Architectural Aspects of Blockchain-Based IoT Economy

Subjects: Computer Science, Information Systems

Contributor: Marco Zecchini

When an IoT ecosystem is equipped with a blockchain, many aspects regarding the blockchain itself and how it is integrated with the rest of the ecosystem have to be carefully considered. The adoption of a blockchain is not free, but it raises problems (e.g., regarding scalability) that have to be addressed to allow a project to develop and work correctly and profitably. On the other hand, it is possible to recognize some blockchain design opportunities that become available when a blockchain is used in an IoT ecosystem.

Internet of Things (IoT) blockchain economy payment

1. Blockchain Nodes: Technology and Deployment Choices

When adopting blockchain for an IoT application, there are three main options:

- leverage an existing general-purpose public blockchain network;
- leverage an existing blockchain technology while creating a distinct dedicated network;
- create a new ad-hoc blockchain technology and a new corresponding network.

In order to reduce production costs and leverage the reputation and reliability of already deployed solutions, the first option is usually considered the most appropriate, and it is easy to find examples of this approach (see, for example, [1][2][3]). As already discussed, public permissionless blockchains also have the advantage of providing the highest guarantee in terms of security—a crucial requirement for the IoT economy—and to facilitate the employment of the obtained incentives in a wider ecosystem of heterogeneous applications and services. The reader can find many examples of projects adopting general-purpose unpermissioned blockchains. The main drawback of this approach is that the IoT application is going to depend on the fluctuations of the public blockchain, and in particular on its network load. Public networks may be congested by usage spikes due to speculations [4] or other applications [5], just to mention two relevant examples. Further, communities governing a general-purpose public blockchain may make choices (e.g., regarding architecture evolution or required node power) that may be in contrast with the needs or the design of the considered IoT application. For these reasons, in some scenarios a dedicated blockchain network may be preferred. In other words, a well-known blockchain technology can be adopted only for a specific application with a dedicated network. In this case, all the issues related to fluctuations can be more easily handled. MedicalChain [6] is an example of this approach implemented as a permissioned blockchain. However, permissioned blockchains have limited decentralization. The most ambitious multivendor IoT ecosystems would probably rely on unpermissioned blockchains.

The third approach is to develop an ad hoc blockchain technology to be deployed as a dedicated blockchain, typically on IoT gateways. This is more costly, but allows greater flexibility. For example, in the Helium [7] network, a specific blockchain is proposed that leverages the physical presence of devices on a territory, and their scarcity, to realize a new proof-of-coverage consensus algorithm. The work done by devices to achieve consensus is not wasted (as occurs in Bitcoin and in all blockchains based on regular proof-of-work), but is reused within the Helium ecosystem, realizing an elegant and efficient use of resources.

Regarding this third approach, a possible criticism is that an ad hoc dedicated blockchain may be considered less reliable than a general purpose public blockchain. In fact, it can be expected a smaller community working on the codebase, and hence governance, bug fixing, and software updates are expected to be less effective. On the other side, an ad hoc technology is expected to be simpler and more focused on the needs of the specific IoT application.

In general, the trade-off between the possible greater efficiency of ad hoc solutions and the time necessary to acquire a satisfactory reputation with the wider public—key ingrediente for the success of an IoT economy—should be carefully evaluated.

2. Accessing a Blockchain from Resource-Constrained Devices

Since things and thing providers are often large in number, it is natural to consider hosting the nodes of the blockchain in the very same IoT devices. However, as remarked, IoT devices are very often resource constrained and thus cannot always satisfy the requirements highlighted. **Figure 1** summarizes the possible roles of things and gateways with respect to blockchain integration. Things are hardly suited to directly speak to the blockchain due to their limits, and usually, with current technology, they just rely on a distinct (trusted) device or service to submit blockchain transactions on their behalf. If they are attached to the blockchain, they can at most play the role of a light node.

