Blockchain Oracles: A Framework for Blockchain-Based Applications

Kamran Mammadzada[1], Mubashar Iqbal[1(✉)], Fredrik Milani[1], Luciano García-Bañuelos[2], and Raimundas Matulevičius[1]

[1] Institute of Computer Science, University of Tartu, Tartu, Estonia
{kamran.mammadzada,mubashar.iqbal,fredrik.milani,rma}@ut.ee
[2] School of Engineering and Science, Tecnologico de Monterrey, Monterrey, Mexico
luciano.garcia@tec.mx

Abstract. Oracles support the access, validation, and transmission of data from external sources to blockchain systems. They are important components of blockchain-based architectures. However, there exists no guidance on how oracles could be used when designing blockchain-based applications. In this paper, based on the results of a systematic literature review, we propose a framework to explain blockchain oracles and their relationships to blockchain-based applications. More specifically, the blockchain oracle framework addresses the origin of data, oracle properties, encryption method, oracle data source, validation procedures, and the integration of oracles to blockchain-based applications. Potentially, this framework can guide developers when incorporating oracles to blockchain-based applications.

Keywords: Blockchain · Blockchain-based applications · Blockchain oracles · Blockchain oracle framework

1 Introduction

Blockchain has been positioned as a technology with the potential to innovate how companies manage inter-organizational processes [21]. The disruptive potential of blockchain has attracted the attention of many companies to explore various commercial use cases for this technology. At its core, blockchain relies on the concept of a distributed ledger. Data is recorded and secured by means of cryptographic algorithms that ensure the data is tamper-proof and propagated to all nodes [16]. In blockchain, transactions are executed in a distributed manner without relying on trusted third parties. These properties enable cryptocurrencies where transactions are managed without the need of a trusted third party, commonly supported by permissionless blockchain. It also enables collaborative execution of inter-organizational processes [21], such as food tracking that is implemented on permissioned blockchain.

Analysts working with implementing blockchain-based applications, particularly for inter-organizational processes, have to consider the exchange of data

A. Asatiani et al. (Eds.): BPM Blockchain and RPA Forum 2020, LNBIP 393, pp. 19–34, 2020.
https://doi.org/10.1007/978-3-030-58779-6_2

between the blockchain and external systems. However, blockchain itself does not access data from external sources. Therefore, a mechanism that provides data to blockchain is required. This can be achieved by using *oracles*. Conceptually, oracles are trusted entities that enable the collection, validation, and transmission of data from external sources [2]. Thus, oracles are important components for design and implementation of blockchain use cases, in particular for commercial applications. However, the research on oracles so far, has not considered types of oracles and when they can be used. In light of this context, we seek to address the overall research objective of gaining an overview of oracles and how they can be used in blockchain use cases. We conduct a systematic literature review (SLR) and, based on the results, propose a *blockchain oracle framework*. The framework focuses on key aspects of oracles, such as origins of data, methods of integration with blockchain, and the types of data transfer to blockchain.

The rest of the paper is structured as follows. Section 2 provides background information describing the major concepts discussed in the research. Section 3 describes the review protocol used to find the relevant studies. Section 4 presents the results whereas Sect. 5 presents the blockchain oracle framework and discusses threats to validity. Finally, Sect. 6 provides some concluding remarks.

2 Background

Blockchain is a distributed ledger technology where transactions are replicated and stored on a multitude of nodes. Each node holds a full or partial copy of the ledger. New transactions can only be added in an append-only manner. Transactions are collected in blocks that are appended to the ledger. The blocks are linked to the preceding and succeeding block by means of a hash. Transactions that have been appended to the ledger are considered as tamper-proof because changing a transaction of an older block requires changing all of the succeeding blocks. Such a computational effort is very costly. Therefore, blockchain is considered to provide secure, immutable, and tamper-proof transaction records. As all nodes can create blocks in an untrusted decentralised system, the nodes must, by means of a consensus algorithm [26], reach an agreement on which block to append to the blockchain.

The participating nodes agree on the state of the ledger by following the consensus mechanism. The Proof of Work (PoW) [15], Proof of Stake (PoS) [15] and Practical Byzantine Fault Tolerance (PBFT) [26] are some of the used consensus protocols. Blockchain could be classified as permissionless (a.k.a. public) and permissioned [2]. In a public blockchain, the ledger is publicly accessible and open for all to join. A permissioned blockchain enforces network participants to be authorised before joining the network (e.g., Hyperledger Fabric) and ledger accessibility is restricted.

