Towards Cross-Blockchain Transaction Verifications

Marten Sigwart*, Philipp Frauenthaler*, Taneli Hukkinen†, Stefan Schulte*

* Distributed Systems Group
TU Wien, Vienna, Austria
{m.sigwart, p.frauenthaler,
s.schulte}@infosys.tuwien.ac.at

† Pantos GmbH
Vienna, Austria
contact@pantos.io

***Abstract*—Interoperability between blockchains remains an open problem with current interoperability approaches providing very limited means of cross-blockchain interaction. In particular, current solutions for cross-blockchain token transfers suffer from limitations such as excessive synchronization of any balance change across blockchains.**

To overcome these limitations, we describe concepts that enable the verification of transactions across blockchains in a trustless and decentralized manner. These concepts can be used as foundation for enabling decentralized applications such as cross-blockchain token transfers.

I. Introduction

The Token Atomic Swap Technology (TAST) research project[1] aims to create a platform for cross-blockchain interoperability. The overarching goal is to investigate possible means of interconnecting various blockchain-related projects [3]. As a first step towards more general blockchain interoperability, we aim to create a cross-blockchain token [2]. Ideally, such a token enables users to freely choose on which blockchain they hold their assets, i.e., users are not tied to particular blockchains and are able to hold different denominations of the token on multiple blockchains at the same time. Further, if a new blockchain technology emerges offering novel functionality, users are able to transfer their assets to this new blockchain taking advantage of the new capabilities in a way that requires no trust in a third party [7].

While a first prototype of such a cross-blockchain token has been developed [4], it suffers from certain limitations that hinder its practicability, e.g., excessive synchronization of balances across blockchains (and therefore high overheads and cost), inability to distribute asset allocation across chains, and difficulty to integrate new blockchains into the ecosystem [7]. To solve these issues, a cross-blockchain token transfer should only incorporate the two blockchains directly involved in the transfer: If a token is to be transferred from chain A to B, only chains A and B should have to communicate in order to finalize the transfer. All other participating blockchains should remain untouched [5].

To enable a solution that restricts the interaction of a cross-blockchain token transfer to the two blockchains directly involved, a couple of requirements need to be fulfilled [7]. Essentially, a client looking to transfer tokens from blockchain A to blockchain B first has to destroy the tokens on chain A and then provide a proof to chain B that the tokens have in fact been destroyed. Only then should the tokens be (re-)created on chain B.

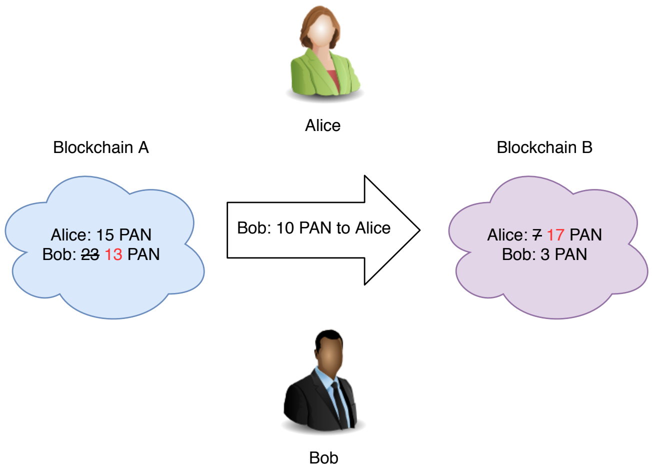

Figure 1: Cross-blockchain token transfer

In the work at hand, we describe fundamental concepts that enable the verification of arbitrary transactions across blockchains. That is, a blockchain can prove the existence of a certain transaction that has occurred on another blockchain in a completely decentralized and trustless manner. In the context of a cross-blockchain token transfer, this allows the destination chain of the transfer to prove the existence of a "destroy" transaction on the source chain of the transfer.

To this end, Section II further provides important background information. Section III describes the fundamental concepts necessary for realizing cross-blockchain transaction verifications. Section IV then gives an outlook on the future work within TAST. Section V concludes the paper.

II. Background

A. Cross-Blockchain Token Transfers

The TAST project aims to enable a cross-blockchain token [2]. Ideally, as stated in Section I, users can choose on which blockchain they keep their tokens with the possibility to freely transfer tokens between blockchains (see Fig. 1). This way, users are not locked-in by particular blockchains and are able to take advantage of new blockchain technologies offering novel capabilities. Furthermore, the distribution of as-

White Paper, TU Wien; May 2019, version 1.0.

