Toward a Policy-based Blockchain Agnostic Framework

Eder Scheid, Bruno Rodrigues, Burkhard Stiller
Communication Systems Group CSG, Department of Informatics IfI, University of Zurich UZH
E-mail: [scheid, rodrigues, stiller]@ifi.uzh.ch

***Abstract*—Blockchain technology is developing at a rapid pace. For consumers that are only interested in simplifying processes and reducing costs by using a blockchain solution a new problem has occurred. It is not only necessary to select the appropriate solution, but also to configure and operate considering blockchain interoperability with different chains. For the industry, there is a need for solutions based on open standards. This paper proposes a Policy-based Blockchain Agnostic Framework that not only connects different blockchains but also facilitates their configuration and operation. Moreover, a case study is presented considering the cold-chain supply-chain, where each actor defines its policies based on blockchain requirements.**

***Index Terms*—Blockchain, Interoperability, Policy, Management, Cold-Chain, OpenAPI**

I. Introduction

Blockchain and Distributed Ledger Technologies have been experiencing a fast development in recent years due the extensive media attention revolving cryptocurrencies and the deployment of the technology in different areas rather than financial, such as supply-chain, health-care, digital identity and others [1]. This fast development led to a spawn of startup companies implementing different blockchain solutions to address issues of particular industry niches. For instance, considering only the supply-chain sector, there are several startups (*e.g.,* modum.io, Everledger, Provenance [2]–[4]) as well as companies (*e.g.,* IBM, Microsoft, and Oracle, [5]–[7]) providing blockchain-based solutions for increasing transparency and traceability.

The availability of many different solutions for the same application area creates, as a natural consequence, a dispersion of solutions used in the market, a new problem is introduced to decide not only which blockchain-based solution is appropriate, but also whether this solution is interoperable with other solutions in the market [8]. This issue becomes evident considering that different blockchains have different characteristics and different usage scenarios. For example, one blockchain design may favor data privacy over higher transaction rates. Whereas another design guarantees faster transactions rates, but no data privacy. In this context, there is a non-trivial trade-off that has to be considered when deciding which blockchain to use.

Making blockchains interoperable is becoming a key to connect independent blockchain networks [9]. For example, interoperability would mean that one could send Bitcoins and another person would be able to receive an equivalent amount in Ethers without the need for a third party, such as an exchange. Alternatively, it is possible to imagine a hospital, which has its medical records on its blockchain interacting with the social security blockchain to validate the identity of a patient. Furthermore, users would be able to access a wide range of features natively of each chain, without the need to download large files for each blockchain that they might want to use. However, enabling such cross-chain communication is not a straightforward task. For instance, although Bitcoin [10] and Ethereum [11] are cryptocurrencies, they have different characteristics and are implemented in different programming languages, which turns the simple task of sending Bitcoins and receiving Ethers into a complex task.

Many startups have been created with the goal of creating an ecosystem capable of interconnecting these different blockchains. Cosmos [12] provides a network of blockchains, called zones, which are interoperable via the so-called Inter-Blockchain Communication (IBC) protocol. A centralized message-broker called Cosmos Hub is used to connect to different zones (blockchains) as long as they implement the IBC protocol. As with a similar direction, Aion [13] provides a trust-less mechanism for cross-chain communications by developing a blockchain that supports custom blockchain architectures. Interoperability is achieved without a centralized message-broker by using so-called bridges and connectors between blockchains.

The work herein proposes a Blockchain Agnostic Framework and an API based on open standards (namely OpenAPI) that allows, based on the paradigm of Policy-based Management (PBM) [14], to perform the cross-chain communication in an agnostic and transparent way. Therefore, the Policy-based Blockchain Agnostic Framework is not only capable of dealing with blockchain interoperability, but also to ease the management of such framework through PBM enabling users and applications to operate over different blockchains transparently. It is worth mentioning that the work herein described is a work in progress; thus, its implementation is still in early stages.

