy provider to save time in energy distribution by distributing energy only locally. Additionally, costs are being reduced by minimizing taxes and fees that are normally required by large providers. Thus, in this use case, blockchain replaces the large energy providers as intermediaries by saving time and reducing costs. In the traditional model, the energy provider acts as the intermediary between power plants and energy consumers and the consumers trust them: Thus, even with the previous intermediary, a trustworthy basis for transactions is available and the blockchain is not meant to establish one that was not existent before. Furthermore, although from the point of view of the energy provider, his business model can be replaced by the blockchain-based smart grid, in this example the use case is evaluated from the point of view of the energy consumer, thus there is no business model endangered. Looking at the second category, *data*, the data stored in the blockchain includes the amount of energy traded and the associated price for every transaction. To ensure a transparent documentation of every trade and to prevent discrepancies, the transaction data needs to be safe from manipulation and needs to be stored long-term in order to enable the verification of previous transactions. However, since the data are not necessarily as sensitive as for example financial or medical data, the importance of the category is not rated with the highest value. Regarding the third category *process* on the other hand, the automation potential can be rated with the highest possible potential. The use case does not require the energy consumers or producers to actively trade energy and therefore ensures the same comfort the consumers enjoy with traditional energy providers, while at the same time relieving the producers from doing any work. The smart meters installed at every house are able to measure energy deficiencies and surpluses and thus know exactly how much energy the network participant can sell or needs to buy. Smart contracts can then automate the energy transactions by arranging the auctions which match the buy and sell offers, thus, the whole process can be automated. In summary, the second use case also benefits from blockchain technology, since one scenario in the first category is rated as true, and both the second and third category are rated positively. Although the importance of secure data only receives the second highest value, the automation potential for this use case is extremely high. In this case, the process category influences the overall result more strongly than the data category, as the use case can make full use of smart contracts. Therefore, smart grids would benefit very much from blockchain technology.

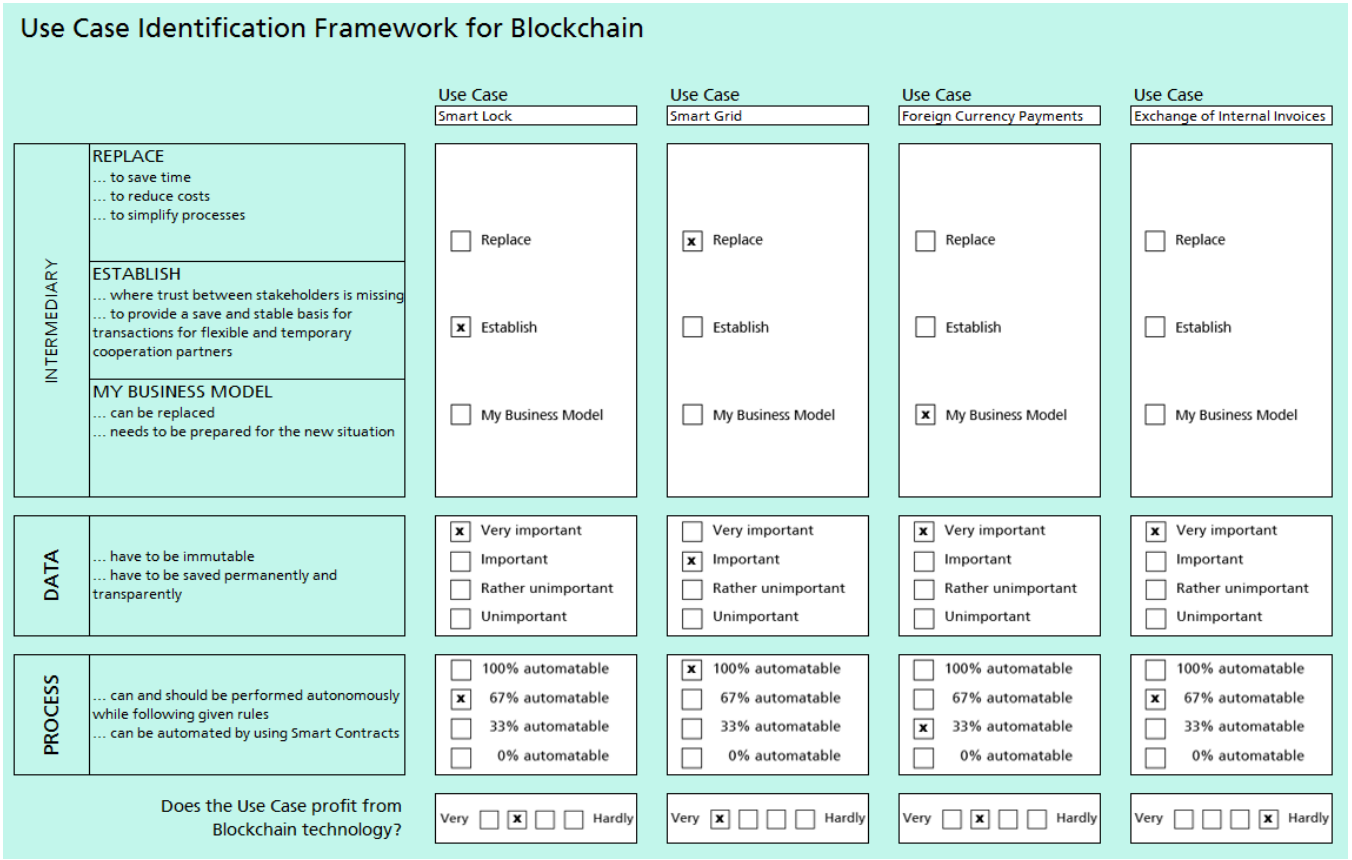


Figure 3: Use Case Identification Framework with examples

The third use case, foreign currency payments, presents another interesting example. Regarding the *intermediary*, the blockchain is being established as an intermediary between banks, however, the bank as the visible intermediary between two parties of a financial transaction does not get replaced, although time and cost benefits can be realized. Therefore, the blockchain does not replace an existent intermediary. Furthermore, since even without the blockchain an intermediary for international financial transactions is existent, no new intermediary is being established by using a blockchain. Nonetheless, for the banks, thus the intermediaries themselves, the blockchain provides a new, comfortable basis for flexible transactions. However, the motivation for this use case lies in the danger of the bank's business model being completely replaced by blockchain. Thus, to prevent two parties from making foreign currency transactions on a public blockchain without a bank as an intermediary, the bank itself needs to find a way to keep its customers, in this case with the private blockchain between banks. As a result, for this use case the third scenario in the first category is true. Concerning the *data* category, data from financial transactions are very sensitive, thus it is very important that those data are protected from manipulation. Additionally, banks need to be able to verify their payments, thus all payment transactions need to be transparent and made available permanently. Therefore, a documentation of transactions as it is possible with the blockchain is rated as being very important. Taking a look at the possible *automation*, the automation potential of this use case is not very high. Although the blockchain would enable banks to transfer money fast and cost-effectively within the blockchain, a complete process automation from the customer request to the transfer between banks to the transfer of the foreign bank to the foreign customer can probably not be completely automated. Nevertheless, a partial automation is possible and necessary for the fast processing of international payments. As a conclusion, although this use case would make international payments not as fast as direct transactions between two parties on a blockchain and although the process is not very automatable, the blockchain fulfills all requirements necessary for the handling of sensitive financial data. Additionally, it is a very important use case for banks to consider, since it enables them to compete with public blockchains and keep their customers. Therefore, banks would benefit from blockchain technology for this use case.

The fourth and last use case describes the exchange of internal invoices between subsidiaries and the parent company. In regard to the first category, *intermediary*, a blockchain would replace an already existing internal system of documenting transactions. However, it could not completely replace this system, since not all data can be saved on the blockchain and thus an external database to which the blockchain could reference would still be necessary. Furthermore, time and cost savings could probably not be realized since transaction data would still need to be manually entered and the process of doing this would not be simplified. Additionally, all subsidiaries would most likely still be required to maintain their own, separate accounts. Therefore, replacing the existing system would not be beneficial. Since without the blockchain as an intermediary, an internal system already exists, no new intermediary can be established. Also, the intermediary being used as well as the users of the system are both part of the same company, thus the possibility of a replacement is no threat to the company since it can always actively decide against it. Therefore, the potential replacement of an intermediary is no motivation for the company to realize a blockchain use case. Analyzing this first category, none of the three scenarios can be rated as true, which means that the use case would hardly benefit from blockchain technology and in general, the analysis could stop at this point. However, in this evaluation the second and third category are also being assessed to provide a complete analysis. The data being handled in this use case are sensitive financial data that need to be protected from manipulation. Additionally, transparency and long-term availability is necessary to avoid discrepancies in the records of different subsidiaries, thus, the importance of data immutability and transparency is very high. Regarding the process, the use of blockchain for the exchange of internal invoices could not offer significantly more automation potential than any other system. Although smart contracts can be used to automate parts of the process, this can also be done with different systems, thus, the automation potential is not a motivation for using blockchain technology in this process. In conclusion, since no scenario in the first category can be rated as true and additionally the automation potential by using smart contracts is not very high compared to other systems, this use case would hardly benefit from blockchain technology.

Applying the canvas

After all 4 use cases have been evaluated, 3 use cases are considered as being able to benefit from blockchain technology, while one use case is not suitable for using blockchain. In this next step, one use case will be analyzed more deeply by applying the use case canvas and evaluating the 5 categories explained in the canvas. For this purpose, the smart locks use case will be used.

Firstly, there are two main characteristics of the blockchain application in this use case that provide added value compared to any existent solution: The first benefit is the absence of any physical key: Instead, the virtual key can be transferred to and from users without the requirement of the physical presence of an intermediary. The second benefit is that the owner of the object is guaranteed to receive the rental money and deposit amount, since the smart contracts ensure that the virtual key is only transferred if the money has been paid. If the rental period is over, the key is automatically taken away from the phone of the one using the object. Thus, he cannot keep using the object unless he transfers more money to the owner on the blockchain. The immutability of smart contracts is one of the main values of blockchain

technology, once the rules of a smart contracts are set, they cannot be manipulated. Furthermore, since the absence of a physical key is a new and unique improvement compared to previous processes, and the owner does not need to worry about missed payments, both benefits have a high rating.

Secondly, four types of data need to be stored on the blockchain: The deposit amount, rental price and rental duration for which the rental price is valid are necessary for the smart contracts to correctly release the virtual keys. The owner of the object needs to determine these values and save them on the blockchain in a smart contract. Additionally, since the smart contract triggers the transfer of the virtual key to the user of the object as soon as he made a correct payment, data related to this virtual key needs to be saved on the blockchain, to make sure that the correct key gets transferred and to ensure the security of the key. All four types of data have a high ranking, since transparency and immutability of the transactions can only be guaranteed if all of these data are saved on the blockchain.

Thirdly, the blockchain is a decentralized network with several partners, so it needs to be determined who needs to have access to the blockchain. On the one hand, the owners of every object secured with a smart lock need to be able to access the blockchain to create smart contracts related to their smart lock. On the other hand, people who want to rent these objects also need access, to be able to make the necessary transactions to trigger the smart contracts. Since the concept only works with both network partners, both have a high ranking. This concept suggest the use of a public blockchain, to ensure the availability to every person interested in the secure renting of objects. However, for closed communities a private blockchain could also be suitable. An example for this kind of community could be carsharing, where users need to register to be able to rent a car whenever they need one, which could provide the car owners with an additional feeling of security.

The fourth aspect of the canvas is concerned with the values or rights that get transferred on the blockchain, i.e. the purpose of the transactions stored on the blockchain. In this use case, the right to use an object for a specific price and duration gets transferred to the one renting the object. This right can be received by transferring money on the blockchain, and will be revoked after the rental period is over, therefore, it also receives a high ranking.

Lastly, the automation aspect investigates which parts of the process can be automated with smart contracts. With smart locks, the key transfer in the beginning and the end of the rental period is completely automated after the transfer has been triggered by the payment of the price and deposit. Additionally, after the person renting the object has made the payment, the further handling of the money is also automated, meaning the deposit amount is kept on the blockchain and later automatically re-transferred after the virtual key has been returned. However, for the advantages the blockchain offers for the use case, the first automation potential is more important than the other one, therefore it receives a high ranking, while the importance of the second automation scenario is ranked as medium.

EVALUATION OF FRAMEWORK AND CANVAS

Both the use case identification framework and the use case canvas were developed in an iterative way. A first version of the canvas was designed to provide a tool for a better understanding of the requirements that a certain use case would have for a blockchain. This version already included the five categories. After some testing, it was realized that sometimes several possibilities exist to implement a suitable blockchain, therefore it was decided to add a rating to every aspect of every category to prioritize which aspects are must-haves (high rating), should-haves (medium rating) or could-haves (low rating). Additionally, the description of every category has been refined in several steps in order to achieve the best possible guidance on evaluating the use case. However, after several use cases had been utilized to test the canvas, it became clear that the canvas is most useful if a basic understanding of the usefulness of blockchain technology for a specific use case is already present. Therefore, to prepare use cases for the application of the canvas, a framework was developed to identify suitable use cases. This framework should use three main characteristics of blockchain as categories to find out if these characteristics are needed in the use case. In the first design, open spaces were used for each category and scenario to enable the user to describe how each aspect gets evaluated. However, after testing the framework, feedback emerged that with the open text boxes, it does not become clear what the user should include, and additionally redundancies between the application of the framework and the canvas can occur. Therefore, it was decided to adapt the framework and use multiple choice questions to better fit the need for a simple-to-understand, useful tool to evaluate blockchain suitability.

Evaluation of the framework and the canvas was conducted both internally and externally as soon as the first designs were finished. Internal testing included the research of possible use cases in several domains, which would then be used to apply the framework and the canvas. This internal testing lead to some revisions and redesigns. After the internal testing, an external workshop with a large German telecommunication provider which was organized by the authors of this paper was used to receive further feedback. The workshop participants made a list of 9 use cases which they encounter in their every-day work life, and then applied the use case identification framework to 7 of these use cases while working together in small groups. After 5 use cases have been identified as being generally suitable for blockchain technology, the use case canvas was applied. After this application, those use cases were discussed with all groups, to receive 4 prioritized use cases for blockchain applications. During the workshop, the participants were observed to find out how well the framework and the canvas suited their needs for the task they had to perform. Additionally, feedback was collected after the workshop.

This external evaluation was then used to make further adjustments to both the framework and the canvas.

DISCUSSION AND CONCLUSION

Value of Framework and Canvas

The identification framework and use case canvas presented in this paper will aid in guiding practitioners in exploring blockchain opportunities and deciding which parts of their business could benefit from blockchain technology. There are two possible scenarios in which framework and canvas would help practitioners in the beginnings of their blockchain implementation: In the first scenario, a practitioner might be interested in blockchain, but is still unsure which use cases of his company could benefit from the new technology. He can then use the framework to gain a first overview on possible use cases and to rate the suitability of each of those use cases to prioritize future implementations. In a second step, he can then use the canvas to further explore highly prioritized use cases and gain an understanding on how a blockchain for this use case could be used. As a result, he has a list of suitable use cases and knows how blockchain could improve those use cases. He can use this input to further guide the implementation of one or several blockchain applications.

In a second scenario, another practitioner might have a specific use case in mind that she would like to use blockchain for, but she has no real knowledge on how the implementation of blockchain technology for this use case would look like. To change this, she can use the canvas to explore the different aspects of a blockchain and use this as a first draft for a later implementation.

While both the framework and the canvas can be applied separately, they also work together very well. The framework explores the basic suitability for blockchain, while the canvas uses the results from this analysis to elaborate on the specific advantages and functions the blockchain enables for a certain use case.

Integration into the research environment

The proposed use case identification framework and use case canvas can be integrated into the research environment by comparing them to the related work. The Business Model Canvas [7] as well as the design thinking approach [12] have very exploratory characteristics. They guide the user through different categories by explaining them and inviting the users to apply them to their own use cases. While the Business Model Canvas has nine distinct categories which in combination provide the user with a complete understanding of the business model, the design thinking approach guides the user through an iterative refinement process resulting in a finished prototype. While both approaches are rather creative and open, the use case canvas described in this paper is comparable to the Business Model Canvas, as it also explains 5 distinct categories of blockchain application which need to be understood and applied separately to gain a complete understanding of the use case. Since the Business Model Canvas is now a broadly used tool in the business sector, it can be anticipated that the use case canvas could meet the same approval and usefulness.

A comparison with the framework proposed in the paper by Wüst and Gervais [14] is especially interesting since they discuss the same topic: Blockchain adoption. They exclusively use yes/no questions for their framework which could be answered independently, guiding the user through several questions to reach a conclusion regarding which type of blockchain, if any, should be used. In the framework proposed in this paper, the first category also consists of yes/no questions, however, they are structured differently, because the user needs to understand all three scenarios in order to decide which scenario fits his use case. For the evaluation of the other two categories, the user needs to decide on an answer based on a four-point scale, meaning he has to carefully evaluate the category before deciding on an answer. In general, the framework described in this paper is more open in that it does not simply offer a yes/no answer, but the possibility of prioritizing several possible blockchain use cases and understanding why each use case is more or less suitable for blockchain.

Evaluating the type of questions asked in both frameworks, the one proposed in this paper presents three main characteristics of blockchain (the role of an intermediary, data immutability and transparency, smart contracts) and aims at exploring how much the use case that is being analyzed needs those characteristics. Wüst and Gervais also explore the intermediary characteristic by asking questions about the amount of partners in the network, which directly leads to the necessity of an intermediary, whether or not they are known, the amount of trust that these partners enjoy and the existence of a trusted third party, the intermediary himself. They also mention the data category, but only ask a question about the requirement of long-term storage, without explaining data immutability and transparency. They also do not mention the possibility of process automation by using smart contracts. Instead, they give a recommendation about which type of blockchain would be the most suitable (permissionless, public or private permissioned). Furthermore, they use 6 questions in their framework but in most cases the user only answers a few of them, since at almost every decision point, one answer leads to the “Don’t use Blockchain” decision. This means that the decision process might be terminated after only one question, which is probably not suitable to provide a high-quality blockchain decision. Instead, it might be more suitable to first decide whether or not a blockchain should be used, then to understand what characteristics this blockchain should have and then use this knowledge and understanding to make a well-founded decision about the type of blockchain. The proposal of first using an identification framework and then a canvas supports this viewpoint and aims at providing comprehensive guidance on the application of blockchain technology to a specific use case.

Limitations

There are some limitations to the analysis framework presented in this paper. Firstly, other than the paper by Wüst and Gervais [14], no decision on which type of blockchain would be the most suitable is being made. However, a well-founded decision on this question can only be made after the information gathered in the canvas has been completed, and would therefore needed to be realized as a third step to the analysis framework. It is therefore outside the scope of the framework and canvas presented in this paper.

A second limitation would be that there are no clear rules regarding which results in the three categories of the identification framework lead to which decision regarding the final question of how much the use case would profit from blockchain technology. This ambiguity could however benefit the user, since it requires him to really understand the use case and its requirements.

