Connect API with Blockchain: A Survey on Blockchain Oracle Implementation

AMIRMOHAMMAD PASDAR and YOUNG CHOON LEE, School of Computing, Macquarie University, Australia
ZHONGLI DONG, Aglive Pty Ltd, Australia

A blockchain is a form of distributed ledger technology where transactions as data state changes are permanently recorded securely and transparently without the need for third parties. Besides, the introduction of smart contracts to the blockchain has added programmability that has revolutionized the software ecosystem toward decentralized applications (DApps). Although promising, the usability of smart contracts is primarily limited to on-chain data without access to the external systems (i.e., off-chain) where real-world data and events reside. This *connectability* to off-chain data for smart contracts and blockchain is an open practical problem referred to as the "oracle problem" and is defined as how real-world data can be transferred into/from the blockchain. Hence, *Blockchain oracles* are introduced and implemented in the form of application programming interfaces (APIs) connecting the real world to the blockchain for mitigating such a limitation. This paper studies and analyzes how blockchain oracles provide final feedback (i.e., outcome) to smart contracts and survey blockchain oracle technologies and mechanisms regarding data integrity and correctness. Since the existing solutions are extensive in terms of characteristics and usage, we investigate their structure and principles by classifying the blockchain oracle implementation techniques into two major groups voting-based strategies and reputation-based ones. The former mainly relies on participants' stakes for outcome finalization, while the latter considers reputation and performance metrics in conjunction with authenticity proof mechanisms for data correctness and integrity. We present the result of this classification with a thorough discussion of the state-of-the-art and provide the remaining challenges and future research directions in the end.

CCS Concepts: • **Security and privacy → Distributed systems security**.

Additional Key Words and Phrases: Blockchain, Decentralized oracle, Blockchain oracle, Data feed, Smart contracts.

ACM Reference Format:
Amirmohammad Pasdar, Young Choon Lee, and Zhongli Dong. 2021. Connect API with Blockchain: A Survey on Blockchain Oracle Implementation. *ACM Comput. Surv.* 0, 0, Article 0 (2021), 36 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The use of blockchain in the past decade has been exponentially noticed in nearly every industry, including, but not limited to, supplying products [69] or food security [2]. This technology has also become a good fit for decentralized finance (Defi), providing immutability and transparency to the software ecosystem. In addition, it has attracted financial and investment institutions, e.g., Santander UK bank officially announced the first blockchain-based international money transfer

Authors' addresses: Amirmohammad Pasdar, amirmohammad.pasdar@hdr.mq.edu.au; Young Choon Lee, young.lee@mq.edu.au, School of Computing, Macquarie University, Sydney, NSW, Australia, 2109; Zhongli Dong, Aglive Pty Ltd, Sydney, Australia, andrew@aglive.com.

0360-0300/2021/0-ART0 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

service in 2018 [80]. Hence, blockchain technology is a growing area of interest for research communities, companies, and industries as the backbone of their distributed systems in a trustless and permissionless way [92].

A blockchain is a form of distributed ledger technology where transactions are replicated and saved onto many nodes. Transactions are a set of data packages for storing monetary value, parameters, and function calls. Cryptographic techniques ensure the integrity of transactions, and transactions are collected in the form of blocks where they become immutable records. Each block is linked to the succeeding block through the hash, and blocks are appended to the ledger by means of consensus algorithms such as proof of work (PoW) [36], proof of stake (PoS) [68], or practical byzantine fault tolerance (PBFT) [49]. In addition, programmability has been added to the blockchain by introducing smart contracts. Smart contracts are programs (i.e., functions and states) that are encoded and compiled into bytecode, and upon deployment on the blockchain, they are given unique addresses. Smart contracts are executed on the blockchain to digitally facilitate the transaction process as they are event-driven, self-executable, and resistant to tampering. They consume transaction fees based on the complexity of code (e.g., gas for deployment in the Ethereum blockchain) and only use resources available on the blockchain network [25].

Smart contracts and blockchain are similar to an enclosed system that *does not* have access to the information outside (i.e., connectability to off-chain data), meaning interactions are limited to the data available on the blockchain. This is an open practical problem referred to as the "oracle problem" defined as how real-world data can be transferred into/from the blockchain. Suppose the blockchain is assumed as a component of a larger software system [88]. In that case, smart contracts need the external information (e.g., the highly volatile exchange rate) relevant to contractual agreements or practical applications, e.g., data availability verification for decentralized applications or adjudication mechanisms. A superficial solution might be storing the real-world information onto the blockchain providing access to data on-chain. However, there are flaws and drawbacks to this approach. Firstly, data might be big that storing it on the blockchain would be impossible. Secondly, in addition to assessing the authenticity and reliability of data, its confidentiality and privacy due to third parties' internal procedures and policies would be a considerable hurdle for transferring owned data to the blockchain. Last but not least, it would be costly to store all the information on the blockchain due mainly to its limited storage space and charges incurred per transaction on the blockchain.

Therefore, blockchain oracles (also known as *data feeds*) shown in Figure 1 are introduced as trusted third-party services. They send and verify the external information and submit it to smart contracts for triggering state changes in the blockchain. Blockchain oracles are a combination of smart contracts implemented in the form of an application programming interface (API) and off-chain components (i.e., data providers) for serving data requests by other contracts [93] to connect the world to the blockchain. They resolve connectability issues as they may not only relay information to smart contracts but also send it to external resources and broaden the scope of smart contracts operation [11]. In addition, monetary incentives support blockchain oracles to encourage users to participate in network governance and improve blockchain oracle security and functionality.

There have been several surveys on blockchain oracles [4, 9, 33, 45, 52, 90] each of which studied particular aspect(s) of oracles. Muhlberger et al., [52] examine oracles from two dimensions which are data flow direction and the data initiator. Heiss et al., [52] present a key requirement set for trustworthy data on-chain and how related challenges should be addressed. Beniiche [9] describes widely used blockchain oracles and human oracles, and Al-Breiki et al., [4] study the trust used in the leading blockchain oracles. Xu et al., [90] discuss oracle roles and provide the benefits and

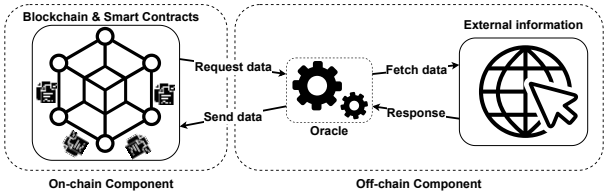

Fig. 1. The oracle role and its off-chain and on-chain components as trusted third-party services for sending and verifying the external information. Blockchain oracles can also be a pool of oracles interacting with the external world.

drawbacks, and Mammadzada et al., [45] provide general characteristics of a blockchain framework to be considered when designing blockchain-based applications.

In this paper, we take a novel approach to reviewing and analyzing blockchain oracles revolving around their technical perspective and implementation as APIs that previous studies have overlooked. In particular, we conduct research on blockchain oracles in terms of providing final feedback (i.e., outcome) to the smart contracts, and we mainly seek an answer for the following research question: *How are blockchain oracles designed and implemented to provide the outcome to smart contracts?*. The answer to this research question will contribute to providing deep insights into the design and development, implementation requirements, and usage of blockchain oracles. Therefore, we conduct the survey and review on blockchain oracles' implementation through the multivocal literature review (MLR) technique, defined as a form of systematic literature review (SLM) in which "grey" literature and "white papers" are included [29]. The initial search keywords are selected based on a combination of *blockchain oracles*, *data feed*, and/or *smart contracts*, *design*, *implementation* and/or *pattern* for covering the vast majority of related studies. During the paper collection, snowballing technique [86] is employed to collect relevant studies for the literature review. Several digital libraries are used to extract a fine set of papers on the proposed topic such as IEEE Xplore[1], ScienceDirect[2], ACM Digital Library[3], DBLP:Computer Science Bibliography[4], and Google Scholar[5].

Given the volume of research studies, including practical uses cases on the blockchain oracles, we focus on technical implementation, technologies, and mechanisms used to provide feedback by highlighting their characteristics, structures, and principles. In this regard, through a comprehensive review of cutting-edge studies and the monetary incentive mechanisms employed in the blockchain oracles, it is uncovered that the blockchain oracle design, implementation, and usage fall into two major groups; voting-based strategies reputation-based ones. The first group is oracles that employ participants' votes and related mechanisms. These mechanisms can be putting stakes or using the blockchain-specified tokens for data aggregation and outcome but may lack data integrity techniques. The second group leverages reputation and performance metrics to assess and selects the oracle for reporting the outcome to the blockchain. They may employ authenticity proof mechanisms to prove the integrity and correctness of obtained data from external resources. Thus, the contributions of this article revolve around: (1) a breakdown of blockchain oracle implementation and design patterns, including a comprehensive review and discussion of their challenges, advantages, and disadvantages, (2) a coarse-grained classification of existing authenticity proof

[1]https://ieeexplore.ieee.org/Xplore/home.jsp
[2]https://www.sciencedirect.com/
[3]https://dl.acm.org
[4]https://dblp.org/
[5]https://scholar.google.com.au/

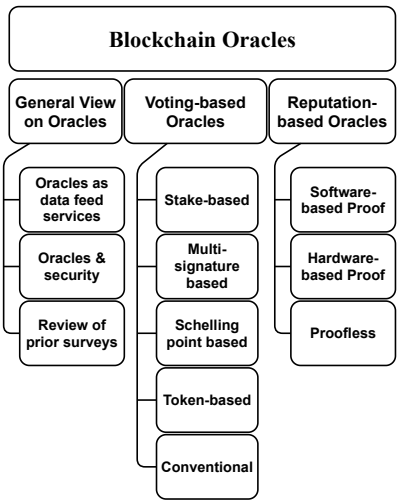

Fig. 2. The structure of research study.

mechanisms for data integrity and correctness and how they benefit blockchain oracles, and (3) providing essential requirements and principles in the design and implementation of blockchain oracles. Figure 2 shows the structure of this research survey.

The paper is organized as follows: Section 2 briefly overviews the high-level classification of oracles, and their security and reviews survey papers on blockchain oracles. In Section 3 we present a finer classification of voting-based oracles as a way of providing oracle feedback which is followed by Section 4 in which authenticity proof mechanisms are also studied. Section 5 presents research challenges and future research directions, and we conclude the paper in Section 6.

2 GENERAL VIEW ON ORACLES AND PRIOR RESEARCH

This section presents a general classification of oracles as data feed services, issues, and discussions on prior survey research on blockchain oracles.

2.1 Oracles as Data Feed Services

Oracles (or data feed services) respond to queries, e.g., Ethereum/USD exchange rate. Oracles may consult with different sources (or a single source) to fetch the required information and return it to the smart contracts. These data feed services may act as *computation oracles* such as Truebit [77] or Provable [63] where they perform user-defined computation-intensive tasks off-chain. They provide computing power to the blockchains and enable a decentralized token economy. Hence, by looking into their structure and design, a general classification can be inferred: (1) source; the origin of data, (2) information direction; inbound or outbound, (3) trust; centralized or decentralized, and (4) data-oriented patterns [4, 9].

Oracle sources can be chosen from (a) software oracles where data comes from online sources (e.g., online servers), (b) hardware oracles where data comes from the physical world (e.g., sensors), and (c) human oracles that are also responsible for verifying the authenticity of information and its conversion into smart contracts. *Information direction* means the way information flows; from/to external resources (inbound/outbound oracles).There is the concept of *trust* that can be centralized or decentralized. Centralized oracles are efficient, but they can be risky because a single entity provides information and controls the oracles. A failure makes the contracts less resilient to vulnerabilities and attacks. In contrast, decentralized oracles (i.e., consensus-based oracles) increase

the reliability of the information provided to the smart contracts by querying multiple resources. It should be noted that an oracle is considered decentralized if it is permissionless- users can join or leave, and every user has an equal privilege [47]. Finally, data-oriented patterns are defined as (a) request-response when the data space is huge and can be implemented as on-chain smart contracts and initiated on-chain, or off-chain oracles for monitoring, retrieving, and returning data (b) publish-subscribe when the data is expected to change, e.g., RSS feeds, and (c) immediate read when the data is required for an immediate decision.

2.2 Oracles and Security

Data feed services may have desired properties such as how easily they are parsable, adopted, and deployed. In addition, properties such as their authenticity and being non-equivocation (i.e., being unable to modify or delete data when it becomes published) exist [31]. However, it may come with issues that emerge from putting *trust* in a single third party. It is represented as a single *point of failure* because external malicious actors can break into a single system and alter or delete facts. Oracles are prone to be hacked; their process is vague [10], they can be bribed, and may not be stable [57]. Also, smart contracts lack *direct* network access, and the use of transport layer security (TLS) to fetch information while keeping data untampered during transmission is not enough [66]. Hence, mechanisms should exist to sign the data for verification digitally, and in this regard, oracles are neither tamper-resistant nor trustless.

Oracles do not mainly hold security properties of native blockchain protocols. However, the correctness of the data can be attested through authenticity proof mechanisms, e.g., software-based [66, 78] or hardware-based approaches [42, 93]. Although an ideal oracle is hard to achieve, oracles must provide the same level of security in proportion to the blockchain they support in the form of integrity, confidentiality, and availability. High economic security is defined as financial resources required for compromising a network. It should be so that compromising a network would not be beneficial if the financial benefits are not higher than the cost. Hence, the higher the decentralized oracle platform degree, the more nodes to be compromised. For example, Truebit [77] is believed to be the first scalable off-chain computation protocol designed for the Ethereum blockchain. It employs incentive models as well as proofs via off-chain solvers and challengers. If there is a dispute, solvers and challengers employ an off-chain verification by checking the computation steps. It is powered by an on-chain interpreter recursively to reach a point where they disagree with the state change. The final value is decided on-chain, verifying the validity of one of the state changes. Truebit incentivizes challengers via jackpot repository for auditing purposes, outsourcing computation off-chain while maintaining verification on-chain.

2.3 Prior Research on Blockchain Oracles

Although the given classification may provide information about the oracle's role, it does not offer *technical* aspects of blockchain oracles, including design patterns [58]. Xu et al., [89, 90] provide insights about the oracle roles, benefits, and drawbacks from another perspective. They argue that oracles can be implemented as smart contracts in the blockchain network where an external state is periodically injected into the oracle by an off-chain injector. This type of oracle imposes drawbacks on the blockchain as all participants involved in the transaction should trust the oracle. Also, the injected external states cannot be fully verified by other validators, i.e., miners.