This work is structured as follows. Section II presents a background on blockchain interoperability and on PBM. Section III provides a description of the approach and details technical aspects. Section IV describes a case study in the cold-chain supply-chain scenario. Section V describes related works regarding the approach. Finally, Section VI concludes this paper and presents future work directions.

978-3-903176-15-7 © 2019 IFIP

II. Background

This section presents the approaches to how to achieve interoperability of blockchains, and the Internet Engineering Task Force (IETF) policy framework, which is the basis of the proposed framework.

A. Blockchain Interoperability Approaches

Interoperability is the ability to transparently share information and transact across different blockchain. There are three different ways to achieve blockchain interoperability, namely Notary, Sidechains and Hash-locking [9]:

1) **Notary**: is the simplest way to facilitate cross-chain interoperability. In this approach, a trusted or a set of trusted entities is used to claim whether an event happened on a target blockchain and update the main chain accordingly.
2) **Sidechain or Relay**: another blockchain is used to validate data of a target blockchain. Whenever an event happens on a target blockchain, a sidechain is updated, and the main chain can react based on changes of the sidechain.
3) **Hash-locking**: it is a simple approach for atomic operations. For example, if A wants to exchange assets with B, A would first create a random secret *s*, compute its hash *h* and send to B. Then, A lock his asset into a smart contract and B also will lock his counterpart asset once A is locked. Lastly, it comes to the stage of claiming assets. If B receives the correct secret *s* from A within *X* seconds, B's asset will be transferred to A.

B. Policy-Based Management (PBM)

A PBM-driven framework can help to reduce complexity in managing the cross-chain interoperability. The policy framework [15] describes a general policy framework for managing policies in a vendor-independent, interoperable, and scalable manner. The PBM architecture is composed of a Policy Decision Point (PDP) where decisions on which policy to use are made, a Policy Enforcement Point (PEP) where policy decisions are enforced, and a Policy Management Tool (PMT) which is used by an administrator to configure and select policies stored in a Policy Repository, which hold policies templates in the Event-Condition-Action format.

III. Policy-based Blockchain Agnostic Framework

This section describes components and the functioning of the Blockchain Agnostic Framework and details its OpenAPI, which enables the cross-chain communication.

A. Blockchain Agnostic Framework

Figure 1 presents the design of the proposed framework, which is designed as a server-side component based on the notary interoperability approach. In practice, this means that a client supporting a blockchain X will send transactions to the framework, which will convert the transaction to the desired blockchain transparently.

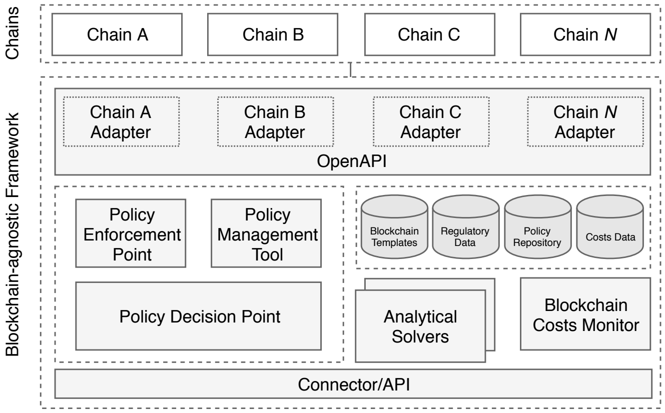

Fig. 1: Blockchain-agnostic Framework Architecture

- **Connector/API** is the entry point for the interaction with the framework. It provides an interface for decentralized applications to connect with one or multiple chains.
- **Blockchain Costs Monitor** performs periodical queries on selected chains to retrieve data such as transaction fees and rates.
- **Analytical Solver** periodically calculate usage scenarios based on user-defined thresholds for costs, pre-selected chains, and monitored blockchain data.
- **Database** stores data such as templates for different chains, policies, chain costs, and specific regulatory aspects.
- **OpenAPI** implements adapters, which are specialized components that create one or more transaction inputs into readable transactions on the desired chain.
- **Policy-Based Management** comprises PDP and PEP which are responsible for deciding over which policy is active based on user-defined parameters and enforce these actions into OpenAPI calls.