Lastly, the framework does not include a category acknowledging those blockchain characteristics that could negatively influence the decision to adopt blockchain, for example the issue of data privacy which receives increasing importance with the new GDPR regulation. Adding such a category could be included in future refinements of the analysis framework.

Future Work

Future work founding on the results of this paper could include additional studies with practitioners to test the framework and canvas. Workshops with companies could be used to further improve the analysis framework and test its applicability for different industries.

Furthermore, another refinement of the identification framework could include a prioritization of the data and process categories as well as a coding of the answer possibilities with numeric values, which could then be used to calculate an overall score for each use case to aid in the decision making process.

Lastly, the two-step approach of the analysis framework could be extended to a three-step approach, with the third step including a transaction model of which transactions are being written on the blockchain and an overview on the smart contracts used for automation. This step could also include the selection of a suitable blockchain technology, e.g. if a public or a private blockchain would be most suitable. It should also be noticed, that the analysis framework is based on the currently available blockchain technology, and should be adjusted if technological advances are being made.

ACKNOWLEDGMENTS

We acknowledge the help of everyone who internally or externally tested and reviewed the analysis framework. Additionally, we thank the blockchain lab team, which helped with the initial discussions and further refinement of the framework and the canvas as well as the reviewers who provided valuable feedback.

REFERENCES

1. AbjCoin. 2017. AbjCoin Whitepaper. Retrieved May 17, 2018 from https://abjcoin.org/downloads/AbjCoinWhitepaper.pdf
2. Arshdeep Bahga and Vijay K. Madisetti. 2016. Blockchain platform for industrial Internet of Things. *Journal of Software Engineering and Applications* 9, 10, 533-546.
3. Der Treasurer. 2017. *Teil 1: Was blockchain für das Treasury leisten kann.* Retrieved from https://www.dertreasurer.de/.
4. Marco Iansiti and Karim R. Lakhani. 2017. The truth about blockchain. *Harvard Business Review 95*, 1, 118-127.
5. Wolfgang Gräther et al. 2018. Blockchain for Education: Lifelong Learning Passport. In: W. Prinz & P. Hoschka (Eds.), *Proceedings of the 1st ERCIM Blockchain Workshop 2018, Reports of the European Society for Socially Embedded Technologies* (ISSN 2510-2591), DOI: 10.18420/blockchain2018_07
6. Esther Mengelkamp et al. 2018. Designing microgrid energy markets: A case study: The Brooklyn Microgrid. *Applied Energy*, 210, 870-880.
7. Alexander Osterwalder and Yves Pigneur. 2010. *Business model generation: a handbook for visionaries, game changers, and challengers*. John Wiley & Sons.
8. Giulio Prisco. 2015. Slock.it to Introduce Smart Locks Linked to Smart Ethereum Contracts, Decentralize the Sharing Economy [Blog Post]. Retrieved from https://bitcoinmagazine.com/.
9. Wolfgang Prinz and Axel T. Schulte. (Eds.). 2018. BLOCKCHAIN AND SMART CONTRACTS: Technologies, research issues and applications. Retrieved from https://www.iuk.fraunhofer.de
10. Wolfgang Prinz. 2018. Blockchain and CSCW – Shall we care?. *Proceedings of 16th European Conference on Computer-Supported Cooperative Work - Exploratory Papers, Reports of the European Society for Socially Embedded Technologies* (ISSN 2510-2591), DOI: 10.18420/ecscw2018_13.
11. Ripple. 2018. *RippleNet Brochure*. Retrieved on March 8, 2018 from https://ripple.com/files/ripplenet_brochure.pdf
12. Standford DSchool. 2018. The Virtual Crash Course Playbook. Retrieved from: https://dschool.stanford.edu.
13. Don Tapscott and Alex Tapscott. 2016. *Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world*. Penguin.
14. Karl Wüst and Arthur Gervais. 2017. Do you need a Blockchain?. *IACR Cryptology ePrint Archive*, 2017, 375.

On Immutability of Blockchains

Esteban Landerreche
Centrum Wiskunde & Informatica
Amsterdam, Netherlands
esteban@cwi.nl

Marc Stevens
Centrum Wiskunde & Informatica
Amsterdam, Netherlands
marc.stevens@cwi.nl

ABSTRACT

Recently we presented a single-party cryptographic timestamping mechanism based on proof-of-sequential-work, which we proved secure in the universal composability framework [16]. This paper describes this construction and its security claims and uses it to construct a multi-party permissioned blockchain protocol and show that it achieves an immutability notion. Finally we discuss applications of this protocol, including unpermissioned blockchains, and how these may benefit.

ACM Classification Keywords

H.1.m. Models and Principles: Miscellaneous

Author Keywords

blockchain, cryptographic immutability, cryptographic timestamping, proof-of-sequential-work

INTRODUCTION

Blockchain immutability

The primary goal of a blockchain protocol is to achieve consensus over state between all participating nodes. This process is simplified by the append-only nature of the blockchain structure. Whenever new information needs to be added to the state, a new block is created and added to the chain. With the exception of temporary forks, which are resolved by erasing the latest blocks, the state is only updated by adding new blocks. After a certain point in time, a block is considered *immutable* as it becomes unfeasible for it to be erased or changed. This fact allows nodes to reach consensus on the whole state by agreeing only on the latest changes in the state.

Immutability in blockchains has mainly been studied as an element that allows for proof-of-work (PoW) consensus, but has not been studied widely as an individual characteristic. The first comprehensive paper in Bitcoin presented the abstraction of the Bitcoin blockchain and proved its security in a partially synchronous setting [9]. This paper was followed by numerous other papers investigating different aspects of Bitcoin. The same team followed up their work with a proof of Bitcoin with chains of variable difficulty in [10]. In [21], Bitcoin is proved secure in the asynchronous model and [1] presents a fully-composable treatment of Bitcoin.

All of these papers prove that consensus in Bitcoin (more broadly, proof-of-work consensus) works in part because of the immutability provided by the blockchain structured combined with proofs-of-work.

The proof-of-work based consensus of Bitcoin provides strong guarantees for immutability. Informally, adversaries that do not have a majority of the total hashing power invested in the proof-of-work can succeed at rewriting previous blocks in the network only with success probability that is exponentially small in the depth of rewritten blocks. Moreover, even against adversaries with a fraction $\alpha > 0.5$ of the total hashing power, the adversary is computationally restricted in how deep it can rewrite blocks in a certain amount of time, *i.e.* on average after time T the adversary can only successfully rewrite blocks in the network up to $T \cdot \alpha/(1-\alpha)$ time deep.

However, there exist various problems relating to the proof-of-work based consensus of Bitcoin. First, it requires a mining reward as an incentive to compute proofs-of-work and contribute to the security of Bitcoin [20], which is natural for cryptocurrencies, but is not a generic solution for all possible applications of blockchain. Second, the mining reward causes undesired effects threatening the aimed decentralization of Bitcoin, namely it causes centralization via mining pools (for miners to obtain frequent small rewards instead of very rarely obtaining large rewards) as well as via specialization (larger more efficient mining operations can reduce costs and increase profits) [12]. Third, Bitcoin does not seem to be sustainable given that the world-wide energy consumed in computing proofs-of-work for Bitcoin now exceeds that of entire countries[1].

In light of the above issues related to proof-of-work consensus, attention is turning towards other consensus mechanisms, such as proof-of-stake [3] and proof-of-space [8]. These non-PoW systems can ensure consensus for participants which are (almost) continuously connected to the protocol [7, 15, 19]. However, the consensus guarantees achieved in the protocol do not directly apply to parties that are mostly offline. For the same reasons, these systems are vulnerable to long-range attacks where parties can easily construct multiple blockchains that returning parties cannot distinguish from the real one [22]. Proof-of-work based systems do not have this problem, as creating valid blockchains requires an investment in computational work and time.

[1] https://digiconomist.net/bitcoin-energy-consumption

Landerreche, Esteban; Stevens, Marc (2018): On Immutability of Blockchains. In: W. Prinz & P. Hoschka (Eds.), Proceedings of the 1st ERCIM Blockchain Workshop 2018, Reports of the European Society for Socially Embedded Technologies

ERCIM-Blockchain 2018 May 08–09, 2018, Amsterdam, Netherlands

ISSN 2510-2591.

DOI: 10.18420/blockchain2018_04

When getting rid of proofs-of-work, the notion of immutability is maintained by punishing misbeahavior through incentives [3] or by constantly connecting to the the protocol in order to have an updated chain [11]. While this notion might be considered sufficient in certain contexts, it provides weaker guarantees than the PoW setting. We will call this notion *weak immutability* as it is not intrinsic and requires either incentives or monitoring to be achieved. In this paper we will seek immutability in a similar sense to the one found in PoW systems. We will say a blockchain has *strong immutability* if protection against malicious changes is obtained via a computationally hard problem. Note that, similar to a complexity class, strong immutability is parametrized by the difficulty of the computational problem, meaning that a blockchain is not necessarily more secure by having strong immutability. In particular, for a PoW blockchain to be secure it must maintain honest majority against the strongest possible adversary. In practice, this means that given one choice of a PoW function, at most one blockchain which uses that function can be considered secure.

Besides this distinction being theoretically interesting, we believe that it has consequences in the practical sense as well. Most of the users of a blockchain are not involved in the protocol as miners or maintainers of the chain. In practice, these parties connect sporadically to the network to read the blockchain in order to create new transactions or verify ones that they receive. Given this behavior, strong immutability is required.

Cryptographic timestamping

The immutability of blockchains is closely linked to the timestamping problem. In the seminal Bitcoin paper [20] the author presents the blockchain as a timestamp server which requires no trusted party. In particular, efficiently rewriting blockchains can be prevented by cryptographic timestamping.

Originally, Bayer, Haber and Stornetta considered timestamping digital documents as well as digital signatures [2, 14]. The security of proposed solutions relies on trusted parties and broadcasting blockchains where any malicious behaviour will be caught.

Other proposed solutions include encoding messages in blockchain transactions, in particular in Bitcoin transactions [5, 13], where security relies on the immutability of the underlying blockchain.

Recently we presented a single-party cryptographic timestamping mechanism based on proof-of-sequential-work, which we proved secure in the universal composability framework [16]. This paper will briefly review this construction and its security claims and apply it to achieve certain notions of immutability in the multi-party blockchain setting.

PRELIMINARIES

Time.

We consider a setting where time is essentially continuous, but it may be divided into intervals of time of a certain length which will be context-dependent. For instance, when a party computes a certain slow function at a rate of γ, then a time-step for this process will be $1/\gamma$ long, but for rounds of a (network) protocol this may be a pre-agreed length of time. Parties are equipped with synchronized clocks with at most an insignificant difference in time with respect to rounds of network protocols. We assume that timestamps can be described in bitstrings of length θ at a sufficient granularity.

Public-key signatures.

We assume a public-key infrastructure Σ for digital signatures that is existentially unforgeable. I.e., we assume that no attacker will ever be able to create any kind of forgery for a public-key where he does not know the corresponding private key. Given a public key pk message m and a signature s, any party may verify it by calling the function $\Sigma.\mathsf{verify}(pk, m, s)$.

Cryptographic hash function.

Let $\mathsf{H} : \{0,1\}^* \to \{0,1\}^\lambda$ be a collision-resistant cryptographic hash function, i.e. we assume no collisions will ever be found.

Merkle Trees.

Merkle Trees are balanced binary trees, where the ordered leaf nodes are each labeled with a bitstring, and where each non-leaf node has two child nodes and is labeled by the hash of its children's labels. The root hash of a Merkle Tree equals the label of the root node. Merkle Trees allow for short set membership proofs of length $O(log(N))$ for a set of size N. For convenience we define some interface functions that deal with Merkle Trees in some canonical deterministic way.

$\mathsf{MT.root}(T)$ computes the root hash h of the Merkle Tree for some ordered finite sequence $T \in (\{0,1\}^*)^*$ of bit strings and outputs $h \in \{0,1\}^\lambda$.

$\mathsf{MT.path}(T, v)$ outputs the Merkle path described as a sequence of strings $(x_0, \ldots, x_l)$ where $x_0 = v$, $x_l = \mathsf{MT.root}(T)$, $x_i \in \{0,1\}^\lambda$ and either $x_{i+1} = \mathsf{H}(x_i \| \mathsf{H}(x_{i-1}))$ or $x_{i+1} = \mathsf{H}(\mathsf{H}(x_{i-1}) \| x_i)$ for all $i > 0$.

$\mathsf{MT.verify}(P)$ given an input sequence $P = (x_0, \ldots, x_l)$ outputs accept if P is a valid Merkle path. It outputs reject otherwise.

With a slight abuse of notation we also use $\mathsf{MT.root}(T)$ recursively, *i.e.*, if one of the elements S of T is not a bitstring but a set or sequence, we use $\mathsf{MT.root}(S)$ as the bitstring representing S. E.g., if $T = (a, b, S)$ with bitstrings $a, b \in \{0,1\}^*$ and a set of bitstrings $S = \{c, d, e\}$, then $\mathsf{MT.root}(T) = \mathsf{MT.root}((a, b, \mathsf{MT.root}(S)))$. This similarly extends to $\mathsf{MT.path}(T, v)$, *e.g.*, where $v \in S$ in the previous example.

SINGLE-PARTY CRYPTOGRAPHIC TIMESTAMPING

Proofs of Sequential Work (PoSW).

Informally proofs of sequential work are proofs that some long and inherently sequential computation was performed, whereas any verifier can quickly verify the correctness of the proof. We use an extended notion of proof of sequential work to make it variable time, where one does not have to choose the strength in advance, but whose strength continuously increases with time spent computing it. More formally, we consider a non-interactive variable-time PoSW to be a triple of algorithms ($\mathsf{PoSW.gen}$, $\mathsf{PoSW.extend}$, $\mathsf{PoSW.verify}$) with security parameter μ and parameters $g, v \in \mathbb{N}$ as defined below.

$\mathsf{PoSW.gen}(x, s)$ is a slow cryptographic algorithm that for an input $x \in \{0,1\}^*$ and strength $s \in \mathbb{N}$ computes an output $(p, s) \in \{0,1\}^\mu \times \mathbb{N}$ in $s \cdot g$ parallel time steps.

$\mathsf{PoSW.extend}$ is a slow cryptographic algorithm that for inputs $x \in \{0,1\}^*$, $(p, s) = \mathsf{PoSW.gen}(x, s)$ and $s^* \in \mathbb{N}$ returns the output $(p^*, s+s^*)$, where $(p^*, s+s^*) = \mathsf{PoSW.gen}(x, s+s^*)$, in $s^* \cdot g$ parallel time steps.

$\mathsf{PoSW.verify}(x, p, s)$ is a fast cryptographic algorithm that for inputs $x \in \{0,1\}^*$, $p \in \{0,1\}^\mu$, and $s \in \mathbb{N}$ outputs accept if $(p, s) = \mathsf{PoSW.gen}(x, s)$, and reject otherwise, in at most $s \cdot v$ time steps.

We require perfect **correctness**:

$$\mathsf{PoSW.verify}(x, \mathsf{PoSW.gen}(x, s)) = \mathsf{accept}$$

for all $x \in \{0,1\}^*$ and $s \in \mathbb{N}$. The PoSW is called **secure** if no efficient adversary given a challenge x with sufficient min-entropy can compute values (s, p) in less than $s \cdot g$ parallel time steps for which $\mathsf{PoSW.verify}(x, p, s) = \mathsf{accept}$ with non-negligible probability. The **usability** of the PoSW is the factor g/v by which verification is faster than generation of the proof.

A candidate construction that satisfies this notion is the Sloth construction by Lenstra and Wesolowski [17] that iterates modular square root and (keyed) binary permutation functions.

In this work we will assume that every party and the adversary have access to certain computational resources (a CPU running at some clock speed) or some specific optimizations which implies that they each can compute proofs of sequential work at a certain (potentially distinct) rate γ. So for every party we model their capability to compute PoSW as a slow oracle $\mathcal{F}_\gamma^{\mathsf{PoSW}}$ as defined in Algorithm 1 that beneath interacts with a global random oracle PoSW.

Algorithm 1: Oracle $\mathcal{F}_\gamma^{\mathsf{PoSW}}$

Setting: The oracle is parametrized by a PoSW-rate $\gamma > 0$. Let $\mathsf{PoSW} : \{0,1\}^* \times \mathbb{N} \to \{0,1\}^\mu$ be a global random oracle each oracle instance has access to. The oracle also has access to a global clock clock (to exactly measure time elapsed computing the proof of sequential work).

The oracle functions as follows:

Let $Q := \emptyset$ be the (initially empty) query log;
On input (start, x) *at time* t*:*
  Update $Q \leftarrow Q \cup \{(x, t)\}$;
On input (output, x) *at time* t_o*:*
  Let t_i be the earliest time such that $(x, t_i) \in Q$, return $\perp$ if there is no such t_i;
  Let $s := \lceil (t_o - t_i) \cdot \gamma \rceil$ be the strength of the resulting proof and $p := \mathsf{PoSW}(x, s)$;
  Return (p, s) at time $t_i + s/\gamma =: t_o + \varepsilon$, with $0 \le \varepsilon < 1/\gamma$;
On input $(\mathsf{verify}, x, p, s)$*:*
  Return accept if $\mathsf{PoSW}(x, s) = p$ and reject otherwise;

The SingleLipwig protocol

In this section we present our recent single-party SingleLipwig protocol [16] and its security claims.

Consider party $\mathcal{P}$ with public key pk that can compute PoSW at rate $\gamma > 0$, who thus has access to a functionality $\mathcal{F}_\gamma^{\mathsf{PoSW}}$.

Party $\mathcal{P}$ will run protocol SingleLipwig as described in Algorithm 2.

He will wait till he obtains some msg to output, at which point he will create his next $block$ containing a hash pointer to the previous block, the msg, a PoSW and a signature.

The output of party $\mathcal{P}$ running SingleLipwig can be verified by any (γ, ε)-SingleLipwig-verifier as described in Algorithm 3, where γ is the PoSW-rate γ of $\mathcal{P}$ and $\varepsilon \ge 0$ is the time $\mathcal{P}$ requires to execute steps 3–5 of SingleLipwig. We call ε the *PoSW-interrupt time* of $\mathcal{P}$.

Algorithm 2: SingleLipwig

Setting: Assume that party $\mathcal{P}$ with public key pk has access to $\mathcal{F}_\gamma^{\mathsf{PoSW}}$ for PoSW-rate $\gamma > 0$.