The authors state *reverse oracles* are also in need. It is due to sometimes off-chain components may need to have access to data stored on the blockchain (or the smart contracts running on the blockchain) to provide data or checks. From their point of view, one crucial aspect of the reverse oracle is interactions that should be *non-intrusive*. A non-intrusive interaction is defined as not changing the system core design while it should be through the configuration of smart contracts

function or visibility of the transaction on the blockchain. However, adding such a component in a non-intrusive may not be possible, e.g., the Nakamoto consensus algorithm may be inconsistent with normal transaction semantics in enterprise systems. Finally, they explain that a bidirectional binding can exist between off-chain legal contracts and on-chain smart contracts. Digitizing legal contracts and smart contracts could be done on the blockchain, where smart contracts implement some conditions. This model comes with drawbacks such as expressiveness since some items of a legal contract cannot be translated into the code. Also, enforceability would be questioned by using a public blockchain, and different interpretations of the conditions and coding them into the smart contract may exist.

Furthermore, Xu et al., in another study [88] consider the blockchain as a software connector, which could be a decentralized solution to centralized shared data storage. Although information transparency and traceability are improved, communication latency increases due to the mining mechanism resulting in a poor user experience. The authors propose that a good practice for public blockchain is to keep the raw data stored off-chain and only meta-data be injected into the blockchain.

In contrast, Muhlberger et al., [52] examine blockchain oracles based on four different scenarios for the data flow direction and the initiator of the data. The inbound oracle data fetches data from the outside world and pushes data onto the blockchain network. Based on the data initiator, it can be pull-based or push-based inbound oracle. Upon receiving the request, the former collects the state from off-chain components and sends the result back to the blockchain (via a transaction). In the latter, the off-chain state is sent to the on-chain component by the off-chain component. The outbound mode is where information from the blockchain is transmitted to the external world. If the outbound becomes pull-based, the off-chain component retrieves on-chain states from an on-chain component. The on-chain component sends the off-chain state to an off-chain component in the push-based. Muhlberger et al., through quantitative analysis, reveal that the pull-based inbound oracle is the fastest, and the push-based outbound oracle is the slowest.

Beniiche [9] reviews the most widely used oracle services, such as Provable and ChainLink, and provides the general architectures of the oracles. Beniiche also considers human oracles with an introduction to Augur and Gnosis as the leading prediction markets. Based on the discussed architectures, it classifies oracles into three design groups; publish-subscribe, immediate read, and request-response. In contrast, Al-Breiki et al., [4] study leading blockchain oracles (and services) in terms of the trust. The authors review their system architectures along with the advantages and disadvantages. Mammadzada et al., [45] present a blockchain oracle framework that assists developers and decision-makers with the design of blockchain-based applications. The framework considers data origins, how data is processed during transactions, validation, and integration into the applications as the fundamental criteria for the framework.

In comparison to studies, [4, 9, 45, 52, 90], Heiss et al., [33] provide a set of key requirements for trustworthy data on-chain, explaining the challenges and the solutions for them. They argue that in addition to safety and liveness as the characteristics of a distributed system, *truthfulness* is necessary as it does not allow execution of blockchain state transition by untruthful data provisioning. Safety refers to avoiding triggering blockchain state transition by incorrect data and liveness is defined as blocking blockchain state transition when data is unavailable.

Table 1 provides the summary of the existing literature review on the blockchain oracles.

3 VOTING-BASED ORACLES

Although oracles can provide feedback to the submitted queries, there may be inconsistencies and discrepancies between the received responses from the oracles to the blockchain and smart contracts. This issue can be resolved by involving users to form a set of voters and/or certifiers.

Table 1. Summary of existing literature review on blockchain oracle

Literature	Key Research Outcomes
Xu et al., [88]	A discussion on validation strategies for oracles which can be internal or external. The former discusses injecting the external state into the blockchain causing latency and trust management issues, but the latter's issue is trusted third parties.
Muhlberger et al., [52]	Classifying oracles into four groups based on the information flow direction and data initiator; inbound pull-based, inbound push-based, outbound pull-based, and outbound push-based.
Heiss et al., [33]	Data on-chain trustworthy requirements and challenges are explained.
Beniiche [9]	Describing widely used oracles and human oracles, and with respect to their architecture classifies oracles into publish-subscribe, immediate read, and request-response.
Al-Breiki et al., [4]	Studying trust in the leading blockchain oracles, and reviewing their system architecture, advantages, and disadvantages.
Mammadzadea et al., [45]	Describing blockchain oracle framework for assisting developers with the design of blockchain-based applications taking into account data origins, processing data during transactions, validation, and integration to the applications.
Xu et al., [89, 90]	Classifying oracles into three groups; conventional oracles, reverse-oracles, and legal and smart contract pairs.

They can participate in the process of data correctness approval that is shown in Figure 3. Each voter and certifier put stakes on responses to verify the data. If the outcomes are matched, rewards are distributed between them; otherwise, they are penalized. It may also inherit the game theory concept Nash Equilibrium (e.g., [1, 38, 53]). It is defined as the determination of the optimal solution in a non-cooperative game in which each player does not have any incentive to alter the initial strategy. It leads to nothing from changing their initially chosen strategy if other players keep their strategies unchanged.

Use cases of voting strategies can be seen in prediction market platforms implementation, e.g., Augur, Gnosis, and X Predict Market [30, 44, 59]. Prediction markets are platforms where participants create, share, and exchange financial shares in outcomes or facts. These platforms enable users to bet on anything, e.g., political forecasting, and receive compensation or become penalized if they are correct or wrong. Prediction markets are resistant to manipulation, are largely scalable, and can help with the aggregation and distribution of unlimited information. Data in the prediction markets depends on the number of users participating. The more participants, the more data and, consequently, the more effective the prediction markets are. Prediction markets can be based on distributed oracles, e.g., the Delphi-based prediction market called Omphalos [21]. Markets should have a tradable market price at all times, known as market liquidity. Prediction markets can also be multi-dimensional markets in which users trade on the state probabilities and the relationship between dimensions.

The voting-based strategy and its implementation raise issues in the incentivized platform. There is a term called lazy equilibrium- a form of verifier's dilemma- in which voters always return the same answer to questions to secure profits without performing works for correctness. The other issue is that Sybil attacks are defined as attackers out-vote honest nodes on the network by creating multiple fake nodes to take over the network. A Sybil attacker can also employ mirroring that makes oracles post individual responses based on a single data-source query. Freeloading is another issue defined as a cheating oracle obtaining and copying another oracle's response without paying per-query fees. This issue threatens the response time of oracles but can be addressed with the commit and reveal strategy. The strategy is about sending cryptographic commitments to the responses and revealing the responses in the next round.

The voting-based oracles in a more refined classification and based on the most significant *characteristic* are categorized into the following groups. However, they might also be fallen into the other groups. In the following sections, each group will be discussed in detail, providing information about their advantages or disadvantages.

- Stake-based Systems: Participants should provide their stake to contribute to the outcome in these systems. Therefore, based on the outcome, they may receive rewards or be penalized.
- Multi-signature based Systems: These systems are designed to finalize the outcome by putting their stakes in a safe address. Once the outcome is ready, participants are rewarded or penalized.
- Schelling Point bases Systems: The basic concept is choosing a set of data providers' answers based on the median value.
- Token-based Systems: These systems employ tokens for finalizing the outcome while ensuring the integrity of data provided by the participants.
- Conventional: In this group, the blockchain oracles are to transfer data into the blockchain without employing any techniques mentioned above.

3.1 Stake-based Systems

Astraea [1] is a general-purpose decentralized oracle running on the public ledger that relies on a voting-based game strategy. This framework employs monetary and staking fees and ensures the system is immune to Sybil attacks. This oracle has entities that may have one or more roles, such as submitters, voters, and certifiers. Boolean propositions are submitted to the system based on paying a fee by submitters. Voters play a low-risk/low-reward game by placing a small stake of their confidence on the truth of random propositions. In contrast, certifiers play a high-risk/high-reward game by placing a large stake in the outcome of the selected voting and certification process. The outcome of voting and certification is the stake-weighted sum of votes or certifies, respectively, and due to the random nature, it is resistant to manipulation. In weighted votes, the weight (and reward) is affected by the level of the deposit made, because the higher the deposit, the heavier the weight, and the higher rewards and penalties. If the outcome of voters and certifiers are matched, they are rewarded. Otherwise, the players who take the opposite position are penalized. Hence, this oracle encourages players to place bets/votes on propositions with high confidence. The voting and certification deposit should be small or large relative to the total voting stake on the proposition. The former could not control the outcome, and the latter could be penalized and could not tamper with the outcome.

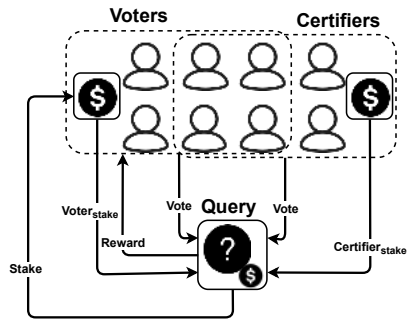

Fig. 3. Stake-based systems overall structure. The reward is only distributed when certifiers and voters' outcomes are matched. Otherwise, certifiers are penalized. Each query comes with a bounty.

Kamiya presents Shintaku [38] as an end-to-end decentralized oracle that is blockchain-agnostic for deciding on the outcome of binary propositions. It relies on a stake-based voting scheme that rewards voters for being honest. The work is an extension to Astraea [1], in which the verifier's dilemma is handled. The issue with the main contribution (Astraea [1]) is that the reward pool is non-zero most of the time, which is filled in with the penalties obtained from the system. Hence, the system could *lazily* vote and certify toward a single outcome forever. To eliminate this, Kamiya argues that the payout must be zero. Submitters and voters exist in the system, and each voter receives two propositions. They are only rewarded when their choice for the propositions differs, which would lead to returning their bond. The author suggests that voting pools (similar to mining pools but cheaper) can be constructed via an off-chain frontend to have such a decentralized implementation. It makes on-chain transactions, and voters can freely move between the pools to reduce the risk of centralization.

Similar to Shintaku [38], Merlini et al., [47] also present a paired-question oracle protocol to extract true answers from the public. It shows a Nash equilibrium of truthful reporting with the advantage of re-balancing. The user submits a pair of antithetic questions with a bond. The voting users answer them to obtain a reward. The oracle collects votes and checks whether the two questions converged to different answers. If so, the submitter regains their bond, and voters are rewarded (penalized) for agreement (disagreement) with the majority answer. Otherwise, the submitter loses the bond, and voters receive nothing. In comparison to [1, 38], truthful voters receive larger expected payoffs.

Cai et al., [15, 16] present a peer prediction-based protocol with a non-linear stake scaling for decentralized oracles. In comparison to [47], a lightweight scoring rule controls the rewards for voters, and it considers the behaviors of the other voters for their answers. Authors argue that Astraea [1] has drawbacks such as enhancing the expected rewards by herding. It reports the opposite of voters' belief when they would find their true opinions belong to the minority group. They also point out that the paired-question protocol used in [38, 47] is susceptible to Sybil attacks. The authors present a non-linear way to control the voting weight and award weight for the submitted stakes that are sub-linear and super-linear scaled, respectively. The oracle assigns questions to voters, and reports in the form of a binary answer, including a popularity prediction, are collected. The majority of the information determines the oracle answer weighted by the associated stakes and adjusted by a sub-linear function. Then, a score is assigned to each report based on the accuracy and degree of agreement with peers. Only top-scored voters are awarded, while the share of the award is determined by their stake adjusted by a super-linear function. In comparison to Astraea [1], the system encourages minority voters to vote based on their true opinion to receive an award. The next benefit of the approach is non-linear stake scaling. An honest voter is incentivized to stake more onto a single report while increasing the penalty for a participant to bias the outcome with Sybil attacks.

Similar to Astraea [1], Nelaturu et al., [53] propose a voting-based game oracle that evaluates the truth or falsity of a query. This framework leverages the crowd-sourced voting mechanism agnostic to the blockchain consensus protocol. It is deployed on existing platforms such as Bitcoin and Ethereum. There are user roles; submitters who send the queries to the blockchain in conjunction with funding it, randomly selected reporters playing a low-risk/low-reward game. Upon participation, stakes must be deposited, and certifiers play high-risk/high-reward games. They have the choice to choose the query they want to put the deposit into. Voters and certifiers have outcomes defined as a function- the sum of the votes weighted by the deposits. The termination takes place when the query has attached sufficient funds. Based on the proposed protocol, a light version of the protocol is presented where only submitters and reporters exist.

Band protocol [8] is a blockchain-agnostic framework that has its native token for connecting public blockchains to the off-chain information. This framework supports generic data requests and on-chain aggregation with WebAssembly. For retrieving data from Bandchain's oracles, an oracle script is necessary that is defined as an executable program that encodes raw data requests and aggregates final results. Participants in the framework are validators and delegators. The former is based on a random selection responsible for proposing and committing new blocks to the blockchain. They participate in the consensus protocol by broadcasting votes and supporting external data queries. The latter stakes their holding on the network of validators, and they can join in the network governance as their voting power is proportional to the size of the stake they hold. Each oracle script requires outlining the data sources and sending the request to the chain's validators for fetching data. Also, it aggregates the returned results into final results such that the creator controls the aggregation policy. Validators also have voting power in which the tallied voting should be greater than two-thirds of the system. Upon storing onto the BandChain, an oracle data proof is created.

Razor [34] is a decentralized oracle network that offers maximum game-theoretical security without compromising speed. Razor network consists of stakers responsible for responding to the queries from a queue, fetching the information from the real world, and being rewarded for reporting honestly. Razor employs proof of stake and has its token named Razor used by stakers whose stake amount can influence the network. Razor protocol relies on high economic security. Hence, it uses a proof of stake chain with Honey Badger BFT as a consensus algorithm network where many individual stakers can participate. Values are reported in consensus with the majority of stakers. Razor uses Median Absolute Deviation (MAD) to measure the consensus. Based on that, votes with absolute division higher than that are penalized. Leveraging the proof of stake consensus protocol reduces malicious behaviors by reporting inaccurate data points to influence the result. Razor also employs game-theoretical and cryptographic strategies such as a commit-reveal scheme to provide further collusion and censorship resistance. The Razor architecture consists of four parts; oracles composed of stakers for processing queries, a job manager responsible for accepting and prioritizing queries based on fees, client applications or smart contracts, and users. For each query, in addition to fees for using oracles, there is a validity bond incentivizing clients to provide valid and reliable sources equal to the maximum potential lost due to the incorrect source. Providing a reliable source to users can be another point of failure and data integrity issue.