The first step involves the configuration of users, supported blockchains, and policies. A user can have 1 to N accounts associated to their blockchain accounts. Policies can be defined (using defined templates) to, *e.g.,* optimize costs (P_{COST}) or performance (P_{PERF}). These policies can be seen in Equation 1, where b is the blockchain and c the cost associated to the transaction fee.

$$F_{(cost,time)} = bc\sum_1^N \begin{cases} min(estimateFee(b,c)), & \text{if } P_{COST} \\ max(estimateFee(b,c)), & \text{if } P_{PERF} \end{cases} \qquad (1)$$

where

$$cost \leq MAX\_COST$$

$$time \leq MAX\_TIME$$

If one desires to optimize costs, *i.e.,* minimize costs related to transaction fees, the framework decides which supported blockchains to store data that optimizes cost, or it is bounded to a maximum cost. For instance, a MAX_COST threshold can define a target amount to be spent, and a MAX_TIME

can define a maximum waiting time to have all transactions mined. However, if one desires to prioritize performance, the framework will be configured to store transactions as fast as possible, which includes increasing transaction fees. However, a MAX_COST can be defined to bound costs to such an operation.

Security policies have priority over costs and performance policies. Data confidentiality may inhibit companies to store specific data on public blockchains, and thus, private or consortium blockchains should be declared for that specific operation. This can be defined by defining allowed operations (*e.g.,* write, read, migrate, estimate) on allowed types of data associated to the set of $ALLOWED\_CHAINS$ to express a set of allowed chain IDs per operation, which in practice restrict the group of blockchains available. Also, the use of private blockchains influences the performance and cost policies, since these blockchains do not have the computational overhead caused by public blockchains based on Proof-of-Work (PoW). Based on the policy definitions, the PDP will decide, which policy will be active for which blockchain and type of operation requested by the OpenAPI.

Once policies were configured, the Blockchain Costs Monitor is used to calculate the cost and waiting time information from available blockchains. This can be performed periodically (triggered by an Analytical Solver) to obtain an average transaction cost based on past transactions on N blocks or as a response of an OpenAPI call to estimate costs on average waiting time of a single transaction being sent as a parameter. Public APIs (*e.g.,* blockcypher, bitcoinfees, coinmarketcap) are used to query such information by an estimated fee on the selected currency. Once information concerning cost and time are collected, the PEP, based on the current policy, forwards the operation request to the OpenAPI.

B. *OpenAPI Workflow*

Based on the user policies defined, the framework selects the most suitable blockchain implementation to store and retrieve the data. This store/retrieve interaction is performed within the framework using the OpenAPI component. The code for the OpenAPI prototype can be found at the IFI GitLab Repository[1].

The workflow of how the OpenAPI performs the `store()` function is depicted in Figure 2. The *API* is the entry point for the OpenAPI, once it receives the `store()` function, it delegates the request to the corresponding adapter based on the chosen blockchain using the `BC_ID` value. Thus, each blockchain requires a *Blockchain Adapter*, which implements specific methods to create, sign, and send transactions (TX). The core of each *Blockchain Adapter* are the `store` and `retrieve` methods. The former, shown in Listing 1, creates a transaction based on the correct blockchain template and encodes the input data. It signs this transaction with the private key (retrieved from the *Credentials* database) and broadcasts it, as a raw transaction, to multiple nodes in the blockchain. The broadcast is performed using the defined blockchain *Remote Procedure Call (RPC) server*. Finally, it stores the transaction in the *TXs database* before returning the transaction hash to the user. The latter, shown in Listing 2, searches the transaction information from the blockchain using the transaction hash as the search parameter. Then, it extracts the data from the transaction information and returns the restored text message to the user. The described workflow is the same for all *Blockchain Adapters*. However, specific methods, such as to construct a transaction, are implemented accordingly to the design details of each blockchain supported.

