




Towards a Blockchain Architecture for Cultural Heritage Tokens

Aristidis G. Anagnostakis^(✉) 

University of Ioannina, 45110 Ioannina, Greece
arian@teiep.gr

Abstract. Disputes over Cultural Heritage tokens and collections claims among collectors, organized social groups, countries, ethnicities, even civilizations, are but uncommon over time. Universal ontologies such as the CIDOC Conceptual Reference Model (ISO 21127:2014) have emerged as global standards over the past years, to allow for seamless structuring and interchange Cultural Heritage artifacts documentation information, in spite of their actual nature (physical, intangible, digital, etc.).

Yet, no objective universal procedure exists to safeguard the originality of the records and the archives; the immunity of essential sensitive data of the documentation tokens (dates, places, owners, etc.) is still questionable.

Aiming toward a coherent, effective Blockchain architecture to establish an immune, objective, collective archive of the documented Cultural Heritage tokens, the present proposes an implementation based on a comparative analysis of the prominent Blockchain architectures.

Keywords: Blockchain use cases · Blockchain for cultural heritage · Hyperledger · Cultural Heritage Documentation · CIDOC blockchain

1 Introduction

Currently, Cultural Heritage data are being structured, processed and archived in libraries and services such as the Archaeology Data Service (ADS) [5]. Even the best structured and formatted data suffer from serious deficiencies; there is no objective way to verify that the data has not been altered over time, neither intentionally, nor as the result of an attack. There is no objective mechanism to safeguard the ownership of the data, leading in many times to misconceptions upon the ownership of the heritage token itself. Different “documentalists” may describe the same artifact subjectively. There is no objective way to merely verify the succession of the recordings in time!

Yet, the historian of the future should have no doubt that the “Physical, man-made thing” of subtype “building” [1] entitled “Acropolis”, was situated at the referred location known as “Athens” at the time of existence or at the time of the first recorded documentation, (*even this has been altered/vanished in the meantime).

At a minimum, an assimilated hash fingerprint of the XML structured document describing “Acropolis” (which has the form of a graph), has to be stored in an immune public index, allowing for verification of the original documentation. The actual content of the documentation may still relay under the copyright of the “documentalist”.

Blockchain can offer such an immutable public record for cultural heritage archives, upon which everyone will be able to verify at low to no cost the validity of the documentation process.

2 Methods

Blockchain is a series of public records, the sequence and the content of which is extremely hard to question, thus forming a “universally” accepted ledger. It may be stored in a central point or be distributed over the network. For a number of reasons, i.e. to robustly operate among non-trusting third parties, to avoid “single point of failure” vulnerability, etc. the distributed ledger has evolved rapidly over the past years [8–25].

One of the key operations of a distributed ledger mechanism is to ensure that the entire network collectively agrees on the contents of the ledger; the mechanism is known as “consensus” mechanism. A consensus mechanism assures that each next block added to the chain, is actually representing the most recent link, thus preventing arbitrary infinite “forking”.

The state of arbitrary forking suggests that different copies of the initial Blockchain evolve over time; the issue is known as “*double spending*” or “*nothing at stake*” under the context of the *Proof of Work (PoW)* and *Proof of Stake (PoS)* methodologies respectively.

The most popular and validated Blockchain consensus method is the Proof of Work (PoW). The Proof of Stake (PoS) is emerging to tackle specific inefficiencies of the PoW [3–8].

2.1 Consensus via Transparent Proof of Work

Transparent Proof of Work (PoW) as a consensus mechanism was first published by Dwork and Naor in 1993 [9], however, it wasn’t until 1999 the actual term “Proof of Work” was coined by Jakobsson et al. [10]. Irrespective of the implementation variances, the PoW has some major properties:

- There is a Prover (i.e. the “minter-miner”) and a Verifier; the Prover solves a mathematical puzzle at the work-cost w , and the Verifier verifies the puzzle is correctly solved at the cost of z .
- The puzzles are asymmetric; it is difficult for Prover to solve but the solution is easily verified by a Verifier (one-trap-door). “A POW may be regarded as efficient if the Verifier performs substantially less computation than the Prover” Cai et al. [12], Such a proof is considered to have “large advantage” if $\frac{z}{w} \ll 1$, or if z is asymptotically lower than w (as in Jakobsson et al. [10]).
- The puzzles are statistically independent; no skills are involved, they require brute force. This ensures certain Provers do not gain an unfair advantage over others. The only way for a Prover to improve his odds of solving a puzzle is to acquire additional computational power (more w at the solution time interval (ts, tc)); something that is very energy and capital-demanding. The only way to overpower the network strength of Blockchain networks is through a “51% attack” [11].