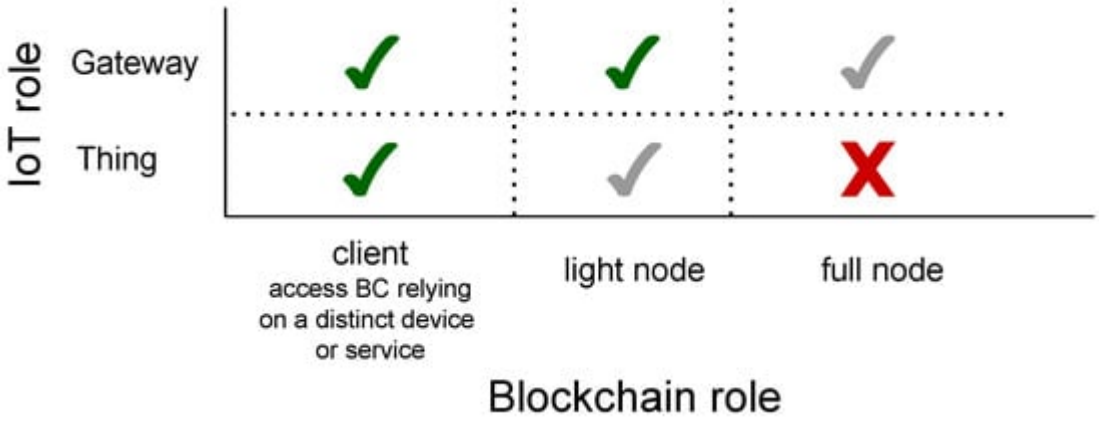

Figure 1. Summary of the possible blockchain roles that IoT devices can have.

If the IoT architecture encompasses a nearby gateway or a server of a fog computing layer, these can be directly exploited to interface the blockchain, surely in the role of a light node and, in certain cases, even as a regular full node. In this case, a gateway or a server can be used by its nearby things not only for connecting to the Internet but also to submit blockchain transactions.

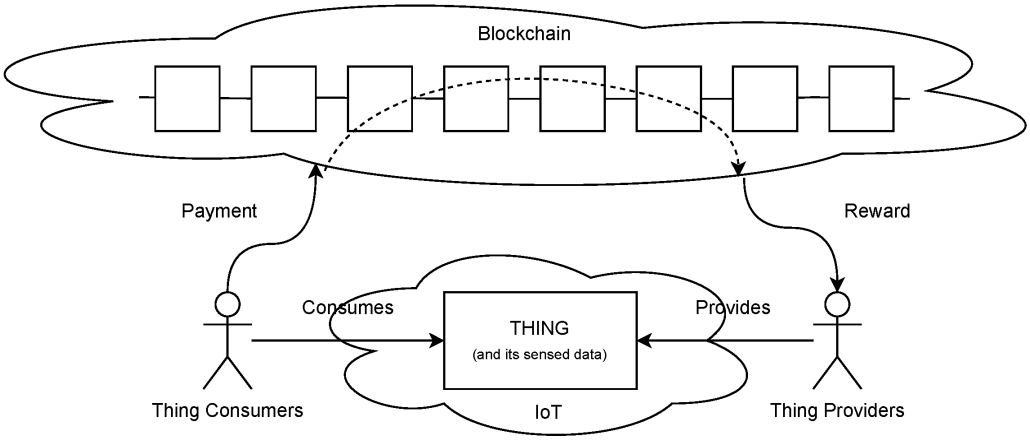

Figure 2. A schematic representation of the reference scenario. Thing consumers pay to use things or the data they gather. They are entitled to do that only if their payment was recorded in the blockchain. Thing providers get a reward to make their things or data available. The reward is autonomously dispensed by a transaction on the blockchain.

In view of **Figure 2**, there are at least two fundamental use-cases to be supported regarding the interaction between IoT devices and the blockchain: a device should be able to interact with the blockchain (1) to perform payments and (2) to assess that a payment has been performed. Ideally, any device that has to perform these tasks should have access to the whole blockchain status (or history, depending on the technology). This is clearly unfeasible even for moderately powerful devices, such as, for example, mobile phones. To overcome this problem, light nodes adopt *simplified payment verification*, where Merkle proofs [8] are used as a means of verification of the information collected from untrusted nodes.

However, even simply collecting and storing these proofs is still well above the power of many IoT devices. The work in [9] analyzes this problem and surveys results about different SPV implementations in the context of healthcare applications. It is worth mentioning a new technology, Mina [10], which offers an elegant solution using advanced cryptography and recursive zk-SNARKs to reduce the size of the blockchain to tens of KB (instead of hundreds of GB). Instead of verifying the entire chain from the beginning of time (full node), participants fully verify the network and transactions using recursive zero-knowledge proofs (or zk-SNARKs). Nodes can then store the small proof (of constant size), as opposed to the entire chain. While very promising, the Mina protocol can be considered still in its infancy.

A more drastic solution that eases the adoption of very small IoT devices is to avoid having them store any proof. This means relying on an external centralized service to access the blockchain, which has to be considered trusted. An example of this approach is given by Helium [11]. While this may be considered secure enough for many

applications, the introduction of a centralized element in the architecture has been regarded us unsatisfactory by some authors. For example, the INCUBED protocol [12] and other competing solutions [13][14] have the objective to provide very small devices with access to a blockchain without relying on a trusted third party.

Figure 3 summarizes the possible relations between things and a blockchain.