Initialize $prev := \mathsf{H}(pk)$, $t_0 := \mathsf{clock}()$;
Send $(\mathsf{start}, t_0 \| prev)$ to $\mathcal{F}_\gamma^{\mathsf{PoSW}}$;
for $i = 0, \ldots$ **do**
  Wait till message (record, msg_i);
  Retrieve (p_i, s_i) by querying $\mathcal{F}_\gamma^{\mathsf{PoSW}}(\mathsf{output}, t_i \| prev)$;
  Create signature sig_i for $\mathsf{MT.root}(data_i)$, where $data_i = (\mathsf{H}(pk), i, prev, ((t_i \| prev), (p_i \| s_i)), msg_i)$;
  Set $block_i \leftarrow (data_i, sig_i)$, $prev \leftarrow \mathsf{MT.root}(block_i)$, and $t_{i+1} \leftarrow \mathsf{clock}()$;
  Send $(\mathsf{start}, t_{i+1} \| prev)$ to $\mathcal{F}_\gamma^{\mathsf{PoSW}}$;
  Output $(block_i, t_{i+1})$;
end

Algorithm 3: (γ, ε)-SingleLipwig-verifier

Input: A sequence $C = ((block_0, t_1), \ldots, (block_l, t_{l+1}))$ for public-key pk output by SingleLipwig
Output: $\perp$ or a sequence $(msg_0, age_0), \ldots, (msg_l, age_l)$

Output $\perp$ if not all $block_i$ are correctly signed: $block_i = (data_i, sig_i)$ where sig_i is a valid signature of $\mathsf{MT.root}(data_i)$ for public key pk;
Output $\perp$ if not all $data_i$ are correctly formed: $data_i = (\mathsf{H}(pk), i, prev_i, posw_i, msg_i)$ where $prev_0 = pk$, $prev_i = \mathsf{MT.root}(block_{i-1})$, and $posw_i = ((t_i \| prev_i), (p_i, s_i))$ for some msg_i, p_i, s_i;
Output $\perp$ if not all $posw_i$ are correct PoSW: $posw_i = ((t_i \| prev_i), (p_i, s_i))$ where $p_i = \mathsf{PoSW}(t_i \| prev_i, s_i)$;
Output $\perp$ if not all PoSW are strong enough: $s_i \ge (t_{i+1} - t_i - \varepsilon) \cdot \gamma$;
Output $\{(msg_i, age_i = t_{l+1} - t_{i+1}) \mid 0 \le i \le l\}$;

The following properties of the algorithm SingleLipwig together with a (γ, ε)-SingleLipwig-verifier have been proven in the universal composability framework for sufficiently large security parameters κ, λ and μ.

THEOREM 3.1 ([16]). *(1) The output of an honest $\mathcal{P}$ running the algorithm* SingleLipwig *with PoSW-rate γ and PoSW-interrupt time ε will be accepted by any (γ, ε)-*SingleLipwig*-verifier.*

*(2) When a honest $\mathcal{P}$ is corrupted at time T_{corr} by an adversary with PoSW-rate $\gamma \cdot \alpha$ and PoSW-interrupt time ε/α with $\alpha \geq 1$, any sequence $C = ((block_0, t_1), \ldots, (block_l, t_{l+1}))$ output by the adversary at time T_{output} that is accepted by any (γ, ε)-*SingleLipwig*-verifier satisfies the following properties except with negligible probability:*

1. *Let $A = T_{output} - T_{corr}$ be the time passed since corruption, then either C contains all $block_i$ created by the honest $\mathcal{P}$ at least $A \cdot \alpha$ time ago or none.*
2. *Any $block_i$ created by the adversary has claimed age $age_i = t_{l+1} - t_{i+1}$ (cf. Algorithm 3) at most $A \cdot \alpha$.*
3. *Any $block_i$ created by the adversary at time T_i has claimed age age_i at most $(T_{output} - T_i) \cdot \alpha$.*

PERMISSIONED BLOCKCHAINS

Permissioned blockchains could be a solution to many practical problems faced by governments and enterprises. Cooperation between mutually mistrusting entities can only emerge when a party has assurances that power will not be abused. In practice, these cooperations can only happen through trust on a third party (most likely the government). These processes take considerable time, energy and money, elevating costs and affecting efficiency. Blockchains can solve these issues by providing a trusted ledger which provides the necessary assurances.

Permissioned blockchains should provide a ledger that any participant can add to according to preset rules. Distributed consensus means that no single entity has enough power to arbitrarily modify the ledger. There exist multiple efficient protocols for consensus in a permissioned network. Permissioned consensus requires a trust in identities (through a PKI or otherwise), trust that enough parties will not collude. Distributed consensus has existed for a long time, so what exactly do blockchains offer beyond distributed consensus?

Compared to classical consensus protocols, blockchains offer two primary advantages: the ability to function in an unknown network and cryptographic immutability. The former is irrelevant in the context of a permissioned network, especially because this comes at a large efficiency (and economic) cost. Therefore, the main feature of blockchains in this setting is immutability. Blockchains prevent any entity from arbitrarily rewriting anything written in it. The blockchain structure ensures that any change propagates throughout the whole chain. Changing a block becomes equivalent to recreating the entire succeeding section of the chain. If creating each block in the blockchain is *hard* enough, we can consider it immutable.

Immutability is generally studied through its relationship with consensus. In non-PoW blockchains, it is shown that the adversary cannot disrupt consensus on the recent state of the chain. However, these systems are vulnerable to long-range attacks, as it is easy to emulate the execution of the protocol [22]. If a sufficient number of identities become adversarial at some point in time then in principle they can efficiently rewrite the entire ledger almost immediately. In a permissioned network consisting of a known (possibly small) number of known parties, the risk of an adversary corrupting enough parties is especially relevant (particularly if all parties use the same software, which should be expected) [23].

Bitcoin's PoW paradigm avoids this problem because rewriting a chain implies re-doing all the necessary work. Beyond hash collisions, rewriting a block is equivalent to disrupting consensus. In PoW-based systems, even if consensus breaks down because of an adversarial majority, the blockchain still offers some guarantees against rewriting. Alternatively, if a blockchain does not use proofs-of-work, an adversarial majority can, in principle, efficiently and arbitrarily rewrite everything whenever it gains control of a majority of the network. We postulate that this resistance to modification even with a majority adversarial corruption is the strongest advantage of a blockchain in a permissioned setting.

Unfortunately, proof-of-work consensus does not provide these guarantees in a permissioned setting. Security in proof-of-work blockchains comes from the amount of computational power that the adversary has access to, in comparison with the network. In large permissionless networks like Bitcoin and other cryptocurrencies, it is unfeasible for an adversary to have access to more computational power than the rest of the network combined. Therefore, a new blockchain must have a network with enough computational power to be resilient to attacks from the networks that maintain these cryptocurencies. For example, this is what causes merged-mining sidechains to be vulnerable to rewriting attacks [6].

Sensing this attack vector, multiple cryptocurrencies have adopted different proof-of-work functions that weaken these attackers. Bitcoin mining today relies on application-specific integrated circuits (ASICs); it is possible to choose a function where this specialized hardware provides little advantage. Multiple cryptocurrencies have chosen this road in order to sidestep this problem [4]. This solution does not work in a permissioned setting, as these networks are likely to be small. A malicious party could easily get access to enough computational power to rival that in the network. The only way to prevent this is to invest considerable amounts of computational power in the network, which is not cost-effective. As permissioned networks do not require proofs-of-work to achieve consensus and do not achieve any immutability from them, we believe that proofs-of-work have no place in the permissioned setting, but their role can be filled by proofs-of-sequential-work.

A Simple Protocol

Using proofs-of-sequential-work prevents arbitrary rewriting the content inside of a blockchain, even if this blockchain is maintained by only one party. However, this is not enough to ensure immutability and provide the trust guarantees that are expected from a blockchain. Proofs-of-sequential-work prevent against rewriting something in the history, but cannot prevent real-time forking. The creator of the chain could secretly maintain several forks of its blockchain and choose

which one to present depending on the situation. A corrupt agent cannot arbitrarily rewrite a chain but can still make modifications to it in certain conditions. Like any other blockchain protocol, we require multiple parties to cooperate to achieve a stronger immutability. What is more interesting is that a single honest party can prove whether or not a blockchain has been rewritten in a definitive way.

In our previous model there was only one party, which meant that the time spent between instances of the proof of sequential work (denoted by ε) was as small as possible.

We are interested in minimizing this time, but run into a problem in the multi-party setting where achieving consensus on the next block takes significant time. If we compute our PoSW over blocks in the chain, the time between a block being proposed and it being confirmed is time where PoSW cannot be computed. While getting a larger γ requires an investment to acquire a faster processor, an adversary can gain an advantage in ε by creating the block on its own. Even if in practice this process is almost instantaneous, it forces us to allow *weaker* PoSWs; proofs that do not span all the time needed to create the block. Computing PoSW over the blocks of the chain gives the adversary a strong advantage for rewriting blocks. Instead, we propose each participant will maintain their own personal PoSW chain and together maintain a ledger chain that interacts with the personal chains.

We will present a simplified model with favorable network conditions and access to an arbitrary consensus mechanism that has certain desirable properties as described in Algorithm 4. Our goal will not be to show that consensus is achieved but instead to show the immutability properties provided by the blockchain. Besides the advantages provided by the PoSWs to SingleLipwig, this protocol will additionally benefit from additional guarantees provided by signatures of each block. In this simplified construction, we will assume the existence of some protocol Consensus that achieves certain minimum security properties which determines the block created at each round of the protocol.

The protocol MultiLipwig, presented in Algorithm 5, will interact with other parties using Consensus to determine the next ledger block and locally use SingleLipwig to record its copy of the ledger block. So each party will record two distinct chains: its personal chain and the ledger chain. Blocks in the ledger chain will be represented by $block_i^L$, while blocks in party $\mathcal{P}_j$'s personal chain will be represented by $block_i^j$.

In order to link the personal blockchains with the ledger chain, we will add Merkle pointers to the blocks. For simplicity, we will act as if the participants simply copy the entire blocks. Each personal block will contain the ledger block created that round. The new ledger block output by Consensus will contain the personal blocks created in the previous round[2], in order to provide the ledger chain with the security provided by the proofs-of-sequential-work. While the PoSW are computed over the personal chains and not the ledger chain, every personal block contains the ledger block. Therefore, we can create a Merkle path between the root of the ledger block up to the root of a personal block, which is the input to the PoSW. Because we assume that no collisions can be found for the hash function underlying our Merkle trees, the query to $\mathcal{F}_\gamma^{\mathsf{PoSW}}$ could only come after the ledger block was created. This allows the ledger chain to *borrow* the proof-of-sequential-work from the personal chains.

[2]In practice, personal chains will contain a simple Merkle root while the ledger chain will contain Merkle paths to the PoSW, the pointer to the ledger block and the signature.

Algorithm 4: Consensus for n parties

Setting: Let $\mathcal{P}_j$ running this protocol in a network of n parties $\mathcal{P}_1, \ldots, \mathcal{P}_n$ with corresponding public keys $pk_1, \ldots, pk_n$. The desired consensus protocol is parametrized by a minimum number $n/2 < Thr \leq n$ of contributing parties and a maximum run time T.

Input: Ledger chain $C^L = (block_1^L, \ldots, block_{i-1}^L)$ and SingleLipwig chain $C^j = ((block_0^j, t_1^j), \ldots, (block_{i-1}^j, t_i^j))$.

Result:
The protocol runs within time T and either outputs $(block_i^L, sig_{i,j}^L)$ or abort. The new block $block_i^L$ is of the following form:

$$\left(pkH, i, prev_i, \{block_i^k \mid k \in P_i\}, msg_i\right)$$

where $pkH = \mathsf{MT.root}(pk_1, \ldots, pk_n)$, $prev_1 = pkH$ for $i = 1$ or $prev_i = \mathsf{MT.root}(block_{i-1}^L)$ otherwise, $|P_i| \geq Thr$ and $block_i^k$ is the last block in the SingleLipwig chain C^k of $\mathcal{P}_k$. The content msg_i is also decided by the protocol, however the format and validity of msg_i is application specific and therefore left unspecified here.
If there are at least *Thr* honest parties then all honest parties receive the same output $block_i^L$ and every honest party $\mathcal{P}_k$ receives a set $sig_{i,k}^L$ of signatures on $block_i^L$ of at least *Thr* distinct parties.
In all cases, even if all other parties are corrupt, if $\mathcal{P}_j$ is honest then it signed at most one candidate $\widetilde{block_i^L}$ that must be valid in the above form including a valid $\widetilde{msg_i}$.

The Consensus algorithm must wait until the threshold of participants submit their blocks. We require the threshold to be at more than half the players to allow for external verification. After that, it waits for an random amount of time before outputting a block to all participants, containing all the blocks for the round that it received. Note that if the block is not correctly formatted, does not contain a pointer to the latest block Consensus created or the signature is not valid, then it is not added to the block.

As we did for SingleLipwig, we will define a (γ, ε)-MultiLipwig-verifier to verify chains output by MultiLipwig in Algorithm 6. This verifier will certify the ledger chain (found encoded in the personal chain) and call the SingleLipwig-verifier on the personal chain.

Theorem 4.1. *(1) If there are at least Thr honest parties, then the output of any $\mathcal{P}_j$ running* MultiLipwig *with PoSW-*

Algorithm 5: MultiLipwig

Setting: Let $\mathcal{P}_j$ running this algorithm in a network of n parties $\mathcal{P}_1, \ldots, \mathcal{P}_n$ with corresponding public keys $pk_1, \ldots, pk_n$. Assume each has the same PoSW-rate γ and PoSW-interrupt time ε.

$\mathcal{P}_j$ starts running SingleLipwig;
Retrieve $(block_0^j, t_1^j) \leftarrow$ SingleLipwig(record, $\perp$);
Set $C^j \leftarrow ((block_0^j, t_1^j))$, $C^L = \varnothing$;
for $i = 1, \ldots$ **do**
  Call Consensus(C^L, C^j) to obtain $(block_i^L, sig_{i,j}^L)$;
  Call SingleLipwig(record, $(block_i^L, sig_{i,j}^L)$) to obtain $(block_i^j, t_{i+1}^j)$;
  Append $block_i^L$ to C^L;
  Append $(block_i^j, t_{i+1}^j)$ to C^j;
  Output $(block_i^j, t_{i+1}^j)$;
end

Algorithm 6: (γ, ε)-MultiLipwig-verifier

Input: An MultiLipwig sequence $C^j = ((block_0^j, t_1^j), \ldots, (block_{l+1}^j, t_{l+2}^j))$
Output: $\perp$ or a sequence $(msg_0, age_0), \ldots, (msg_l, age_l)$

Call (γ, ε)-SingleLipwig-verifier for C^j;
if *Verifier outputs* $(((block_1^L, sig_{1,j}^L), age_1), \ldots, ((block_{l+1}^L, sig_{l+1,j}^L), age_{l+1}))$ **then**
  Output $\perp$ if not all $block_i^L$ are correctly formed: $block_i^L = \left(pkH, i, prev_i, \{B_i^k \mid k \in P_i\}, msg_i\right)$ where $pkH = \mathsf{MT.root}(pk_1, \ldots, pk_n)$, $prev_1 = pkH$, $prev_i + 1 = \mathsf{MT.root}(block_i^L)$, $|P_i| \geq Thr$ and **if** $j \in P_i$ **then** $B_i^j = block_{i-1}^j$;
  Output $\perp$ if $|sig_{i,j}^L| < Thr$ for some i;
  Output $\perp$ if there are invalid signatures in some $sig_{i,j}^L$ There exists $s \in |sig_{i,j}^L|$ for some i such that for all $k \in [n]$, $\Sigma.\mathsf{verify}(pk_k, block_i^L, s) = \mathsf{reject}$;
  Output $(((msg_1), age_1), \ldots, (msg_{l+1}), age_{l+1}))$
else
  Output $\perp$
end

*rate γ and PoSW-interrupt time ε will be accepted by any (γ, ε)-*MultiLipwig*-verifier. Furthermore, the output of any adversarial party that is accepted by any (γ, ε)-*MultiLipwig*-verifier contains the same ledger chain C^L as the output of* all *honest parties.*

*(2) If there are less than Thr honest parties but also less than Thr corrupted parties, then the output of any adversarial party that is accepted by any (γ, ε)-*MultiLipwig*-verifier contains a ledger chain C_i^L where each $block_i^L$ is verified and signed by* some *honest party. No consensus is guaranteed.*

*(3) Suppose that Thr parties fall under adversarial control at time T_{corr}. Given an adversary with PoSW-rate $\gamma \cdot \alpha$ and PoSW-interrupt time ε/α with $\alpha \geq 1$, any sequence $C = ((block_0, t_1), \ldots, (block_l, t_{l+1}))$ output by the adversary at time T_{output} that is accepted by any (γ, ε)-*MultiLipwig*-verifier satisfies the following properties except with negligible probability:*

1. *Let $A = T_{output} - T_{corr}$ be the time passed since corruption, then every $block_i^L$ in C of at least $A \cdot \alpha$ time ago is verified and signed by* some *$\mathcal{P}$ that was honest at that point in time.*
2. *Any $block_i^L$ created by the adversary has claimed age $age_i = t_{l+1} - t_{i+1}$ at most $A \cdot \alpha$.*
3. *Any $block_i^L$ created by the adversary at time T_i has claimed age age_i at most $(T_{output} - T_i) \cdot \alpha$.*

Proof. Part (1) follows from correctness of consensus and SingleLipwig. Part (2) follows from the fact at least *Thr* signatures are required by the MultiLipwig-verifier, requiring participation of at least one honest party. Part (3) follows from Theorem 3.1 and part (2). □

Note that while Consensus requires an honest *Thr*-majority to ensure agreement, without a corrupted *Thr*-majority the adversary still cannot create new ledger blocks without the verification and signature by some honest party. Thus the adversary must actually control a *Thr*-majority of parties to start rewriting blocks without any verification by an honest party. But most importantly, even if the adversary completely corrupts the entire permissioned network then still it is limited by how many blocks it can rewrite over a period of time that would be accepted by any external verifier.

Our construction of the verifier assumes that every ledger block contains a certificate of correctness (the set of signatures) which can be forged the moment the adversary gains control of enough parties. Different consensus mechanisms may have different certificates of correctness, like the block hash in proof-of-work systems. The verifier can be modified for different consensus mechanisms, but in that case the adversary must not only corrupt enough parties but also be able to create valid certificates. The need for certificates can be avoided by requiring the verifier to certify that the personal blocks contained in each ledger block are correctly constructed and contain both a signature and a pointer to the previous ledger block. Signatures in personal blocks can be considered a commitment to the ledger block that they point to. The alternative construction implies a higher verification cost in exchange for a possibly cheaper consensus mechanism.

APPLICATIONS

We have extended our single party protocol to a permissioned blockchain. We now briefly present possible applications for our constructions, including some outside the classical realm of blockchains.

Semi-private Databases

Advantages of blockchain technology can be also applied to a centralized setting, in cases where a party is partially trusted. Assume that an entity (for example a government) maintains a database of important records (for example, a land registry) for which temporality is important. This entity is interested in maintaining the entirety of this database private, while making entries available to the right parties. In the case of a land registry, if a party wants to know the particular status of a piece of land, they can request this information and get a response. As the database is not public, this party has no direct way to verify that the information is actually in the registry. There are ways to prove that the record is in the database, but it is harder to prove *when* the record was added.