The distributed architecture of blockchain allows for independent entities to directly interact with each other without depending on a central system or authority. Blockchain is commonly categorised as public or permissioned [38]. A public blockchain is fully decentralised and open for all to access and join. Such a solution is, therefore, suitable for digital currencies [34]. A permissioned blockchain,

on the other hand, is partially decentralised (managed by several organizations) with restrictions on who can join and access the data. Permissioned solutions are can, therefore, be used for commercial cases where mutually distrustful entities have shared interest. Such solutions have been implemented, for instance, in the insurance domain and in trade settlement between financial institutions [31].

In blockchain, smart contracts enhance the transaction process and automation. A smart contract is a program that is self-verifying, self-executing, tamper-resistant and executes on the blockchain platform. Smart contracts have been defined as programs that digitally facilitate, verify, and enforce contracts made between two or more participants on the blockchain [26]. Smart contracts are event-driven, meaning that they can be activated when a predefined condition is met [26]. However, smart contracts can only use resources available on the network and cannot access or interact with the external data [2]. To address this, blockchain oracles can be introduced to enable an exchange of data available external to the blockchain.

Smart contracts are often required to have relevant information from the outside world to execute the agreement (or to meet certain conditions) [3,4]. Here, blockchain oracles come into play because smart contracts cannot, by themselves, interact with external sources [3]. According to Al-Breiki et al., *"blockchain oracle is an external data agent that observes the real-world events and reports them back to the blockchain to be used by smart contracts"* [3]. Accordingly, oracles are trusted entities that bring external information into the blockchain [1,27] and serve as a bridge between blockchain and the outside world [4]. Furthermore, the role of oracles is not limited to simply querying the information from outside of the blockchain, but can also verify the authenticity and validity of that data. Blockchain oracles can directly interact with smart contracts. Therefore, it is important that oracles provide reliable and valid information to ensure consistency and validity of smart contract execution. Therefore oracles can be essential for blockchain implementations.

3 Review Protocol

The objective of this paper is to propose a framework for oracles and, in particular, when they can be used in blockchain use cases. To this end, a SLR is suitable as the method enables a systematic review of relevant literature. We followed the guidelines proposed by Kitchenham [17]. Accordingly, we specify the research questions, design a search protocol to search, and identify relevant papers. We defined the following six research questions, each covering a different aspect of oracles in blockchain-based applications.

RQ1: *What are the origins of data that oracles provide to blockchain-based applications?*
In order to describe the nature of information oracles provide to blockchain and relationship between oracles and blockchain, it is valuable to explore various origins of data from where through oracles communicate to blockchain.

RQ2: *What are the properties of oracles for use in blockchain-based applications?*
Exploring oracles properties ensures understanding the characteristics that are needed to possess and inject information into the blockchain.

RQ3: *How is data received and sent by oracles encrypted?*
Protecting the data transfer from an external data source to oracle and then oracle to blockchain is critical to the integrity of blockchain-based applications thus discussion of encryption is important to understand methods used to ensure reliable data transfer. The encryption methods represent the cryptography technology that used to secure the communication between entities.

RQ4: *How many sources are used by oracles to collect data from?*
Oracle data source captures the mechanism behind how the decision for passing the data into blockchain is made. It is important to explore how oracle data sources are used by oracles to gather data sent to the blockchain.

RQ5: *How do oracles validate the data they provide to blockchain-based applications?*
Data validation within blockchain oracles is critical since information recorded in blockchain cannot be deleted. In these perspectives, data validation ensures that information collected from external sources to blockchain is legitimate and correct.

RQ6:*How are oracles integrated with blockchain platforms?*
Oracle integration into blockchain platforms contributes to blockchain widespread implementation since oracles help solve the issue of bringing external data into the network. Investigating various oracles integration ways could provide good basis for making necessary decisions when developing blockchain-based applications.

The overall search strategy is to find a body of relevant studies. For this SLR two search strategies were used, as recommended by Okoli et al. [25], Fink et al. [10] and Levy et al. [18], to secure identification of relevant studies. Accordingly, in the first step, called primary search, search strings were used to identify an initial set of papers [10]. Several electronic databases were used for this step. In the second step, a secondary search was performed by means of backward and forward tracing [18,25].

The search strings included the keywords *"blockchain"* in combination with *"oracle"*, *"internet of things"*, or *"IoT"*. We tested the search strings and found that the terms *"internet of things"* and *"IoT"* are extensively included in blockchain solutions and often used, in the context of blockchain solutions, as a quasi-term for an oracle. Thus we decided to include this term in the search string. We applied the search strings on *ACM Digital Library*, *IEEE Xplore*, *Scopus*, *Web of Science*, *Wiley*, and *Google Scholar*. We included google scholar to

identify publications by companies and other non-academic organizations (grey literature) as proposed in [17].