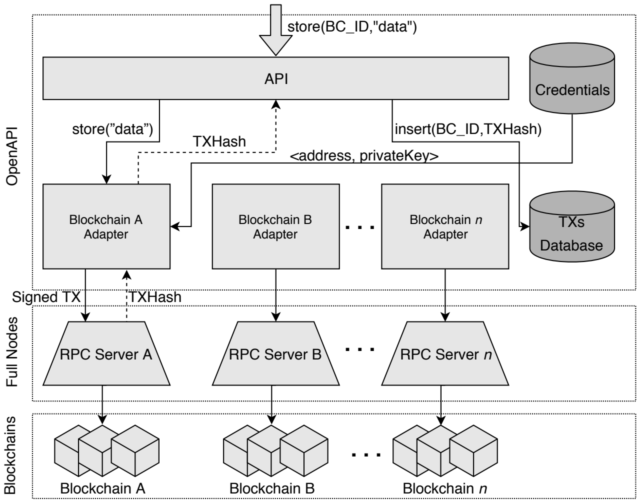

Fig. 2: OpenAPI Store Function Workflow

```
def store(text):
        tx = create_transaction("data")
        signed_tx = sign_transaction(tx)
        transaction_hash = send_raw_transaction(signed_tx)
        add_transaction_to_database(tx_hash)
        return tx_hash
```

Listing 1: Store method of the Blockchain Adapter

```
def retrieve(tx_hash):
        tx = get_transaction(tx_hash)
        data = extract_data(tx)
        return to_text(data)
```

Listing 2: Retrieve method of the Blockchain Adapter

IV. Cold-Chain Interoperability Case Study

Until temperature-sensitive products (*e.g.,* drugs) reach their final destination, they must be handled by different stakeholders involved in the cold-chain supply-chain (*cf.* Figure 3). These stakeholders are considered during the development of the proposed framework, because they have different points of views and requirements.

Each one of these stakeholders has different requirements that impact on the choice of the blockchain, resulting in different policies. For example, companies follow specific standards and regulations in their manufacturing line. Therefore, recording in the blockchains that the goods left the company

[1] https://gitlab.ifi.uzh.ch/scheid/bcio

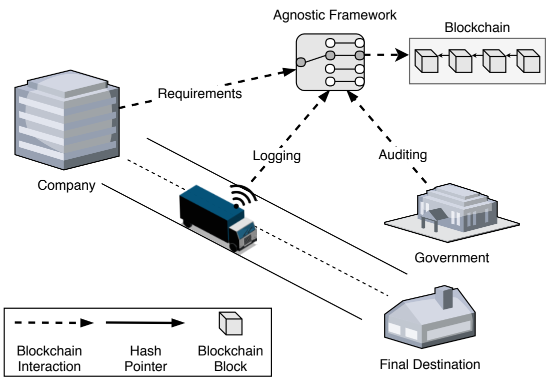

Fig. 3: Overview of the Case Study Stakeholders

under such standards is crucial to avoid any disagreement or Service Level Agreement (SLA) violations. What impact on the blockchain choice for such a stakeholder is not the cost involved, but rather how fast can the records (*e.g.,* temperature sensor readings) be included in the blockchain and how secure is the blockchain implementation (*e.g.,* a private blockchain). Thus, the MAX_COST parameter in the policy should be set to the highest possible so that the framework will store these records as fast as possible.