Oraichain [56] is recognized as a data oracle platform that employs artificial intelligence models and uses the ORAI token for payments and governance. The Oraichain core technology is similar to Tellor [76], or DIA [55], but it focuses more on the AI. The issue with the use of AI in smart contracts roots in the languages such as Solidity or Vyper [81] which are not suitable for AI model implementation. Hence, Oraichain tries to bridge the gap by enabling secure access between smart contracts and AI. Oraichain is a public blockchain that employs a delegated proof of stake (dPoS) consensus protocol providing fast transaction times and completion of data requests quickly. AI models in Oraichain are constantly tested for quality. They come with each data request test case (e.g., face authentication). The AI provider must pass these test cases before receiving payment for sourcing the data request. These test cases are the incentive for providers to keep their AI models more accurate. ORAI token holders, by staking their tokens, can take part in securing the network and can be rewarded. Oraichain is a community-driven platform in which ORAI tokens give holders voting power.

Table 2 provides a summary of studies that employ a stake-based strategy for finalizing the oracle outcome. The use of stakes on the outcomes can mitigate the Sybil attack in the presence or absence of a consensus algorithm. However, it may be prone to the verifier's dilemma issue.

Table 2. The summary of stake-based studies for voting-based oracles

Literature	Key Research Outcomes	Query Type	Sybil Attacks	Verifier's dilemma
Merlini et al., [47]	Presenting a paired-question oracle in which two antithetic questions are submitted and based on the answers bonds and rewards are managed.	Binary	✓	✓
Adler et al., [1]	Presenting a general-purpose decentralized oracle referred to as Astraea with entities such as submitter, voter, and certifier each of which holds stake. Voters play a low-risk/low-reward game while certifiers play the high-risk/high-reward one.	Binary	✓	Partially
Kamiya [38]	An extension to Astraea, in which two propositions are submitted, and based on different responses to the propositions rewards are distributed.	Binary	✓	✓
Nelaturu et al., [53]	Presenting a framework based on the crowd-sourced voting mechanism employing two strategies for oracles; a version similar to [1], and a light version in which only voters (reporters) exist in the system.	Binary	✓	✓
Cai et al., [15, 16]	Presenting a peer prediction-based protocol with non-linear stake scaling. It leverages a light-weight scoring rule for controlling rewards for the voters.	Binary	✓	✓
Band protocol [8]	A blockchain-agnostic framework with a native token, and validators and delegators are the main participants, the former broadcasts votes and the latter stakes their holdings on the validators governing the network.	(Non)Binary, scalar, categorical	✓	✓
Razor [34]	A decentralized oracle framework that employs a weighted-voting mechanism with respect to the stakers' stake.	Categorical or scalar	✓	Partially
Oraichain [56]	A community-driven platform that provides a connection between the smart contracts and AI models utilizing ORAI token.	(Non)binary	✓	✓
Synthetic [84]	An Ethereum-based protocol issuing synthetic assets that maintains its token SNX employing decentralized oracles for price discovery.	Scalar	✓	✓

3.2 Multi-signature based Systems

A multi-signature address is an address on the blockchain associated with more than one private key. A multi-signature transaction needs to have more than one private key for transaction authorization. They are considered as m-out-of-n addresses requiring m keys out of a total of n keys to sign a transaction for adding into the blockchain shown in Figure 4.

Orisi [57] is a Bitcoin-based distributed system for the creation of oracle sets run by independent and trustworthy parties. The Orisi aims to mitigate flaws and issues arising from a single (server) oracle, i.e., the point of failure. In this framework, most oracles need to agree on the outcome for a transaction to be finalized, as it would be costly and hard to bribe more than half of the oracles. For this purpose, Orisi leverages multi-signature addresses (oracles and sender/receiver). The money from senders and receivers is placed into the addresses (i.e., safe address). To increase security, Orisi uses Bitmessage, which is a trustless, decentralized peer-to-peer protocol for sending and receiving messages securely [83]. The Bitmessage protocol employs a hash of the public key. It has a message transfer mechanism similar to Bitcoin transactions, such that each message requires proof of work. Messages are broadcast, and each recipient should apply its private key to decode

them. Hence, employing Bitmessage protects IP addresses for communication with oracles, and senders use Bitmessage to broadcast the transaction on the network. Oracles check the validity of transactions and rules. Then, upon realizing the oracles' acknowledgments of the transaction validity, the sender and receiver send the fund to the "safe". Once oracles notice the condition, they add their signature to the transactions that are broadcast to the network. Also, Orisi uses a timelock verdict when the source becomes unavailable, and it leverages dedicated oracle data feeds or mediation protocol for mitigating hacks. Mediation protocol is defined within a time frame, the receiver can challenge the verdict, and a human operator delivers arbitration.

Gnosis [30] is an open-source infrastructure for building prediction markets on the Ethereum platform, aggregating relevant information from human and artificial intelligence agents. The outcome of events is exchanged in the prediction markets. Gnosis provides the ability to trade cryptocurrencies represented as the outcome of events on the platform, which can be categorical or scalar. Gnosis consists of three primary layers; (1) Core Layer, which interacts with the Ethereum blockchain providing the base functionalities for event contracts that monitor and set the outcome token creation and settlement and a market mechanism, (2) Service Layer that offers optimization tools such as chatbots and stablecoins, and (3) Application Layer that is the Gnosis frontend and targets a particular prediction market or customer segment. Third-party applications in the layer may charge additional fees or use alternative business models, e.g., market making, information selling, or advertising. Consensus can be done via voting, i.e., it requires multiple signatures for approval. Oracles can be on-chain and centralized. The ultimate oracle is triggered by staking 100 Ethereum if users disagree with the reported value.

Delphi [21] offers a light-client strategy in which event filters and social application functions are bundled in helping users to build and deploy distributed oracles. Delphi employs Pythia's weighted signature framework, leading to faster input arbitration, understandable oracle interfaces for developers, and flexibility and extensibility. These distributed oracles can resist Sybil attacks and rely on multi-signature contracts highlighting authorization from more than one entity. It is necessary to generate the oracle output for the weights and threshold, making the consensus trivial. This platform also allows for re-weight signatures, e.g., to vote out misbehaving oracles, and a decaying weight strategy may be applied to the involved oracles such that the weight replenishment requires being honest for providing the truth. The platform leverages a compound token; (1) a minimal token that is atomic, lightweight, easy to use and understandable, and compatible with existing token solutions. (2) The signal component gives the rich functionality for market signaling, seamless and permissionless feature upgrades, and tracking values or rankings over time. It makes it suitable for voting, leading to a Sybil attack-resistant mechanism based on the coin. Finally, (3)

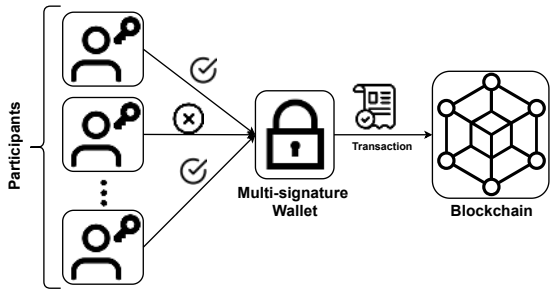

Fig. 4. Multi-signature overall structure. A transaction requires the minimum keys to unlock, i.e., m-out-of-n addresses meaning more signatures for transaction execution will make the decision making more distributed.

the trustless peg component unites two tokens into a single token architecture that gives users toggling balance freely.

Table 3. The summary of Multi-signature based studies for voting-based oracles

Literature	Key Research Outcomes	Query Type	Sybil Attacks	Verifier's dilemma
Orisi [57]	A bitcoin-based distributed system for creating a set of oracles run by independent and truth-worthy parties leveraging multi-signature addresses.	Non-binary	✓	None
Gnosis [30]	An open-source infrastructure to build prediction markets on the Ethererum platform in which event outcomes are traded in the prediction markets.	Categorical or scalar	Partially	Partially
Delphi [21]	Offering a light-weight strategy that employs a weighted signature framework called Pythia and a compound token.	Non-binary	✓	✓
Moudoud et al., [51]	Employing an oracle network for data veracity where follows m-out-of-n multi-signature transaction should be reached for the consensus.	Non-binary	Partially	Partially
DOS Network [54]	A layer-2 protocol provides off-chain computation in a decentralized way, and has two partitions on-chain and off-chain as a client software.	Non-binary	✓	Partially

Moudoud et al., [51] present a permissioned-based and lightweight blockchain architecture for supply chain use cases consisting of distributed internet of things (IoT) entities. It has two blockchains; private for storing private information and the public for tracking production and providing general information to the public. It is a peer-to-peer overlay network involving supply-chain members identified by a public key. Any new member is added when the minimal number of members reaches an agreement. Since data is collected from different locations, oracles are employed to check the correctness of data. Hence, the proposed blockchain uses multiple oracles-the oracle network- for data veracity approval to be divided and approved by various parties. Due to the limited block size, the metadata is kept on-chain. The consensus used for the oracle follows m-out-of-n multi-signature transactions that should be reached among oracle parties.

DOS Network [54] as a layer-2 protocol that provides off-chain computation in a decentralized way to feed the results to the blockchain. A layer-2 protocol is defined as a secondary framework or protocol built on an existing blockchain. Node operators are incentivized by DOS token for providing honest services to receive rewards and providing unlimited decentralized, verifiable computation oracles to mainstream blockchains. The DOS network is resistant to Sybil attacks and chain-agnostic (i.e., it can deal with any smart contract platform). It is also horizontally scalable, offering more capability and computation when more nodes run the DOS client software. DOS network consists of two partitions; (1) on-chain for providing a variety of functionalities and (2) off-chain as client software for implementation of the core protocol. The latter is used by third-party users to obtain economic rewards and constitute a distributed network. The consensus among off-chain clients in the DOS network is achieved by employing unbiased, verifiable randomness generation and non-interactive and deterministic threshold signatures. Computation oracles are

equipped with zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK)[6][65]. It enables a decentralized computation marketplace for commercial computation applications monopolized by tech giants (e.g., video/audio transcoding or machine learning model training). Upon availability of a query, randomly selected nodes to reach consensus by the t-out-of-n threshold signature algorithm (in addition to verifiable random function (VRF)). The agreed result is reported back to the DOS on-chain system, as long as more than t members are honest. The nodes' identity and quality of service (QoS) (e.g., responsiveness/correctness) performance are recorded on-chain for monitoring and data analysis purposes. Since the honest nodes earn an even split of the payout, the DOS network is also protected against freeloading. Each node to join the network needs to deposit DOS token mitigating Sybil attacks and enhancing security. The DOS token is natively supported for payment, and an extra payment for stablecoins (e.g., USD coin) exists.

Table 3 shows a summary of the studies based on the multi-signature strategy.

3.3 Schelling point Systems

Buterin [14] presents a mechanism that relies on the concept Schellingcoin for the creation of a decentralized data feed. The mechanism works as follows; users submit a hash of data (e.g., price feeds) along with their Ethereum address, and in the block, after users provide the value (plus assigning a deposit to it), the submitted values are sorted. Each user whose submitted value is correct and is between the 25th and 75th percentile is rewarded. In other words, deposits are reassigned so that reported values far from the median are penalized while values closer to the median are rewarded. The mechanism is not immune to Sybil attacks, but proof of work or proof of stake mechanisms can be used for this purpose. This approach also has a limitation because if an entity controls more than 50% of the votes, the median can be set to any wanted value. The other issue is micro-cheating, defined as when slight changes are frequently applied to the value. Participants can slightly tweak their answer in one direction, pushing the median toward their desired point. This can be addressed in a centralized way, e.g., defining a value unambiguously or a coarse-grained approach for the value to mitigate the slight changes.

Usage of the median point can be seen in [46] that is known as Maker Protocol or the Multi-Collateral Dai (MCD) system built on the Ethereum blockchain for the creation of currency. One element of the system is oracles (assumed trusted and approved) being responsible for the real-time market price of the collateral assets. These oracles are decentralized and have independent individual nodes called Oracle Feeds. Oracles have a security module and medianizer that is a smart contract for collecting price data from Feeds and providing a reference price by a median. Each oracle feed has a Setzer tool for pulling median exchange prices and pushing them to a secure network (i.e., database protocol) where relayers aggregate the price data. Medianizer receives a transaction from the relayers, determines the median of the reported values, and publishes it as a queued reference price that the Oracle Security Module delays.

The Oracul system [12] is also based on the SchellingCoin concept that Vitalik has introduced. A δ represents a spread tolerance range for the reported value in this system. If the δ is zero, every reported value except the median one is penalized for its distance from the median. When δ is bigger than zero, all the reported values are valid and will receive a share of penalties produced based on the reported values outside the range.

Table 4 presents a summary of the studies based on the Schelling point concept, and it is understood they are not entirely resilient to Sybil attacks.

[6]It is a cryptographic proof that assists a party with proving specific information it possesses without revealing the information.

Table 4. The summary of Schelling point based studies for voting-based oracles

Literature	Key Research Outcomes	Query Type	Sybil Attacks	Verifier's dilemma
Buterin [14]	An Ethereum-based blockchain for the currency creation assisted by trusted oracles for fetching data feeds.	Scalar	✗	✗
MarkerDAO [46]	Presenting a mechanism that relies on the concept of Schellingcoin for the creation of a decentralized data feed.	Scalar	✗	✗
Oracul [12]	Utilizing a spread tolerance range for rewarding or penalizing the price reporters.	Scalar	Partially	✗

3.4 Token-based Systems

Augur [59] is a decentralized oracle and platform for prediction markets and is believed to be an early prediction market implementation. It was initially designed as an extension to the Bitcoin Core source code employing Bitcoin Script-based logic, but later on, it switched to the Ethereum smart contract architecture. Users in Augur select the outcomes of events, and they hold reputation tokens. Token holders post Progressively-larger reputation bonds, divided into multiple versions for disputing the proposed market outcome. The token (REP) is required for the market creators and reporters who stake their REP on a market's outcome, similar to Truthcoin. Suppose a reporter's outcome does not match with the other reporters'. In that case, Augur re-distributes its stake in the outcome to other reporters whose outcomes are matched. In the Augur system, the market creator posts two bonds; the validity for incentivizing creators for creating well-defined events and the creator (paid in REP) for choosing a reliable reporter. There is a period for the designated reporter to report the outcome. If it fails, the bond will be distributed to the first reporter. Upon receiving the tentative report, there is a dispute window such that REP holders may participate in creating a dispute that consists of staking REP on an outcome other than the tentative one. The dispute is resolved successfully when the dispute's stake in some outcomes meets the dispute bond size for a dispute round. The Augur system functions as a single oracle that leverages an iterative commit-and-reveal process where token holders are free to participate. The collected platform fees are shared among all the voters requiring relatively active participation (e.g., voting and appealing). The system may let voters be settled as long as their ability to collude is minimized. Also, Augur leverages a validation-dispute protocol in which token holders report or challenge the outcome.