[1] http://www.infosys.tuwien.ac.at/tast/

sets across the participating blockchains can give an indication about the significance of a particular blockchain [7].

Naturally, cross-blockchain token transfers should only be successful (i.e., the specified amount of tokens is created on the destination chain) if the same amount of tokens has been burned (i.e., destroyed) on the source chain. If this was not the case, tokens could effectively be created out of nothing since there is no assurance that tokens that are being created on the destination chain have actually been burned on the source chain. Hence, the destination chain has to be certain that the amount of tokens has been destroyed on the source blockchain before (re-)creating the same amount on the destination chain [7].

Fully replicating one blockchain within another blockchain is not feasible in practice [1]. Therefore, a solution is necessary that supplies enough information to the destination blockchain so that it can be certain that the transferred amount of tokens has actually been destroyed on the source blockchain. This information transfer can be done either via oracles [7] or using cryptographic proofs, so-called Simplified Payment Verifications (SPVs) [5]. In the next section, we briefly explain this kind of proof.

B. Simplified Payment Verification

SPVs can be used to cryptographically prove that a certain transaction is part of a blockchain [6]. In general, an SPV consists of two parts. First, it needs to be verified that a certain transaction is part of a specified block. Second, it needs to be verified that the specified block is valid and part of the valid blockchain.

In blockchains like Bitcoin or Ethereum, the transactions of a block are stored as leaves in a Merkle Tree or an extension thereof [6, 8]. Hence, proving that a transaction is part of a block can be done by constructing a so-called Merkle proof of membership [5]. Anyone in possession of the hash pointer to the root node of the Merkle tree can verify such a proof of membership by recalculating the hashes of all nodes along the path from the leaf (i.e., the transaction) up to the root node. If the calculated root hash matches the stored root hash of the verifier, the membership of the transaction has been successfully proven.

However, a Merkle proof of membership is not enough to prove that a certain block itself is valid and part of a blockchain. The validity of important data of a block, such as the block's hash, number (or height), difficulty, and timestamp, is largely determined by the block's parent [8]. Hence, to determine the validity of a block, the verifier essentially needs to know about preceding blocks. Further, forks are a common occurrence in blockchains [8]. While these forks usually consist of valid blocks, only one fork will eventually be accepted as the valid branch of the blockchain (i.e., in proof of work (PoW) blockchains, the branch with greatest total difficulty). Hence, the verifier also needs to determine whether a block is part of the valid branch of the blockchain.

SPVs can be leveraged for cross-blockchain token transfers by executing an SPV on the destination chain of the transfer certifying the existence of the correct "burn" transaction on the source blockchain. However, executing SPVs on-chain encompasses several issues. First, the blockchain executing the SPV (i.e., the destination chain) potentially needs to have a copy of all blocks of the source blockchain. Therefore, participants need to be incentivized to continuously submit new blocks of the source chain to the destination chain [5]. Second, depending on the underlying blockchains, verifying that a block is a valid successor of another block can be computation and storage intensive. Hence, fully validating every submitted block of the source chain on the destination chain can become very expensive. The benefits of being able to execute SPVs on the destination chain need to outweigh the associated cost. Finally, the destination chain needs a way to allow forks of the source chain while at the same time being able to determine the valid branch of the source chain.

In the next section, we describe concepts that enable the on-chain execution of SPVs.

III. Enabling Cross-Blockchain Transaction Verifications

As stated above, verifying SPV proofs on-chain requires the provision of some block information of the source blockchain to the destination chain so that the destination chain can prove that a particular transaction has actually been included in the source chain. However, having to execute a full validation of every submitted block header can become expensive. To keep the cost for the preparation and verification of SPVs comparatively low, it is a good idea to take a liberal approach when accepting new block information of the source chain while also taking the possible blockchain forks into account.

A. Replicating the source blockchain

The destination chain needs to know about the blocks from the source chain in order to execute SPVs. Hence, block information of the source chain has to be continuously submitted to the destination chain. The information required to determine the validity of a block and its membership in the longest PoW chain, as well as the root hash of the Merkle tree are all stored in the so-called block header. Storing these headers requires less space than storing the complete blocks [5]. Thus, whenever new blocks are appended to the source blockchain, relayer nodes forward the block headers instead of the complete blocks to the destination chain (see Fig. 2).