We applied *exclusion (EC)* and *inclusion (IC)* criteria to identify relevant papers. Papers that were duplicates, not in English, shorter than 5 pages, inaccessible (via University subscriptions or Internet search), or published before 2008, were excluded. Papers less than 5 pages were excluded as short papers would not contain enough information for our evaluation. Papers within the domain of blockchain *(IC1)*, covering the integration of oracles with blockchain *(IC2)*, and providing a description of the oracle solution *(IC3)*, were included.

The search resulted in 3015 hits from all sources. Having removed the duplicates, 2356 papers remained and their publishing date is between 2008–2018. After several iterations of filtering, considering the exclusion criteria and the first two inclusion criteria *(IC1 & IC2)*, a total of 70 papers remained. These were subjected to full-text examination *(IC3)*, which resulted in a *total of 23 studies* (Fig. 1) remaining (including backward tracing references). The corpus of papers consisted of *65% academic publications* and *35% grey literature*.

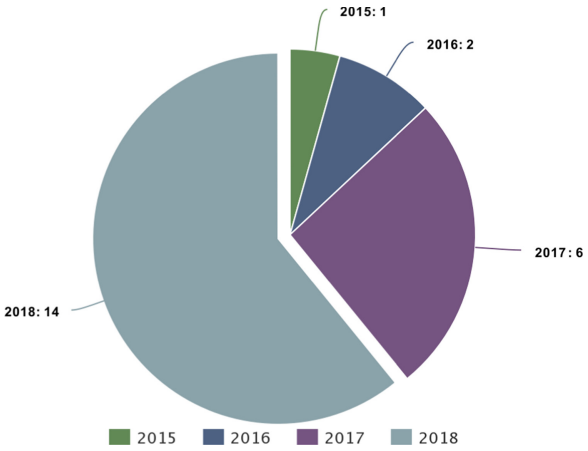

Fig. 1. Papers distribution per year

Following the identification of the final list of papers, relevant data were extracted. To ensure unbiased data extraction strategy, it has been recommended [10] to develop a data extraction form and strategy. The data extraction form was developed after the screening process, allowing for utilising the insights drawn during the screening phase. Three types of data were extracted. The first relates to data about the paper. The second was related to the context of the study and finally, the third type related to the availability of developed solutions. The complete review protocol, list of final papers, and detailed results of the SLR are available here: https://doi.org/10.5281/zenodo.3605157.

4 Results

In this section, we present the results obtained from the SLR in relation to each research question respectively.

4.1 Origin of Data

In determining the best-suited oracle design, the first question to consider is about the origin of the data that the blockchain-based application requires. Our review shows that the data which is sent by oracles, come from two types of origins: from web content and collected using sensors (Table 1).

A large body of the literature describes oracles that provide data, collected by third parties, through web services [15,36]. Web content differs from sensor data in that it is easily available via the browser *(e.g., generic HTTP(s) data)* and does not directly originate from a physical device. Web content covers data such as financial information, sports results, weather updates, operational data, and user inputs that are categorised under generic http(s) data. For instance, EdenChain [2], an asset tokenization platform, and Town Crier [12,37], an authenticated data-feed service for smart contracts, aiming to deliver trusted data input through web services. Boolean propositions and scalar measurements are a subtype of web content and mostly used in prediction markets [1,15,36]. Hence, web content ranges from binary responses to discrete or continuous responses [28,36].

Table 1. Origins of data

Origin of data	Sub-type	Papers
Web content	Generic HTTP(S) data	[2,8,9,12,29,37]
	Boolean or scalar	[1,15,28,33,36]
Sensor data	IoT device readings	[13,26]
	Energy readings	[20]
	Vehicular sensor readings	[16,22]
	Biometric readings	[11]
	Health readings	[35]
	Product readings	[14,24]
	Visual feed	[19]

Oracles providing sensor data to the blockchain are commonly generated by physical devices (e.g., IoT & vehicle sensors [2,12]). In an example from the domain of vehicle data [16,22], traffic-related data and data on road conditions (using embedded sensors) are collected. Similarly, Uddin et al. [35] record health data (e.g., blood pressure, pulse & body temperature) using wearable and implantable medical devices. In [24], a unique RFID chip is used to collect and track clothing-related data (e.g., geolocation & product details). These examples illustrate how sensors are applied as oracles to record and send data to the blockchain.

4.2 Oracle Properties

In this section, we explored oracle and blockchain types that are essential components when designing blockchain-based applications.