Tellor [76] is an Ethereum-based decentralized oracle that employs proof of work (resistant to Sybil attack) and fetches any data requests in the Tellor smart contract. In addition to enabling developers to query Tellor's on-chain database for data, Tellor holds a token named the Tributes (TRB) to incentivize miners to provide data legitimately and vote for data validation in a dispute. Users use the token to request data and to reward miners. Participant miners deposit the token in the Tellor's smart contract and are rewarded or penalized in case of providing correct or incorrect data, respectively. Tellor chooses the first five miners to provide the proof-of-work solution and the five off-chain data points to be rewarded with newly minted tokens and the accumulated tips for the specific data requests. When other users request the same data, they need to pay a "tip" to incentivize the miners more. The Tellor's smart contract picks the most funded query in an interval-based manner. When values become available, they are sorted, and the first five values are selected, of which the median value is saved onto the chain, and miners are rewarded.

Decentralized Information Asset (DIA) [55] as an Ethereum-based ecosystem for an open financial ecosystem acts as a bridge between smart contracts on-chain and off-chain data sources in a verifiable and reliable way. DIA employs crypto-economics to incentivize and validate data from

data providers and use the community wisdom for validation and data sourcing. DIA has three main building blocks (1) data collection mechanisms known as scrapers supported by a centralized backend, (2) a flexible database layer for handling all different kinds of data streams, and (3) distribution through REST API and oracles operating on multiple blockchains. Stakeholders for the information unavailable on the DIA blockchain submit a funded request that becomes public. The requester pays the bounty in DIA token for the data provision upon validating the information. Scrapers are created by data providers that may be connected to the on-chain smart contracts or APIs to retrieve the requested data. Analysts are responsible for verifying the submitted code by staking mechanisms. In case of incorrect submitted data, the code is challenged via staking DIA tokens, and based on a voting strategy, the DIA community evaluates the right solution and who should be rewarded. The outcome is kept in a database which is an immutable and open-source database and is also published on the DIA platform. Historical data can be accessed free of charge on this platform, while DIA tokens pay specific APIs and live prices.

Sztorc presents Truthcoin [73], renamed to Hivemind, a blockchain-based platform for prediction markets. Hivemind is a peer-to-peer decentralized oracle protocol that acts as a side chain to Bitcoin and inherits all the Bitcoin assumptions. It provides the ability of multi-factor decision governance in the prediction market. It aims at the information aggregation problem with the help of the monetary aspect, transparency, and censorship resistance of the blockchain. The platform uses dual tokens in which the Bitcoin (i.e., CashCoins) serves as the interface for the users, and the VoteCoins as the reputation layer indicating a user reputation on the platform. Hivemind can host many oracles named branches (for a topic), each of which holds a set of VoteCoins; the higher VoteCoins percentage in the branch, the higher degree of voting influence. Weighted votes assisted by Votecoins provide the outcomes, and the malicious behavior is controlled by collapsing the coin market value, miner vetoes, and overrides. Owners' VoteCoins are prone to be lost due to refusing to participate in voting or voting differently from the majority. Decisions are resolved by the voters, which can be Boolean or scalar, leveraging the VoteCoin for a decision on the outcome, similar to Augur's process using Reputation tokens (REP). Hivemind applies singular value decomposition (SVD) for outcomes calculation. Market decisions are divided into branches having their parameters and VoteCoins, deciding for the branch.

Polkadot [60] is software that incentivizes a global network of computers to operate a blockchain on top of user-defined blockchains. Polkadot maintains two types of them; (1) the main network that is called a relay chain on which transactions are permanent. The second type is (2) user-created networks, referred to as parachains employing a variation of proof of stake consensus known as nominated proof of stake (NPoS). One advantage of parachains is their customization for any number of use cases and feeding data into the main blockchain. It provides the parachain transactions with benefits in the same level of security as the main chain and keeps the transactions secure. This only leverages computing resources that are necessary for running the main chain. In addition to the main chain and parachains, the bridge blockchain exists that assists the Polkadot network with interacting with other blockchains. There are different roles for those who stake DOT (i.e., the native token) as they can be validators for voting and validation of data. Also, they can be nominators for selecting trustworthy validators. Collectors are responsible for storing the history of each parachain and aggregating parachain transaction data into blocks. Eventually, they are for monitoring the network to report to validators. DOT token holders in the network can use their coins to prove/reject changes proposed by others to the network.

Kylin Network [39] employs the Polkadot platform to create a cross-chain decentralized oracle network at a low cost. It holds the native token KYL and provides the application, blockchain, or parachain of any form of access to external data. It also provides a wide variety of data feeds, e.g., weather or the stock market, connecting to APIs. This token assists the on-chain governance and

Table 5. The summary of token-based studies for voting-based oracles

Literature	Key Research Outcomes	Query Type	Sybil Attacks	Verifier's dilemma
Augur [59]	A decentralized oracle and platform for prediction markets that uses token (REP) for market creators and reporters.	Non-binary or scalar	✓	Partially
Tellor [76]	A decentralized oracle that employs proof-of-work and its native token to return the outcome data based on the median value of first five data providers.	Any type of data	✓	✓
Decentralised Information Asset (DIA) [55]	A decentralized oracle that employs crypto-economics to incentivize and validate data, and has three main building blocks such as (1) data collection mechanisms, (2) a flexible database layer, and (3) distribution.	Any type of data	✓	✓
Sztorc [73]	A peer-to-peer decentralized oracle protocol acting as a side chain to Bitcoin, employing dual tokens CashCoins to serve as the interface for the users, and the VoteCoins as the reputation layer.	Binary or scalar	✓	Partially
Polkadot [60]	A platform for creation of user-defined blockchains that maintains two blockchains; relay chain and parachain that employs a form of consensus and the DOT token for voting.	Non-binary	✓	✓
Kylin Network [39]	A low-cost cross-chain decentralized oracle based on Polkadot platform that holds the token KYL for operating as an oracle node.	Categorical or scalar	✓	✓
Mobius [50]	A Stellar blockchain-based network assisting developers with creation of their decentralized applications and oracle systems via series of APIs. It holds MOBI token and employs proof of stake for incentivizing/penalizing users.	Any type of data	✓	✓
Zap Protocol [72]	A decentralized oracle and a permissionless protocol, which permits oracles to be built on the protocol as a form of investment via ZAP/DOT tokens.	(Non)Binary	✓	Partially

keeps the platform decentralized as it develops. KYL token is also a requirement (through staking) for operating as an oracle node or opening a dispute. Additionally, KYL is used for payment to access private data APIs. In the Kylin network, there are four major components; (1) analytics for improving the efficiency of applications, (2) a query engine for the public and API access, (3) data oracle as a decentralized data feeding protocol powered by Polkadot, and (4) marketplace as an open platform for pricing and trading data. Kylin network employs a network of data providers, oracle nodes, and arbitration nodes to keep the data sourcing decentralized.

Mobius [50] is a Stellar blockchain-based network [75]. It enables developers to create their decentralized applications and oracle systems through a series of APIs connecting applications to the blockchain. Mobius has a MOBI token to facilitate transactions and supplement Mobius protocol (i.e., cross-blockchain standards). This is to assist payments, logins, and oracle management through simple APIs and developer frameworks. Mobius employs the proof of stake model, requiring participants to stake a certain amount of tokens to be granted the privilege of contributing to the network's maintenance and growth. There is a Universal Proof of Stake Oracle Protocol for

connecting the real-world data to the blockchain, and by staking Mobius tokens, high authenticity data transmissions to the blockchain and high-throughput data transfer to secure smart contracts are achieved. This proof of stake protocol incentivizes/penalizes oracles for providing correct/incorrect data. In contrast, the Mobius Universal Proof of Stake oversees the vesting and staking of MOBI tokens. In addition, a quality threshold for the selection of oracles is used so that separate markets are created, and quality scores are aggregated based on past performance, staking MOBI, and Proof of Verification for auditing oracles.

Zap Protocol [72] is a decentralized oracle and a permissionless protocol. This oracle is based on Ethereum, focusing on three main components as data, tokens, and bonding curves. These components are mechanisms powered by on-chain smart contracts for controlling the direction of a decentralized autonomous organization. By this protocol, in addition to buying/selling data, liquid tokens and pricing curves can provide the ability to (un-)bond money to the curves. This protocol, unlike [76], permits oracles to be built on Zap as a form of investment. It means the oracle model and the underlying data can be monetized, and the better the oracles, the more money bonded with them. Thus, users can detect which oracles are more reliable and profitable. Although the Zap protocol employs tokens (i.e., ERC20 [26]) on its network, they are defined in a way that can be liquid, and the supply can dynamically be adjusted to demand, and these tokens can be traded on decentralized exchanges. Bonding ZAP to the oracle by subscribers (i.e., smart contracts that require data) results in receiving DOT tokens for querying the oracle. The ZapMarket smart contract helps the exchange of InterPlanetary File System (IPFS) [40] public keys for the creation of a private IPFS publish-subscribe channel for pushing data to the subscriber by the oracle. A data provider can be an oracle by being registered with the ZapMarket smart contract that needs to define the DOT/ZAP supply curve determining the distribution of DOT per ZAP.

Table 5 provides a summary of the studies for token-based systems, and it illustrates that employing a token for the network/protocol governance is of great importance.

3.5 Conventional Systems

While stake-based systems are beneficial to be used for oracles, there are studies whose platforms rely on either a single data source, e.g., [24] or multiple data sources, e.g., [95], without any mechanism to verify the data integrity and correctness, e.g., [7]. Also, data can be directly submitted to the blockchain without a need for third-party data providers or obtaining processed data from a distributed ledger [27]. One can assume that they are trusted entities whose *vote* for data verification is reliable. However, this reliability comes with flaws, such as a single point of failure or tampering.

Eskandari et al., [24] present Velocity as an Ethereum-based decentralized market to trade a custom type of derivative option. It employs a tool called PriceGeth to fetch the price information in real-time. A derivative is a contract between two or more parties, and its values are determined based on the agreed underlying financial assets. The price feed consists of PriceFetcher saving exchange prices into a database at specific intervals, BlockListener for monitoring the Ethereum blockchain for new blocks, and a PriceGeth server for sending data to the PriceGeth smart contract updating the latest price. The PriceGeth interacts with the PriceFetcher module for exchange prices. While Velocity employed Provable (in addition to TLSNotary for authenticity proof- see Section 4), issues such as delay and the insufficient amount of gas led to presenting PriceGeth for nearly real-time price acquisition. However, the structure of PriceGeth by employing the PriceFetcher module is a point of failure as there is no mechanism to avoid data manipulation while PriceFetcher can be assumed to be a trusted entity.

Zhang et al., [95] extend the industrial IoT (IIoT) framework for providing a solution for trustless data sharing by employing an encrypted ledger for reducing the risk of data tampering. The framework consists of a blockchain controlled by the consensus rule of byzantine fault tolerance

and layers such as IoT, fog, micro-service, and decentralized applications. The cloud-based micro-service layer (i.e., APIs) provides data feed to smart contracts. It acts as Platform-as-a-Service (PaaS) layer in IIoT to provide computational power and hold APIs for smart contracts. The framework uses a fog layer to alleviate the slow response due to the massive data-producing rate at the IoT layer. Upon receiving data requests and approval by the customer, the framework allows IoT data to be fetched by net gates (i.e., in the form of hardware oracles), which are encrypted and anonymized by the user's private key.

Arts et al., [7] present Aternity as an open-source blockchain-based framework. It uses proof of work cuckoo cycles (a graph-theoretic and memory-intensive problem for finding cycles in the graph). It leverages state channels as (on)off-chain encrypted peer-to-peer communication for a smart contract execution which may not be recorded on-chain. This allows the framework to use an off-chain contract with an on-chain oracle for providing on-chain data. The framework has Sophia- the smart contract language- and contracts are compiled into bytecode executed on highly efficient virtual machines FATE. The framework has a native token that is for coordination between participants. Their amount of tokens represents their vote influence on the system, and it is required for any operations on its blockchain. Generalized accounts provide flexibility for transaction authentication managed by a smart contract. Upon transaction execution, the authentication function in the smart contract and the account evaluate the authentication data. If it fails, the transaction is discarded; otherwise, it incurs charges. Moreover, there are registered oracle transactions for the announcement of oracles to the chain specifying queries and response format associated with a fee. Publicly available oracles monitor the blockchain for queries, and since the response on the chain is public, it causes privacy issues.

Hyperledger Fabric is an open-source blockchain but permissioned framework hosted by the Linux Foundation [6, 35] for the enterprise context in which participants' identities are authenticated. The framework has a configurable and modular architecture. It does not need to have a native cryptocurrency that allows smart contracts (i.e., chaincodes) to run within a container and be coded with general-purpose programming languages. It supports customized consensus protocols and benefits industries, e.g., track-and-trace supply chains, healthcare, banking, or insurance, where data cannot be exposed to unknown entities, increasing privacy. The framework employs execute-order-validate architecture for transactions. The "execute" checks the correctness after execution, the "order" applies a customized consensus protocol, and "validate" determines transactions against an application-specific endorsement policy. It reveals how many peer nodes and which peer nodes are required to confirm the correctness of smart contract execution. In Hyperledger Fabric, there are channels to communicate between members. These channels add an extra layer of access control, improving confidentiality. Participants in these channels establish a sub-network where each member can access a particular set of transactions. Chaincodes on the Hyperledger Fabric run on the peers and create transactions. Invocation of chaincodes can lead to updating or querying the ledger. Based on the proper permission, another chaincode can be invoked to access the state in the same or different communication channel. Chaincodes have endorsement policies for selecting peers to execute the chaincode by checking whether enough endorsements are present and versioning checks are done. They are derived from eligible entities and verify the result to ensure that the transaction is valid. Hence, they can be considered platform-supported (distributed) oracles for the Hyperledger Fabric.