In parts of Mexico City there are residential areas where land must only be used for housing unless the land was used for something other than a house from before this law was passed. In order to use these residences as commercial units, bureaucrats are bribed to forge documents that state that the property has always been used for commercial purposes [18] . Having a history as a commercial unit allows the owner to "legally" rent it as a commercial space. If there was a PoSW-secured database containing these records, it would be possible to prove that the forged documents are not as old as they claim to be, preventing this instance of corruption.

Permissionless Blockchains

On the other side of the spectrum, a permissionless blockchain can also benefit from *immutability blockchains* created by SingleLipwig or MultiLipwig, in particular ones without proofs-of-work.

A non-intrusive method is to use such an immutability blockchain that collects hashes of blocks from the permissionless blockchain. It can act as a sort of *immutability beacon* providing time-lock guarantees about the permissionless chain to any verifier interacting with the immutability blockchain, for instance during bootstrapping.

In a more intrusive manner, any permissionless blockchain can *borrow* the immutability from the permissioned blockchain by embedding its blocks into the permissionless blockchain, much in the same way that the permissioned ledger chain *borrows* the immutability from the personal blockchains [16].

In both cases, this requires minimal commitment and trust of the set of parties executing the immutability blockchain. As long as this immutability blockchain continues to include hashes of blocks and correctly compute proofs-of-sequential work, both easily verifiable, any party can benefit from the proofs of age provided by it. Moreover, it is easy to start a new trusted permissioned group that does the same and that can actually use the entire immutability chain of any previous group up to that point in time[3], making it easy to switch between immutability blockchains if desired.

In both cases, it can also aid light-weight clients and in fast bootstrapping, since the immutability chain is significantly smaller in data size than the permissionless chain and is quickly verified. Then light-weight and/or bootstrapping parties can more easily rely on the validity of the history and focus on verification of recent blocks.

REFERENCES

1. Christian Badertscher, Ueli Maurer, Daniel Tschudi, and Vassilis Zikas. 2017. Bitcoin as a Transaction Ledger: A Composable Treatment.. In *Annual International Cryptology Conference*. Springer, 324–356.
2. Dave Bayer, Stuart Haber, and W Scott Stornetta. 1993. Improving the efficiency and reliability of digital time-stamping. *Sequences II: Methods in Communication, Security and Computer Science* (1993), 329–334.
3. Iddo Bentov, Ariel Gabizon, and Alex Mizrahi. 2016. Cryptocurrencies without proof of work. In *International Conference on Financial Cryptography and Data Security*. Springer, 142–157.
4. Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A Kroll, and Edward W Felten. 2015. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *Security and Privacy (SP), 2015 IEEE Symposium on*. IEEE, 104–121.
5. Jeremy Clark and Aleksander Essex. 2012. CommitCoin: Carbon Dating Commitments with Bitcoin - (Short Paper). In *Financial Cryptography and Data Security - 16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012, Revised Selected Papers (Lecture Notes in Computer Science)*, Angelos D. Keromytis (Ed.), Vol. 7397. Springer, 390–398. DOI: http://dx.doi.org/10.1007/978-3-642-32946-3_28
6. Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, and others. 2016. On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*. Springer, 106–125.
7. Phil Daian, Rafael Pass, and Elaine Shi. 2016. Snow White: Provably Secure Proofs of Stake. Cryptology ePrint Archive, Report 2016/919. (2016). http://eprint.iacr.org/2016/919.
8. Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. 2015. Proofs of Space. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II (Lecture Notes in Computer Science)*, Rosario Gennaro and Matthew Robshaw (Eds.), Vol. 9216. Springer, 585–605. DOI: http://dx.doi.org/10.1007/978-3-662-48000-7_29

[3]This requires only a minimal change to MultiLipwig and its verifier where the genesis ledger block includes this borrowed chain which has to be verified in a recursive manner.

9. Juan Garay, Aggelos Kiayias, and Nikos Leonardos. 2014. The Bitcoin Backbone Protocol: Analysis and Applications. Cryptology ePrint Archive, Report 2014/765. (2014). http://eprint.iacr.org/2014/765.
10. Juan Garay, Aggelos Kiayias, and Nikos Leonardos. 2017. The bitcoin backbone protocol with chains of variable difficulty. In *Annual International Cryptology Conference*. Springer, 291–323.
11. Peter Gaži, Aggelos Kiayias, and Alexander Russell. 2018. Stake-Bleeding Attacks on Proof-of-Stake Blockchains. Cryptology ePrint Archive, Report 2018/248. (2018). https://eprint.iacr.org/2018/248.
12. Adem Efe Gencer, Soumya Basu, Ittay Eyal, Robbert van Renesse, and Emin Gün Sirer. 2018. Decentralization in Bitcoin and Ethereum Networks. *arXiv preprint arXiv:1801.03998* (2018).
13. Bela Gipp, Norman Meuschke, and André Gernandt. 2015. Trusted Timestamping using the Crypto Currency Bitcoin. *iConference 2015 Proceedings* (2015).
14. Stuart Haber and W. Scott Stornetta. 1991. How to time-stamp a digital document. *Journal of Cryptology* 3, 2 (01 Jan 1991), 99–111. DOI: http://dx.doi.org/10.1007/BF00196791
15. Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. 2017. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*. Springer, 357–388.
16. Esteban Landerreche, Marc Stevens, and Christian Schaffner. 2018. Cryptographic timestamping through sequential work. Preprint. (2018). https://marc-stevens.nl/research/papers/preprint_LSS18_Cryptographic-timestamping.pdf.
17. Arjen K. Lenstra and Benjamin Wesolowski. 2015. A random zoo: sloth, unicorn, and trx. Cryptology ePrint Archive, Report 2015/366. (2015). http://eprint.iacr.org/2015/366.
18. Patricia López Moreno. 2017. Irrregularidades en los Procesos y Autorizaciones de las Manifestaciones de Construcción. (February 2017). http://www.impunidadcero.org/uploads/app/articulo/23/archivo/1486526151A57.pdf Impunidad Cero.
19. Silvio Micali. 2016. Algorand: The efficient and democratic ledger. *arXiv preprint arXiv:1607.01341* (2016).
20. Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).
21. Rafael Pass, Lior Seeman, and Abhi Shelat. 2017. Analysis of the blockchain protocol in asynchronous networks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 643–673.
22. Andrew Poelstra. 2014. Distributed consensus from proof of stake is impossible. (2014).
23. Emin Gün Sirer. 2017. What Could Go Wrong? When Blockchains Fail. (2017). http://events.technologyreview.com/video/watch/emin-gun-sirer-cornell-when-blockchains-fail/ Business of Blockchain.

DEFenD: A Secure and Privacy-Preserving Decentralized System for Freight Declaration

Vos, Daniël; Overweel, Leon; Raateland, Wouter; Vos, Jelle; Bijman, Matthijs; Pigmans, Max; Erkin, Zekeriya
Delft University of Technology
(D.A.Vos,L.P.Overweel,W.Raateland,J.V.Vos,M.D.Bijman,M.Pigmans)@student.tudelft.nl,Z.Erkin@tudelft.nl

ABSTRACT

Millions of shipping containers filled with goods move around the world every day. Before such a container may enter a trade bloc, the customs agency of the goods' destination country must ensure that it does not contain illegal or mislabeled goods. Due to the high volume of containers, customs agencies make a selection of containers to audit through a risk analysis procedure. Customs agencies perform risk analysis using data sourced from a centralized system that is potentially vulnerable to manipulation and malpractice. Therefore we propose an alternative: DEFenD, a decentralized system that stores data about goods and containers in a secure and privacy-preserving manner. In our system, economic operators make claims to the network about goods they insert into or remove from containers, and encrypt these claims so that they can only be read by the destination country's customs agency. Economic operators also make unencrypted claims about containers with which they interact. Unencrypted claims can be validated by the entire network of customs agencies. Our key contribution is a data partitioning scheme and several protocols that enable such a system to utilize blockchain and its powerful validation principle, while also preserving the privacy of the involved economic operators. Using our protocol, customs agencies can improve their risk analysis and economic operators can get through customs with less delay. We also present a reference implementation built with Hyperledger Fabric and analyze to what extent our implementation meets the requirements in terms of privacy-preservation, security, scalability, and decentralization.

KEYWORDS

System security, Blockchain, Privacy, Freight declaration, Customs Auditing, Logistics

ACM Reference format:
Vos, Daniël; Overweel, Leon; Raateland, Wouter; Vos, Jelle; Bijman, Matthijs; Pigmans, Max; Erkin, Zekeriya . 2018. DEFenD: A Secure and Privacy-Preserving
Decentralized System for Freight Declaration. In *Proceedings of W. Prinz & P. Hoschka (Eds.), 1st ERCIM Blockchain Workshop 2018, Amsterdam, Netherlands, May 2018 (ERCIM),* 8 pages.
DOI: 10.18420/blockchain2018_08

1 INTRODUCTION

In today's economy, many countries have specialized in producing certain goods: the Netherlands grows the most tulips, China assembles iPhones, and Honduras is the biggest producer of coffee [6]. These exports are consumed by people all over the world, so millions of containers full of goods move in and out of the world's ports every day [22].

Before any goods may enter a trade bloc, they must be cleared by the relevant customs agency. The customs agency at the destination country of the goods taxes the goods, and attempts to prevent forbidden goods from entering their trade bloc. However, customs agencies are only able to audit on the order of 1% of incoming containers due to the high volume of containers they are processing [24]. Therefore, customs agencies must determine which small portion of the containers to examine.

To decide which containers to examine, the customs agencies estimate for each container the risk that it is carrying illegal or mislabeled goods. Customs agencies audit those containers with the highest estimated risk. This risk analysis depends heavily on the available data and its quality and reliability.

In a typical scenario, there are two types of parties involved. The *economic operators* move goods in and out of containers and move containers around the world. The *customs agencies* need reliable data on these goods and container movements for their risk analysis calculations. The economic operators create data about their goods and containers, and customs agencies consume that data. The transfer of data from economic operator to customs agency in the current system is based around a *bill of lading*.

A bill of lading is an aggregate of information about all the goods on a single shipment of containers coming into a port. It is created by the economic operator in charge of that shipment, and sent to the relevant customs agency at least 24 hours before the ship arrives in the port. This system has three major shortcomings:

- The bill of lading does not tell a customs agency through which other ports a container of goods may have traveled.

- The bill of lading is an aggregation, so it is not the original source of data.
- The bill of lading is only required to be received 24 hours before a ship's arrival.

These shortcomings make it more difficult for customs agencies to predict which containers must be audited.

A naive solution to this problem would be to create a central trusted authority that collects data from all economic operators from the various online locations where it is available [10]. Such a centralized system exist: ConTraffic, a "web-based geographical information system enabling interactive visualization of container movements" collects data by mining public data repositories of economic operators [11]. This centralized approach has several drawbacks. A central authority could alter the data, and could decide to exclude or mistreat specific economic operators. Therefore, such a centralized system requires *trust*, which cannot be expected of all economic operators and customs agencies in the world.

These security concerns raise the need for a decentralized system, in which involved parties put their trust into a system rather than an organization. This system should enable different, mutually untrusted entities to collaborate and be privacy-preserving, secure, scalable, and decentralized.

We propose a decentralized system named DEFenD: *a secure and privacy-preserving DEcentralized system for Freight Declaration*, which enables economic operators and customs agencies to collaborate in an environment that does not require centralized trust, also when they do not have a direct business connection.

In our proposal, economic operators share data about containers and the goods within them through the network. When an economic operator inserts or removes goods from a container, they send this information to the network as an encrypted *claim*. This claim can only be decrypted by the customs agency at the goods' country of destination. Whenever a container changes hands, the involved economic operators create a signed unencrypted claim specifying the involved parties, as well as when and where this happened. Any customs agency can then observe the whole history of the container in question.

While the entire network can observe the container movements, the package information is only available to the destination agency. This preserves the privacy of the economic operators. Any alterations or mismatching data about the container movements can then easily be detected, resulting in a significant increase in the detection of high-risk containers.

In this paper, to the best of our knowledge, we propose the first decentralized system for freight declaration. The mechanism currently in use is not reliable, causing significant loss in container fraud detection accuracy. Our proposal presents a secure, privacy-aware, scalable system that solves the fundamental trust problem in the container shipment industry. We present the design details of the system using available blockchain technology, and introduce the key data partitioning schema that allows for validation while preserving key privacy requirements. We believe our proposal will enable a number of customs agencies and economic operators to start collaborating in freight declaration without having to trust any one authority to keep their data secure.

The rest of the paper is organized as follows. In Section 2, we present the building blocks of our system; in Section 4, we describe our proposal; in Section 5 we detail and analyze our system design; and finally, we conclude in Section 6.

2 BACKGROUND

A *blockchain* is often referred to as a "shared ledger", which is a type of distributed database technology. Many *nodes* form a peer-to-peer network that maintains this *shared ledger* consisting of *transactions*. These nodes use a *consensus algorithm* to determine which data may be added to their copy of the shared ledger.

Nodes in a blockchain network are responsible for maintaining the data in the shared ledger. In a *permissionless* blockchain, such as Bitcoin [19], any internet-connected computer, capable of understanding the blockchain network's protocol can participate in the network. In a *permissioned* blockchain, only select nodes may participate.

A *shared ledger* is a distributed append-only database present on each node. Data is added to the ledger in the form of *blocks*. As a new block is added to the ledger, the nodes synchronize their copies of the shared ledger by applying the consensus algorithm. In a *public* blockchain, anyone can read the data in the shared ledger, while in a *private* blockchain, all or some of the data in the shared ledger is encrypted.

Data is transmitted as *transactions* from one party to another. In Bitcoin, for example, a transaction consists of some amount of bitcoin being sent from one address to another [19].

The *consensus algorithm* determines how each node adds blocks of new transactions to its copy of the shared ledger. Examples of consensus algorithms include Byzantine Fault Tolerance replication [23] and Proof-of-Work [19] (see Table 1 for more consensus algorithms).

The key feature of blockchain is that it enables trust without requiring a central authority, since the *truth* is determined by

Table 1: Comparison of blockchain implementations.

Name	Maintainer	Permission	Consensus	GitHub
Chain	Chain Inc	Permissioned	Federated Consensus [4]	chain/chain
Corda	R3	Permissioned	(Custom)	corda/corda
Ethereum	Ethereum Foundation	Both	Proof-of-Work [19]	ethereum/go-ethereum
Hyperledger Fabric	The Linux Foundation, IBM	Permissioned	(Custom)	hyperledger/fabric
Hyperledger Iroha	The Linux Foundation	Permissioned	Byzantine Fault Tolerance [23]	hyperledger/iroha
Hyperledger Sawtooth lake	The Linux Foundation, Intel Corporation	Both	Proof of Elapsed Time [3]	hyperledger/sawtooth-core
Kadena	Kadena LLC	Permissioned	ScalableBFT [18]	(closed source)
MultiChain	Coin Sciences Ltd	Permissioned	Practical BFT [14]	multichain/multichain
OpenChain	Coinprism	Permissioned	Partionned Consensus [2]	openchain/openchain
Quorum	JPMorgan Chase & Co.	Permissioned	Raft [20]	jpmorganchase/quorum
Ripple	Ripple	Permissioned	Ripple [21]	ripple/rippled
Tendermint	All In Bits, Inc.	Permissioned	Byzantine Fault Tolerance [23]	tendermint/tendermint

an agreed-upon consensus algorithm. As long as the majority of a network is not collaborating to pollute the blockchain with incorrect data, its integrity is guaranteed [19].

After Bitcoin's rapid rise in popularity in the past few years, many different blockchain implementations have emerged; Table 1 contains a comparison of key aspects of a selection of such blockchain implementations.

3 RELATED WORK

In recent years, many companies have explored blockchain projects and prototypes for their respective industries. The container shipping industry is no exception: A.P. Mollerfi?!-Maersk Group ("Maersk" hereinafter) has launched a project in collaboration with EY (Ernst & Young) and Microsoft Corporation to launch a marine insurance blockchain that would help reduce the market's inefficiencies. A prototype of the platform has been built on Microsoft Azure to "make auditing aspects of a shipping supply chain easier, to improve the tamper-resistance and sharing of data in realtime, and to enable many different parties to settle upon the terms of premiums in a more timely fashion" [15]. The platform was to be deployed in January 2018, but no further news has emerged about it.

4 DEFEND- A DECENTRALIZED SYSTEM FOR FREIGHT DECLARATION

DEFenD is a secure and privacy-preserving decentralized system for freight declaration. It is built on a blockchain consisting of a network of certified nodes managed by *customs agencies* and *economic operators*. For DEFenD, we assume that:

(1) *Economic operators trust the customs agency of their own country.* DEFenD must be implemented on a permissioned blockchain to ensure that only verified economic operators can submit data; the party that issues the certificates to enable this participation is the customs agency of the economic operator's country of origin; so they must be trusted to not abuse those certificates.
(2) *Customs agencies do not trust customs agencies outside their trade bloc.* If there was complete trust between all customs agencies, a centralized system could be maintained by one of them; this is not the case.
(3) *Packages in the system only move by shipping container.* We do not consider the movement of packages by other modes of transportation.

Based upon these assumptions, we aim to achieve the following goals with DEFenD:

Privacy-preserving. In order to preserve the privacy of economic operators DEFenD must support visibility restrictions on the claims that economic operators post about the goods

Table 2: Fields stored in container and package claims

Field	Container claims	Package claims
Container ID	The ISO 6346 container identification number [16]	
Shipment ID	Concatination of the ship's IMO number [5] and date of departure in ISO 8601 format [17]	(N/A)
Package ID	(N/A)	This package's identification number, defined by the shipper
From	Economic operator who has this container	(N/A)
To	Economic operator who receives this container	(N/A)
Sender	(N/A)	Person / company who sends this package
Receiver	(N/A)	Person / company who receives this package
Time	Time of claim submission in ISO [17] format	
Location	Longitude and latitude of the location at which the claim was made	
Weight	Weight of this container, in kilograms	Weight of this package, in kilograms
Action	(N/A)	`INSERT` if package was inserted into container; `REMOVE` if it was removed"
Contents	(N/A)	Description of this package's contents

they are transporting. Most importantly, economic operators must not have the ability to read the data in each others' claims. Furthermore, only the customs agencies that are in the trade bloc that the package is destined for should be able to read the data in the claims, since economic operators do not necessarily trust every customs agency.

Secure. Non-repudiation is key to the security of the system because we want to make sure an economic operator can never deny a claim they made. To achieve this DEFenD must enforce that no economic operator in the system can alter previously submitted data. Economic operators can only submit new data when they have been granted access to the system by customs agencies. Claims about containers should only be accepted as long as the economic operators that submitted them are likely interacting with the containers.

Scalable. DEFenD must be able to handle enough transactions to support economic operators submitting transactions at any time. To support a gradual rollout, DEFenD must be able to track containers and packages even when not all economic operators in the supply chain participate in the system.

Decentralized. DEFenD must be decentralized to avoid the shortcomings of a centralized system, such as the potential for a central database to be manipulated.