However, in the context of logistic services, the requirement that impacts the most is how much it will cost to append a new record to the blockchain, independent of the time that it will take. This is because the conditions (*e.g.,* temperature and humidity) within the storage compartment of vehicles must continuously be monitored to ensure that temperature-sensitive drugs produced were not affected. This active monitoring produces a considerable amount of blockchains transaction. Thus, reducing the cost of data storage becomes crucial to this stakeholder. In this sense, it must define a policy with the MAX_COST parameter set to the lowest possible value.

For stakeholders that only verify/audit the data appended in the blockchains, which is the case for the government and the final product destination, the framework allows the retrieval of data based on an input hash. These stakeholders do not define any policies. However, they have to be registered in the system and must have permissions to read data from private blockchains, which contain sensitive information.

V. Related Work

There have been efforts regarding the interoperability between blockchains in different areas such as supply-chain, finance, and identify management, as it can be seen in Table I (non-exhaustive list). Most of the current work on interoperability follow a Sidechain/Relay approach [12], [16]–[22]. The Sidechain/Relay approach, as described in Section II, maintains a decentralized character by implementing the same consensus mechanisms, and protocols of blockchains. However, this approach introduces one additional layer of management complexity, increasing implementation difficulty.

Approaches, such as [23] or [13], which rely on Notary-schemes, are less complex to implement than Sidechains due to their straightforward character. The interoperability goal of this approach is achieved through the placement of a trusted entity that actively manages inter-blockchain assets or data. The shortcoming of such interoperability approach is that a trust model must be defined and all stakeholders must be compliant.

Hash-locking approaches can achieve the exchange of assets (cryptocurrencies) using atomic operations between two different blockchains by following a defined mechanism. This type of approach is employed in [24]. However, hash-locking alone does not provide full interoperability between blockchains, it must be combined with other approaches, such as Sidechains/Relays.

TABLE I: Interoperability Approach of Related Works

Work	Interoperability Approach	Area
BC Agnostic	Notary-scheme	Supply-Chain
Herdius [23]	Notary-scheme	Finance
AION [13]	Notary-scheme	General Purpose
Originaltrail.io [16]	Sidechain/Relay	Supply-Chain
Aelf [17]	Sidechain/Relay	General Purpose
Cosmos [12]	Sidechain/Relay	General Purpose
Polkadot [18]	Sidechain/Relay	General Purpose
Ark.io [19]	Sidechain/Relay	General Purpose
Crowdmachine [20]	Sidechain/Relay	Cloud Computing
VirtualChain [21]	Sidechain/Relay	Identity
ICON Project [22]	Sidechain/Relay	General Purpose
Wanchain [24]	Hash-locking	Finance

VI. Conclusion and Future Work

Many companies address particular niches applying specific solutions, creating a myriad of blockchain implementations. Two problems arise with the expansion of implementations: the interoperability between them and how to allow non-technical users to choose which blockchain implementations to use. This paper, presented a proposal for an agnostic framework toward addressing such issues. The framework allows users to specify policies regarding the desired maximum cost and the minimum performance requirements for data storage. These policies guide the selection of which blockchain to use in a transparent manner using an OpenAPI provided by the framework.

Moreover, the employment of the framework was described in a use case in the cold-chain supply-chain scenario. The scenario is composed of stakeholders with different requirements to choose blockchains that will store their data; thus, they define different policies in the framework, either to store data as fast as possible or to minimize costs during storage. As the work is in an initial stage, further investigations are still required in the context of *(i)* security, *(ii)* policy decision mechanism, *(iii)* access control, and *(iv)* overall performance.

Acknowledgments

This work has been supported by Komission für Technologie und Innovation (KTI), Switzerland (CTI-No. 26996.L PFES-ED). Also, the authors would like to acknowledge manifold discussions with modum.io, especially Stefan M. Weber, Sacha Uhlmann, and Thomas Bocek from the Hochschule für Technik Rapperswil (HSR).

References

[1] T. Bocek and B. Stiller, "Smart Contracts–Blockchains in the Wings," in *Digital Marketplaces Unleashed*. Springer, 2018, pp. 169–184.