Compound protocol [61] as an Ethereum-based system is recognized as an interest market. It allows borrowers to obtain loans when lenders put their crypto assets into the protocol earning variable interest rates. The protocol has its native token called cToken, a form of ERC20 token, and requires approval to be minted initially. The protocol employs the Open Price Feed as a decentralized price oracle built on the Ethereum blockchain. This price feed consists of entities such as Reporters,

Table 6. The summary of conventional studies for voting-based oracles

Literature	Key Research Outcomes	Query Type	Sybil Attacks	Verifier's dilemma
IOTA oracle [27]	A First Party Oracle where data is sent to the IOTA Tangle directly without a need for third-party data providers.	Scalar	✗	✗
Eskandari et al., [24]	An Ethereum-based decentralized market for trading a custom type of derivative option (e.g., price feeds) from a single data source.	Scalar	✗	✗
Zhang et al., [95]	Extending the industrial IoT framework to provide trustless data sharing by an encrypted ledger for reducing the risk of data tampering.	Non-binary	✗	✗
Arts et al., [7]	An open-source blockchain-based framework that employs proof of work cuckoo cycles and uses state channels as (on)off-chain encrypted peer-to-peer communication.	Non-binary	✓	✓
Hyperledger Fabric [6, 35]	An open-source blockchain but tokenless and permissioned framework for the enterprise context in which participants' identities are authenticated.	(Non)Binary, scalar, or categorical	✓	✓
Compound protocol [61]	An Ethereum-based system and known for interest markets. It employs cToken as the native token and Open Price Feed as a decentralized price oracle.	Scalar	✓	✓

Posters, and a View, a set of Reporters used to obtain the finalized prices. Posting/storing price data can be done by any user who has access to a reliable source.Posters post the data on the chain, and this responsibility is shared among many Posters. The Compound protocol leverages a View contract employing a single Reporter. It verifies all the reported prices within an acceptable time, and in the presence of enough reporters approved through governance, a median price can be used. These price feeds via the View contract can be configured for developers, while the Reporters in the Compound View contract may require approval by the Compound governance. The Open Price Feed allows price data reported by reporters to be signed via a known public key. Posters can be any Ethereum address that can put a value on the chain. The interest rates paid and received by borrowers and lenders are determined by the supply and demand.

Table 6 provides a summary of the studies in this category for voting-based strategies.

4 REPUTATION-BASED ORACLES

Oracles may employ different data sources/providers. Hence, this necessitates using evaluation mechanisms for selection, monitoring the truthfulness of the provided data, and keeping the retrieved information intact. Information retrieval by oracles may firstly necessitate mechanisms to ensure the received data is untampered. Secondly, *identify* which oracles have more potential to be trusted for the outcome. The former can be achieved by *authenticity proof* mechanisms attached to the data, while the latter can be managed by a reputation component that is responsible for oracles' evaluation. The reputation-based oracles may be assisted by authenticity proof mechanisms to verify the data retrieved from external resources and ensure that the data is untampered and genuine. In addition, blockchain oracles should handle *confidentiality* of the data (e.g., passwords) with extra care, avoiding the exposure of sensitive data to non-authorized parties or entities.

Providing proof can be in the form of; non-repudiation of origin, non-repudiation of receipt, and non-repudiation of conversation [66]. Non-repudiation of origin supplies proof that a message

comes from a specified originator blocking attempts to deny having sent the message. In contrast, non-repudiation of receipt proves that a message is received by a specified recipient falsifying the recipient's false claim. Finally, non-repudiation of conversation generates proof of the occurrence of a conversation between parties.

Reputation-based oracles may employ software or hardware to supply authenticity proof for the data. In the former, they mainly rely on TLS protocol, while in the former, they employ a built-in trusted execution environment embedded in hardware devices such as Intel SGX enclaves.

Oracles may employ a secure HTTP connection (i.e., HTTPS) powered by the TLS protocol to retrieve data from sources. However, the TLS protocol cannot fully guarantee that the content of the HTTP session is not tampered with. Three related studies for improving the TLS protocol are explained in the following.

Ritzdorf et al., [66] propose TLS-N. It is a TLS extension for providing a decentralized, seamless, and standardized internet-wide non-repudiation mechanism to securely share data feeds. TLS-N produces proofs about the content of a TLS session providing an efficient way of verification by third parties and blockchain-based smart contracts. Based on the generated proofs, TLS-N allows parts of a TLS session, e.g., passwords, to be hidden for increasing privacy while the remaining content becomes verifiable. There are requester, generator, and verifier. The process starts with establishing a TLS connection and negotiating the TLS-N parameters in the handshake. During the TLS session, a small TLS-N state holds information about the hash value of the previous records. In addition, an ordering vector exists (i.e., a bit vector encoding the interleaving requester and generator records), including a timestamp (to mitigate time-shifting attacks). They are kept updated by the generator that signs its TLS-N state by its private key. Upon asking for the evidence by the requester, the evidence window that consists of records to be included closes. The requester maintains full control of the retained records in the proof; by checking the proof, the verifier can access the content of the TLS session. To reduce the size of proof, Merkle Tree is used, and if the session contains sensitive information to be hidden, independent random values called salts are needed that are derived from the TLS traffic secret using a record-based nonce. Hence, the verifier for the proof verification re-generates the evidence such that in the absence of sensitive information, it constructs the Merkle (and salt) Tree. Otherwise, it produces the partial Merkle Tree for the provided plain text, commitments, and hashes.

TLSNotary [78] is a service that introduces third-party auditors for validating TLS session data exchanged between the client and server. It uses TLS protocol to help a client (auditee) provide evidence of specific web traffic between the client and the server to a third party (i.e., auditor). TLSNotary facilitates verifying obtained data from external sources in an oracle against tampering by splitting the TLS master via the RSA encryption. Auditors hold a portion of the TLS secret key to generate the Message Authentication Code (MAC) key for setting up the HTTP session key. It is for intercepting the auditee's attempt to fabricate traffic from the server. Without the MAC key, the client cannot decrypt and authenticate the traffic received from the server. Upon commitment to the encrypted content of the server's response, the auditor provides the portion to the auditee to complete the authentication steps of TLS. Although promising, there are issues such as it is not supported by most web servers (websites must support TLS 1.0/1.1), and there has to be a trusted auditor for the process.

Android proof [62] uses Google technologies such as SafetyNet Software Attestation and Android Hardware Attestation (implemented in a Trusted Execution Environment (TEE)[7]) to notarize web pages or certify data served by HTTPS APIs. The former evaluates the application runs on a safe and not rooted physical device, i.e., unmodified root certificate authorities (CA). Besides, it checks

[7]It is defined as a computational environment which is heavily isolated from the main operating system running on a device.

the application code hash and makes sure that the source is untampered. The latter verifies that the device is running on the latest OS version for preventing any potential exploits. Hence, both technologies assure that the device is a secure environment for making an untampered HTTPS connection with a remote data source. When a request for Android proof becomes available, the given URL by the user is forwarded to the Android device, an HTTPS connection is established, and the entire HTTP response is retrieved. The SHA256 hash of the response is signed with the hardware attested key pair available on the device. The service application calls SafetyNet's API, and the nonce parameter of the API becomes the SHA256 hash of the HTTP response key, the signature, and the request identifier formatted as a JSON Web Signature (JWS). By full validation of the proof, the data in the HTTP response is parsed and distributed to the user with the SafetyNet Attestation Response and Hardware Attestation Object. Although promising, there is a quota (10k requests per day) for Google SafetyNet API that limits the system's scalability.

It should be noted that there are studies whose reputation-based oracle design does not practice authenticity mechanisms (e.g., [3, 82]) as the data sources are assumed to be trusted and the data integrity is intact during data retrieval. Figure 5 depicts the overall structure of reputation-based oracles assisted by the authenticity proof mechanisms, and oracles may return data to the blockchain without authenticity proof mechanisms.

The following sections present how these proofs are employed and developed for data authenticity in the reputation-based oracles.

4.1 Software-based Proof

Guarnizo et al., [31] present PDFS as a data feed system that allows data to be authenticated over the blockchain without breaking TLS trust chains or modifying TLS stacks. Content providers can specify data formats freely; thus, data can be easily parsable and tailored for smart contracts. Also, PDFS provides content providers with a payment framework to be incentivized, but it does not allow content providers to misbehave by equivocating or censoring queries. They are defined as retrospectively revising or deleting the published content, influencing a contract execution by censoring some required content. In PDFS, content providers create authoritative contracts enabling other contracts to verify the authenticity of the content and providing functionalities to mitigate misbehavior. Content providers then create a signed manifest containing information about the blockchain address, the authoritative interface, and metadata of the content. The manifest's signature is computed with the help of the private key corresponding to the public key from the content provider's TLS certificate allowing contract parties to verify the manifest's authenticity directly. Content providers create a tamper-evident data structure (TDS) (i.e., Merkle Tree) storing served data entries, including the manifest. Per each update, the data structure is re-computed, and its consistency proof is sent to the authoritative contract for validation. If contract parties

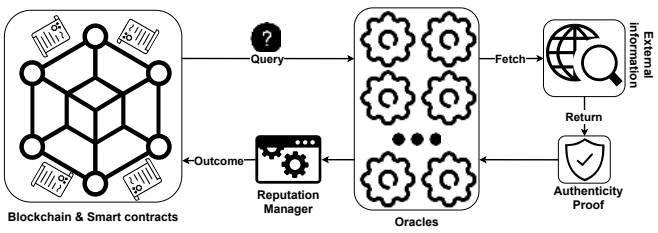

Fig. 5. Reputation-based oracles overall structure. Malicious oracles can be blocked by the reputation manager, and authenticity proof may be attached to the data.

wish to deploy relaying contracts (i.e., smart contracts require data feeds from external websites), they should find and agree on a content provider necessitating verification of its manifest and authoritative contract. The relying contract is called by a contract party, and interacts with the authoritative contract's membership verification method for the produced data by the content provider. PDFS is resilient to attacks such as TLS Public Key Infrastructure (PKI) compromise and malicious content providers. In the former, data verification is done by the correct deployed authoritative contract, while in the latter, the TDS consistency is enforced by authoritative contracts noting that the content provider's response is visible.

Zhang et al., [94] present Decentralized Oracle (DECO) assisting users with proving data accessed via TLS that comes from a particular website and provides statements (zero-knowledge proofs (ZKPs)) about the data. Authors argue that mandating installing TLS extensions at servers suffers from two issues: legacy compatibility becomes broken and reduces wider adaptability. The next issue is the limitation of data exportability as the web servers determine what data to export, resulting in censoring export attempts. Hence, the oracle does not require trusted hardware or server-side modification and provides a three-party handshake protocol to mitigate forging arbitrary TLS session data. This is due to the TLS nature generating symmetric encryption and authentication keys shared between users and web servers. Moreover, DECO reduces context-integrity attacks (i.e., specific data exists in the server's response and appears in the expected context) via a two-stage parsing scheme, as attacks can be thwarted if the session content is structured and parsable. DECO consists of three phases; a three-party handshake phase for establishing session keys in a particular format for unforgeability. It also has a query execution phase such that the server is queried for data based on a query built from the template with the private parameters. Finally, a proof generation phase in which the query is proved that it is well-formed and the desired condition is satisfied by the response. In the first phase, the session key used in the TLS session with a server is distributed between the prover and verifier in a secret-share form. In the next phase, since the session key is secret-shared with the prover and verifier, both must interact and execute a two-party computation protocol to construct TLS records encrypting the query. Finally, upon receiving a response from the server, the prover commits to the session by providing the ciphertext to the verifier to obtain the MAC to verify the response integrity and prove its statements.

He et al., [32] present SDFS as a scalable data feed service employing a reputation-evaluation strategy for malicious node detection that leverages a blockchain to preserve the data processing. The proposed data feed service consists of smart contracts to invoke the interface for requesting the data, a server that has multiple nodes for data fetching, and an auditor blockchain assisted by TLSNotary [78] to verify that obtained data is untampered. The service maintains a reputation mechanism for data feed nodes updated per data feed that keeps a verification pass. Based on the pass, the reputation value of the node is increased or decreased in each round.

Bridge Oracle (BRG) [74] is an *unsuccessful* public oracle technology on the Tron network [79]. The Bridge oracle can attach authenticity proofs and deals with various APIs and parsing helpers while employing TRON (i.e., the Tron network token) and project-purpose tokens on the network for payments. The bridge oracle consists of three main smart contracts; (1) the Bridge API contracts (e.g., public, decentralized, and enterprise) for connecting the client smart contract to the Bridge oracle, (2) The bridge oracle address resolver that is in charge of redirecting requests to the correct services (e.g., public oracle system or decentralized oracle system), and (3) the bridge oracle connector for processing requests and outputting specific data to be accessed by oracle data carriers. A queuing system is employed for the off-chain bridge oracle architecture to balance the load of data carriers. A random-access memory (RAM)-based database is used for temporarily logging queries to avoid losing queries during load conditions while the query's feedback is stored in the permanent database.

Table 7. The summary of software-based studies for reputation-based oracles

Literature	Key Research Outcomes	Limitation	Authenticity	Confidentiality
PDFS [31]	A data feed system for data authentication without breaking TLS trust chains or modifying TLS stacks.	None	✓	✗
Ritzdorf et al., [66]	Providing a TLS extension that is compatible with TLS 1.3 and produces proofs for the content of a TLS session.	Adding minor overhead to the TLS	✓	✓
TLSNotary [78]	Verification of the obtained data from external resources against tampering through splitting the TLS master.	Only works with TLS 1.0/1.1.	✓	✗
Zhang et al., [94]	Requires no-server side cooperation, and having support for TLS 1.2 and 1.3.	None	✓	Partially
He et al., [32]	Reputation-based scalable data feed service assisted by TLSNotary for data verification against tampering.	Restricted to TLSNotary limitations	✓	✗
Bridge Oracle [74]	A dedicated oracle technology on the Tron network providing the ability to attach authenticity proofs to data.	Unreliable [10]	✓	✗
JustLink [37]	A decentralized oracle where a single result obtained from trusted sources is calculated by an aggregator contract.	Not resilient to data tampering	✗	✗
Pyth Network [64]	A Solana-based cross-chain market employing different entities such as delegators for providing a confidence level in the data from a particular Pyth data provider.	Not resilient to data tampering	Partially	✗
API3 [13]	A decentralized autonomous organization aiming for the creation and monetizing of a decentralized API network for connecting blockchains to the existing data provider APIs governed by API3 token.	Not resilient to data tampering	Partially	✗
PolkaOracle [43]	A Polkadot-based oracle that employs POT for governance and Substrate 2.0 Off-chain Worker for secure integration of data to the blockchain applications.	Not resilient to data tampering	✓	Partially

JustLink [37] is a decentralized oracle network [10] that is an under-development project deployed on the Tron network. Data requests interact with the on-chain JustLink open-source and verifiable interface that includes smart contracts. There is an on-chain aggregator contract for which users choose nodes and services. The final result is computed and finalized through *trusted* sources for requesting contracts. Oracles are selected based on the requirements, and the aggregator that can be different for each demand outputs the result by, e.g., calculating the weighted average after removing abnormal data. The nodes obtain off-chain data separately, and a single result is finally calculated in the aggregator contract. Each assignment can be divided into subtasks (e.g., HTTP requests or JSON parsing) customizable within external adaptors as services with a minimal REST API. Subtasks pass their results to the next subtasks they run end-to-end to obtain the final result. In JustLink, the freeloading issue is handled by commit/reveal, and JustLink employs the token named Just (JST) for paying JustLink Node operators that retrieve data from off-chain data feeds.