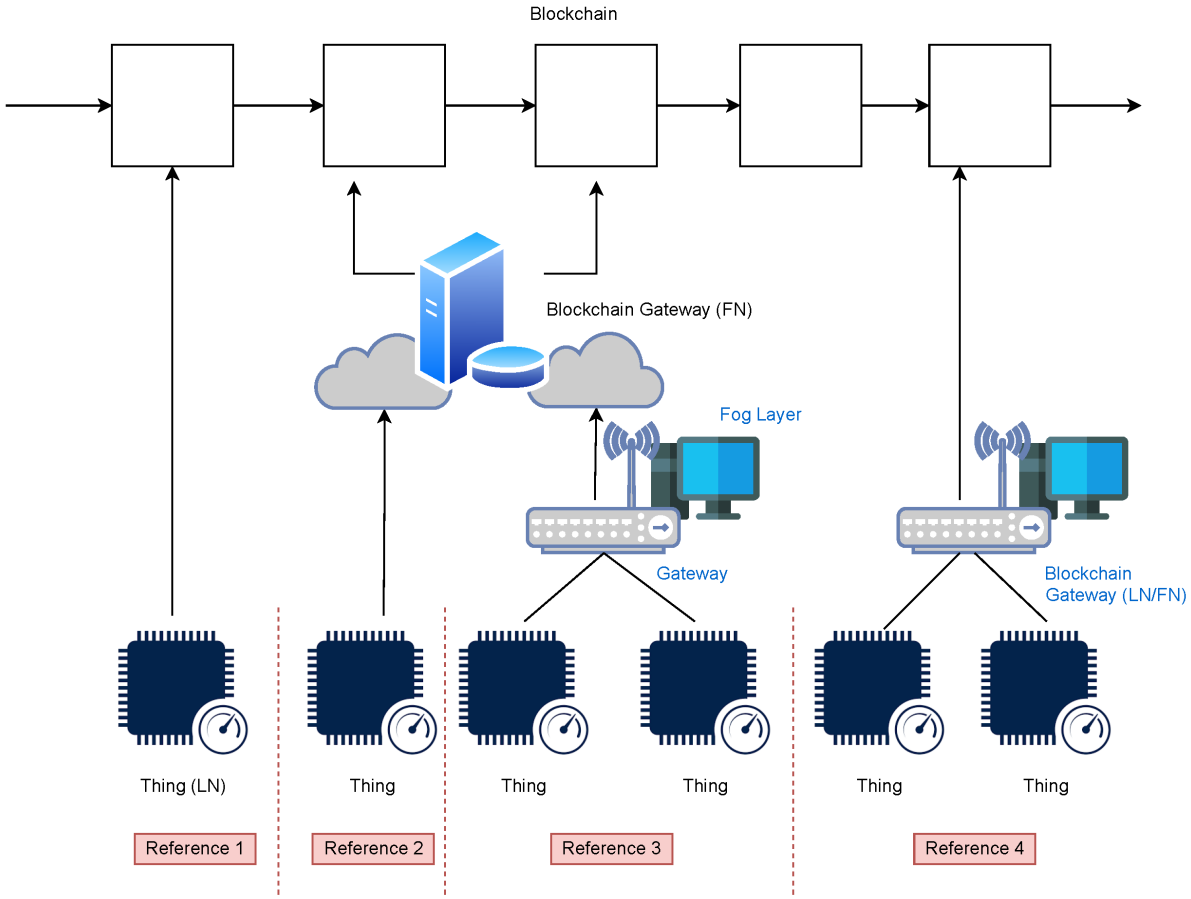

Figure 3. There are four main types of reference scenarios. In Reference 1, things have sufficient resources to operate autonomously as light nodes (LN). In Reference 2, things can autonomously be connected to the Internet, but their limited resources require them to rely upon third-party blockchain services to interact with the blockchain. In Reference 3, things need to rely on a gateway to access the Internet, but also in this case, the gateway or a server of a fog computing layer does not have sufficient resources to interact with the blockchain, and thus it relays on a third party. In Reference 4, things still need a gateway or an intermediate fog computing layer, but in this case, the gateway or the server has sufficient resources to run a light (LN) or full (FN) blockchain node.

3. Oracles: Interfacing the Blockchain with Off-Chain Data and Devices

Blockchains and smart contracts can only access data stored within the blockchain itself; on the contrary, IoT applications are ultimately designed to provide access to the physical world. This occurs, for example, in vehicle rentals [15][16], smart cars [1], and Industry Marketplace [3]. Blockchain technologies are designed to be

deterministic, that is, when the whole transaction history is replayed it always ends up with the same results. Determinism is important so that blockchain nodes can come to a consensus [17]. If a smart contract requires accessing the measure of a smart meter, the value could differ from time to time, or even from place to place, causing nodes in the future, or without access to a certain site, to reach different conclusions about the state of the network, thus breaking the consensus. *Oracles* are components that allow a blockchain, or a smart contract, to get inputs from outside the blockchain. They inject data coming from outside the blockchain into regular blockchain transactions. In this way, they become part of the blockchain history and can be handled deterministically by all blockchain nodes.