- The puzzle parameters are periodically updated in order to keep the next-block time consistent (lower bound the [ts,tc] interval, thus limiting the occurrence of inflation). As an example, the Bitcoin protocol [11] sets the desired “*block generation time*” to 10 min. If the average “*block generation time*” gets below this interval during a specific timeframe, the network automatically increases the difficulty of the puzzle (Fig. 1).

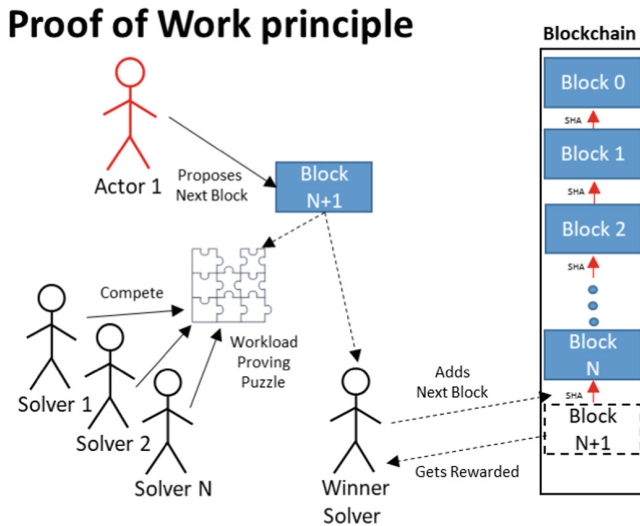


Fig. 1. Proof of work blockchain principle.

PoW has proved its’ validity; yet it suffers from deficiencies (i.e. 51% type of attack vulnerability, increased latency, high energy consumption) that makes it hard to apply generally.

2.2 Consensus via Proof of Stake

Proof of Stake, is currently gaining popularity [8, 14, 24, 25] as a consensus mechanism for validating Blockchain links and achieving consensus, bearing a number of comparative advantages over the PoW, notably security, reduced risk of centralization, and energy efficiency [13].

Basic properties of the PoS include:

- Consensus finality: “if a valid block appends to a Blockchain at some point in time, it can never after be removed” [17]. PoW Blockchains does not satisfy consensus finality (see the “double spending” issue).
- Energy efficiency: In PoS, there is no mathematical puzzle, instead, the creator of a new block is chosen based on his stake, or even randomly.

- Reduced centralization risk: The “stake” policy on a PoS blockchain can be designed to provide “blind” equality to all nodes irrespective of their power.
- Penalization: *Greedy & malicious* behaviors and various forms of 51% attacks can be made unaffordably expensive; “it’s as though your ASIC farm burned down if you participated in a 51% attack” Vlad Zamfir [22].
- Node Identity: The BFT PoS approach to consensus typically requires each node to know the entire set of the nodes participating in the consensus process. PoS is in many cases closely bound to the notion of “Smart-contracts” [17] (Fig. 2).