Oracle Type: The need for defining oracle types is important for organising the oracles to facilitate developers in navigating through the landscape of potential options. Some [5,6], have divided the blockchain oracles into the following categories:

- Software oracles – oracles that push information available online to blockchain
- Hardware oracles – oracles that push information from physical devices (e.g. sensors, RFID chips, etc.) into the blockchain
- Inbound oracles – oracles that provide smart contracts with data from the external world
- Outbound oracles – oracles that send information to the outside world
- Consensus-based oracles – data passed to blockchain is treated as a result of consensus of multiple oracles

In the blockchain oracle framework, when defining oracle type we consider physical attributes of oracle. For example, the oracle is pushing the information from a physical device *(tangible entity)* or an oracle is a piece of code *(intangible entity)* collecting information from intangible sources (e.g., websites).

Blockchain Type: Blockchain platforms can be categorised as permissionless, permissioned, or hybrid. Our review shows that oracles are used with all types of blockchain platforms; for example, Ethereum (permissionless) [37], Hyperledger Fabric (permissioned) [30] and BlockID (hybrid) [11].

4.3 Encryption Method

The third column of the framework represents the encryption methods (Table 2). This component concerns ensuring data confidentiality when data is transferred from external data sources to the oracles, and from oracles to the blockchain. It is noteworthy that most studies mention encryption methods briefly (e.g., [30]) or does not discuss it at all (e.g., [11,19]).

The most commonly implemented encryption method for data transfer from external sources to oracles is a public key infrastructure (PKI). For PKI, the most common encryption technique is transport layer security (TLS). TLS provides authentication, privacy, and data integrity between communicating entities [32] and is the prevalent form of secure communication on the internet [7]. One paper presented a solution called TLS-N, a novel communication protocol which acts as an oracle and is built on top of TLS [29].

Table 2. Encryption methods when data transmit from external sources to Oracle

Encryption method	Technique	Papers
PKI	TLS	[1,2,9,28,33,37]
	TLS-N	[29]
	Not discussed	[13,19,22,36]
Symmetric cryptography	Not discussed	[35]
Asymmetric cryptography	ECC	[16]
	Not discussed	[11,24]

Apart from receiving data from external sources, oracles also transfer information to the blockchain. The only encryption method proposed from oracles to the blockchain is asymmetric encryption (e.g., [12,35]) (Table 3). Elliptic curve cryptography (ECC) is a form of asymmetric cryptography and is used in Bitcoin and Ethereum. ECC with threshold cryptography (ECC-TC) is a protocol with a cooperative property where data necessary for decryption is shared among participants so that encrypted data can be decrypted only when data of other participants is present as well as yours [2]. This process enables secure decentralised exchange of information.

Table 3. Encryption methods when data transmit from oracle to blockchain

Encryption method	Technique	Papers
Asymmetric cryptography	ECC	[15,16,29,33,37]
	ECC-TC	[2]
	Not explicitly discussed	[12,14,24,35]

4.4 Oracle Data Source

Oracle data source refers to the data sources used by the oracles to gather data sent to the blockchain. If a single data source is used, it is called a single-source oracle; and if multiple data sources are used – multi-source oracle.

In this work, we found studies that employ a single-source oracle where, for instance, smart meters used an IoT enabled smart grid [20] and on-body sensors used to enable tracking of vital information [35]. These examples of sensor data oracles rely on a single source of data. Single-source oracles can also be used for web content, for example, oracles that receive data from single trusted content provider [12,37].

Multi-source oracle receives data from several sources. For instance, a set of roadside units (RSU) collect vehicle data and send the aggregated data to the blockchain. Also, the RSUs interact with each other to verify vehicle identity or request specific data (e.g., reputation score, authorization & data sharing settings, etc.) [16].

4.5 Data Validation

We identify data validation as the method by which oracles ensure that the data provided to the blockchain is correct (Table 4). Some studies propose a consensus-based solution where data is validated by means of majority voting, i.e., on the basis of the wisdom of the crowd [36]. Another consensus-based method relies on weighted voting where each individual vote has a specifically assigned weight [28]. Finally, one study proposes a hybrid of PoW and PoS solution [15].

Table 4. Data validation approaches

Data validation	Mechanism	Papers
Consensus	Majority voting	[1,8,9,36,37]
	Weighted voting	[28,33]
	Hybrid of PoW & PoS	[15]
No data validation	Trusted third party	[11,12,16,19,29,35,36]
	Not discussed	[14,20]
Self-validation	RFID signature validation	[24]

We found that the most common approach is to rely on trusted data providers. As such, the oracles do not have any method for validating the data. Such a strategy operates under the assumption that the data source is trustworthy. For instance, vehicular blockchain networks [22] trust the central governments' authority for issuance of legitimate vehicle plates, while service platforms provide data as trusted web content providers [29]. Some [23] use trusted data exchanges or incorporate certified equipment [30] when deploying IoT devices.