Protocol Overview

To meet these goals given our assumptions, we define DEFenD with the following entities:

- *Economic operators* are companies that either insert or remove packages from containers or transport containers. Economic operators submit data about the goods that they are handling to the blockchain network in the form of *claims*.
- *Customs agencies* process these claims and attempt to reach consensus over whether or not to append submitted claims to their shared ledger.
- *Containers* carry one or multiple packages of goods in them, and are identified by a container number that is specified according to the ISO 6346 standard [16].
- *Packages* are identified by the combination of container number, time and a number identifying them inside the container.
- Data that is submitted to the blockchain network about containers or packages are referred to as *container claims* and *package claims*. Claims are data objects that are signed by an economic operator.

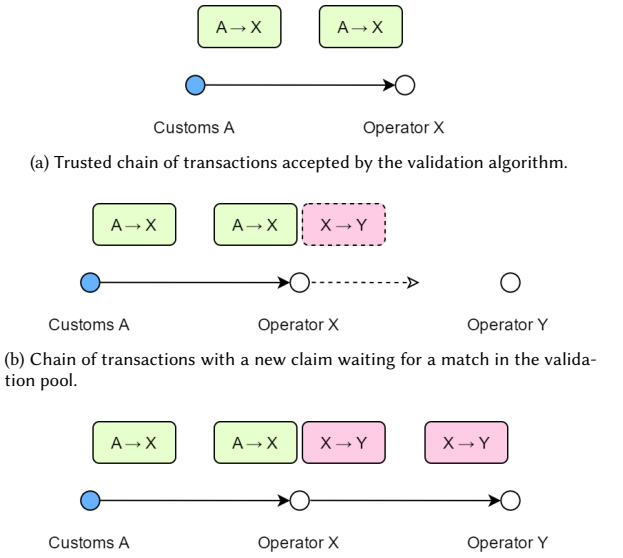
The DEFenD protocol consists of the following three subprotocols:

Algorithm 1 Validate container claim $C_X = X \xrightarrow{c} Y \mid X$ by operator X about container c, with validation pool P_c.

```
procedure VALIDATE(C_X, P_c)
    T ← QUERY(c)                                   ▷ Query blockchain for latest accepted claim about c
    if isCUSTOMS(X) then
        ACCEPT(C_X)                                ▷ Start a new trusted chain
    else if T exists and T.to = C_X.from then
        if C_Y ∈ P_c and C_X.from = C_Y.from and C_X.to = C_Y.to then   ▷ If matching claim by Y exists in P_c
            ACCEPT({C_X, C_Y})                     ▷ Accept both claims
            P_c ← ∅                                ▷ Clear P_c of wrong claims to save memory
        else                                       ▷ If matching claim by Y does not (yet) exist in P_c
            P_c ← P_c ∪ {C_X}                      ▷ Add C_X to P_c
    else
        REJECT(C_X)                                ▷ Only customs agencies may create new trusted chains
```

(1) The *claim submission protocol* (Section 4) specifies how economic operators submit data to the blockchain. It is run on nodes belonging to economic operators.
(2) The *container claim validation protocol* (Section 4) determines whether data submitted by economic operators is valid and should be added to the shared ledger. It is run on nodes belonging to customs agencies.
(3) The *economic operator certification protocol* (Section 4) lets customs agencies allow or revoke access to economic operator in the blockchain network. The certification protocol is run on nodes belonging to customs agencies.

We describe each sub-protocol in detail in the following sections.



(a) Trusted chain of transactions accepted by the validation algorithm.

(b) Chain of transactions with a new claim waiting for a match in the validation pool.

(c) Trusted chain of transactions accepted by the validation algorithm with a new transaction of two claims appended.

Figure 1: Chains of transactions submitted by economic operators, same colors refer to the same transaction but made by different parties. a → b → c shows the process of adding a new transaction to the chain.

Claim Submission Protocol

Economic operators submit container claims and package claims to the network. Container claims tell customs agencies how containers are moving around the world, and package claims tell customs agencies what packages of goods are inside these containers. The data fields in each of these types of claims are described in Table 2. We depict a *container claim* as $X \xrightarrow{c} Y \mid S$, where X is the economic operator that hands container c to economic operator Y. S is the signer of this claim and must be the same as either X or Y. In the case that the next operator does not participate in the system, an economic operator A should claim $A \xrightarrow{c} \epsilon \mid A$. Then, When an operator B that participates in the system receives c again, B should claim $\epsilon \xrightarrow{c} B \mid B$. Before claims are appended to the shared ledger, customs agencies will run the validation protocol on claims. Package claims must be encrypted using the public key of the destination's customs agency before they are submitted. Economic operators also add a plain-text field to package-claims to indicate which customs agency has the private key that can be used to decrypt the claim.

Validation Protocol

Algorithm 1 validates a submitted container claim and determines whether it should be added to the shared ledger. It first checks that the economic operator in the *from* part of the new claim is actually in possession of the claim. It then makes sure that both economic operators involved in the

claim agree on what happened. If both of these conditions are true, the claims are added to the shared ledger.

Given a container claim $C_X = X \xrightarrow{c} Y \mid X$, a previously accepted transaction of A to X, and container c's validation pool P_c, Algorithm 1 first queries the blockchain for the latest claim T about c, as shown in Figure 1a. If T exists and c has been given to X (which is the case in the example), then the new claim will be added to the pool of to-be-validated claims P_c. This scenario is shown in Figure 1b. The claim $X \xrightarrow{c} Y \mid X$ will only be accepted when $X \xrightarrow{c} Y \mid Y$ is submitted as shown in Figure 1c. Claims are not required to be submitted in the order of from-operator then to-operator, they are always added to P_c when no matching claim is found.

If c has not been given to X in the previous transaction, the new claim can only be accepted if a customs agency has submitted the claim to reset the chain. This is required when operators can not confirm transactions and therefore the trusted chain is broken.

To save memory in the validation, all claims in the to-be-validated pool P_c about container c can be cleared when a claim is accepted for c. Also claims that contain impossible data can be left out of the validation pool.

Certification Protocol

When an economic operator is to be added to the network, the customs agency in its country can certify that operator. This is done by generating a digital certificate that is signed by the newly added economic operator to prove that it has the correct key. When an economic operator misbehaves, the customs agency in their country can revoke the operator's certificate to restrict access to the blockchain. A revocation list is kept by the certificate authority, and blockchain nodes verify that claims have signatures that are generated using certificates that do not occur in the revocation list.

To add or remove customs agencies from the system, the customs agency nodes participate in a vote to reach consensus.

5 SYSTEM DESIGN AND ANALYSIS

We present a reference implementation for DEFENDD, which implements all the protocols in Sections 4, 4 and 4. As shown in Figure 2, several *nodes* exist on the blockchain network. Such nodes may be *customs agency nodes* (e.g. nodes run by customs agencies that follow the customs agencies protocol) or *economic operator nodes* (e.g. nodes run by economic operators that follow the economic operator protocol). These nodes communicate with each other over gRPC[1]. Each node has its own web server, which wraps its functionality in an API, with which it communicates over gRPC as well. The

[1]gRPC stands for "gRPC Remote Process Call." See http://www.grpc.io.

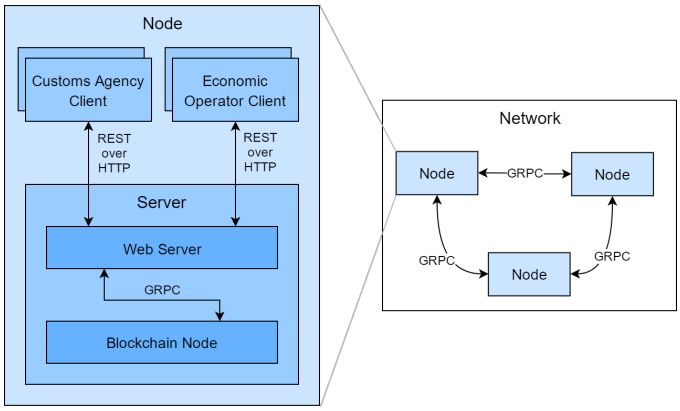


Figure 2: System architecture for PassPort.

clients each communicate with the web server via REST over HTTP.

In the next sections, we describe each component in detail.

System Design

Blockchain. We implement the blockchain component using Hyperledger Fabric [9]. We compare several blockchain implementations in Table 1. Hyperledger Fabric is ideal for our protocol since it offers a private permissioned blockchain that supports at least 1,000 transactions per second (see Section 5) and allows for a pluggable custom consensus algorithm.

Client-Server Interaction. For customs agencies and economic operators to interact with their peers in the blockchain network we implement a web server that wraps networking complexity in a RESTful API. The server receives API calls from two GUIs: One for economic operators and one for customs agencies. The customs agencies get an overview of shipments and containers, along with their estimated risk level. The GUI for economic operators mainly comprises of forms that are used to submit claims.

Analysis

In Section 4 we have put forward some goals for DEFENDD that we have addressed in our implementation. We now evaluate DEFENDD in regards to these requirements.

Privacy-preserving. Since package data is encrypted using asymmetric encryption, only entities that have access to the private key can read the data. In the case of our protocol only the customs agency that will receive a package in their port has access to that private key. This means that economic operators can never read package data from others, only customs agencies that they will definitely interact with can.

Secure. To ensure that economic operators can only submit claims if they have been granted access to the blockchain network, customs agencies run a *certificate authority node*, which grants the economic operators a key, which is part of their certificate, that they can use to sign their claims. Nodes in the system will immediately reject claims that are signed by revoked certificates.

We also introduce a protocol that ensures that economic operators can only make claims about containers they likely interacted with; economic operators' claims are in fact transactions. Economic operator X can therefore only make a claim about container c if there is another economic operator Y that made a claim saying, Y provided X with c, and thus confirms X's claim.

Scalable. Because economic operators must have the ability to make claims to other economic operators that are not part of the system we introduce an ϵ operator that represents a hole in the system. This means that containers can leave and reenter the system.

For a scalable system we must support sufficient throughput. We determine the amount of transactions that DEFenD must support as follows. As of 2012, there are approximately 32.9 million TEU[2] shipping containers globally [1] (order of magnitude: 10^8). Because a single voyage takes days or weeks to complete, we generously estimate that a single container may 'switch hands' up to 10^2 times per year. Each time a container 'switches hands', this requires a transaction. We estimate the amount of containers switching hands per second in Equation 1:

$$\frac{10^8 \text{ containers} \times 10^2 \ \frac{\text{moves}}{\text{year}}}{3600 \times 24 \times 365 \ \frac{\text{seconds}}{\text{year}}} \approx 317 \ \frac{\text{container moves}}{\text{second}} \quad (1)$$

The only nodes that participate in the consensus algorithm are customs agencies, in the World Customs Organization 182 countries are included[7]. In practice, however, countries have formed customs unions reducing the need for every country to run their own node. Currently shipping is dominated by trade between 24 customs unions[8]. Therefore the system must be able to support a maximum of approximately 24 nodes.

Our system must also consider what goods go into containers. If containers have a Full Container Load (FCL), this is one 'package' per container per trip, but they have a Less than Container Load (LCL), this means multiple 'packages' per container per trip.

So DEFenD must have a throughput of at least 10^3 transactions per second and be able to support around 24 nodes. Hyperledger Fabric has been shown to support this throughput in recent benchmarks and currently supports up to 16 nodes[12]. With the release of Hyperledger Fabric version 1.1 a promise for higher scalability and performance have been made[13].

Decentralized. Our implementation is decentralized as it uses a blockchain framework that runs on multiple nodes, it ensures that no single party controls the data in the system. By doing so, we remove trust that is required between parties.

6 CONCLUSION

The shipping industry is responsible for the movement of millions of containers every day. Before these containers may enter a trade bloc, they must be cleared by the relevant customs agency. Because of the high volume of containers that must be processed each day, customs agencies perform risk analysis to decide which containers to audit. Risk analysis requires lots of data, which in the current system is potentially vulnerable to manipulation and malpractice because it is centralized and collected by a single authority.

In this work we have presented DEFenD, *a secure and privacy-preserving decentralized system for freight declaration* that does not require the trust between entities that is required in centralized systems. In DEFenD, economic operators make *claims* about the packages of goods and containers with which they interact, customs agencies validate those claims. Customs agencies and economic operators participate in a blockchain that validates this data and stores it in a secure and privacy-preserving manner. Our two key contributions are a data partitioning scheme and several protocols to enable this, and a reference implementation built on Hyperledger Fabric.

Firstly, our data partitioning scheme and protocols allow DEFenD to take advantage of the powerful validation principles enabled by blockchain, while hiding certain parts of the data to preserve the privacy of the involved economic operators. In our system, claims about the movement of containers are unencrypted, and can be validated to ensure that 1) the claim fits in the preceding chain of claims about that container and that 2) both parties involved in the claim agree on its contents. Claims about packages are encrypted so that only the customs agency at the goods' country of destination can see them. Hiding this critical link in the data means that only the appropriate customs agency can recreate the exact path that goods took to get to their country. This knowledge can improve the customs agency's risk analysis.

Secondly, our reference implementation built on Hyperledger Fabric shows that it is possible to implement DEFenD on a blockchain that meets our privacy-preservation, security, scalability, and decentralization requirements.

[2]Twenty-foot Equivalent Unit, a standard shipping container size

In future work, the combined container claim and package claim data provided by DEFENDD could be used to further automate customs agencies' taxation procedures.

REFERENCES

[1] 2013. Global Container Fleet. (2013). http://www.worldshipping.org/about-the-industry/containers/global-container-fleet

[2] 2015. How does it work? (2015). https://www.openchain.org/

[3] 2015. Introduction — Proof of Elapsed Time (PoET). (2015). https://intelledger.github.io/introduction.html

[4] 2017. Federated Consensus. (2017). https://chain.com/docs/1.1/protocol/papers/federated-consensus

[5] 2017. IMO identification number schemes. (2017). http://www.imo.org/en/ourwork/msas/pages/imo-identification-number-scheme.aspx

[6] 2017. The World Factbook. (2017). https://www.cia.gov/library/publications/the-world-factbook/

[7] 2017. World Customs Organization - 182 members. (2017). http://www.wcoomd.org/-/media/wco/public/global/pdf/about-us/wco-members/list-of-members-with-membership-date.pdf

[8] Soamiely Andriamananjara. *Customs Unions*. https://siteresources.worldbank.org/INTRANETTRADE/Resources/C5.pdf

[9] Christian Cachin. 2016. Architecture of the Hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*. https://www.zurich.ibm.com/dccl/papers/cachin_dccl.pdf

[10] Tatyana Dimitrova. 2013. Survey of availbale web services for maritime tracking. *Int. J of Computer and Information Technology* 2, 2 (2013). http://ijcit.com/archives/volume2/issue2/Paper020201.pdf

[11] Tatyana Velikova Dimitrova, Aris Tsois, and Elena Camossi. 2013. Visualization of container movements through a web-based geographical information system. In *Intelligence and Security Informatics Conference (EISIC), 2013 European*. IEEE, 182–185. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6657150

[12] Tien Tuan Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi, and Kian-Lee Tan. 2017. BLOCKBENCH: A Framework for Analyzing Private Blockchains. In *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 1085–1100. http://www.comp.nus.edu.sg/~ooibc/blockbench.pdf

[13] Christopher Ferris. Performance and scale improvements for 1.1. (????). https://jira.hyperledger.org/browse/FAB-6421

[14] G Greenspan. 2015. MultiChain Private Blockchain - White Paper. (2015). http://www.multichain.com/download/MultiChain-White-Paper.pdf

[15] Robert Hackett. 2017. Maersk and Microsoft Tested a Blockchain for Shipping Insurance. *Fortune* (Sep 2017). http://fortune.com/2017/09/05/maersk-blockchain-insurance/

[16] ISO 6346:1995 1995. *Freight containers – Coding, identification and marking*. Standard. International Organization for Standardization, Geneva, CH. https://www.iso.org/standard/20453.html

[17] ISO 8601:2004 2004. *Data elements and interchange formats – Information interchange – Representation of dates and times*. Standard. International Organization for Standardization, Geneva, CH. https://www.iso.org/standard/40874.html

[18] W. Martino. 2016. Kadena - The first scalable, high performance private blockchain. (2016 2016). http://kadena.io/docs/Kadena-ConsensusWhitePaper-Aug2016.pdf

[19] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008). https://bitco.in/pdf/bitcoin.pdf

[20] Diego Ongaro and John K Ousterhout. 2014. In Search of an Understandable Consensus Algorithm.. In *USENIX Annual Technical Conference*. 305–319. https://www.usenix.org/system/files/conference/atc14/atc14-paper-ongaro.pdf

[21] David Schwartz, Noah Youngs, and Arthur Britto. 2014. The Ripple protocol consensus algorithm. *Ripple Labs Inc White Paper* 5 (2014). https://ripple.com/files/ripple_consensus_whitepaper.pdf

[22] United Nations Conference on Trade and Development. 2016. *Review of Maritime Transport 2016*. United Nations. http://unctad.org/en/PublicationsLibrary/rmt2016_en.pdf

[23] Marko Vukolić. 2016. *The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication*. Springer International Publishing, Cham, 112–125. DOI: http://dx.doi.org/10.1007/978-3-319-39028-4_9

[24] E.D. Wiebes. 2017. Antwoord op vragen van de leden Swinkels, Belhaj en Van Veldhoven over de berichten 'Druk aan de Poort' en 'Zelfverrijking normale gang van zaken bij Rotterdams containerbedrijf'. (Jan 2017). https://www.tweedekamer.nl/kamerstukken/kamervragen/detail?id=2017D01350

Privacy by BlockChain Design: A Blockchain-enabled GDPR-compliant Approach for Handling Personal Data

Christian Wirth
Senior Blockchain Architect (IBM)
Freie Universität Berlin
wirth@protectivecircle.com

Michael Kolain
Research Associate (Law)
German Institute for Public Administration
Speyer
michael.kolain@posteo.de

ABSTRACT

This paper takes an initial step forward in bringing to life the certification mechanisms according to Art. 42 of the General Data Protection Regulation (GDPR). These newly established methods of legal specification act not only as a central vehicle for overcoming widely articulated and discussed legal challenges, but also as a sandbox for the much needed close collaboration between computer sciences and legal studies. In order to illustrate, for example, what data protection seals could look like in the future, the authors propose a methodology for "translating" legal requirements into technical guidelines: architectural blueprints designed using legal requirements. The purpose of these blueprints is to show developers how their solutions might comply with the principle of Privacy by Design (Art. 25 GDPR). To demonstrate this methodology, the authors propose an architectural blueprint that embodies the legal concept of the data subject's consent (Art. 6 sec. 1 lit. a GDPR) and elevates best practice to a high standard of Privacy by Design. Finally, the authors highlight further legal problems concerning blockchain technology under the GDPR that will have to be addressed in order to achieve a comprehensive certification mechanism for Privacy by Blockchain Design in the future.