For each block header that the destination chain receives, it performs a light validation:

1) Verify that the block's parent exists on the destination chain.
2) Verify that the block's number is incremented by one.
3) Verify that the block's timestamp is correct.
4) Verify that the block's gas limit is correct.
5) Verify that the block's difficulty is valid.

If these checks are successful, the destination chain accepts the block and stores it internally. Note, the destination chain does not verify the PoW for each block header it receives,

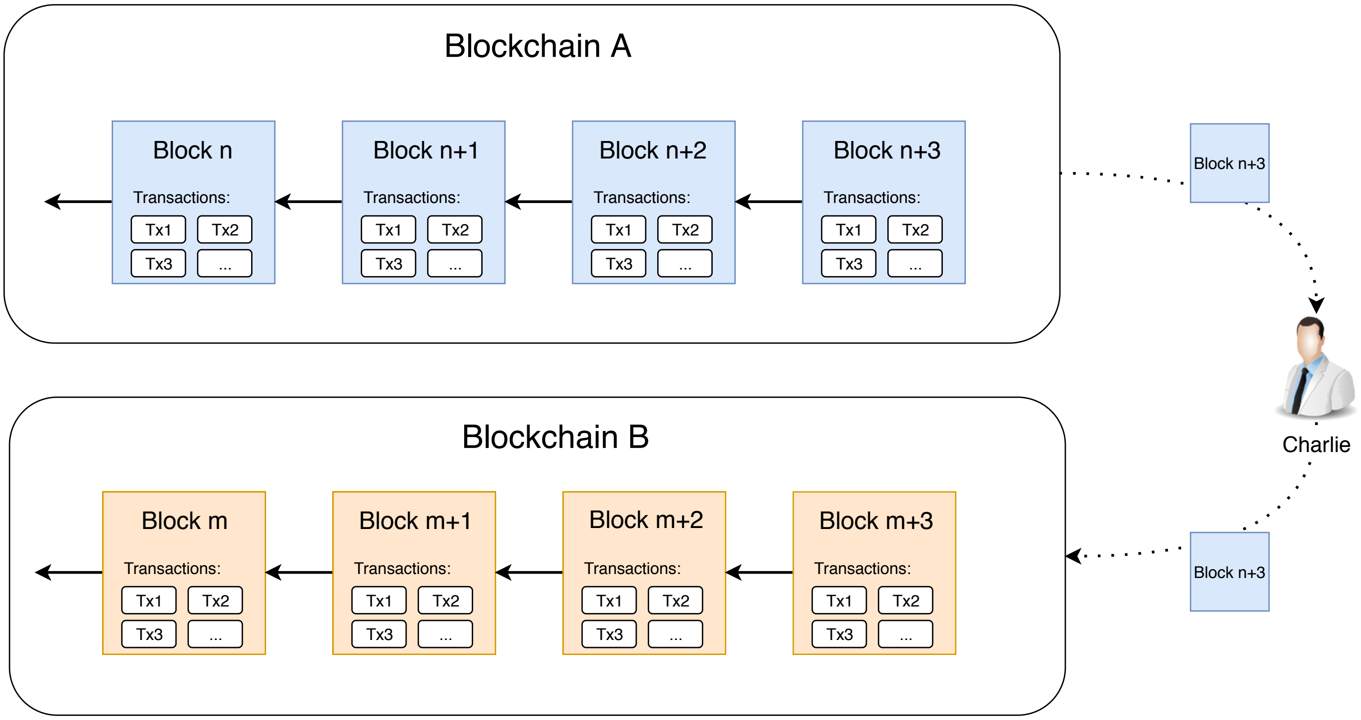

Figure 2: Relayer "Charlie" forwarding block headers of the source chain ("A") to the destination chain ("B")

as validating the PoW for every block header becomes very expensive. Instead, a liberal approach is taken by accepting all block headers in the beginning but assigning a lock period for each block header. Before this period is over, clients cannot request the verification of transactions on this block header. Within this period, clients have the possibility to dispute any block header they think is illegal. In case of a dispute, the full PoW verification is carried out. If the verification fails, the block header and all its successors are removed from corresponding smart contract on the destination chain (see Fig. 3).