In addition, JustLink plans to devise a reputation system for controlling the quality of oracles and leverages a certification service to mitigate Sybil attacks. Statics from the validation system is collected in this system, and the service performs after spot-checking of on-chain answers.

Pyth network [64] is a solution for providing a cross-chain market of verifiable data in a decentralized way that is powered by Solana-based blockchain [71]. Solana can achieve heavy and fast data processing as it is recognized as the only chain providing the computing bandwidth. Pyth network employs formal likelihood methods to output a suitable price considering all the received information through high-quality data providers. Also, the Pyth network can include information such as historical quality or potential stake at risk. The Pyth network consists of entities responsible for a particular task; data providers can be data owners or sources for new datasets and obtaining data on-chain, respectively. Delagotors, as the other entity, provide a confidence level in the data from a particular Pyth data provider, which can be done through evaluation and consideration of their historical performance and accuracy and can be rewarded or penalized. Also, there are curators for determining which data should be sourced through paying tokens into a bonding curve to signal interest. Increasing the interest in symbols will lead to receiving the most significant share of rewards, incentivizing the data providers to provide prices for them.

API3 [13] is recognized as a decentralized autonomous organization (DAO) that aims to create and monetize a decentralized API (dAPI) network to act as a bridge for connecting blockchains to the existing data provider APIs. Each dAPI has oracles managed by decentralized API providers and holds API3 tokens enabling holders through staking the tokens to practice governing rights over the API3 DAO with the rewarding opportunity. Hence, unlike the general oracles, API3 aims to be governed by DAO, meaning the API3 ecosystem players are in charge of securing the network. As first-party oracles build a data feed, over-redundant decentralization would not be necessary, and it would be more immune to attacks resulting in better transparency. Also, a user could easily identify data owners as on-chain identities of API providers are published via off-chain channels. In addition, funds are transferred to them because of performing the actual work instead of fee-paying third-party oracles. Data is signed by the API provider and becomes accessible via a common API endpoint that third-party oracles can query to fetch the data, and their public keys verify the authenticity of the data.

PolkaOracle [43] is a Polkadot-based oracle network that aims to be a community-driven oracle system. PolkaOracle employs POT as the native token in the network to pay the data providers, to govern the network via voting and deposit purposes. In comparison to Chainlink [11], PolkaOracle provides flexibility and reliability through leveraging Substrate 2.0 Off-chain Workers for the infrastructure that can act as parachain or parathread for connecting to the Polkadot blockchain. It uses on-chain operations such as on-chain computation, encryption and decryption, data verification, and random challenge for credible and reliable real-time data feeding. PolkaOracle has a layered architecture; (1) a cross-chain application layer for providing data interfaces based on the cross-chain technology for applications and public tools (e.g., data display panels), (2) the on-chain infrastructure for security and transparency of the network. Also, (3) there are Off-chain Workers to integrate data to the blockchain applications securely. It utilizes verifiable random functions (VRF) to randomly select network nodes for off-chain calculation and verification to ensure the data is not wrong or tampered with. Finally, a (4) data source layer is responsible for obtaining third-party off-chain data via APIs and employs techniques (e.g., filtering or screening) to ensure the accuracy and authenticity of the data.

Table 7 presents the summary of studies that employed software-based strategy to provide proof of the reputation-based oracles. Most studies may fail to satisfy the data integrity as data could be tampered with during transmission between source and oracles.

4.2 Hardware-based Proof

Schaad et al., [69] present a Hyperledger-based blockchain design with a local secure element (Wibu CmDongle for cryptographic software protection [85]) as an external hardware-based oracle. They applied a use-case study where a 3D printer is rented and loaded with a specified amount of printing credit. Per each print, the local counter embedded in the dongle is decreased, and in parallel, the counter unit is maintained on the Hyperledger blockchain. Upon meeting the threshold, a chaincode on the blockchain is triggered to set the counter again based on the payment processing and update the local counter on the device for further printing.

Ledger proof [42] leverages hardware wallets owned by the Ledger company. These hardware devices employ Blockchain Open Ledger Operating System (BOLOS) that provides Software Development Kit (SDK) and enables developers to code applications (i.e., cryptocurrency wallets) for installation on the hardware. It provides an isolated environment as each application has its memory region operating in the user mode and interacting with the operating system in the superuser mode. There are also similar hardware wallets such as Trezor [70], or Coldcard [18] for securing digital assets in an offline manner, but they may not support altcoins, e.g., Coldcard wallet.

Town Crier [93] is an authenticated data feed system that acts as a bridge between smart contracts on the Ethereum blockchain and commonly trusted websites' data known as datagrams. Town Crier ensures *confidentiality* that is referred to as requesting private data with encrypted parameters, e.g., accessing online accounts. Town Crier employs a combination of a front-end smart contract and Intel Software Guard Extension (SGX) technology, a set of instructions granting hardware protections on the user-level code. Town Crier has three components; the town crier contract, the enclave, and the relay. The enclave and relay reside on the Town Crier server while the contract is on the blockchain. The relay functionality is defined as handling the network traffic on behalf of the enclave. The front-end smart contracts respond to requests from contracts on the blockchain with the attestation holding characteristics of datagram parameters, HTTPS website, and the time frame. A relaying contract can verify the datagram considering trust in the SGX security, Town Crier code, and validity of the source data in the time frame. In addition to data authenticity, Town Crier proves gas sustainability (i.e., Ethereum service never runs out of gas), and trusted computing base code minimization by authenticating the enclave outputs on the blockchain. There are still some issues; (1) an enclave requires network capability (which can be done by splitting TLS code between the enclave and untrusted host environment). The next issue is (2) compromising a single website or an enclave addressed by the majority voting.

Woo et al., [87] propose a distributed oracle for safely importing time-variant data into the blockchain where the response time is important. The proposed oracle employs multiple oracles to support data availability and data integrity with the help of Intel SGX. Each oracle verifies the data pulling procedure such that other oracle nodes pull data from external data sources through remote attestation[8] provided by Intel SGX. The proposed oracle resolves malicious oracles with the help of a reputation system to block selfish oracles from obtaining benefits.

Edenchain [22] is a permissioned blockchain platform technology for capitalizing assets of any form into a token. Edenchain employs namespaces with Merkle Tree and isolates transactions based on the namespaces (the type of transactions) for performance improvement. Also, it uses Proof-of-Elapsed-Time (PoET) implemented in the SGX enclave as a consensus algorithm using CPU commands to randomly select a leader with the shortest wait time without requiring to consume excess energy for solving a hash problem. Edenchain has three layers; a distributed ledger layer based on the Hyperledger for storing data, a validation layer for execution and verification of a transaction, and a bridge layer for securely importing required data by on-chain smart contracts.

[8]It is method by which hardware and software configuration of devices are authenticated to remote hosts.

In the bridge layer, on-chain and off-chain nodes exist, and reliable communication between nodes is managed by E-Protocol that implements an encryption technique called Elliptic Curve Cryptography–Threshold Cryptography (ECC-TC). Threshold cryptography is a protocol with a cooperative property in which data for decryption is shared among participants. On-chain nodes interact with the smart contract, while off-chain nodes are designed to interact with the external system and connect the on-chain and off-chain modules. The Edenchain uses the E-Bridge layer to fetch data from multiple data sources, encrypts this data, and leverages the median voter theorem (MVT) to secure trust and improve security. The bridge layer's core technology is E-Bridge, which has a modular design with components such as an oracle server and an SGX enclave located off-chain for serving requests and providing the trusted execution environment. The other components are an executor situated in the chain for transaction execution and E-Oracle for forwarding data access requests from the smart contracts. The E-Oracle has a client and server such that the former provides parameters to be executed on the server, and the server runs external data requests on separate spaces named SGX enclaves for security enhancement. It also collects external data, selects appropriate values, and forwards them to the client. The E-Oracle servers can have discrete and continuous types of data, each evaluated by the majority voting, i.e., the most common value and median voter theorem (MVT). In MVT, the result is chosen by a median voter and consensus algorithms that make MVT suitable for continuous data types.

Hearn presents Corda [19] as a decentralized global database platform without a mining concept for recording and processing financial agreements. Corda aims to provide a distributed ledger consisting of mutually distrusting nodes to let a single global database keep the deal states and obligations between people and institutions free of disparate ledgers synchronization. The Corda employs SGX enclaves for attestation, supports smart contracts, and leverages cryptographic hashes for data and parties identification. It also defines state objects as a digital document recording the existence, content, and current state of an agreement between two or more parties. It employs a consensus algorithm for transaction validity (the parties involved) by independently checking that the associated code runs successfully with the required signatures and refereeing transactions are valid and unique. Contracts are executed in the Java virtual machine, which eases reusing the existing code in the contracts. The Corda network has one or more notary services and zero or more oracle services. Oracles are implemented in two ways; by using commands in which a fact is encoded in a command embedded in the transaction itself, and the oracle becomes a co-signer of the entire transaction. If a transaction includes the fact, it must be returned to the oracle for signing. The other way is to use attachments that facts are encoded as attachments and are considered separate objects to the transaction and referred to by the hash. The transaction content becomes accessible from oracles by employing the Merkle Tree that reveals only necessary parts of transactions.

Provable [63] is a platform-agnostic bridge between the blockchain and the internet and behaves as a data carrier to provide a reliable connection between Web APIs and DApps. Provable employs cryptographic proofs such as TLSNotary to enforce reliability, and the platform can be used in public and private blockchains and even in non-blockchain contexts. Provable also provides the ability for encrypted queries (does not support private or custom datagrams) via the provable public key, and the Elliptic Curve Integrated Encryption Scheme can protect the plain-text queries. Moreover, Provable presents ProofSheild assisting smart contracts to verify on-chain authenticity proofs provided by Provable. It provides tools and services for connecting oracles (data providers) with distributed applications; however, it is more suitable for centralized data centers as oracle solutions. Provable facilitates the data verification through returning data with a document named authenticity proof which can be produced by technologies such as auditable virtual machines and trusted execution environments. Provable assumes that data fetched from the sources are genuine and untampered and provides various parsing helpers to extract a data type value. Provable employs

Android proof for authenticity proof, and the verification and proof process consists of a series of verification such as SafetyNet Authenticity verification, SafetyNet Response verification, and Hardware Attestation verification. Additionally, these services ultimately rely on the reputations of their (small) providers to ensure data authenticity.

Gray et al., [48] present Bletchley; a Microsoft Azure-based enterprise consortium blockchain ecosystem that is similar to the Microsoft Coco framework. Russinovich et al., [67] present the Coco framework as an open-source system that is a high-scale and confidential blockchain designed explicitly for the confidential consortium. It employs Intel SGX and Windows Virtual Secure Mode (VSM) to create a trusted execution environment. Bletchley consists of two major components; (1) blockchain middleware providing core service functionalities in the cloud, such as operation management and data services, and (2) Cryptlets that enable secure communication between Microsoft Azure, middleware, and customers for providing information for the transaction execution. Cryptlets are assumed as a secure blockchain middleware tier that provides the oracle functionality and is defined as off-chain components written in any language. They execute within a secure and trusted container and communicate using a secure channel. They can be used in smart contracts by an adaptor (CryptoDelegate) such that the adaptor in the smart contract calls Cryptlets, which extends the secure and authentic envelope for the transaction. Two types of Cryptlets exist; utility and contract; the former is organized into services or libraries to provide common functionalities, e.g., encryption. In contrast, the contract Cryplets run within an enclave and provide all the execution logic, securely storing data in the smart contract. It can also function as autonomous agents or bots for interaction with the off-chain world (i.e., acting as multiple oracles) while maintaining the blockchain's integrity and smart contracts. Cryptlets can be accessed by a trusted attested host and can employ enclaves for process isolation and encryption. Cryptlets can be event-driven or control-driven as the former provides notifications based on events and securely passes data. The latter is followed by Cryptlet Contracts allowing Cryptlets to perform the required business logic.

Chainlink [23] proposes a general-purpose and token-based framework for building secure decentralized input and output oracles for complex smart contracts on any blockchain. A Chainlink node can have multiple external adapters for different data sources, and its token protocol is blockchain agnostic that can run on other blockchains simultaneously. Chainlink has two major components that are on-chain and off-chain components. The on-chain component has contracts such as (1) reputation; for tracking the performance metric, (2) order-matching, taking and logging a proposed service level agreement, and collecting bids from oracle providers. Also, the last contract (3) aggregating is in charge of response collection from oracle providers to calculate the final collective result of the query and is also responsible for feeding the oracle provider metrics, i.e., the reputation contract. The Chainlink on-chain component follows a workflow defined as query parameters, the number of needed oracles, and an oracle service purchaser prepares reputation and aggregates contracts for Service Level Agreement (SLA) proposal. Then, the purchaser submits the SLA to an order-matching contract on which oracle providers filter the SLAs based on their capabilities and service objectives. The Chainlink nodes decide whether to bid (i.e., stake) on the proposal or not, and only bids from nodes satisfying the SLA's requirements are accepted. The bid on a contract means commitments within a bidding window and is subject to penalties because of misbehavior. Once the biding window ends and enough qualified bids are received, the requested number of oracles is selected from the pool of bids. Then, the finalized SLA record is created, and chosen oracles are notified for performing the assignment and reporting. The aggregating contracts calculate a weighted answer, and the validity of each oracle response is reported to the reputation oracle. Chainlink has components such as core, external adaptors, and subtask schemes for the off-chain contract. The core node software is responsible for interacting with the blockchain or

Table 8. The summary of hardware-based studies for reputation-based oracles

Literature	Key Research Outcomes	Limitation	Authenticity	Confidentiality
Schaad et al., [69]	A Hyperledger-based blockchain design with a local secure element (Wibu CmDongle).	Hardware requirement	Partially	✗
Town Crier [93]	An enclave-based oracle for pull-based data provisioning helped by Intel SGX technology.	Hardware (Intel CPUs) dependent	✓	✓
Hardware wallets [18, 42, 70]	Providing functionalities to the developers for coding cyrptocurrency wallets.	Hardware requirement and prone to fail [41]	✗	Partially
Woo et al., [87]	Presenting a distributed oracle for time-variant data to be recorded onto the blockchain by using enclaves (Intel SGX).	Hardware (Intel CPUs) dependent	✓	Partially
Edenchain [22]	A permissioned blockchain platform technology for capitalizing assets of any form into a token.	Hardware (Intel CPUs) dependent	✓	Partially
Corda [19]	A decentralized global database platform for recording and processing financial agreements.	Hardware (Intel CPUs) dependent	✓	Partially
Provable [63]	A platform-agnostic bridge between the blockchain and the internet, i.e., web APIs and dapps, that employs Android proof for data authenticity.	Centralized and operating system based	✓	Partially
Bletchley [48]	A Microsoft Azure-based enterprise blockchain that uses Cryptlets to provide oracle functionality defined as off-chain components.	Hardware (Intel CPUs) dependent	✓	Partially
Chainlink [23]	A token-based framework for building secure decentralized input and output oracles for complex smart contracts on any blockchain.	Hardware (Intel CPUs) dependent	✓	✓

work (i.e., assignments) balancement across multiple external services. Each assignment consists of subtasks that are processed as a pipeline. Also, adaptors can create custom subtasks defined as external services with a minimal REST API. Chainlink has a validation system for monitoring the on-chain oracle behavior regarding availability and correctness and provides performance metrics. It also has a reputation system for collecting user ratings of oracle providers and nodes and presenting their historical performance. It also has a certification service responsible for endorsing high-quality oracle providers and employs an optional contract-upgrade service to create a new set of oracle contracts in case of vulnerabilities.