ACM Classification Keywords

10002978.10003029.10011150 Security and privacy: Privacy protections

Author Keywords

Blockchain; Privacy by Design; GDPR; Personal Data; Smart Contract; Data Protection

INTRODUCTION

Handling Personal Data in the digital world and the opportunities of blockchain technology

The more digitalized our every day life becomes, the more important it is from a privacy perspective to have control over the data we emit. According to the GDPR, the procession of personal data by any party requires either the consent of the data subject or a legal basis (Art. 6 GDPR). However, today's IT-systems are not capable of providing this functionality in an ideal sense. Rather, the status quo remains as follows: Personal data is purported to be stored according to legally binding security standards in company-owned or cloud-based data silos, without mutual confirmation between data subject and recipient that this information is being handled responsibly.

The consent of the data subject is the central vehicle for ensuring everyone's right to the protection of personal data (Art. 16 section 1 of the Treaty on the Functioning of the European Union, TFEU). Consent links the processing of personal data to the free decision of the data subject–in an ideal sense for both initial and subsequent processing. However, under the architecture currently in use, the data subject must have confidence that the recipient of the consent processes his personal data lawfully and that the data protection authorities otherwise perform their supervisory function responsibly. However, given today's methods of storing and accessing data, the individual usually cannot directly retrace what happens to his personal information in the IT-system of the controller. As a result, the individual is mostly limited to giving his or her consent beforehand, in a way that is based on an abstract clause (e.g. "... for purposes of health improvement") rather than on a more transparent case-by-case basis.

Although the supervising bodies of the EU member states monitor the data market and can sanction infringements, they underlie the same restrictions as the data subject in order to detect a unlawful subsequent processing not covered by consent. Until now, their ability to guarantee a 'consistent and high level of protection of natural persons' (Recital 10 GDPR) has been severely limited–one of the main reasons being the technical status quo just described.

Despite widespread concern about the safety of the digital sphere, the Web 3.0 [8], in combination with blockchain-technology and modern cryptography, can bring personal data management to a level of privacy and security that prioritizes individual sovereignty and shared transparency. The semantic web gives meaning to data in the digital space, allowing it to be classified and encrypted accordingly. The blockchain can then act as a tamper-proof ledger to record digital interactions; the data subject can verify where his personal data is stored and put to (commercial) use. Today, data is present

Wirth, Christian; Kolain, Michael (2018): Privacy by BlockChain Design: A Blockcchain-enabled GDPR-compliant Approach for Handling Personal Data. In: W. Prinz & P. Hoschka (Eds.), Proceedings of the 1st ERCIM Blockchain Workshop 2018, Reports of the European Society for Socially Embedded Technologies

ISNN 2510-2591.

DOI: http://dx.doi.org/10.18420/blockchain2018_03

in vast multiplicity, with each copy representing the state of this data at the point in time when it was saved. The Web 3.0 allows us look at this from a new perspective: Instead of saving copies of the relevant data, which could potentially become outdated from the second it is stored, users should keep pointers to the origin of the data, which they know will always provide the most up-to-date version of the information. In addition, smart contracts can now be executed completely automatically on behalf of digital identities, which enables us to provide personal data to a third party whenever access to it is required. Third parties can file a request for access, and a smart contract will check the validity of this request and handle it accordingly–transparently for all parties involved.

For a GDPR-compliant blockchain solution predefined by the specific requirements of a certification mechanism, giving and withdrawing consent will form a necessary base element. In the experiment documented by this paper, the goal was to determine what architecture we would end up with if we used the law as the base requirement for designing a minimal, sufficient architectural blueprint representing the legal concept of the data subject's consent. In order to allow the reader to follow our interdisciplinary journey we will first present the concept of Privacy by Design according to the GDPR and explain our methodology of architectural blueprints. Next, to demonstrate our methodological approach, we will introduce our own blueprint focusing on the data subject's consent in a blockchain-enabled and GDPR-compliant manner. We will then outline further legal challenges that could not be covered in this paper, but will play a crucial part in the further development of certification mechanisms in accordance to the GDPR. We will conclude with some general thoughts on why blockchain is an important technology enabling us to rethink obsolete design models and establish new standards for trust, transparency and privacy under which personal data could be handled in the future.

PRIVACY BY DESIGN UNDER THE GDPR AND CERTIFICATION MECHANISMS

The GDPR came into effect within all member states of the European Union on May 25th, 2018. One of the major requirements when it comes to handling personal data is that the underlying IT-systems follow the concept of **Privacy by Design** (Art. 25 GDPR). In its most basic sense, it asserts that "privacy should be promoted as a default setting of every new IT system and should be built into systems from the design stage" [7]. In the more complex words of Art 25 sect. 1 GDPR, it obliges the controller to implement "appropriate technical and organisational measures (...) which are designed to implement data-protection principles (...) in an effective manner and to integrate the necessary safeguards into the processing in order to meet the requirements of this Regulation and protect the rights of data subjects". As a simple example, one consequence would be that personal data must not be stored in plaintext on blockchains.

While these legal requirements remain highly abstract and, thus, open to interpretation, Art. 25 sect. 3 GDPR leaves room for specifications: "*Approved certification mechanisms* pursuant to Article 42 may be used as an element to demonstrate compliance with the requirements." The concept of EU-wide certification mechanisms is new to EU data protection law. Their purpose is "to enhance transparency and compliance with this Regulation" and allow "data subjects to quickly assess the level of data protection of relevant products and services" (Recital 100 GDPR). Especially for new technologies, like blockchain, which occupy the margins of the GDPR's legal doctrines, the process of certification–which includes **data protection seals and marks**–can serve as a means for legal certainty, as it fulfills the "purpose of demonstrating compliance with this Regulation" (Art. 42 sec. 1 GDPR).

However, being certified does not guarantee, but rather only indicates, the legal processing of data. What's more, it will not be mandatory for software to be certified. Still, the certification mechanisms bear the potential–even as a "voluntary" (Art. 42 sec. 3) measure–to set standards, thereby boosting technological advancement in the market. It is therefore foreseeable that certification mechanisms will play a crucial and practical role in defining compliant ways of processing personal data under the GDPR [2]. It is likely that the supervisory bodies and the newly established European Data Protection Board (Art. 68 ff. GDPR) will take steps in this direction fairly soon–and they will need scientific help from the blockchain community. In order to master the herculean task of defining technology-specific standards, the fields of computer science and law must align themselves more closely–a perspective that supplies the impetus for our interdisciplinary work [10] and [11].

METHODOLOGY: ARCHITECTURAL BLUEPRINTS AS AN ELEMENT OF CERTIFICATION MECHANISMS

The architectural blueprint introduced in the following section tries to give a first methodological answer to how the certification mechanisms of the GDPR could narrow down specific standards. As there are no particular specifications in Art. 25 GDPR for how privacy by design should work or which properties a system inspired by or built on blockchain-technology should have, this paper aims to provide a general approach for architectural system designs.

Rather than propose an entire framework for a certification mechanism (which would go widely beyond the limits of this paper), we attempt to integrate a single legal requirement into a blockchain-enabled-architecture that builds upon the ideas of Privacy by Design.

As a central vehicle to protect personal information in the digital world (see Introcution), **the data subject's consent** (Art. 6 sec. 1 lit. a GDPR) seems like a good starting point. This is so because, firstly, the legal outlines of the consent have already undergone a thorough academic and practical discourse. This makes it relatively simple to model consent in a distributed architecture in contrast to other more controversial legal requirements, such as an implementation of the Right to Erasure. Secondly, consent forms a fundamental legal category in order to justify the processing of personal data under EU legislation. Therefore, future data protection certificates, seals and marks will have to cover this aspect as a base element.

Our technical proposal, however, does not stop with defining solely minimal standards for implementing the concept of consent into an IT-solution. It rather aims to contribute to a (potential) data protection seal that marks a high standard of Privacy by Design. By doing so, we also want to propose a new generation of systems handling personal data. Consequently, we utilize an architectural blueprint that both guarantees compliance with the GDPR concerning consent and sets new standards that embrace the core ideas nurturing the concept of Privacy by Design.

TECHNICAL DETAILS

First we start with the "translation" from the legal requirements about consent to technical requirements.

Broken down to the technical level, consent means that the data subject, first, **shall be asked once for approval** when someone wants to process his or her personal data and, second, has to be able to **withdraw their consent** given to a specific party (Art. 7 sect. 3 GDPR).

In addition to the minimal requirements to be GDPR-compliant mentioned above, the solution supports and reflects that data subjects should be given control over and allowed to reclaim their personal data as they see fit. Therefore data subjects are **notified whenever their personal data is processed**; changes to access rights are instantaneously reflected on the endpoint providing the personal data to the controller.

To make a distinction from personal data in a legal sense, the technical representation of it shall be called *Set of Personal Data (SoPD)*.

Actors of the Use Case

- **Issuing Party**: the entity guaranteeing that a particular SoPD is authentic
- **Data Subject**: the person to whom a SoPD relates to
- **Third Party**: the party requesting a SoPD

Explanation of Cryptographic Expressions:

- $*R$ Denotes a pointer to a resource R
- $H(X)$: The hash of a SoPD X
- $Enc()$: The function of an encryption scheme used to encrypt the SoPD in plain text using the public key of the data subject
- $Dec()$: The function of an decryption scheme used to decrypt the SoPD in cyphertext using the private key of the data subject
- $Enc(P)$: The resulting Cyphertext using $Enc()$ on the SoPD P in plaintext
- $Dec(C)$: The resulting Plaintext using $Dec()$ on the SoPD C in cyphertext

The data subject is the only person who should be able to decrypt the SoPD; therefore we use a suitable asynchronous-public key encryption scheme, where the issuing party encrypts the verified SoPD using the public key of the data subject, sends the $Enc(SoPD)$ to the data subject, and keeps only the Hash of the Set of Personal Data $H(SoPD)$.

Issuing Party

The **issuing party** stores a set of the following information on their blockchain or blockchain-compatible data storage:

- $H(SoPD)$: The Hash of the SoPD
- $*SC(SoPD)$: A pointer to the access-point of a smart contract to request the required SoPD

The Hash of the SoPD is stored on a blockchain-compatible data store of the issuing party in order to allow any third party to check the validity of the decrypted SoPD that has been delivered directly by the data subject's smart contract. The pointer reveals the access point to a smart contract handling every request for a SoPD. This ensures that the data subject is notified every time his or her data is requested in order to be processed. From the perspective of the self-determination about one's personal data, this is an ideal situation: The data subject can give specific consent case-by-case (or inspect if the smart contract was applied correctly in each case) rather than having to declare his or her consent beforehand in an abstract way without being able to control each processing. In contrast, from the perspective of a company working with personal data, this could lead to a higher administrative burden.

Data Subject

The **data subject** provides a smart contract, allowing third parties to request a subset of or a full SoPD. This service allows the data subject to decide on how to react to requests – and which subsets of personal data he or she wants to share. The following Figure 1 shows the connection flow and the underlying interaction between the smart contract provided by the data subject, the third- and the issuing party. The interaction parts for the third party requesting the SoPD will be described in the next subsection.

Smart Contract

The smart contract handles only a single type of SoPD by one issuing party to be provided to third parties. The third party initializes contact to the smart contract once, requesting a certificate for future access to the SoPD. Given that the data subject gives his consent to the recipient's request (and does so "freely" as demanded by Recital 11 GDPR), an up-to-date SoPD can from now on be requested just in time, when it is needed for processing. The smart contract will provide the SoPD immediately as long as the certificate of the third party is valid. There is now no more need to store the actual personal data for the third party.

The smart contract has to meet the following minimal requirements:

- The smart contract has an interface that can handle the initial request of a certificate for future requests of an SoPD.
- The smart contract has access to a securely hosted decryption function, which will provide the function $Dec(X) = Enc(SoPD)^{-1}$.

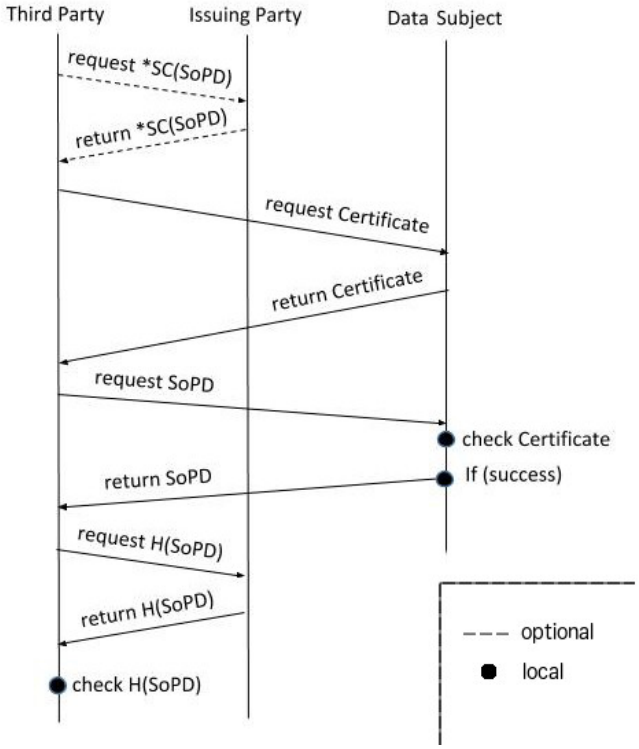


Figure 1. Flow diagram of communication between data subject, issuing party and third party

The critical part here is the key used for decryption, as security stands and falls with the secrecy of this private key [6]. In order to ensure that the data subject is notified whenever a SoPD is accessed, it would be required that the data subject is the **single source**, providing the smart contract capable of decrypting the SoPD in question. In favor of practicality, this functionality can be handled by a blockchain functioning as an immutable access-log as Zyskind and Nathan have shown in their paper [12].

The following model describes a minimal interface for a smart contract that allows a third party to request a SoPD ensuring that the data subject is always notified whenever one of his SoPD is disclosed to any third party:

- RequestCertificate(ThirdPartyID, ReasonForRequest)
- RequestSoPD(Certificate, RequestedSubsetOfSoPD)
- Access to Oracle for $Dec(X)$, where X is element of $Enc(SoPD)$
- CheckValidityOfCertificate(Certificate) Checks if the requesting party is allowed to be given access to the SoPD based on the certificate provided with the request.

It would be unrealistic to expect a service provided by the data subject to be highly available, or to assume the average user is capable or willing to set up such a service and maintain it. However, this is where blochckain comes into play: it is the missing puzzle piece in achieving high availability while maintaining full control over one's personal data. Zyskind et al. have shown that an architecture using blockchain can solve this problem quite elegantly [12].

In addition, there are no guarantees that copies of this SoPD are still up-to-date as soon as the hash of the SoPD $H(SoPD)$ has changed. This can easily be achieved by modifying a timestamp in the SoPD whenever it is requested, as changing only the timestamp without touching the relevant personal data results in another hash, and forces third parties to file a new request against the smart contract provided by the data subject if they want to make sure that they process an up-to-date SoPD. This would also allow them to identify any processing of an outdated SoPD by the third party without requesting it for just-in-time processing. Through this mechanism it is possible to identify third parties that store personal data without the data subject's consent. Changing the hash $H(SoPD)$ in combination with invalidating the third party's certificate also can serve as a tool to withdraw a once given consent (Art. 7 sect. 3 GDPR).

For security reasons, a separate key pair is to be generated for every smart contract. This allows us to invalidate the public key in case the private key for this particular SoPD is compromised. It also minimizes the effort to mitigate the damage as only the SoPD encrypted with the compromised key has to be encrypted with a newly generated keypair.

For efficiency reasons, the service of the data subject could also provide a smart contract informing a third party if a previous request of a SoPD is still up-to-date. This would not conflict with the requirement of every request to notify the data subject, as each inquiry is linked to the recipient by the Certificate.

FURTHER LEGAL CHALLENGES CONCERNING BLOCKCHAIN TECHNOLOGY

While the terminological foundation of the GDPR is only hardly compatible with decentralized database structures like Distributed Ledger Technology (blockchain in particular), it also comes with many innovations [9]. Still, legal uncertainty is one of the main obstacles for a widespread adoption of blockchain solutions, especially in the common market of the EU. A closer look at data protection law can, however, show a way out of the legal deadlock. The academic debate could lead into additional architectural blueprints which can be used in the certification processes ensuring Privacy by Design under the GDPR.

Personal Data

First of all, the scope of the GDPR applies only if **personal data** (Art. 4 sect. 1 GDPR) is involved. At first glance, a blockchain handles no names, addresses, or e-mail IDs - only hashes and encryption keys. Therefore, especially in the non-legal debate, blockchain-data is often referred to as "anonymous" - and since anonymous data is not subject to the GDPR [1], blockchain could thus per se fall out of the scope of

[1] According to Recital 26 GDPR information is anonymous if it "does not relate to an identified or identifiable natural person" or if "personal data is rendered anonymous in such a manner that the data subject is not or no longer identifiable"

data protection law and its regulatory corset. However, this is too simplistic. In many cases, there will be an entity that can identify the person behind a private key - e. g. when the data subject is buying an item using a cryptocurrency (and leaves his address for delivery) or for someone using methods such as Chainanalysis to mine the data in a public blockchain making sense out of the usage of a specific private key. Therefore, many use cases of blockchain are not anonymous [3]; rather, they are - in the legal sense - examples of *pseudonymity*. This is the case if personal data "can no longer be attributed to a specific data subject without the use of additional information" (Recital 26 GDPR). In other words: data is pseudonymous if someone has the possibility to combine it with other available information and can thus identify a person; and it is anonymous if this possibility does not exist. Since means of pseudonymity are still "personal data", the scope of the GDPR will still apply for a wide range of blockchain-solutions.

However, blockchain-data could, theoretically, fall through the cracks of data protection law if the person behind the data cannot be identified directly or indirectly by a person trying to do so. Data is considered anonymous if the identification of a person - even if theoretically possible - would need disproportionatly high measures, taking into account "all the means reasonably likely to be used," including "all objective factors, such as the costs of and the amount of time required for identification, taking into consideration the available technology at the time of the processing and technological developments" (Recital 26 GDPR). In a legal sense, the GDPR would not be applicable in such a scenario of anonymity. However, the "means likely to be used" and "available technology" come with legal uncertainty for IT-architects. What could be considered anonymous data today, could be personal data in five years or some entities could have a high computing power available to attack the encryption while others don't. Furthermore, as many blockchain-applications will operate in use-case scenarios that make it necessary to identify a specific person, true anonymity would not be a feasible design-decision there anyway.

From the standpoint of Privacy by Design, we propose as a groundrule: the more difficult an IT-system renders the option to identify a person behind blockchain-data and the closer it comes to anonymity, the more compliant it is with the virtues of the GDPR. We therefore suggest to develop a design that clearly appoints certain actors to be able to point (pseudonymous) blockchain-data to a specific person by using additional information while keeping the dataset as unidentifiable as possible for other actors.