Furthermore, the destination blockchain keeps track of the currently longest PoW chain. Whenever a new block header is submitted, the destination chain checks whether the corresponding block header is part of the currently longest chain or if the block header is part of a fork of the source blockchain. In case the fork of the newly submitted block header is longer than the currently longest chain, it becomes the longest chain. Even if a block header is part of a fork which is not the longest chain, it is accepted by the destination chain. However, transaction verifications are only successful on these block headers if they become part of the longest PoW chain in the future. By accepting forks, an attacker might try to overtake the longest chain by submitting a lot of block headers. However, since block headers are assigned a lock period before they can be used for transaction verifications, clients can dispute these illegal block headers before transaction verifications are allowed on the headers.

By taking a liberal approach when accepting submitted block headers, the source blockchain can be closely replicated including potential forks while keeping computational requirements at a minimum since the expensive PoW verification is only carried out if block headers are disputed by clients. This way, the destination chain can reliably provide an answer to the question whether or not a block is part of the source blockchain and whether the block belongs to a fork that is currently accepted as the longest PoW chain.

B. *Verifying Transactions*

Clients can request the verification of a transaction on the destination chain. For that, they send a request in the form of "Is transaction x of block b included in the source chain and confirmed by at least n succeeding blocks?" (see Fig. 4). When the destination chain receives the request, it performs multiple checks. First, the destination chain checks that the block header of block b exists (i.e., the block header has been submitted to the destination chain). Second, it is verified that block b is currently part of the longest PoW chain. Then, it is verified that block b is followed by at least n succeeding blocks (all of these blocks' headers must be known to the destination chain). Further, the destination chain checks that the header of block b and the n succeeding block headers are not locked anymore. That is, they have passed the lock period without having been disputed in the meantime.

If all of the above verifications are successful, the Merkle proof of membership certifying the inclusion of transaction x in block b is verified. The corresponding Merkle proof has to be generated beforehand and submitted together with the verification request by the client requesting the verification. If the Merkle proof is valid, the destination chain confirms to the

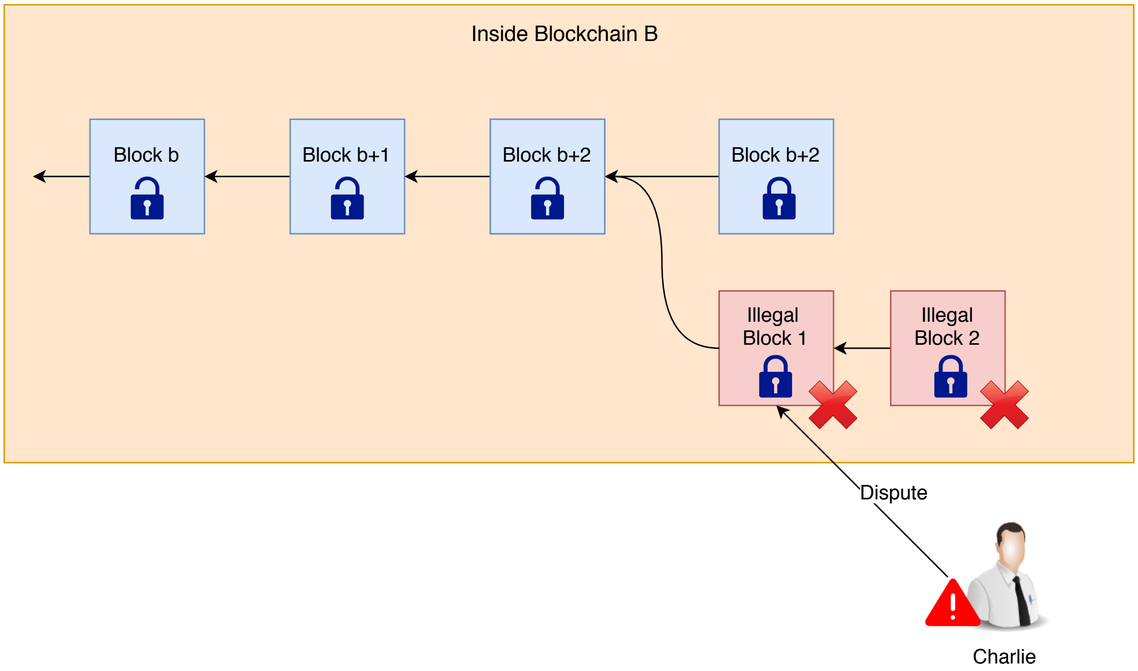

Figure 3: Charlie disputing a submitted illegal block header which is still “locked”