The Chainlink technology has also progressed [11] toward providing key oracle functions, e.g., an extensive collection of on-chain financial market data or verifiable randomness backed by on-chain cryptographic proofs. Chainlink 2.0 is a decentralized oracle network (DON) to create a decentralized meta layer for enhancing smart contracts. There are hybrid smart contracts where DONs offer capabilities to fill in the blockchain limitations and connect to the off-chain systems. By the advanced off-chain computation, DONs provide a blockchain-agnostic gateway for smart contracts not only for off-chain access but also for providing an execution code environment to address blockchain limitations. The use of Chainlink can also be seen in Ampleforth [5] that is

recognized as a piece of software running on the Ethereum blockchain aiming to incentivize a network of users to maintain a value of crypto-asset equal to the U.S. dollar. The Ampleforth employs a token called AMPL such that its supply is adjusted programmatically by the software that reduces the reliance on deposits or issuing and redeeming debt. This process is called "rebasing" and takes place every 24 hours in a way that if the demand for AMPL tokens is high and the price of each AMPL token exceeds $1, there will be an increase in supply. Otherwise, the supply will be adjusted and will decrease; hence, the AMPL token is recognized as a cryptocurrency token that is elastic and non-dilutive. In other words, despite changes in the supply, users keep possession of the same proportion of the overall supply.

Table 8 illustrates that the usage of hardware for data authenticity proof may lead to limiting the oracle to employ specific hardware. Although promising, it is not a generalized strategy and does not provide flexibility.

4.3 Proofless

Wang et al., [82] present a blockchain oracle based on Application-Specific Knowledge Engines (ASKE) that is a framework to acquire and analyze information. Open-source information in specific domains is collected and unified, and the framework analyzes the collected data in different dimensions by integrating data analysis methods. The analysis uses knowledge configuration files (KCF) for specifying keywords, search sequences, topics, and schedules for query processing and helping users with accurately finding the required information. This framework facilitates data collection from authoritative websites via web crawlers automatically and aggregates data off-chain to produce the final results for sending to the smart contracts. In this proposed oracle, authoritative domains for fetching the information are recognized by asking domain experts (i.e., evaluating their reputations). The oracle determines the domain of a request, and with the help of the ASKE framework, the results are extracted, analyzed, and returned to the blockchain.

Al Breiki et al., [3] present a decentralized access control for IoT data assisted by blockchain and trusted oracles. It leverages smart contracts to shift access management toward a decentralized, secure, and scalable management for IoT data access. It employs multiple oracles to provide decentralized but trusted source feeds for IoT data. The proposed system consists of entities; admins for user control access, end-user (i.e., DApps or wallet), smart contracts for verifying IoT user data access, oracles for providing information about the registered oracles, aggregator, and the reputation smart contract. They are responsible for sending a data request to the set of oracles and computing the hashes for the requested data. They then compare hashes and report to the reputation smart contract for averaging.

Fujihara [28] presents a blockchain-based open data platform that relies on mobile crowdsourcing. It employs a decentralized oracle to extract the accurate binary information for saving recorded onto the blockchain via the Expectation-Maximization (EM) algorithm. There are task requesters, workers, and tasks in the proposed platform, and the algorithm is done in E and M steps. In the former step, for each task, the corresponding E_i value is determined by the Bayes' theorem. In the M step, the probability of the correct answer for tasks considering the E-step is calculated. By step repetition, the estimated values gradually converge to fixed values resulting in the determination of workers' reliability scores. This score is used for incentivizing workers and is proportional to the score.

Pedro et al., [20] present Witnet, which is a decentralized oracle network that runs on a blockchain with a native protocol token for incentivizing miners. It enables any software to retrieve content published at any web address (HTTP/HTTPS) with complete and verifiable proof of its integrity while immune to Sybil attacks and laziness (e.g., verifier's dilemma). The ledger on the Witnet is based on a directed acyclic graph (DAG) where multiple blocks can exist at a time while enforcing a

Table 9. The summary of proofless-based studies for reputation-based oracles

Literature	Key Research Outcomes	Limitation	Authenticity	Confidentiality
Wang et al., [82]	Leveraging Application Specific Knowledge Engines (ASKE) for information acquisition and analysis.	Not decentralized and lack of data authenticity mechanisms	✗	✗
Al Breiki et al., [3]	A decentralized access control for IoT data that is assisted by blockchain and trusted oracles.	No authenticity proof mechanism	Partially	✗
Fujihara [28]	Presenting open data platform that is assisted with a decentralized oracle for correct information extraction.	No authenticity proof mechanism	Partially	✗
Pedro et al., [20]	A decentralized oracle network that employs truth-by-consensus protocol for obtaining the truth, and uses miners for retrieving, attesting, and delivering (RAD) of web contents which heavily depends on the reputation.	No authenticity proof mechanism	✓	Partially

legit ledger. In addition, miners who are witnesses earn by retrieving, attesting, and delivering (RAD) web contents to the users via a deterministic algorithm that heavily relies on reputation. Demurrage gradually affects this reputation, meaning a deduction on reputation scores is proportionally applied. They are required to work honestly as contradicting with the majority of miners would lose their reputation. In the Witnet, witnesses compete to earn a reward, and with respect to their mining power, rewards become proportional to their previous honesty and trustworthiness, i.e., their reputations. The Witnet employs truth-by-consensus protocol to obtain the "agreed truth". This protocol is based on singular value decomposition (SVD) to analyze a matrix that contains all the claims produced during epochs. Moreover, the network scalability is guaranteed by the sharding feature of the Witnet and allowing clients to choose several witnesses for the RAD tasks. Miners use a scriptable headless browser with no interface to retrieve information from websites. In addition to clients and miners, there are bridge nodes in charge of watching other blockchains in case of RAD requests and replicating the results upon requests.

Table 9 presents the summary of proofless strategies, and it illustrates that they barely utilize authenticity mechanisms for verifying data integrity.

5 RESEARCH CHALLENGES AND FUTURE RESEARCH DIRECTIONS

This section presents research challenges and future research directions in blockchain oracle design and usage. Although blockchain oracles are well-studied, there are still unaddressed research and technical questions in practice.

5.1 Operating Cost and Speed

Smart contracts use resources for execution, e.g., gas in the Ethereum blockchain; hence, it necessitates developing very efficient and optimal code for smart contracts and provides a faster response time for incoming queries. While there have been studies for developing cost-effective blockchain-based applications (e.g., [91]), there is still a need for designing high-performance blockchain oracles. For example, Chainlink has recently presented Off-Chain Reporting (OCR)

that significantly improves data computation across Chainlink oracles while reducing the operating cost [17]. Moreover, the design and deployment of oracles should be relied on employing high-performance and low transaction fee blockchains, e.g., Solana or Polkadot blockchains. However, blockchain oracles tend to be deployed on Ethereum-based blockchains, and performance comparison of blockchains assists developers in choosing the proper blockchain.

5.2 Decentralized Oracles and Security

Although presented oracle techniques may sound sophisticated and novel, they still require data integrity and authenticity mechanisms for enforcing security and privacy. Although decentralized oracles benefit data acquisition, data security challenges are still the issues. Hence, the design of oracles should provide an acceptable level of data integrity, security, and privacy. In Section 3, we discussed different strategies for the related studies in voting-based oracles; however, the majority, if not all of them, barely employed authenticity proof mechanisms for data integrity and correctness.

Moreover, with the ever-increasing introduction of blockchain oracles and their customized tokens to the market for investment, detecting a legitimate project is essential. It requires more attention from the research perspective. For example, Mate Tokay, co-founder of bitcoin.com, has filed legal action against the Bridge.link founders due to misleading Bridge token (BRG) investors [10]. Hence, a set of requirements for identifying a legitimate blockchain oracle should be defined to mitigate such issues.

5.3 Blockchain-agnostic Oracles

Majority of studies presented in Sections 3 and 4 employed a certain blockchain (i.e., Ethereum blockchain). In contrast, few studies have presented blockchain-agnostic oracles, e.g., Band protocol [8] or Chainlink [17] oracle. Therefore, blockchain oracles' adaptability should be studied further to propose crucial rules and requirements for developers. Also, it will make the blockchain oracle design more flexible and functional to the constant changes in such a growing ecosystem, assisting decentralized applications with execution across multiple blockchains. Such flexibility will allow users to easily handle transactions without using different exchange platforms, contributing to the lower operating cost.

5.4 Query Types and Uncovering Dishonest Data Providers

Oracles designed for fetching off-chain information should be capable of dealing with different query types, e.g., binary, scalar, or categorical. Processing non-binary query types are challenging as the diversity of responses may be big enough; hence, it requires techniques to manage data aggregation at the blockchain efficiently. Moreover, there should be reliable while fast authenticity proof mechanisms attached to the data for data integrity verification.

In addition, providing incorrect information by the data providers may happen. Thus, there should be policies, procedures, or mechanisms universally defined that assist developer to take them into account. As explained in Section 3, some studies were not sensitive to Sybil attacks or verifier's dilemma, which led to questioning the integrity and correctness of the final outcome.

5.5 The Use of AI in Blockchain Oracles

Limited studies are focusing on the use of AI in blockchain oracles, e.g., Oraichain [56]. Advantages of employing AI include profiling the data providers when choosing them to fetch a query's response and detect dishonest participants. Although AI models cannot be implemented on-chain, an AI-enabled off-chain component can be a game-changer to ensure the security and correctness of

responses. A comprehensive study on AI-based blockchain oracles will reveal essential information regarding their performance, adaptability, and usage for developers.

6 CONCLUSION

Blockchain technology has disrupted digital interaction in our economy and society in the last few years. Blockchain has enabled data to be shared among nodes connected over the internet through distributed ledger technology. In addition, by introducing smart contracts to the blockchain, programmability is added to this disruptive technology and has changed the software ecosystem by removing third parties for administration of (non)business purposes. Although promising, blockchain and smart contracts do not have access to the external world; hence, they need *trusted* services referred to as blockchain oracles for sending and verifying external information to smart contracts. These blockchain oracles are smart contracts implemented in the form of an application programming interface (API) and off-chain components (i.e., data providers) connecting the world to the blockchain. This paper presented an overview of blockchain oracles implementation technologies and mechanisms based on the feedback (i.e., the blockchain oracle outcome) with respect to data integrity and correctness. Hence, we investigated the blockchain oracle structure and principles by classifying their implementation techniques into two major groups: voting-based oracles and reputation-based oracles. While the first group leverages voting strategies, e.g., stake on outcomes, for data aggregation and outcome, the latter considers reputation and performance metrics, e.g., the influential participant(s) on the network, for choosing the oracle for reporting the outcome to the requester. Oracles may employ authenticity proof mechanisms to check the correctness and integrity of the obtained data from external sources. Our discussion and review showed that the existing strategies should keep the integrity of data obtained from external resources, and oracles should honestly work toward providing the truth back to the blockchain and smart contracts. Moreover, although existing studies sound promising, blockchain oracles still need further research from different perspectives, such as operating cost, processing speed, security, and handling of different query types.

REFERENCES

[1] J. Adler, R. Berryhill, A. Veneris, Z. Poulos, N. Veira, and A. Kastania. 2018. Astraea: A Decentralized Blockchain Oracle. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 1145–1152.

[2] Selena Ahmed and Noah Ten Broek. 2017. Blockchain could boost food security. *Nature* 550, 7674 (2017), 43–43.

[3] H. Al Breiki, L. Al Qassem, K. Salah, M. Habib Ur Rehman, and D. Sevtinovic. 2019. Decentralized Access Control for IoT Data Using Blockchain and Trusted Oracles. In *2019 IEEE International Conference on Industrial Internet (ICII)*. IEEE, 248–257. https://doi.org/10.1109/ICII.2019.00051

[4] H. Al-Breiki, M. H. U. Rehman, K. Salah, and D. Svetinovic. 2020. Trustworthy Blockchain Oracles: Review, Comparison, and Open Research Challenges. *IEEE Access* 8 (2020), 85675–85685. https://doi.org/10.1109/ACCESS.2020.2992698

[5] Ampleforth. 2021. Ampleforth. https://www.ampleforth.org/basics/.

[6] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. 2018. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the thirteenth EuroSys conference*. ACM, 1–15.

[7] T. Arts, Y. Malahov, and H. Sascha. 2020. Æternity Open source blockchain for scalable and secure smart contracts. https://raw.githubusercontent.com/keypair/white-paper/master/aeternity-whitepaper.pdf.

[8] Bandchain. 2020. Band Protocol. https://docs.bandchain.org/whitepaper.

[9] Abdeljalil Beniiche. 2020. A study of blockchain oracles. *arXiv preprint arXiv:2004.07140* (2020).

[10] Bitcoin.com. 2021. Bitcoin.com Co-founder Files Legal Action Against Bridge.link Token Project Over Market Manipulation. https://news.bitcoin.com/bitcoin-com-co-founder-files-legal-action-against-bridge-linktoken-project-over-market-manipulation/.

[11] Lorenz Breidenbach, Christian Cachin, Benedict Chan, Alex Coventry, Steve Ellis, Ari Juels, Farinaz Koushanfar, Andrew Miller, Brendan Magauran, Daniel Moroz, et al. 2021. Chainlink 2.0: Next Steps in the Evolution of Decentralized Oracle Networks. https://research.chain.link/whitepaper-v2.pdf.

[12] Roman Brodetski. 2017. Oracul System. https://gist.github.com/RomanBrodetski.