[2] T. Bocek, B. B. Rodrigues, T. Strasser, and B. Stiller, "Blockchains Everywhere - a Use-Case of Blockchains in the Pharma Supply-Chain," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, May 2017, pp. 772–777.

[3] Everledger, "Everledger - A Digital Global Ledger," Aug 2018. [Online]. Available: https://www.everledger.io/

[4] Provenance, "Provenance - Every Product has a Story," Aug 2018. [Online]. Available: https://www.provenance.org/whitepaper

[5] IBM, "Blockchain for Supply Chain," Aug 2018. [Online]. Available: https://www.ibm.com/blockchain/industries/supply-chain

[6] Microsoft, "How Blockchain will Transform the Modern Supply Chain," Aug 2018. [Online]. Available: https://goo.gl/zKzdXF

[7] U. Koester, "How Blockchain Can Transform Your Supply Chain Ecosystem," Aug 2018. [Online]. Available: https://goo.gl/GMcj7c

[8] B. Rodrigues, T. Bocek, and B. Stiller, "The Use of Blockchains: Application-Driven Analysis of Applicability," in *Advances in Computers*. Springer, 2018, pp. 163–198.

[9] V. Buterin, "Chain Interoperability," September 2013. [Online]. Available: https://www.r3.com/blog/2017/01/23/chain-interoperability/

[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system."

[11] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[12] J. Kwon and E. Buchman, "Cosmos: A Network of Distributed Ledgers," August 2018, available at https://cosmos.network/docs/resources/whitepaper.html.

[13] M. Spoke and N. E. Team, "Aion: Enabling the decentralized Internet," July 2017. [Online]. Available: https://aion.network/media/en-aion-network-technical-introduction.pdf

[14] D. C. Verma, "Simplifying network administration using policy-based management," *IEEE network*, vol. 16, no. 2, pp. 20–26, 2002.

[15] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry, "The cops (common open policy service) protocol," Tech. Rep., 1999.

[16] B. Rakic, T. Levak, Z. Drev, S. Savic, and A. Veljkovic, "First Purpose Built Protocol for Supply Chains based on Blockchain," October 2017, available at https://origintrail.io/storage/documents/OriginTrail-White-Paper.pdf Accessed 24 August 2018.

[17] aelf Team, "aelf - A Multi - Chain Parallel Computing Blockchain Framework," June 2018, available at https://grid.hoopox.com/aelf_whitepaper_EN.pdf?v=1.6 Accessed 24 August 2018.

[18] G. Wood, "Polkdaot: Vision for a Heterogeneous Multi-Chain Framework," November 2016, available at https://polkadot.network/PolkaDotPaper.pdf Accessed 24 August 2018.

[19] T. A. Crew, "ARK White Paper - A Platform for Consumer Adoption," available at https://ark.io/Whitepaper.pdf Accessed 24 August 2018.

[20] C. Sproule, "Crowdmachine - White Paper," February 2018, available at https://www.crowdmachine.com/wp-content/uploads/2017/11/Crowd-Machine-Whitepaper.pdf Accessed 24 August 2018.

[21] J. Nelson, M. Ali, R. Shea, and M. J. Freedman, "Extending Existing Blockchains with Virtualchain," in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, 2016.

[22] ICON Foundation, "ICON - Hyperconnect the World," January 2018, available at https://icon.foundation/resources/whitepaper/ICON-Whitepaper-EN-Draft.pdf Accessed 22 August, 2018.

[23] D. Balzs, "Herdius - Next Generation Decentralized Blockchain Financial Infrastructure," available at https://herdius.com/whitepaper/Herdius_Technical_Paper.pdf.

[24] Wanchain Foundation Ltd., "Wanchain - White Paper," 2017, available at https://wanchain.org/files/Wanchain-Whitepaper-EN-version.pdf Accessed 24 August 2018.