There are several oracle services providing APIs to allow smart contracts to access external data. Examples include Chainlink [18], Provable [19], BandChain [20], and Tellor [21]. Oracle functionalities can even be part of an IoT ecosystem. For example, in Helium [11], certain nodes of the network are in charge of providing information about the exchange ratio of the Helium native token to keep the service price constant. This is a form of special-purpose oracle included in an IoT ecosystem.

Oracles can be classified according to the following aspects.

Origin of off-chain data. There are *software oracles* and *hardware oracles*. A software oracle handles information data that originates from online sources, like the prices of commodities and goods, flight or train delays, and so on. Therefore, it extracts the needed information from an online resource and pushes it into the smart contract. Hardware oracles allow smart contracts to gather information directly from the physical world, for example, a car crossing a barrier where movement sensors must detect the vehicle and send the data to a smart contract [1], or RFID sensors in the supply chain industry [3].

Inbound/outbound oracles.*Inbound oracles* pull in-chain data from the external world. *Outbound oracles* provide smart contracts with the ability to send data to the outside world. An example would be a smart lock in the physical world, which receives payment on its blockchain address and needs to unlock automatically.

Degree of decentralization. Oracles can be centralized entities getting data from the off-chain world. However, using only one source of information could be risky and unreliable. For further security, a combination of different oracles may be used, where, for example, three out of five oracles could determine the outcome of an event. This combination of multiple oracles is called *consensus-based oracles*. ChainLink [18] and Tellor [21] are two examples of decentralized oracles. The special-purpose oracle of Helium mentioned above is consensus-based but has limited decentralization, since currently, only 11 fixed members can submit exchange ratio data (nine of them are anonymous for security reasons).

Figure 4 summarizes two methods of interaction of an IoT device with the blockchain. The thing can autonomously initiate the interaction with a smart contract. In this case, it acts as the source of a “standard” transaction invoking the smart contract; consequently, oracles are not necessary. If the thing is queried by the smart contract, oracles are required to guarantee the determinism and provide a consistent data view of the observed thing.

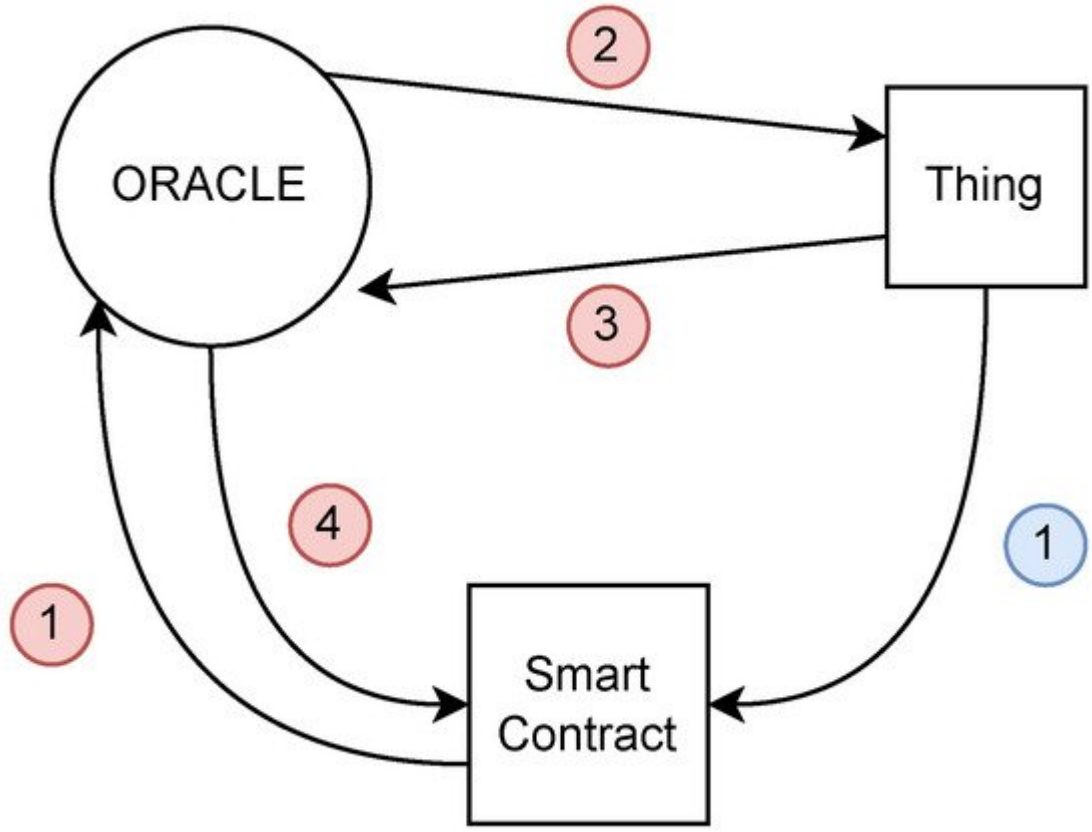

Figure 4. Methods of interaction of an IoT device with the blockchain. When the thing pushes data into the blockchain, it can autonomously start a transaction (1). In all the cases where a smart contract needs to access data available on a thing, it has to make a request to an oracle (1) that collects the data from the thing (2 and 3) and makes them available for any subsequent request (4), guaranteeing consistency.