4.6 Oracle Integration Method

Oracle integration is a method by which an oracle is interfaced to a blockchain to provide data (Table 5). We found that smart contracts, software modules, custom solutions, and built-in solutions are approaches used to integrate oracles with blockchain platforms.

The most common integration method is by using a custom smart contract. Several studies [2,36] use this approach to integrate decentralised web applications (dApps), such as implementing a pair of smart contracts, one on-chain and other off-chain. Another approach relies on custom software modules. Such modules provide additional functionality to achieve integration between the oracle and the blockchain. Software modules are often used when physical devices communicate with a blockchain. The software module serves as an intermediate agent that, according to rules, manipulates the incoming data from oracles before sending it. For instance, in [24], they use RFID readers and a PC with a blockchain node to deliver tracking data.

Table 5. Blockchain oracles integration methods

Integration method	Mechanism	Papers
Custom smart contract interface	On-chain and off-chain smart contract	[2,9,36]
	Off-chain smart contracts deployed on-chain	[26]
	Data storage smart contract (DSSC)	[16]
	Information sharing smart contract (ISSC)	[16]
	Chaincode (specialised smart contract)	[30]
	On-chain smart contract accessing Data Cubes	[23]
	Smart contract able to verify TLS-N proofs	[29]
	Server + on-chain smart contract	[37]
	On-chain smart contract + Bridge node	[8]
	TLS Identities linked to content contract	[12]
Custom software module	RFID Reader + PC with blockchain module	[24]
	Software module (ETSE) + Adapter	[20]
	Control system + Blockchain client	[19]
	Patient centric agent	[35]
Custom solution	Blockchain identity bound to government ID	[11]
	OriginStamp	[14]
Built-in		[15,33]

Some studies propose a custom solution i.e., a unique and separate solution built to cater for the specific needs of the project. Two cases, one involving scanning images for combating counterfeit products [14] and the other, fingerprint scanning for identity management [11], use this method of integration. Finally, some studies [33] developed the oracle inside the blockchain network. In these solutions, the smart contracts self-execute when certain conditions are met.

5 Framework

In this section, we present the framework and the associated components (Table 6) that are derived from the results of the SLR. The goal of this framework is to summarise the results of the SLR in a clear and concise manner that represents the state of the art of blockchain oracles. It serves developers and decision-makers when making decisions in their blockchain-based applications regarding blockchain oracles. The framework covers the possible scenarios of combinations where certain data passed through a specific oracle & blockchain type using a pre-defined oracle data source mechanism and data validation approach could add value to a blockchain network. A visual representation (Fig. 2) of the framework aims to provide visual cues to the reader and communicate the interaction flow.

Table 6. Blockchain oracle framework

Origin of Data	Oracle Type	Blockchain Type	Encryption	Oracle Data Source	Data Validation	Oracle Integration Method	Ref.
Web Content	Intangible	Permissioned	PKI	Single-source Oracle	Trusted Third Party	Custom Smart Contract Interface	[2]
		Permissionless	PKI	Single-source Oracle	Majority Voting	Custom Smart Contract Interface	[37]
					Trusted Third Party	Custom Smart Contract Interface	[12]
				Multi-source Oracle	Hybrid of PoW & PoS	Built-in	[15]
					Majority Voting	Custom Smart Contract Interface	[8,9,36]
						Not explicitly discussed	[1]
					Trusted Third Party	Custom Smart Contract Interface	[29]
					Weighted Voting	Built-in	[33]
						Custom Smart Contract Interface	[28]
Sensor Data	Tangible	Hybrid	Asymmetric	Single-source Oracle	Trusted Third Party	Custom Solution	[11]
		Permissioned	Asymmetric	Multi-source Oracle	Trusted Third Party	Custom Smart Contract Interface	[16]
			Not Covered	Single-source Oracle	No Data Verification	Custom Software Module	[20]
					Trusted Third Party	Custom Smart Contract Interface	[30,26]
			PKI	Multi-source Oracle	Trusted Third Party	Not explicitly discussed	[22]
			Symmetric	Single-source Oracle	Trusted Third Party	Custom Software Module	[35]
		Permissionless	Asymmetric	Single-source Oracle	RFID Signature	Custom Software Module	[24]
			Not Covered	Single-source Oracle	No Data Validation	Custom Solution	[14]
					Trusted Third Party	Custom Smart Contract Interface	[23]
			PKI	Single-source Oracle	Trusted Third Party	Custom Software Module	[19]
		Hybrid	PKI	Single-source Oracle	Trusted Third Party	Not explicitly discussed	[13]