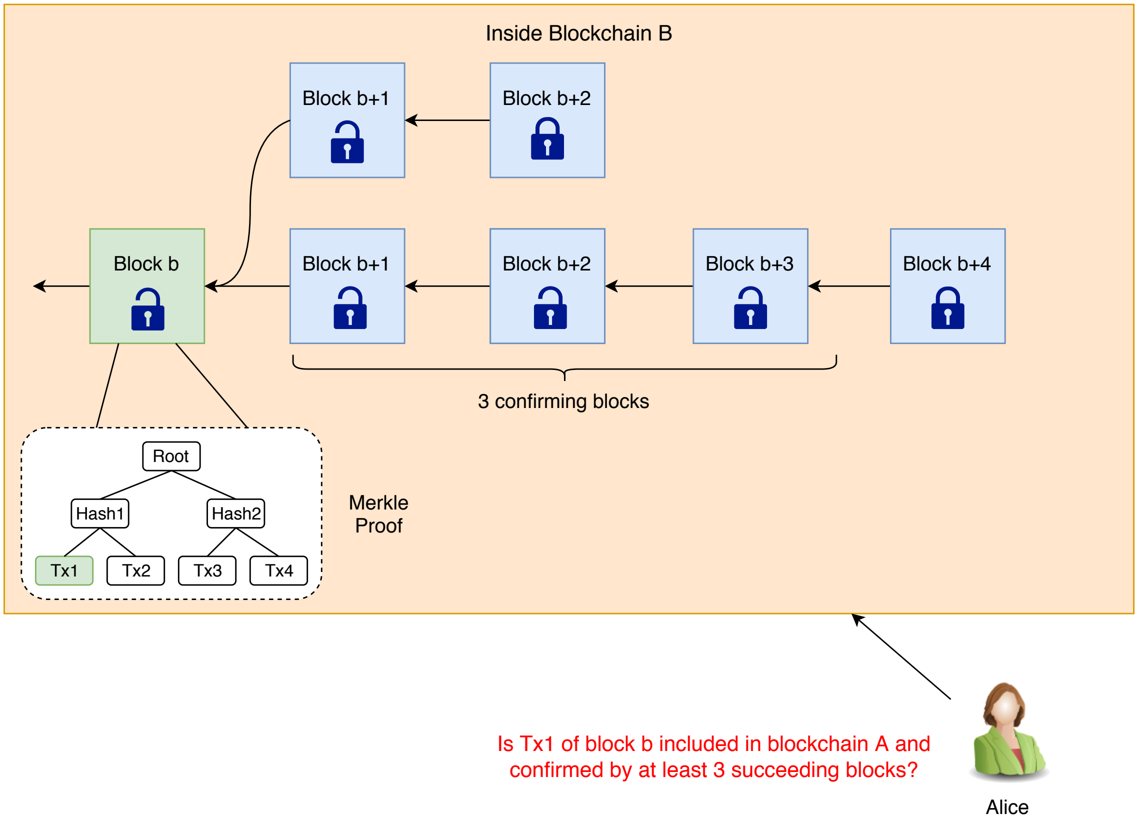

Figure 4: Alice requesting the verification of transaction *Tx1* of the source chain (“A”) on the destination chain (“B”)

client that transaction x is part of block b and that block b is part of the longest PoW chain of the source blockchain with at least n confirming blocks. Since all verifications are executed on the destination chain, no trust in a third party is required.

C. Incentive Structure

The destination chain relies on relayers regularly submitting block headers of the source chain to the destination chain as well as on clients that dispute any submitted illegal blocks. Since submitting and disputing blocks incurs cost, an incentive structure has to be in place that compensates relaying and disputing clients for their efforts. Otherwise, clients have no incentive to submit or dispute blocks. Furthermore, the biggest cost is caused by the destination chain executing a full PoW verification. Hence, the incentive structure should further keep the occurrence of full verifications to a minimum.

The following incentive structure can be employed. Clients requesting the verification of a transaction have to pay a small fee. Whenever a transaction verification is executed, this fee is transferred to the relayer that submitted the block header to the destination chain. Further, relayers are required to provide a deposit or stake. If they submit a block header that is later disputed and deemed illegal in the course of the full PoW verification, they lose their stake. The lost stake is distributed to the client that requested the full verification. In addition, clients are unlikely to dispute any legal blocks since they would have to pay for the full PoW verification without the chance to win the stake as reward.

These potential rewards motivate relayers to submit and clients to dispute block headers. At the same time, relayers are disinclined to submit illegal block headers that would subsequently cause a full PoW verification since they could potentially lose their provided stake.