4. Transactions Throughput, Fees, and Sidechains

As already observed, scalability (i.e., supported transactions per second) is a major issue when blockchain is applied to the IoT. It easy to observe that most of the sample application can scale to a huge number of devices and require very high transaction throughput. Bitcoin, the first blockchain, is able to sustain only a small number of transactions per second (about 7). A vast amount of literature is available on blockchain scalability [22][23][24]. Newer technologies may sustain even several thousands transactions per second. However, since it can be expected a large number of micropayments in many applications, depending on the application and on the size of the network, even the faster blockchain technology might represent a bottleneck that imposes a strong limit on the expansion of an IoT ecosystem.

When resorting to a general-purpose public blockchain network, this problem is exacerbated by the fact that the blockchain is shared with a plethora of users that are unrelated with IoT application. For optimal functioning of the blockchain, they collectively have to generate a frequency of transactions below the maximum blockchain throughput.

Since resources of a publicly shared blockchain are scarce, and they are paid by users, the price users (or things) pay for their transactions is governed by the law of supply and demand. When the demand of transactions is close to the maximum transaction throughput, the nodes of the blockchain start picking transactions to be included in the next block, favoring those that pay more. For the most successful blockchains, this has led to very high transaction fees [25].

Further, the actual transaction cost depends on the exchange rate of the blockchain native token with respect to fiat currency, which may greatly vary over time. Certain unpermissioned blockchains have overcome this problem by proposing an approach in which transactions are feeless. Some of them are EOS [26], Nano [27], and IOTA [28]. They achieve this result by different approaches: moving the cost onto developers (EOS), asking for the users to participate in transaction confirmation (IOTA), and assuming operators of nodes have other interests beyond fees (Nano). Other approaches achieve low fees for most transactions (e.g., NEO [29]). However, even in those cases, scalability limits remain.

One solution to this problem is the adoption of *sidechains*, namely secondary blockchains connected to the main one, with a mechanism that allows bidirectional transfer of assets between the two chains. Sidechains may have their own consensus protocols specifically designed to improve scalability and interact programmatically [30] with the *mainchain* to provide the highest security guarantees and take advantage of well-reputed tokens and technologies.

Communication between the sidechain and mainchain are governed by a protocol that has to be realized with smart contracts and off-chain devices. A large number of proposals of protocols and technologies are available in the literature and as open projects [31][32][33][34][35]. Some IoT-specific contributions regarding sidechains are also present in literature [36][37][38][39].

In any case, it is important to note that, at the time of writing, current blockchain technologies do not provide higher transactions throughput when the number of nodes increases. This means that any blockchain imposes an upper bound on the frequency of transactions that can be processed; hence, it is important to choose the blockchain technology in accordance with the growth plans of the IoT network.

In certain cases, it is possible to adopt special high-transactions-throughput solutions for payment transactions based on payment channels.

5. State and Payment Channels

In certain IoT applications, the problem of limited maximum transactions throughput of blockchain technologies can be effectively tackled with the adoption of the so-called payment channels. A typical problem is charging for the use of a service on the basis of how much it is used and doing that continuously while the service is running.

This was initially considered for incremental payment of video streaming, but the problem is relevant in typical IoT applications, such as vehicle renting [15][16].

Payment channels are one of the main ideas behind micropayment off-chain solutions, such as the Lightning Network [40]. In a *payment channel*, two entities (nodes or IoT devices), which are supposed to make a large number of small payments, agree to stake an amount of tokens to guarantee that they behave correctly in managing all micropayments off-chain. The blockchain is used when the channel is opened and the two parties stake their funds, and when the channel is closed and actual settlement is performed. Each micropayment is executed off-chain by exchanging partially signed transactions that commit each party to the new value of the settlement. These transactions are supposed not to be submitted for acceptance in the blockchain unless one of the two parties misbehaves and the channel has to be closed unilaterally, freezing the current balance. The complete technical details of this approach are very clearly explained in [41], and the performance of the Lighting Network in terms of efficiency and fee reduction are optimized for the IoT ecosystem in [42].