5.1 Example of Oracle Framework

The framework (Table 6 and Fig. 2) is to be read from left to right, following the natural flow of data from external sources to the blockchain. When an oracle is required in a blockchain solution, the first step is to identify the origin of data that needs to sent to the blockchain. This decision paves the way for other choices regarding, for instance, oracle properties (e.g, oracle type). Before navigating further, discussing the scope of the project and stakeholders would aid to identify the blockchain type to use. There are frameworks that assist in this regard. Commonly, if there is a limited number of participants, then a permissioned blockchain would be appropriate. Otherwise, a public permissionless network would perhaps be the better fit. Next, it is important to ensure that data is securely transported to the oracle, thus the developers need to identify an encryption method that best serves their purpose. Although there are few methods, it is important to carefully think about this step. Now that the origin of data and how to secure the data transmission has been determined, it is important to explore the level of trust necessary to handle this information by exploring oracle types and oracles data sources approach. While a single-source approach can be beneficial in projects that are limited in scope or already use a permissioned blockchain, the multi-source approach might be useful for efforts involving multiple actors or to augment a public permissionless blockchain network. Due to the immutable nature of the blockchain, it is critical to set up a data validation mechanism to ensure truthful and correct information is injected into the blockchain. Choosing to trust the third party or relying on a form of consensus mechanism will ensure trusted information is injected. Lastly, an integration approach is identified based on the above parameters.

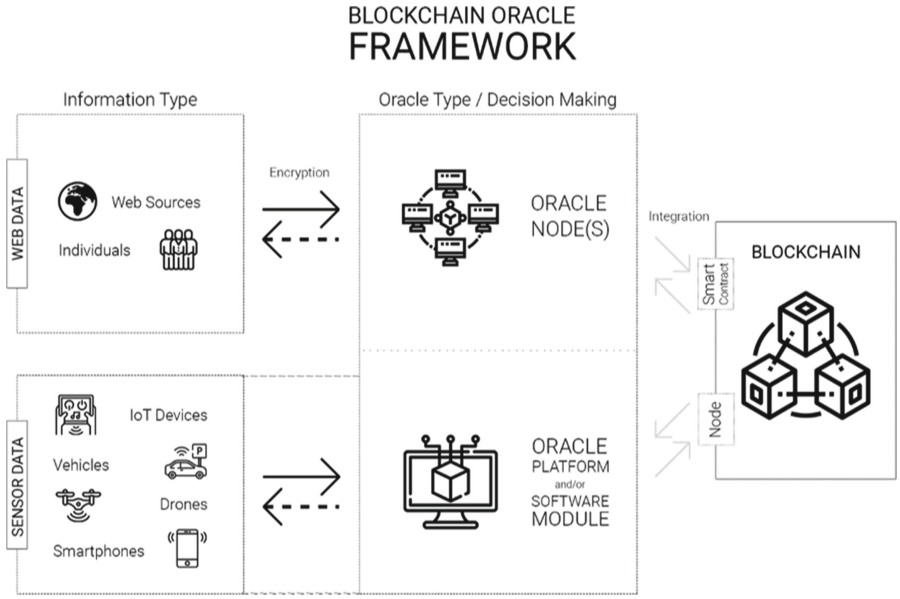

Fig. 2. Blockchain oracle framework (a graphical representation)

5.2 Practical Applicability

Consider an analyst working with developing a blockchain-based solution or a software engineer exploring a potential approach of bringing external data to the blockchain. In this context, the analyst will need to consider the requirements on data being, such as reliability and trust, sent and retrieved from the blockchain. Here, the framework can aid the analyst in understanding what types of oracles are available, their properties, and how they relate to different blockchain solutions, such as permissioned or public ones. An engineer needs to consider, for instance, the origin of data, the level of trust required, and how the data is to be validated. To this end, the framework can assist the engineer in finding a suitable solution by drawing on existing research. For instance (first row of Table 1), in case of web content data (e.g. stock price data) is required to be sent to a permissioned blockchain, PKI being sufficient as encryption, the data can be gathered from a single source provided by a trusted third party e.g., a stock exchange, and the oracle integrated via smart contracts to the blockchain, EdenChain, a programmable economy platform [2], becomes a viable solution to consider. Thus, the framework can aid both analysts who work with conceptual design and engineers focusing on technical implementations in their work to develop blockchain-based solutions.