[13] Sa˘sa Mili´c Burak Benligiray and Heikki V¨anttinen. 2021. API3 Decentralized APIs for Web 3.0. https://raw.githubusercontent.com/api3dao/api3-whitepaper/master/api3-whitepaper.pdf.

[14] Vitalik Buterin. 2014. SchellingCoin: A Minimal-Trust Universal Data Feed. https://blog.ethereum.org/2014/03/28/schellingcoin-a-minimal-trust-universal-data-feed/.

[15] Y. Cai, G. Fragkos, E. E. Tsiropoulou, and A. Veneris. 2020. A Truth-Inducing Sybil Resistant Decentralized Blockchain Oracle. In *2020 2nd Conference on Blockchain Research Applications for Innovative Networks and Services (BRAINS)*. Wiley, 128–135. https://doi.org/10.1109/BRAINS49436.2020.9223272

[16] Yuxi Cai, Nafis Irtija, Eirini Eleni Tsiropoulou, and Andreas Veneris. 2021. Truthful Decentralized Blockchain Oracles. *International Journal of Network Management* (2021), e2179.

[17] Chainlink. 2021. Chainlink Achieves Major Scalability Upgrade With the Mainnet Launch of Off-Chain Reporting (OCR). https://blog.chain.link/off-chain-reporting-live-on-mainnet/.

[18] Coldcard. 2021. Coldcard Hardware Wallet. https://coldcardwallet.com/.

[19] Corda. 2019. Corda: A distributed ledger. https://www.corda.net/content/corda-technical-whitepaper.pdf.

[20] Adán Sánchez de Pedro, Daniele Levi, and Luis Iván Cuende. 2017. Witnet: A decentralized oracle network protocol. *arXiv preprint arXiv:1711.09756* (2017).

[21] Delphi. 2017. Delphi. https://delphi.systems/whitepaper.pdf.

[22] Edenchain. 2018. Edenchain. =https://edenchain.io/wp-content/uploads/2018/08/EdenChain-Whitepaper_v1.2.pdf.

[23] S. Ellis, A. Juels, and S. Nazarov. 2017. ChainLink A Decentralized Oracle Network. https://link.smartcontract.com/whitepaper.

[24] Shayan Eskandari, Jeremy Clark, Vignesh Sundaresan, and Moe Adham. 2017. On the feasibility of decentralized derivatives markets. In *International Conference on Financial Cryptography and Data Security*. Springer, 553–567.

[25] Ethereum. 2020. Ethereum Blockchain. https://ethereum.org/en/whitepaper/.

[26] Ethereum. 2021. ERC-20 TOKEN STANDARD. https://ethereum.org/en/developers/docs/standards/tokens/erc-20/.

[27] IOTA Foundation. 2021. Introducing IOTA Oracles. https://blog.iota.org/introducing-iota-oracles/.

[28] Akihiro Fujihara. 2019. Proposing a blockchain-based open data platform and its decentralized oracle. In *International Conference on Intelligent Networking and Collaborative Systems*. Springer, 190–201.

[29] Vahid Garousi, Michael Felderer, and Mika V Mäntylä. 2019. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology* 106 (2019), 101–121.

[30] Gnosis. 2017. Gnosis. https://github.com/gnosis/research/blob/\master/gnosis-whitepaper.pdf.

[31] Juan Guarnizo and Pawel Szalachowski. 2019. PDFS: practical data feed service for smart contracts. In *European Symposium on Research in Computer Security*. Springer, 767–789.

[32] J. He, R. Wang, W. Tsai, and E. Deng. 2019. SDFS: A Scalable Data Feed Service for Smart Contracts. In *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 581–585.

[33] Jonathan Heiss, Jacob Eberhardt, and Stefan Tai. 2019. From oracles to trustworthy data on-chaining systems. In *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 496–503.

[34] Hrishikesh Huilgolka. 2019. Razor Network: A decentralized oracle platform. https://razor.network/whitepaper.pdf.

[35] Hyperledger. 2014. Hyperledger Fabric. https://hyperledger-fabric.readthedocs.io/.

[36] Markus Jakobsson and Ari Juels. 1999. Proofs of Work and Bread Pudding Protocols (Extended Abstract). Secure Information Networks (s. 258-272).

[37] JustLink. 2020. BJustLink A decentralised oracle network on TRON. https://docs.justlink.io/whitepaper/justlink_whitepaper_v1.0.pdf.

[38] Ryuuji Kamiya. 2018. Shintaku: An end-to-end-decentralized general-purpose blockchain oracle system. https://gitlab.com/shintakugroup/paper/blob/master/shintaku.pdf.

[39] Kylin. 2021. Kylin Network. https://kylin.network/.

[40] Protocol Labs. 2021. InterPlanetary File System (IPFS). https://docs.ipfs.io/.

[41] Ledger. 2020. Ledger Troubleshooting. https://support.ledger.com/hc/en-us.

[42] Ledger. 2021. Ledger Proof. https://ledger.com.

[43] Guozhu Liang, Wei Wu, and Jingyu Wang. 2021. Polkaoracel A Substrate-based Self-evolving Oracle System. https://polkaoracle-1.gitbook.io/polkaoracle-wiki/.

[44] Wenzhu liang. 2021. Polkadot-based decentralized cross-chain prediction platform. https://x-predict.com/X_Predict_market_Whitepaper_en.pdf?v=1.0.

[45] Kamran Mammadzada, Mubashar Iqbal, Fredrik Milani, Luciano García-Bañuelos, and Raimundas Matulevičius. 2020. Blockchain Oracles: A Framework for Blockchain-Based Applications. In *International Conference on Business Process*

Management. Springer, 19–34.
[46] MarkerDAO. 2014. The Maker Protocol: MakerDAO's Multi-Collateral Dai (MCD) System. https://makerdao.com/en/whitepaper.
[47] Marco Merlini, Neil Veira, Ryan Berryhill, and Andreas Veneris. 2019. On public decentralized ledger oracles via a paired-question protocol. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 337–344.
[48] Microsoft. 2019. Introducing Project "Bletchley". https://github.com/Azure/azure-blockchain-projects/blob/master/bletchley/bletchley-whitepaper.md.
[49] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun. 2017. A review on consensus algorithm of blockchain. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2567–2572.
[50] Mobius. 2021. Mobius Network. https://mobius.network/.
[51] H. Moudoud, S. Cherkaoui, and L. Khoukhi. 2019. An IoT Blockchain Architecture Using Oracles and Smart Contracts: the Use-Case of a Food Supply Chain. In *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 1–6. https://doi.org/10.1109/PIMRC.2019.8904404
[52] Roman Mühlberger, Stefan Bachhofner, Eduardo Castelló Ferrer, Claudio Di Ciccio, Ingo Weber, Maximilian Wöhrer, and Uwe Zdun. 2020. Foundational Oracle Patterns: Connecting Blockchain to the Off-chain World. In *International Conference on Business Process Management*. Springer, 35–51.
[53] K. Nelaturu, J. Adler, M. Merlini, R. Berryhill, N. Veira, Z. Poulos, and A. Veneris. 2020. On Public Crowdsource-Based Mechanisms for a Decentralized Blockchain Oracle. *IEEE Transactions on Engineering Management* 67, 4 (2020), 1444–1458. https://doi.org/10.1109/TEM.2020.2993673
[54] DOS Network. 2019. DOS Network: A Decentralized Oracle Service boosting blockchain usability with off-chain data & verifiable computing power. https://s3.amazonaws.com/whitepaper.dos/DOS+Network+Technical+Whitepaper.pdf.
[55] DIA Network. 2021. Decentralized Information Asset (DIA). https://docs.diadata.org/documentation.
[56] Oraichain. 2021. Oraichain AI-powered Oracle and Ecosystem for Blockchains. https://docs.orai.io/.
[57] Orisi. 2014. Orisi. https://github.com/orisi/wiki/wiki/Orisi-White-Paper.
[58] Amirmohammad Pasdar, Zhongli Dong, and Young Choon Lee. 2021. Blockchain Oracle Design Patterns. *arXiv preprint arXiv:2106.09349* (2021).
[59] Jack Peterson, Joseph Krug, Micah Zoltu, Austin K Williams, and Stephanie Alexander. 2019. Augur: a Decentralized Oracle and Prediction Market Platform (v2. 0). https://augur.net/whitepaper.pdf.
[60] Polkadot. 2021. Polkadot Network. https://wiki.polkadot.network/en/.
[61] Compound Protocol. 2021. Compound Protocol. https://compound.finance/docs.
[62] Provable. 2020. Android Proof: Authenticated Data Gathering using Android Hardware Attestation and SafetyNet. https://provable.xyz/papers/android_proof-rev2.pdf.
[63] Provable. 2020. Provable. https://provable.xyz/papers/random_datasource-rev1.pdf.
[64] Pyth. 2021. Pyth Network. https://pyth.network/.
[65] Christian Reitwießner. 2016. zkSNARKs in a Nutshell. http://chriseth.github.io/notes/articles/zksnarks/zksnarks.pdf.
[66] Hubert Ritzdorf, Karl Wüst, Arthur Gervais, Guillaume Felley, and Srdjan Capkun. 2018. TLS-N: Non-repudiation over TLS Enablign Ubiquitous Content Signing.. In *NDSS*.
[67] Mark Russinovich, Edward Ashton, Christine Avanessians, Miguel Castro, Amaury Chamayou, Sylvan Clebsch, Manuel Costa, Cédric Fournet, Matthew Kerner, Sid Krishna, et al. 2019. CCF: A framework for building confidential verifiable replicated services. *Technical Report MSR-TR-201916* (2019).
[68] Fahad Saleh. 2021. Blockchain without waste: Proof-of-stake. *The Review of financial studies* 34, 3 (2021), 1156–1190.
[69] Andreas Schaad, Tobias Reski, and Oliver Winzenried. 2019. Integration of a Secure Physical Element as a Trusted Oracle in a Hyperledger Blockchain.. In *ICETE*. Scitepress, 498–503.
[70] Slush and Stick. 2021. Trezor Hardware Wallet. https://trezor.io/.
[71] Solana. 2021. Solana: High-performance Blockchain. https://solana.com/.
[72] Nick Spanos. 2021. Zap Protocol. https://zap.org/docs/.
[73] Paul Sztorc. 2015. Truthcoin, peer-to-peer oracle system and prediction marketplace. https://www.truthcoin.info/papers/truthcoin-whitepaper.pdf.
[74] Bridge Oracle Team. 2021. Bridge Oracle System: First Ever Public Oracle System on TRON Network. https://bridge.link/Bridge_White_Paper.pdf.
[75] Stellar Team. 2021. Stellar is an open network for storing and moving money. https://www.stellar.org/?locale=en.
[76] Tellor. 2021. Tellor. https://docs.tellor.io/tellor/whitepaper/.
[77] Jason Teutsch and Christian Reitwießner. 2017. Truebit: A scalable verification solution for blockchains. https://people.cs.uchicago.edu/~teutsch/papers/truebit.pdf.
[78] TLSnotary. 2014. TLSnotary a mechanism for independently audited https sessions. https://tlsnotary.org/TLSNotary.pdf.

[79] TRON. 2021. TRON Network. https://tron.network/.
[80] Santander UK. 2018. Santander launches the first blockchain-based international money transfer service across four countries. https://www.santander.co.uk/about-santander/media-centre/press-releases/santander-launches-the-first-blockchain-based-international-money-transfer-service-across-four.
[81] Vyper. 2021. Vyper: a contract-oriented, pythonic programming language for Ethereum Virtual Machine (EVM). https://vyper.readthedocs.io/en/stable/.
[82] S. Wang, H. Lu, X. Sun, Y. Yuan, and F. Wang. 2019. A Novel Blockchain Oracle Implementation Scheme Based on Application Specific Knowledge Engines. In *2019 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*. IEEE, 258–262. https://doi.org/10.1109/SOLI48380.2019.8955107
[83] Jonathan Warren. 2012. Bitmessage: A peer-to-peer message authentication and delivery system. https://bitmessage.org/bitmessage.pdf. , 5 pages.
[84] Kain Warwick. 2021. Synthetix The Derivatives Liquidity Protocol. https://docs.synthetix.io/litepaper.
[85] Wibu. 2020. Wibu CmDongle. https://www.wibu.com/products/codemeter/cmdongle.html.
[86] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. ACM, 1–10.
[87] Sangyeon Woo, Jeho Song, and Sungyong Park. 2020. A distributed oracle using intel SGX for blockchain-based IoT applications. *Sensors* 20, 9 (2020), 2725.
[88] Xiwei Xu, Cesare Pautasso, Liming Zhu, Vincent Gramoli, Alexander Ponomarev, An Binh Tran, and Shiping Chen. 2016. The Blockchain as a Software Connector. In *13th Working IEEE/IFIP Conference on Software Architecture (WICSA 2016)*. IEEE, Venice, Italy, 182–191.
[89] Xiwei Xu, Cesare Pautasso, Liming Zhu, Qinghua Lu, and Ingo Weber. 2018. A pattern collection for blockchain-based applications. In *Proceedings of the 23rd European Conference on Pattern Languages of Programs*. ACM, 1–20.
[90] Xiwei Xu, Ingo Weber, and Mark Staples. 2019. Blockchain patterns. In *Architecture for Blockchain Applications*. Springer, 113–148.
[91] Abdullah A Zarir, Gustavo A Oliva, Zhen M Jiang, and Ahmed E Hassan. 2021. Developing Cost-Effective Blockchain-Powered Applications: A Case Study of the Gas Usage of Smart Contract Transactions in the Ethereum Blockchain Platform. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 30, 3 (2021), 1–38.
[92] Can Zhang, Liehuang Zhu, Chang Xu, and Kashif Sharif. 2020. PRVB: Achieving Privacy-Preserving and Reliable Vehicular Crowdsensing via Blockchain Oracle. *IEEE Transactions on Vehicular Technology* 70, 1 (2020), 831–843.
[93] Fan Zhang, Ethan Cecchetti, Kyle Croman, Ari Juels, and Elaine Shi. 2016. Town crier: An authenticated data feed for smart contracts. In *Proceedings of the 2016 aCM sIGSAC conference on computer and communications security*. ACM, 270–282.
[94] Fan Zhang, Deepak Maram, Harjasleen Malvai, Steven Goldfeder, and Ari Juels. 2020. Deco: Liberating web data using decentralized oracles for tls. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1919–1938.
[95] Z. Zhang, L. Huang, R. Tang, T. Peng, L. Guo, and X. Xiang. 2020. Industrial Blockchain of Things: A Solution for Trustless Industrial Data Sharing and Beyond. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. IEEE, 1187–1192. https://doi.org/10.1109/CASE48305.2020.9216817