The technique can be extended to any kind of state change, and in this case, channels are more properly called *state channels*.

Payment channels are extremely convenient since transactions are not limited by the maximum throughput of the blockchain but only by network and hardware limits. Fees are not paid for each economic transaction, but only for opening and closing transactions, which makes the adoption of a general-purpose unpermissioned blockchain much safer. In any case, the same technique can be used also in dedicated blockchains. This is the approach of Helium [11], in which payments of the Helium packet-forwarding service are performed using payment channels where the corresponding open and closing transactions are submitted on the Helium dedicated chain.

6. Smart Contracts

One of the fundamental aspects of the blockchain is that it allows the realization of automatic behavior, which usually bring some financial effect, without relying on a trusted centralized third party. This has opened the possibility of realizing automatic versions of well-known economic mechanisms or creating new ones that can exist only in a blockchain-based economic environment. Some of the most relevant, for the IoT contexts.

All blockchains provide a consensus mechanism to accept and order transactions. In principle, transactions may be limited to the simple creation and transfer of tokens. However, the need for more complex transactions was quickly recognized. In general, when designing a blockchain, there is great flexibility in the kind of transaction that can be realized. However, at least for general-purpose blockchains, the spectrum of possible useful kinds of transactions is so wide that it is impossible to realize, natively, all possible kinds of transactions.

For this reason, almost all general-purpose blockchains (starting from Bitcoin) have some form of scripting language that allows the user to adapt the rules to accept transactions according to his/her needs. In general, define a *smart contract* as software that runs in a decentralized manner on a blockchain, allowing the developer to

customize the rules according to which the transaction should be accepted. With the introduction of Ethereum [43], smart contracts acquired enough power and flexibility to allow very general applications: transactions can invoke smart contracts, smart contracts can record data to be used in subsequent invocations (i.e., they have a state), and the application logic can manage funds that are under the control of the smart contract (see, for example, Solidity [44]).

While this flexibility is very appealing, it is worth noting that it has a significant cost. In fact, smart contracts require a very controlled execution environment (a so-called *virtual machine* (e.g., see [45])) that impacts on the efficiency of their execution. Further, the development of smart contracts has been recognized to be quite critical from the security point of view [46], in the sense that it is hard to code safe smart contracts.

Given this difficulty and the fact that smart contracts may control large amounts of tokens (i.e., money), they are among the preferred targets of hacking activities.

As an example, the Helium project encompasses an ad hoc blockchain that does not support smart contracts. Its very specific functionalities are hardcoded in the helium software.

7. Consensus Mechanisms Based on Physical Properties

While this content is mostly focused on the advantages that blockchain can provide to IoT ecosystems, there is also an interesting advantage in the opposite direction. In fact, in an unpermissioned blockchain, the way in which the consensus on the next block is achieved is extremely critical for the security of the whole system. The main problem is that a simple vote-based approach is insecure. In fact, for an attacker, it is easy to emulate a large number of nodes (an approach known as *Sybil attack*) to obtain the majority in a decision. For this reason, it has to require some effort to participate in the consensus. In regular blockchains, the most famous approaches to this problem are the so-called *proof-of-work*, in which participants have to prove that they have solved a cryptographic puzzle, and *proof-of-stake*, in which participants have to prove that have staked (i.e., frozen for a certain amount of time) a certain amount of tokens.

A special-pupose blockchain in an IoT ecosystem can take advantage of the physical existence of IoT devices to obtain a high level of security while asking participants to perform some work that is useful for the ecosystem. For example, in Helium [7], consensus security is based on a so-called *proof-of-coverage*. In this approach, participants regularly challenge *hot-spots* to assess their coverage of a certain area. This kind of work cannot be easily scaled programmatically, since physical presence near the hot-spot is required. At the same time, this monitoring activity is reported to the users as valuable information about areas covered by the Helium network [47].

Certain constraints or tasks that are available in an IoT ecosystem can be used to create special-purpose consensus mechanisms. This is an aspect that is underutilized. For example, SolarCoin [48] encompasses the concept of *verified energy production*; however, this concept is not exploited for consensus.

Other approaches based on physical properties were proposed in the literature and are candidates to be used in IoT ecosystems; see, for example, [49][50] and the surveys [51][52].