5.3 Threats to Validity

In this section, we discuss threats to validity as outlined in [39]. Threats to validity that are particularly relevant for SLR are *restricted time span*, the *bias in study selection* and *bias in data extraction*. The threat to validity concerning restricted time span represents the inability of the researcher to anticipate relevant studies outside the time span defined and prepared in the planning phase. Blockchain is a constantly evolving technology with more applications and technologies introduced on a daily basis. Thus, we could not anticipate other relevant studies simply because they were published after the date of our search and, thereby, not included in the primary papers. While it is difficult to account for this, the review protocol includes the dates for all extractions.

Another threat to validity concerns bias in study selection. Such a bias stands for the subjective conjecture which reviewers have in the process of search. This may result in not fully and consistently applying the inclusion and exclusion criteria. This bias could have affected this review due to the knowledge and experience of the authors in the area of blockchain oracles. This is particularly problematic as research on oracles is still in its beginning phases. The terminology is, at times, used inconsistently. For instance, some authors introduced the term *"verifier"* and *"reverse verifier"* as an alternative term for oracles. However, in their more recent work, they use the term oracles. We reduced this threat to validity by testing the search strings. The testing showed that the term IoT was used and we, therefore, included it in our search strings.

Another threat to validity associated with SLR is that of bias in data extraction. As mentioned earlier, the field is still forming and therefore, certain concepts related to oracles are introduced in the papers but not always properly defined and explicitly discussed. In such cases, we had to discuss and, based on our best understanding, extract the data. It is possible that certain data extracted is not fully accurate. We attempted to reduce this threat to validity by discussing all such cases until a common understanding was formed.

6 Concluding Remarks

In this paper, we propose the blockchain oracle framework. This framework could potentially guide developers of blockchain-based applications when incorporating oracles. The framework explains the origins of data provided to the blockchain, oracle properties (how the data is treated during the transactions), encryption method, oracle data source, how it is validated and integrated to the oracle-based applications. As future research, we wish to validate the framework empirically. We also observed that the current literature does not consider the size of the data transferred from oracles. Given that this affects the technical architecture and performance of oracles, this remains as a venue for future study.

References

1. Adler, J., Berryhill, R., Veneris, A., Poulos, Z., Veira, N., Kastania, A.: Astraea: a decentralized blockchain oracle. In: 2018 IEEE International Conference on Internet of Things & IEEE Green Computing and Communications & IEEE Cyber, Physical and Social Computing & IEEE Smart Data, pp. 1145–1152 (2018)
2. Ahn, J.: EdenChain: the programmable economy platform version 1.2 (2018)
3. Al-Breiki, H., Rehman, M.H.U., Salah, K., Svetinovic, D.: Trustworthy blockchain oracles: review, comparison, and open research challenges. IEEE Access **8**, 85675–85685 (2020)
4. Beniiche, A.: A study of blockchain oracles. http://arxiv.org/abs/2004.07140
5. Bisola, A.: Blockchain oracles explained (2018). https://www.mycryptopedia.com/blockchain-oracles-explained
6. Blockchainhub-Berlin: blockchain oracles (2019). https://blockchainhub.net/blockchain-oracles
7. Cloudflare: what is transport layer security (TLS)? (2019). https://www.cloudflare.com/learning/ssl/transport-layer-security-tls
8. De Pedro, A.S., Levi, D., Cuende, L.I.: Witnet: a decentralized oracle network protocol. CoRR (2017). http://arxiv.org/abs/1711.09756
9. Ellis, S., Juels, A., Nazarov, S.: ChainLink: a decentralized oracle network, September 2017. https://link.smartcontract.com/whitepaper
10. Fink, A.: Conducting Research Literature Reviews: From the Internet to Paper, 5th edn. SAGE Publications, Inc., Thousand Oaks (2019)
11. Gao, Z., et al.: Blockchain-based identity management with mobile device. In: 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems, pp. 66–70 (2018)
12. Guarnizo, J., Szalachowski, P.: PDFS: practical data feed service for smart contracts. In: Sako, K., Schneider, S., Ryan, P.Y.A. (eds.) ESORICS 2019. LNCS, vol. 11735, pp. 767–789. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29959-0_37
13. Hardjono, T., Smith, N.: Cloud-based commissioning of constrained devices using permissioned blockchains. In: 2nd ACM International Workshop on IoT Privacy, Trust, and Security, pp. 29–36 (2016)
14. Hepp, T., Wortner, P., Schönhals, A., Gipp, B.: Securing physical assets on the blockchain: linking a novel object identification concept with distributed ledgers. In: 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems, pp. 60–65 (2018)
15. Hess, Z., Malahov, Y., Pettersson, J.: Æternity blockchain (2017). https://blockchain.aeternity.com/\OT1\aeternity-blockchain-whitepaper.pdf
16. Kang, J., et al.: Blockchain for secure and efficient data sharing in vehicular edge computing and networks. IEEE Internet Things J. **6**(3), 4660–4670 (2019)
17. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering (2007)
18. Levy, Y., Ellis, T.J.: A systems approach to conduct an effective literature review in support of information systems research. Inform. Sci. Int. J. Emerg. Transdiscip. **9**, 181–212 (2006)