IV. Future Work

The concepts explained in Section III allow the verification of transactions across blockchains. While a first prototype implementing the described concepts for Ethereum-based blockchains is available on Github[2], work on the prototype continues to eventually deploy the prototype on an Ethereum test network as destination chain and the Ethereum main network as target chain. Before that, the concepts have to implemented in-depth, an incentive structure has to be integrated, and the corresponding client library has to be extended with further functionality. Moreover, an extensive economic analysis on the incentive structure as well as on the operational cost have to be conducted to determine the viability of the described concepts in practice.

Ultimately, the described concepts can lay a strong foundation for the development of a cross-blockchain token, where the destination blockchain of a cross-blockchain token transfer can prove the existence of a "burn" transaction on the source blockchain of the transfer without requiring trust in a third party. However, note that the described concepts are not only applicable to transactions in a cross-blockchain token but can be generalized to act as the basis for arbitrary cross-blockchain applications.

Finally, since the described concepts have mostly been devised with Ethereum-based blockchains in mind, in future work, we will investigate ways to transfer the concepts to other blockchains to enable interoperability between a wide range of blockchains.

V. Conclusion

In this paper, we explained the fundamental concepts enabling cross-blockchain transaction verifications in a decentalized and trustless manner. The ability to verify transactions across blockchains in a decentralized manner that requires no trust in a third party will be a cornerstone for enabling future cross-blockchain application such as the cross-blockchain token envisioned by TAST. While a first prototype implementing the described concepts has been developed, work on the prototype continues to make truly decentralized cross-blockchain transaction verifications a reality.

Disclaimer

Information provided in this paper is the result of research, partly based on publicly available resources of varying quality. Popular use of cryptocurrencies includes investment and speculation on price developments of currencies and assets. The goal of this paper is to describe technical aspects relevant for the TAST research project. Economic considerations or future price developments are therefore not discussed. Technologies are described from a purely technical point of view. Therefore, the information in this paper is provided for general information purposes only and is not intended to provide advice, information, predictions, or recommendations for any investment. We do not accept any responsibility and expressly disclaim liability with respect to reliance on information or opinions published in this paper and from actions taken or not taken on the basis of its contents.

Acknowledgment

The work presented in this paper has received funding from Pantos GmbH within the TAST research project.

References

[1] M. Borkowski et al. *Caught in Chains: Claim-First Transactions for Cross-Blockchain Asset Transfers*. 2018. URL: http://dsg.tuwien.ac.at/staff/mborkowski/pub/tast/tast-white-paper-2.pdf. White Paper, Technische Universität Wien. Version 1.1. Accessed 2019-05-20.

[2] M. Borkowski et al. *Towards Atomic Cross-Chain Token Transfers: State of the Art and Open Questions within TAST*. 2018. URL: http://dsg.tuwien.ac.at/staff/mborkowski/pub/tast/tast-white-paper-1.pdf. White Paper, Technische Universität Wien. Version 1.2. Accessed 2019-05-20.

[3] M. Borkowski et al. *Cross Blockchain Technologies: Review, State of the Art, and Outlook*. 2019. URL: http://dsg.tuwien.ac.at/staff/mborkowski/pub/tast/tast-white-paper-4.pdf. White Paper, Technische Universität Wien. Version 1.0. Accessed 2019-05-20.

[2] https://github.com/pantos-io/go-testimonium

[4] M. Borkowski et al. “DeXTT: Deterministic Cross-Blockchain Token Transfers”. In: *IEEE Access* 7 (2019), pp. 111030–111042.

[5] P. Frauenthaler et al. *Towards Efficient Cross-Blockchain Token Transfers*. 2019. URL: http://dsg.tuwien.ac.at/staff/mborkowski/pub/tast/tast-white-paper-5.pdf. White Paper, Technische Universität Wien. Version 1.0. Accessed 2019-08-14.

[6] S. Nakamoto. *Bitcoin: A peer-to-peer electronic cash system*. 2008. White Paper.

[7] S. Schulte et al. “Towards Blockchain Interoperability”. In: *BPM Blockchain and Central and Eastern Europe Forum*. Vol. 361. Springer. 2019, pp. 1–8.

[8] G. Wood. *Ethereum: A Secure Decentralised Generalised Transaction Ledger*. 2018. URL: https://ethereum.github.io/yellowpaper/paper.pdf. White Paper. Version 69351d5, 2018-12-10. Accessed 2019-02-14.