References

1. On the Money: Earn as You Drive with Jaguar Land Rover. 2019. Available online: https://www.jaguarlandrover.com/news/2019/04/money-earn-you-drive-jaguar-land-rover (accessed on 16 November 2021).
2. ElaadNL Develops Autonomous Self-Balancing Power Grid Using IOTA. 2019. Available online: https://blog.iota.org/elaadnl-develops-autonomous-self-balancing-power-grid-using-iota-de52e9638548/ (accessed on 16 November 2021).
3. IoTA Marketplace. Available online: https://data.iota.org/#/ (accessed on 29 October 2021).
4. Huberman, G.; Leshno, J.D.; Moallemi, C. Monopoly without a monopolist: An economic analysis of the bitcoin payment system. Rev. Econ. Stud. 2021, 88, 3011–3040.
5. Wilson, K.B.; Karg, A.; Ghaderi, H. Prospecting non-fungible tokens in the digital economy: Stakeholders and ecosystem, risk and opportunity. Bus. Horiz. 2021; in press.
6. Medicalchain. Whitepaper: Own Your Health. Available online: https://medicalchain.com/Medicalchain-Whitepaper-EN.pdf (accessed on 30 November 2021).
7. Haleem, A.; Allen, A.; Thompson, A.; Nijdam, M.; Garg, R. Helium Whitepaper: A Decentralized Wireless Network. 2021. Available online: http://whitepaper.helium.com/ (accessed on 30 November 2021).
8. Tamassia, R. Authenticated data structures. In Algorithms—ESA 2003; Springer: Berlin/Heidelberg, Germnay, 2003; Volume 2832, pp. 2–5.
9. Ray, P.P.; Kumar, N.; Dash, D. BLWN: Blockchain-Based Lightweight Simplified Payment Verification in IoT-Assisted e-Healthcare. IEEE Syst. J. 2021, 15, 134–145.
10. Mina Protocol Overview. 2021. Available online: https://docs.minaprotocol.com/en (accessed on 9 November 2021).
11. Helium, People-Powered Networks. 2021. Available online: https://www.helium.com/ (accessed on 30 November 2021).
12. Käbisch, T. Verification of Bitcoin in the Incubed Protocol; Hochschule Mittweida: Mittweida, Germany, 2020.
13. Danzi, P.; Kalør, A.E.; Stefanović, Č.; Popovski, P. Delay and Communication Tradeoffs for Blockchain Systems with Lightweight IoT Clients. IEEE Internet Things J. 2019, 6, 2354–2365.

14. Le, T.; Mutka, M.W. A lightweight block validation method for resource-constrained iot devices in blockchain-based applications. In Proceedings of the 2019 IEEE 20th International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM), Washington, DC, USA, 10–12 June 2019; pp. 1–9.
15. Valaštín, V.; Košťál, K.; Bencel, R.; Kotuliak, I. Blockchain based car-sharing platform. In Proceedings of the 2019 International Symposium ELMAR, Zadar, Croatia, 23–25 September 2019; pp. 5–8.
16. Zhou, Q.; Yang, Z.; Zhang, K.; Zheng, K.; Liu, J. A decentralized car-sharing control scheme based on smart contract in internet-of-vehicles. In Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 25–28 May 2020; pp. 1–5.
17. Why Can’t Contracts Make API Calls? Available online: https://ethereum.stackexchange.com/questions/301/why-cant-contracts-make-api-calls/334#334 (accessed on 9 March 2022).
18. Blockchain Oracles for Hybrid Smart Contracts|Chainlink. Available online: https://chain.link/ (accessed on 18 January 2022).
19. Provable–Blockchain Oracle Service, Enabling Data-Rich Smart Contracts. 2019. Available online: https://provable.xyz (accessed on 26 January 2022).
20. Band Protocol—Cross-Chain Data Oracle. 2022. Available online: https://bandprotocol.com/bandchain (accessed on 26 January 2022).
21. Tellor. Available online: https://tellor.io/ (accessed on 18 January 2022).
22. Bernardini, M.; Pennino, D.; Pizzonia, M. Blockchains meet distributed hash tables: Decoupling validation from state storage. In Proceedings of the Second Distributed Ledger Technology Workshop, 2019, Pisa, Italy, 12 February 2019; Volume 2334, pp. 43–55.
23. Leung, D.; Suhl, A.; Gilad, Y.; Zeldovich, N. Vault: Fast bootstrapping for the Algorand Cryptocurrency. In Proceedings of the Network and Distributed Systems Security (NDSS) Symposium 2019, San Diego, CA, USA, 24–27 February 2019.
24. Zhou, Q.; Huang, H.; Zheng, Z.; Bian, J. Solutions to Scalability of Blockchain: A Survey. IEEE Access 2020, 8, 16440–16455.
25. Swan, M. Blockchain economic networks: Economic network theory—Systemic risk and blockchain technology. In Business Transformation through Blockchain; Springer: Cham, Switzerlabd, 2019; pp. 3–45.
26. Eosio Documentation. 2021. Available online: https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md (accessed on 20 December 2021).