19. Liang, X., Zhao, J., Shetty, S., Li, D.: Towards data assurance and resilience in IoT using blockchain. In: MILCOM 2017–2017 IEEE Military Communications Conference (MILCOM), pp. 261–266 (2017)
20. Lombardi, F., Aniello, L., De Angelis, S., Margheri, A., Sassone, V.: A blockchain-based infrastructure for reliable and cost-effective IoT-aided smart grids. Living Internet Things Cybersecur. IoT **2018**, 1–6 (2018)
21. López-Pintado, O., García-Bañuelos, L., Dumas, M., Weber, I., Ponomarev, A.: Caterpillar: a business process execution engine on the Ethereum blockchain. Softw. Pract. Exp. **7**(49), 1162–1193 (2019)
22. Michelin, R.A., et al.: SpeedyChain: a framework for decoupling data from blockchain for smart cities. In: 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, pp. 145–154 (2018)
23. Missier, P., Bajoudah, S., Capossele, A., Gaglione, A., Nati, M.: Mind my value: a decentralized infrastructure for fair and trusted IoT data trading. In: 7th International Conference on the Internet of Things (2017)
24. Mo, B., Su, K., Wei, S., Liu, C., Guo, J.: A solution for internet of things based on blockchain technology. In: 2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), pp. 112–117 (2018)
25. Okoli, C.: A guide to conducting a standalone systematic literature review. Commun. Assoc. Inf. Syst. **37**, 43 (2015)
26. Pan, J., Wang, J., Hester, A., Alqerm, I., Liu, Y., Zhao, Y.: EdgeChain: an edge-IoT framework and prototype based on blockchain and smart contracts. IEEE Internet Things J. **6**(3), 4719–4732 (2019)
27. Peck, M.E.: Blockchains: how they work and why they'll change the world. IEEE Spectr. **54**(10), 26–35 (2017)
28. Peterson, J., Krug, J., Zoltu, M., Williams, A.K., Alexander, S.: Augur: a decentralized oracle & prediction market (2019). https://augur.net/whitepaper.pdf
29. Ritzdorf, H., Wüst, K., Gervais, A., Felley, G., Čapkun, S.: TLS-N : non-repudiation over TLS enabling ubiquitous content signing for disintermediation. In: Network and Distributed System Security Symposium (NDSS) (2018)
30. Saleh, G., Draskovic, D.: Datapace: decentralized data marketplace based on blockchain (2017). https://datapace.io/datapace_whitepaper.pdf
31. Santo, A., Minowa, I., Hosaka, G., Hayakawa, S., Kondo, M.: Applicability of distributed ledger technology to capital market infrastructure. In: JPX, vol. 15 (2016)
32. ScienceDirect: transport layer security (2019). https://www.sciencedirect.com/topics/computer-science/transport-layer-security
33. Sztorc, P.: Truthcoin: peer-to-peer oracle system and prediction marketplace (2015). https://github.com/psztorc/Truthcoin
34. Tschorsch, F., Scheuermann, B.: Bitcoin and beyond: a technical survey on decentralized digital currencies. IEEE Commun. Surv. Tutorials **18**(3), 2084–2123 (2016)
35. Uddin, M.A., Stranieri, A., Gondal, I., Balasubramanian, V.: Continuous patient monitoring with a patient centric agent: a block architecture. IEEE Access **6**, 32700–32726 (2018)

36. Yayun, F.: Prophet: the prediction platform based on GXChain (2018). https://bitmart.zendesk.com/hc/en-us/articles/360012745833-Prophetset-PPS-
37. Zhang, F., Cecchetti, E., Croman, K., Juels, A., Shi, E.: Town crier: an authenticated data feed for smart contracts. In: 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 270–282 (2016)
38. Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H.: An overview of blockchain technology: architecture, consensus, and future trends. In: 2017 IEEE International Congress on Big Data (BigData Congress), pp. 557–564 (2017)
39. Zhou, X., Jin, Y., Zhang, H., Li, S., Huang, X.: A map of threats to validity of systematic literature reviews in software engineering. In: 2016 23rd Asia-Pacific Software Engineering Conference (APSEC), pp. 153–160 (2016)