Controller

Apart from many innovative rules that aim to update the law of data protection from the old Directive 95/46/EC to a legal framework that reacts to "rapid technological developments and globalisation" (Recital 8 GDPR), the legal doctrine forming the foundation of the GDPR still reflects a limited technological understanding–at least when it comes to methods of decentralized and distributed IT-systems. The GDPR bears in mind administrators but not Peer-2-Peer-networks. By addressing mainly the **controller** as the target of the duties of the GDPR–defining him as "the natural or legal person (...) which, (...) determines the purposes and means of the processing of personal data" (Art. 4 subsection 7)–the regulation focuses mainly on entities which have the ability to actively control the data-flow of an IT-system. Blockchain-technology breaks with this understanding. While, in permissioned blockchains, the entity who manages the key infrastructure potentially also determines the purpose and means to a certain degree that will in most cases make them the controller, in permissionless blockchains there is no obvious controller: the miners have an economical interest in the transaction but are not concerned with the (personal) content of the distributed ledgers, and the programmers lose their influence after the blockchain is set into motion. As a result, only each individual node is, legally, in control [9].

However, the (new) category of **joint controllers** in Art. 26 GDPR may apply, if the nodes "jointly determine the purposes and means of processing." The provision opens a way to represent more complex computational relationships with equal responsibility–and it could even reach out to cover decentralized scenarios like blockchain technology. However, it is not yet clear whether the duty to transparently "determine their respective responsibilities" in an explicit arrangement (to make available for the data subject, Art. 26 sec. 2 GDPR) is the *cause* of joint controllership or rather its *consequence*. In case only those who explicitly agree to be joint controllers would fall under Art. 26 GDPR, there would be only but a small incentive to actually make use of the new category in blockchain scenarios. Determining whether or not to accept and share a legal responsibility would lie solely in the hands of the nodes of a decentralized network. Before this background, it seems rather likely that the law attempts to solidify objective requirements for joint controllership: the new category would then cover all (factual) situations of equal influences on the purposes and means of processing–and require them to make an arrangement. Every infringement of this duty in Art. 26 GDPR could lead to drastic sanctions (Art. 83 sec. 4 lit. a GDPR).

But the question of whether a blockchain-network is really a case of joint control remains hotly contested. Some voices in academia have argued against it, stating that the rules of a blockchain-network stem not "from an agreement of the nodes, but ultimately merely the sum of their independent behaviour" [4]. Whether a notion of *intention* to agree is necessary can surely be questioned, however. The fact that nodes have equal influence and freedom to choose (or start) a certain blockchain-network–and can, for example with the necessary majority or by a Fork, change the rules–rather argues the opposite. These points make a convincing case for the interpretation that blockchain-networks should be considered a subset of joint controllership (Art. 26 GDPR)–with the result that a transparent agreement about the responsibilities becomes the prerequisite for a compliant application (and otherwise sanctions may apply). Blockchain developers would therefore be forced to consider the liability side of data protection already in the design stage–another layer on top of (other) privacy questions. However, this could lead to a huge downfall for the adaption of blockchain-networks and hamper

the innovative potential of decentralization behind blockchain technology. It would lead to the questionable result, that a supervisory body could just pick any node of a (permissionless) blockchain-network and sanction them for the mutual behaviour with thousand other unknown users.

Since the academic and legal discussion about the question "who is the controller of a (permissionless) blockchain?" is still rather in its infancy, certification mechanisms could play a crucial role in narrowing down architectural decisions while other questions still remain unsolved.

Right to Erasure

Even if–with some effort of legal interpretation–personal data is concerned and a controller is found: How can a blockchain-based system implement the **data subject's rights**, such as "the obligation to erase personal data without undue delay" put forward in Art. 17 sect. 1 GDPR or even "take reasonable steps (...) to inform controllers which are processing the personal data that the data subject has requested the erasure by such controllers of any links to, or copy or replication of, those personal data" (Art. 17 sec. 2 GDPR)? While in permissioned blockchain scenarios technical steps have been identified to erase data without interrupting the functionality of the blockchain [1], in permissionless blockchain scenarios such as Bitcoin, no single node is able to efficiently eliminate a set of personal data requested for erasure or inform the network about such a request [9]. It's one of the main challenges for blockchain developers to comply with "the right to be forgotten" in Art. 17 GDPR [5]. In an ideal scenario, the participants of a blockchain-network would agree on an effective process to (jointly) execute a lawful request to erase personal data from the decentralized ledgers.

INTERMEDIATE RESULT AND FUTURE WORK

A technical framework based on blockchain technology must find ways to cope with and implement the manifold legal requirements of the GDPR, while legislators are called upon to seek new forms of legal doctrine that stay abreast of technological changes–especially if they (like blockchain) bear the possibility of decreasing the dangers of uncontrolled, nontransparent, and often unlawful processing of sensitive personal data.

The certification mechanisms specifiying the Privacy by Design doctrine (Art. 25 sec. 3 GDPR) can serve as a tool to find a common way between legal requirements and technical design decisions. They can mark minimal requirements or high standards for GDPR-compliant IT-solutions. In future research, we will address additional aspects beyond the data subject's consent (as a central vehicle of self-determination) by "translating" legal requirements into architectural blueprints.

CONCLUSION

This paper has shown that architectural blueprints can serve as a methodological tool to translate legal into technical requirements in a comprehensible way. An architectural blueprint's main function is not to be implemented in a specific product, but rather to give a technical audience–those capable of creating software for production–an idea of a technical reflection of legal demands about IT-systems.

Even though blockchain was not the main component of the proposed architecture, we regard it as a cornerstone in enabling decentralized, trustworthy transactions between a multitude of pseudonymous participants and believe it has the potential to make the digital sphere a safer place for personal data. But it has to challenge numerous difficulties complying with the GDPR. The ideas put forward in this paper might serve as a starting point to substantiate the principle of Privacy by Design (Art. 25 GDPR) for the practical use of blockchain technology.

There is one more rather political question to consider in discussing "Privacy by Design." To fully adopt and implement the paradigm of "Privacy by Design," we must recognize transparency as an important attribute of not only the data itself but also the code handling the personal data (open-source). Knowing what a system does with our data is the only way of allowing educated data subjects to identify risks themselves. For this reason, we have deliberately chosen to represent the concept of the data subject's consent such that the responsibility of providing personal data lies, both legally and technically, in his or her own hands. By representing the consent of the data subject in a smart contract ecosystem, we make the processing of personal data a question of control rather than trust. Additionally, we are proposing the design of Test Suites, which allow for a technical verification of compliance to the GDPR of source-code.

The idea proposed in this paper goes beyond a mere reconceptualization of data handling. In times of the emerging Web3.0 and key features of decentralized ledgers, consensus-based transaction endorsement, and trust through transparency instead of accountability, we want to also introduce a new way of thinking about how our IT-systems interact with each other and how we should evaluate data locality and validity. From a technical perspective, the results of our work bear certain similarities to those in [12], indicating that the legal requirements of the GDPR indeed ask for a reconceptualization of data handling that could finally become feasible for the mainstream internet-user. We envision a near-future scenario in which self-hosting one's personal data is as routine as logging in to Facebook. In their paper, [12], Zyskind et. al. showed how a system might be designed that not only aligns with our proposed architecture but also manages to track access to personal data on behalf of the data subjects on a blockchain. From the perspective of constitutional law, we believe that blockchain technology can raise the status quo to the ideal of data self-sovereignty for every citizen in comparison to the current design of the digital world. In other words: a high degree of blockchain-based informational self-determination would allow our digital Alter Ego to become what it's supposed to be: Ours.

ACKNOWLEDGMENTS

The authors acknowledge the very helpful copy editing of Saga Briggs. Michael Kolain would like to thank Prof. Dr. Mario Martini for his support. Christian Wirth would like to thank Anja Grafenauer for her support.

REFERENCES

1. Giuseppe Ateniese, Bernardo Magri, Daniele Venturi, and Ewerton Andrade. 2017. Redactable blockchain–or–rewriting history in bitcoin and friends. In *Security and Privacy (EuroS&P), 2017 IEEE European Symposium on*. IEEE, 111–126.
2. Ulrich Baumgartner and Tina Gausling. 2017. Datenschutz durch Technikgestaltung und datenschutzfreundliche Voreinstellungen. In *Zeitschrift fÃijr Datenschutzrecht (ZD)*. IEEE, 308–313.
3. Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. 2014. Deanonymisation of clients in Bitcoin P2P network. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 15–29.
4. Reiner BÃűhme and Paulina Pesch. 2017. Technische Grundlagen und datenschutzrechtliche Fragen der Blockchain-Technologie. In *Datenschutz und Datensicherheit (DuD)*. 473–481.
5. Elke Kunde, Dr. Markus Kaulartz, Med Ridha Ben Naceur, Samater Liban, Matthias Kunz, Prof. Dr.-Ing. Volker Skwarek, Prof. Dr.-Ing. Katarina Adam, Rebekka Weiß, and Marco Liesenjohann. 2018. Faktenpapier Blockchain und Datenschutz. (2018), 40. https://www.bitkom.org/noindex/Publikationen/2018/Leitfaeden/180222-Faktenpapier-Blockchain-und-Datenschutz.pdf
6. Jonathan Katz and Yehuda Lindell. 2014. *Introduction to modern cryptography*. CRC press.
7. Bert-Jaap Koops and Ronald Leenes. 2014. Privacy Regulation Cannot Be Hardcoded. A Critical Comment on the ‘Privacy by Design’ Provision in Data-protection Law. *Int. Rev. Law Comput. Technol.* 28, 2 (May 2014), 159–171. DOI: http://dx.doi.org/10.1080/13600869.2013.801589
8. Ora Lassila and James Hendler. 2007. Embracing" Web 3.0". *IEEE Internet Computing* 11, 3 (2007).
9. Mario Martini and Quirin Weinzierl. 2017. Die Blockchain-Technologie und das Recht auf Vergessenwerden. *Neue Zeitschrift für Verwaltungsrecht (NVwZ)* (2017), 1251 – 1259.
10. Christian Wirth and Michael Kolain. 2016. Speed Dating on Smart Contracts. In *Proceedings of the International Conference for E-Democracy and Open Government*. Parycek/Edelmann (eds.), 201–204.
11. Christian Wirth and Michael Kolain. 2017. Multichain Governance. In *Recht 4.0, Innovationen aus den rechtswissenschaftlichen Laboren*. Oldenburger Verlag fÃijr Wirtschaft, Informatik und Recht, 833–845.
12. Guy Zyskind, Oz Nathan, and others. 2015. Decentralizing privacy: Using blockchain to protect personal data. In *Security and Privacy Workshops (SPW), 2015 IEEE*. IEEE, 180–184.

TRADE: A Transparent, Decentralized Traceability System for the Supply Chain

Mourad el Maouchi
Delft University of Technology
Delft, The Netherlands
mourad@elmaouchi.com

Oğuzhan Ersoy
Delft University of Technology
Delft, The Netherlands
o.ersoy@tudelft.nl

Zekeriya Erkin
Delft University of Technology
Delft, The Netherlands
z.erkin@tudelft.nl

ABSTRACT

Traceability has become an increasingly important aspect of the supply chain in the last few years due to customer awareness as well as better planning and problem identification. Unfortunately, technological, legal, and organizational concerns limit the possibility to utilize a centralized system to achieve traceability. Trust is one of the most important factors preventing the appliance of a centralized system.

Previous works provided several approaches to create a decentralized traceability system. However, these works do not state the feasibility of their work and its appliance for the supply chain. In this paper, we propose a fully transparent and decentralized traceability system for the supply chain, namely TRADE. The system leverages the actors and supply chain structure to achieve traceability. Moreover, consumers and other parties can view all the data in the system and verify the claims of actors on the products. The latter results in positive brand reputation and auditability.

Author Keywords

Blockchain; decentralized system; traceability; supply chain; transparency; auditability.

INTRODUCTION

The supply chain has experienced several highlights in the traceability aspect throughout the last few decades. Especially in the food industry, there have been severe experiences where tracing the product life cycle is crucial such as the mad cow disease and the Asian bird influenza [11, 7]. Traceability is increasing in importance every day for the actors in the supply chain to improve the performance of the business as well as compliance with (inter)national regulations. Besides the supply chain actors, other parties such as consumers, Non-Governmental Organizations (NGOs), governments, suppliers, and buyers show an increase in demand for information regarding their products and materials.

To achieve traceability, a system is required that records and follows the trail of products [2]. The interconnected nature of the supply chain makes it difficult to introduce a centralized system in control of a third party, requiring a high level of trust. The limited amount of trust resulted in separate systems, limiting the possibility to achieve traceability throughout the entire supply chain.

Blockchain technology, first introduced with Bitcoin in 2009 [18], is rapidly increasing as a key technology to address the trust aspect by removing the necessity of having a trusted third party. Blockchain technology has been successfully applied to several industries throughout the years, such as the energy [3] and finance sector [20]. For the supply chain, approaches have been suggested in a theoretical manner without sufficient analysis [14, 22, 5].

In this paper, we address the trust aspect and propose a transparent, decentralized traceability system for the supply chain. TRADE is, to the best of our knowledge, the only system that provides a fully transparent, analyzed and feasible traceability system for the supply chain. This paper is constructed as follows. First, we discuss previous works in Section 2. We present our proposed system, TRADE, in Section 3, wherein Section 4 we discuss the validation mechanisms used in our proposed system. In Section 5, we analyze the security and performance implications of TRADE, along with experimental results from a proof-of-concept implementation. Finally, a discussion and concluding remarks are provided in Section 6.

LITERATURE REVIEW

In 2016, Kim et al. proposed an ontology-based smart contract design of a proof-of-concept traceability system using blockchain technology for the supply chain [14]. Their work shows the appliance of ontologies in their setting, rather than a focus on the blockchain appliance for the supply chain and its real-world feasibility.

Furthermore, Feng Tian combined RFID tags and blockchain technology to create a traceability system for the agri-food supply chain in China [22]. Tian discussed that a decentralized approach for traceability could solve the issues in a centralized approach, namely: trust, fraud, corruption, tampering and falsifying information. In [22], the analysis discusses the blockchain technology and traceability as separate aspects. However, the combination might introduce deficiencies concerning feasibility and performance. The proposed system has also not been implemented to validate the claims.

Abeyratne et al. provided a broader view of traceability and transparency in their work [5]. In their work, transparency

ERCIM-Blockchain 2018 El Maouchi, Mourad; Ersoy, Oğuzhan and Erkin, Zekeriya (2018): TRADE: A Transparent, Decentralized Traceability System for the Supply Chain. In: W. Prinz & P. Hoschka (Eds.), Proceedings of the 1st ERCIM Blockchain Workshop 2018, Reports of the European Society for Socially Embedded Technologies (ISSN 2510-2591), DOI: 10.18420/blockchain2018_01

ACM ISBN 978-1-4503-2138-9…$15.00
DOI: 10.18420/blockchain2018_01

is argued based on the child labor scandal of Nike in 1996, whereas sustainability of products is built upon the importance of understanding the product's life-cycle [6, 8]. Abeyratne et al. discuss that the characteristics of blockchain technology can enhance trust through transparency and traceability within the supply chain. However, their work examines an example, rather than a practical appliance of blockchain technology in the supply chain.

TRADE

In this section, we design our proposed system, namely TRADE. The goal of the system is to introduce a single system for the actors to transfer product data and track products throughout the supply chain. The minimal trust between the actors makes a centralized system, in control of one party, infeasible. Therefore, we use blockchain technology as a communication network. The blockchain is an immutable record keeping system where data cannot be altered, and a product is in possession of a single actor. In the system, only authorized actors can participate and add information to the system. Nevertheless, everyone can view the stored data. The authorization to the system is handled by a central authority (CA).

This section first describes the preliminaries. Next, the system model and the accompanying actors are discussed. Lastly, the structure of transactions and the process per actor is explained.

Preliminaries

Digital signature schemes are mathematical schemes for demonstrating the authenticity of digital data or documents. Digital signatures are made possible by public-key cryptographic schemes and provide the following properties: *authentication*, *non-repudiation* and *integrity*. In our proposed system, we use the Elliptic Curve Digital Signature Algorithm (ECDSA) for the digital signatures [13], which provide smaller keys and signatures compared to RSA [12]. ECDSA is a NIST-approved digital signature algorithm [19].

System Model

TRADE consists of five actors, namely Producers, Transporters, Processors, Distributors, and Retailers. The actor types are based on the modeled supply chains in [23, 15]. We assume that the actors are distributed across different geographic locations, and they are willing to cooperate in reducing costs and improving planning algorithms by deploying a single system.

We assume that a Producer creates a product and then transports it via a Transporter to a Processor. A Processor performs internal processes on the product, which is further transported to a Distributor via a Transporter. The Distributor then distributes the end product to its final destination: a Retailer.

The actors in the system create transactions, containing product information, that is then broadcasted directly to the nodes in the network. The entire network validates the broadcasted transactions. A set of validated transactions is aggregated, by an arbitrary node, and a *block* is created, which is validated afterward. Note that we define a transaction or block as valid if it fulfills a set of requirements that are described later in Section 4. We use a public permissioned blockchain, denoted as *BC*, as a decentralized solution for TRADE, where BC_{gb} denotes the genesis block. Figure 1 depicts a schematic flow diagram of an exemplary supply chain and the data-flow in TRADE.

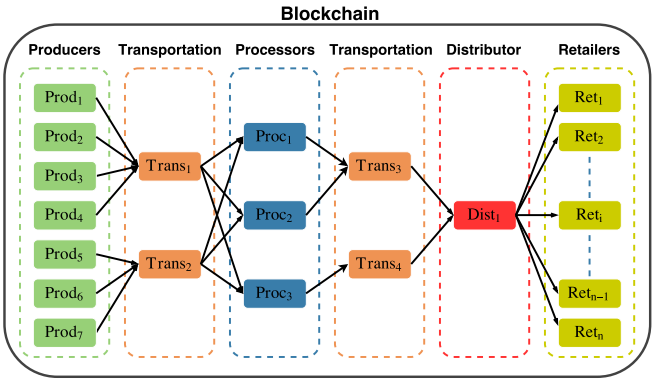


Figure 1: A schematic flow of TRADE.

System Procedures

Each transaction in TRADE is denoted as tx_h, where h denotes the hash value of the transaction. The transaction tx_h is a tuple, where $tx_h = \langle a, p_{id}, k, in, out, info, t, Sig(tx_h)\rangle$. The transaction structure and its description is shown in Table 1. We denote $x[y]$ as the element y in x and `INRF` as "if not, return false" for our proposed algorithms.

Table 1: Transaction Structure

FIELD	DESCRIPTION
a	Actor issuing the transaction.
p_{id}	Unique ID for a product.
k	Number of products.
in	Hash of the previous transaction.
out	Receiver of the transaction.
$info$	List of additional information.
t	Date and time of the transaction.
$Sig_a(tx_h)$	The signature, by a, on tx_h.

Initialization

Each actor in the network performs the key-pair generation algorithm for ECDSA. We denote a key-pair as (pk_a, sk_a), where a denotes the actor. The public key of each actor is shared with the CA. The CA is consulted in case an actor does not hold pk_a of the actor that signed the transaction. Furthermore, each actor holds a list *PID* which contains all the p_{id}'s. The list is used to check if any newly registered p_{id} is unique.