27. Nano. Eco-Friendly and Feeless Digital Currency. Available online: https://nano.org/ (accessed on 17 January 2022).
28. Popov, S. The Tangle White Paper. 2018. Available online: http://www.descryptions.com/Iota.pdf (accessed on 28 March 2022).
29. Neo. Whitepaper. Available online: https://docs.neo.org/v2/docs/en-us/basic/whitepaper.html (accessed on 17 January 2022).
30. Singh, A.; Click, K.; Parizi, R.M.; Zhang, Q.; Dehghantanha, A.; Choo, K.K.R. Sidechain Technologies in Blockchain Networks: An Examination and State-of-the-Art Review. J. Netw. Comput. Appl. 2020, 149, 102471.
31. Inter-Blockchain Communication. Available online: https://ibcprotocol.org/ (accessed on 17 January 2022).
32. Zhao, D. Cross-blockchain transactions. In Proceedings of the Conference on Innovative Data Systems Research (CIDR), Amsterdam, The Netherlands, 12–15 January 2020.
33. Qasse, I.A.; Abu Talib, M.; Nasir, Q. Inter blockchain communication: A survey. In Proceedings of the ArabWIC 6th Annual International Conference Research Track, Rabat, Morocco, 7–9 March 2019; pp. 1–6.
34. Kwon, J.; Buchman, E. Cosmos Whitepaper. Available online: https://cosmos.network/resources/whitepaper (accessed on 17 January 2022).
35. Schulte, S.; Sigwart, M.; Frauenthaler, P.; Borkowski, M. Towards blockchain interoperability. In Proceedings of the International Conference on Business Process Management, Vienna, Austria, 1–6 September 2019; pp. 3–10.
36. Sagirlar, G.; Carminati, B.; Ferrari, E.; Sheehan, J.D.; Ragnoli, E. Hybrid-Iot: Hybrid blockchain architecture for Internet of Things-Pow Sub-Blockchains. In Proceedings of the 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), Halifax, NS, Canada, 30 July–3 August 2018; pp. 1007–1016.
37. Jiang, Y.; Wang, C.; Wang, Y.; Gao, L. A Cross-Chain Solution to Integrating Multiple Blockchains for IoT Data Management. Sensors 2019, 19, 2042.
38. Li, M.; Tang, H.; Hussein, A.R.; Wang, X. A Sidechain-Based Decentralized Authentication Scheme via Optimized Two-Way Peg Protocol for Smart Community. IEEE Open J. Commun. Soc. 2020, 1, 282–292.
39. Ngubo, C.E.; McBurney, P.J.; Dohler, M. Blockchain, IoT and sidechains. In Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, China, 14–16 March 2019.

40. Poon, J.; Dryja, T. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. 2016. Available online: https://www.bitcoinlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf (accessed on 28 March 2022).
41. Antonopoulos, A.M. Mastering Bitcoin: Programming the Open Blockchain; O'Reilly Media, Inc.: Newton, MA, USA, 2017.
42. Robert, J.; Kubler, S.; Ghatpande, S. Enhanced Lightning Network (off-chain)-based micropayment in IoT ecosystems. Future Gener. Comput. Syst. 2020, 112, 283–296.
43. Wood, G. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Proj. Yellow Pap. 2014, 151, 1–32.
44. Dannen, C. Introducing Ethereum and Solidity; Springer: Cham, Switzerland, 2017; Volume 318.
45. Ethereum Virtual Machine (EVM). Available online: https://ethdocs.org/en/latest/introduction/what-is-ethereum.html#ethereum-virtual-machine (accessed on 17 January 2022).
46. Atzei, N.; Bartoletti, M.; Cimoli, T. A survey of attacks on Ethereum smart contracts (Sok). In Proceedings of the International Conference on Principles of Security and Trust, Uppsala, Sweden, 22–29 April 2017; pp. 164–186.
47. Helium Explorer. Available online: https://explorer.helium.com/ (accessed on 18 January 2022).
48. Solarcoin. Whitepaper. Available online: https://www.allcryptowhitepapers.com/solarcoin-whitepaper/ (accessed on 29 October 2021).
49. Amoretti, M.; Brambilla, G.; Medioli, F.; Zanichelli, F. Blockchain-based proof of location. In Proceedings of the 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Lisbon, Portugal, 16–20 July 2018; pp. 146–153.
50. Boeira, F.; Asplund, M.; Barcellos, M.P. Vouch: A secure proof-of-location scheme for Vanets. In Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Montreal, QC, Canada, 28 October–2 November 2018; pp. 241–248.
51. Bodkhe, U.; Mehta, D.; Tanwar, S.; Bhattacharya, P.; Singh, P.K.; Hong, W.C. A Survey on Decentralized Consensus Mechanisms for Cyber Physical Systems. IEEE Access 2020, 8, 54371–54401.
52. Oyinloye, D.P.; Teh, J.S.; Jamil, N.; Alawida, M. Blockchain Consensus: An Overview of Alternative Protocols. Symmetry 2021, 13, 1363.