Production

A Producer, denoted as $PD_i \in PD$, creates a product with a unique ID p_{id}. Afterwards, the Producer creates the additional information $info = \{dest\}$, where *dest* is the Processor $PS_i \in PS$. Since the Producer creates a new product and accompanying p_{id}, he is unable to link it to a previous transaction and

thus links it to the genesis block BC_{gb}. The final transaction is created as $tx_h = \langle PD_i, k, p_{id}, BC_{gb}, T_j, info, t, Sig_{PD_i}(tx_h)\rangle$, where *out* is set as the Transporter $T_j \in T$. Algorithm 1 shows the validation for a tx_h by a Producer.

Algorithm 1 Transaction Validation: Producer

```
1: procedure VALIDATION_PRODUCER(tx_h)
2:     Check in = BC_gb; INRF.
3:     Check p_id ∉ PID; INRF.
4:     Check info[dest] ∈ (PS_i ∈ PS); INRF.
5:     return true
6: end procedure
```

Transportation

The Transporter, denoted as $T_i \in T$, creates a transaction tx_h when he places the product in a means of transportation. He receives a product from a Producer or Processor and transfers it to a Processor or Distributor, respectively. He sets $info = \{src, dest, V_{ID}, SSCC\}$, where *src* is the actor that provided the product, *dest* is the destination actor, V_{ID} is the vehicle ID for transportation and *SSCC* is the Serial Shipping Container Code in which the product is placed, defined by *GS1* [4]. The complete transaction is denoted as $tx_h = \langle T_i, p_{id}, k, in, out, info, t, Sig_{T_i}(tx_h)\rangle$, where *out* is either a Processor or a Distributor, based on *in*. Algorithm 2 shows the validation for a tx_h by a Transporter.

Algorithm 2 Transaction Validation: Transporter

```
1: procedure VALIDATION_TRANSPORTER(tx)
2:     Check info[src] ∈ {PD, PS}; INRF.
3:     Check info[dest] ∈ {PS, D}; INRF.
4:     Check out = info[dest]; INRF.
5:     return true
6: end procedure
```

Processing

A Processor, denoted as $PS_i \in PS$, performs internal processes on p_{id}, such as combining materials, testing or sanitizing the product, denoted as *IP*. The Processor sets $info = \{dest, IP\}$, where *dest* is the final recipient, which is a Distributor. The complete transaction is denoted as $tx_h = \langle PS_i, p_{id}, k, in, T_j, info, t, Sig_{PS_i}(tx_h)\rangle$, where *out* is set to the recipient Transporter $T_j \in T$. Algorithm 3 shows the validation for a tx_h by a Processor.

Algorithm 3 Transaction Validation: Processor

```
1: procedure VALIDATION_PROCESSOR(tx)
2:     Check info[IP] ≠ ∅; INRF.
3:     Check info[dest] ∈ D; INRF.
4:     Check out = (T_j ∈ T); INRF.
5:     return true
6: end procedure
```

Distribution

A Distributor, denoted as $D_i \in D$, creates a transaction upon distribution of p_{id}. The Distributor sets $info = \{src, V_{ID}, SSCC\}$, where *src* is the Processor that sent the product to D_i and recall the definition of V_{ID} and *SSCC* as aforementioned. The complete transaction is set up as $tx_h = \langle D_i, p_{id}, k, in, out, info, t, Sig_{D_i}(tx_h)\rangle$, where *out* is set to a Retailer $R_j \in R$. Algorithm 4 shows the validation for a tx_h by a Distributor.

Algorithm 4 Transaction Validation: Distributor

```
1: procedure VALIDATION_DISTRIBUTOR(tx)
2:     Check info[src] ∈ PS; INRF.
3:     Check out ∈ (R_j inR); INRF.
4:     return true
5: end procedure
```

Retailer

The Retailer, denoted as $R_i \in R$, is the end-actor that eventually sets the products for sale. This actor does not create a transaction. Therefore, retailers do not actively participate in the system, but rather function as an end-station for the products throughout the supply chain.

VALIDATION

Validation of Transaction Authenticity

Digital signatures are applied to prevent forgery and to proof the integrity of a transaction in TRADE. Each transaction holds a signature, made by the creator *a* using his private key sk_a. Anyone with the public key pk_a of *a* can validate the signature. The creator *a* is the only party capable of signing tx_h since he is the only one in possession of sk_a. The integrity of a transaction is held since an altered transaction results in an invalid digital signature. An invalid signature results in an invalid transaction.

Validation of Transactions

The validation of a transaction is dependent on the actor that created the transaction. Recall that a transaction is a tuple containing multiple fields, as shown in Table 1. Actors, upon receiving a transaction, need to check each field of the transaction. In Algorithm 5, we combine our previous proposed algorithms in a single algorithm to validate a transaction. Note that if one of the Check calls returns false, then the execution aborts and returns false. Therefore, in order to return true at the end, all of the Check calls must return true. The same procedure applies for the other validation processes.

Algorithm 5 Transaction Validation

```
1: procedure VALIDATION_TX(tx_h)
2:     ∀x ∈ tx_h, x ≠ null; INRF.
3:     Timestamp of tx_h < current timestamp; INRF.
4:     Validate digital signature of tx_h.
5:     Check Validation_Producer(tx_h) = true; INRF.
6:     Check Validation_Transporter(tx_h)) = true; INRF.
7:     Check Validation_Processor(tx_h) = true; INRF.
8:     Check Validation_Distributor(tx_h) = true; INRF.
9:     return true.
10: end procedure
```

Validation of Blocks

A number of transactions are collected and aggregated in a block, which is broadcasted to the network and requires validation. Note that the validation of a block is different than the validation of a transaction. The block structure is similar to the one described in Bitcoin[1]. Let b be a block and $b[TX]$ be the transaction list in b. We propose an algorithm, described in Algorithm 6, that validates a block.

Algorithm 6 Validation of a Block

```
1:  procedure BLOCK_VALIDATION(b)
2:      Check the syntactic correctness of b.
3:      Check that no duplicate of b exists.
4:      Check length of b[TX] > 1; INRF.
5:      Validate Merkle root.
6:      for each tx_i ∈ b do
7:          Check Validation_TX(tx_i) = true; INRF.
8:      end for
9:      Relay block all actors.
10:     return true.
11: end procedure
```

ANALYSES

In this section, TRADE is analyzed in three dimensions: security, performance and experimental results. Firstly, we discuss the security imposed by the system. Secondly, we provide a theoretical analysis of the performance of the computational and communication complexities. Finally, we discuss the measurements obtained from a proof-of-concept implementation to show the practical performance of the system.

Security Analysis

TRADE does not allow any unauthorized participation since it uses a public permissioned blockchain. The consensus model provides the integrity of the block structure, and the signature algorithm secures the transactions.

For TRADE, the consensus model preserves the integrity of a propagated block. TRADE does not enforce a specific consensus model. There are several models available that can be used for our system [16, 24]. The security of the blocks is thus dependent on the chosen consensus model.

TRADE uses digital signatures to provide authenticity and integrity of each transaction, where ECDSA is used as the digital signature scheme. The security of ECDSA relies on the elliptic curve discrete logarithm problem (ECDLP), which is considered to be computationally hard [13]. Therefore, the security of a digital signature, and thus the transaction, is kept under the ECDLP assumption.

The source and destination of a transaction are viewable to everyone. Also, the throughput of an actor can be derived by using the timestamp in combination with the amount in a transaction. Our proposed system does not take privacy concerns into account and thus is not envisioned to be preserved.

Note that TRADE do not tackle the problem of proving physical delivery of the products in the supply chain. We assume that the delivery can be verified by a tracking item such as RFID tags. Therefore, an malicious actor cannot claim that a false delivery or missing delivery of products because of the proof of the physical delivery mechanism.

[1]Bitcoin block structure: https://en.bitcoin.it/wiki/Block

Computational Complexity

For the analysis of the computational complexity, we list the number of operations performed by each actor in three aspects: (i) the creation of transactions, (ii) validation of transactions and (iii) the validation of blocks. The amount of performed operations depends on a number of variables, listed in Table 2.

Recall that a transaction consists of a set of values. The only computed value is the digital signature. Therefore, we focus on the computation complexity of the digital signature scheme. In Table 3, the amortized number of operations for the aforementioned aspects are listed.

Table 2: Parameters used in the computational analysis.

SYMBOL	DESCRIPTION
$\mathcal{N}$	Number of actors in the network.
γ	Number of transactions per minute, by an actor.
ℓ	Number of transactions in a block.
s	Key-size in bits for the elliptic curve.

Transaction Creation

For the creation of a transaction, a digital signature is created. The digital signature procedure is dependent on the key-size s for the chosen elliptic curve. The computational complexity, per transaction, is thus linear in s.

Transaction Validation

The computational complexity of the validation of a transaction depends on the digital signature. The validation procedure of a digital signature is, equal to the creation, dependent on the key-size s.

Block Validation

The validation of a block has the highest computational complexity. Firstly, the Merkle root is required to be validated, which requires multiple hashing operations and is computed in $\log(\ell)$ [21]. Then, each transaction is validated inside the block. The verification of ℓ digital signatures requires $s\ell$ verifications per block. Since $s\ell \gg \log(\ell)$ for $\ell > 1$, the block validation procedure is dominated by the validation procedure of digital signatures. Consequently, the block validation has a computational complexity of $\mathcal{O}(s\ell)$.

Communication Complexity

To analyze the communication complexity of TRADE, we list the number of communications required on the network for the broadcast of a transaction and a block. The required communication depends on a number of variables in Table 2.

In the initialization phase, each actor sends their public key to the CA and requires $\mathcal{N}$ communication rounds. This procedure only re-occurs if an actor updates their key-pair. The

Table 3: Computational complexity of the operations in TRADE.

PROTOCOL	ACTOR
Transaction Creation	$\mathcal{O}(s)$
Transaction Validation	$\mathcal{O}(s)$
Block Validation	$\mathcal{O}(s\ell)$

public keys of the actors are stored locally by each actor to reduce the communication rounds necessary. Next, each transaction is broadcasted to the network, which requires $\mathcal{N}-1$ rounds with the assumption that each actor knows each other and their addresses on the network allowing a direct connection. The same applies to the broadcast of blocks. Since the initialization phase only occurs at the beginning of the system, the communication complexity is dominated by the broadcast procedure for transactions and blocks. Therefore, the communication complexity of TRADE is $\mathcal{O}(\mathcal{N})$.

Experimental Results

To measure the runtime of TRADE, we created a proof-of-concept implementation of the system in `Python` 2.7 by creating a simple blockchain implementation based on the work of Daniel van Flymen[2] and the `fastecdsa` package[3]. The p_{id} values are represented as 32-bit fixed-point numbers.

The measures of the runtime were executed on our commodity hardware, running macOS 10.13 on a dual-core 3^{rd} generation 2.9GHz Intel® Core i7 processor with 16GB RAM. We measured the runtime for the transaction and validation of a transaction. For accurate measurements, we executed 1000 iterations for each procedure. We use the NIST P-curves for our measurements. Figure 2 shows the impact of s on the runtime for transaction creation and validation. It is clear that the procedures grow quadratically based on s. Using $s = 256$ for an elliptic curve, each actor is able to create approximately $\frac{1}{2.84 \cdot 10^{-3}} = 351$ transactions per second and validate transactions at a speed of $\frac{1}{2.28 \cdot 10^{-3}} = 437$ transactions per second. For the latter, an actor can validate $437/\ell$ blocks per second, depending on ℓ.

Even though our blockchain framework does not rely on a specific consensus protocol, for the experimental results, we implemented a naive Proof-of-work consensus protocol [18]. At the same time, there have been several consensus proposals which achieve 10-100x throughput of the Bitcoin's proof-of-work protocol, such as Bitcoin-NG [9], Honey Badger [17] and Algorand [10]. Therefore, it is important to note that there are consensus protocols which can securely handle the number of transactions required in our TRADE framework.

There are approximately 32.9 million shipping containers globally as of 2013 [1]. Based on the assumption that a container changes possessor up to 100 times per year, and for each time a transaction is made, approximately 317 transactions are made per second. The supply chain requires fewer transactions per second than all containers globally. Given our experimental results, it is clear that our system achieves the required performance to be applied in a real-world setting.

[2]A simple Blockchain Implementation, https://github.com/dvf/blockchain

[3]fastecds: https://pypi.python.org/pypi/fastecdsa

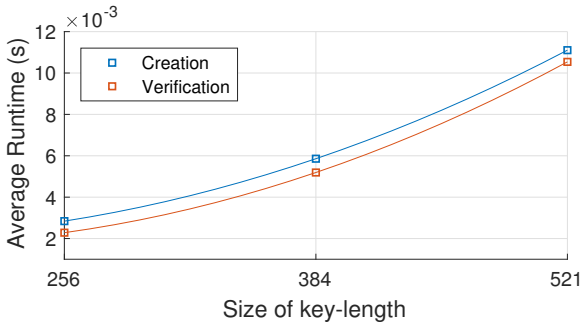


Figure 2: Average computation time for the creation and validation of a single transaction, based on s.

DISCUSSION AND CONCLUSIONS

In previous works [14, 22, 5], researchers proposed several frameworks to achieve a decentralized, traceability system. The previous works are purely theoretical, and thus do not provide any implementation. Furthermore, no complexity of the approaches is given. Therefore, the feasibility of the previous works is missing.

In this paper, we proposed TRADE, a fully transparent, decentralized traceability system for the supply chain. Each actor creates a transaction regarding a product p_{id} containing the full information on the product. The stored data inside a transaction is fully transparent allowing each actor in the network to view the data. Each transaction is signed by the issuing actor using a digital signature, providing a proof of authenticity, integrity, and non-repudiation. The valid transactions are aggregated in a block and broadcasted to the network. Each transaction regarding a product p_{id} is linked throughout the supply chain on the blockchain, providing full traceability and insight for each actor. The insight on the data can be used to improve planning and scheduling, and faster recalls for the supply chain. Also, consumers can also view this data and gain insight into the full life-cycle of products. Standardization is enforced in TRADE since each transaction, depending on the issuing actor, has a corresponding validation procedure.

TRADE achieves a significant performance to create and validate transactions, as well as the validation of blocks. We show that it is feasible to apply blockchain technology for the supply chain to achieve traceability. Moreover, consumers and other parties can view the data to gain knowledge on the procedures performed on their product as well as information on the sustainability, if the actors provide it. Actors are in control to share such information, which is recommended since it aids the company brand and increases the trust of consumers in the company. In case actors are willing to share data in a single system and achieve full traceability, blockchain technology is shown feasible to accomplish this in a real-world setting for the supply chain.

Future Work

TRADE is a generic blockchain framework for traceability systems. As being said, there are open research questions for specific use cases. An important open research question is about privacy: for some supply chain mechanisms, the actors might be competitors of each other and do not enjoy the transparent system. For these cases, privacy-preserving traceability system should be designed. Another research question is about the performance of TRADE: the naive implementation given in the paper does not provide the upper limit of the throughput of TRADE. It needs more research to investigate the performance impact of the parameters like the scripting language, consensus protocol or the blockchain platform itself. Also, performance analysis of TRADE in the existing blockchain platforms can be explored.

REFERENCES

1. 2013. World Shipping Council. (2013). http://www.worldshipping.org/about-the-industry/containers/global-container-fleet
2. 2014. ISO 9000 2015 Definitions. (Nov 2014).
3. 2017. *Blockchain X Energy, A Natural Match*. Standard. Blocklab.
4. 2017. Serial Shipping Container Code (SSCC). (Jan 2017). https://www.gs1.org/serial-shipping-container-code-sscc
5. Saveen A Abeyratne and Radmehr P Monfared. 2016. Blockchain ready manufacturing supply chain using distributed ledger. (2016).
6. Ann Baier. 2005. Organic certification process. *National Sustainable Agriculture Information Service.[Accessed 21 March 2012]* (2005).
7. W Chansud, J Wisanmongkol, and U Ketprom. 2008. RFID for poultry traceability system at animal checkpoint. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-CON 2008. 5th International Conference on*, Vol. 2. IEEE, 753–756.
8. Sara D Elder, Hisham Zerriffi, and Philippe Le Billon. 2013. Is Fairtrade certification greening agricultural practices? An analysis of Fairtrade environmental standards in Rwanda. *Journal of Rural Studies* 32 (2013), 264–274.
9. Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert van Renesse. 2016. Bitcoin-NG: A Scalable Blockchain Protocol. In *13th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2016, Santa Clara, CA, USA, March 16-18, 2016*. 45–59. https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/eyal
10. Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28-31, 2017*. 51–68. DOI:http://dx.doi.org/10.1145/3132747.3132757
11. Elise Golan, Barry Krissoff, and Fred Kuchler. 2004. Food traceability. *Amber Waves* 2, 2 (2004), 14.
12. Nicholas Jansma and Brandon Arrendondo. 2004. Performance comparison of elliptic curve and rsa digital signatures. *nicj. net/files* (2004).
13. Don Johnson, Alfred Menezes, and Scott Vanstone. 2001. The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security* 1, 1 (2001), 36–63.
14. Henry M Kim and Marek Laskowski. 2016. Towards an ontology-driven blockchain design for supply chain provenance. (2016).
15. Bhuwan Maharjan, Bikash Bhagat, Laxmina Shrestha, Madhu Sudan Koirala, Saroj Shrestha, and Supriya Tamrakar. Supply Chain Analysis for Bread. (????). https://www.scribd.com/doc/187747813/Supply-Chain-Management-of-Bread
16. Juri Mattila. 2013. The blockchain phenomenon. *Reuters* (2013), 1–7. http://blogs.reuters.com/felix-salmon/2013/04/09/the-disruptive-potential-of-native-advertising/
17. Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. 2016. The Honey Badger of BFT Protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*. 31–42. DOI: http://dx.doi.org/10.1145/2976749.2978399
18. Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).
19. National Institute of Standards and Technology. 2013. FIPS PUB 186-4 FEDERAL: Digital Signature Standard (DSS). *Processing Standards Publication* July (2013), 1–119. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
20. Paweł Szewczyk. 2016. The Potential Impact of the Blockchain Technology on the Financial Sector. *Finance Today and Tomorrow: Opportunities, Threats, and Challenges* (2016), 63.
21. Michael Szydlo. 2004. Merkle tree traversal in log space and time. In *Eurocrypt*, Vol. 3027. Springer, 541–554.
22. Feng Tian. 2016. An agri-food supply chain traceability system for China based on RFID & blockchain technology. In *Service Systems and Service Management (ICSSSM), 2016 13th International Conference on*. IEEE, 1–6.
23. Sean P Willems. 2008. Data set - Real-world multiechelon supply chains used for inventory optimization. *Manufacturing & Service Operations Management* 10, 1 (2008), 19–23.
24. Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. 2017. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*. 557–564.