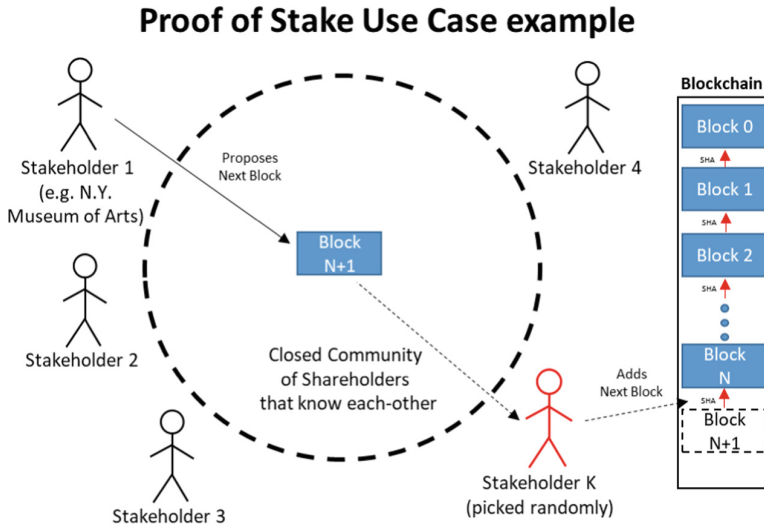


Fig. 2. An example of Proof of Stake consensus Blockchain.

2.3 Smart Contracts

Smart-contract, introduced in 1994 by Szabo [15] is a “computerized transaction protocol that executes the terms of a real-life contract”. The idea behind the Smart-contract is to establish an automated way to fulfill the demands of real-life contracts (e.g. payment terms, confidentiality, conditional execution), to minimize deliberate and accidental “exceptions” and consequently to minimize the need for *trusted intermediaries*.

“*Smart-contract use cases take the Blockchain well beyond its original cryptocurrency orientation, back to the domain of database replication protocols, notably, the classical state-machine replication*” [20]. A Smart-contract can be modeled as a state machine, executing consistently across multiple nodes in an interconnected environment, using *state-machine replication*. A family of such protocols of high interest for Blockchain is the Byzantine Fault Tolerant (BFT) *state-machine replication* protocols, which provide consensus regardless of the existence of malicious (Byzantine) nodes in the network.

Smart-contracts bound to a Blockchain are used as general purpose computations that take place on a Hyperledger. Robust, nearly Turing complete languages have been proposed (i.e. Ethereum Foundation [16], IBM [17], the Hyperledger Fabric [26]), facilitating effective real-life Smart-contract implementations.

2.4 Transparent “Proof of Work” vs BFT “Proof of Stake”

A high-level comparison PoW and the BFT PoS for some important Blockchain properties are summarized after [20] on Table 1:

Table 1. Comparison of important PoW and PoS BFT properties.

	PoW consensus	PoS BFT consensus
Node Identity	Unnecessary, open to anyone without the need of identity awareness	Necessary for every node in the network to know the identity of all nodes
Scalability/distribution	High. Supported and tested on thousands of nodes and millions of clients, e.g. in the Bitcoin Blockchain	Node scalability not extensively tested; proved in practice for thousands of clients
Performance	Poor response time and power consumption. PoW uses power as the valued asset, and in some implementations, specific time duration to down-limit it	High performance in response time and power consumption. PoS BFT introduces minimal latency and computational power need
Attack power vulnerability	25% of the total network power is needed to gain control of the chain Eyal and Sirer [23]	Varies on implementation; Dwork Lynch and Stockmeyer [24] suggest that generally $n/3$ malicious nodes are adequate to gain control
Network synchronization	Physical timestamps required to validate nodes	Not required; temporal synching is needed only to keep the network alive

Perhaps the most important architectural difference between the two summarizes in the fact that in PoS, each node has to be aware of all the node id’s participating in the network. This imposes a “controlled”, centralized group of nodes (i.e. documenting entities in the domain of discourse); to add a new node in the network an additional round of voting may be necessary, to limit the risk of introducing non-reliable members.

In addition, a mechanism to restrict the function of non-reliable nodes -should such exist over time- has to be foreseen; the nodes already in the Blockchain network may act as a central trusted party and re-configure the rules at any future time.

3 Choosing the Appropriate Consensus Model

Building a Hyperledger for Cultural Heritage documents poses some major domain-specific considerations:

1. Administrators are in majority public and non-profit organizations. This calls for transactions of low to no cost and proclaims PoS consensus.
2. Administrators bear a significant know-how in documenting Cultural Heritage artifacts. This suggests a finite, controlled network of nodes and pinpoints the need for membership rules.
3. The members safeguard the validity of the process. They thus may accept or discard a newcomer, acting together as an internal trusted party. The confidence of the network on each has to be constantly re-evaluated.
4. All members have the same “rights” in terms of functionality, yet the relative significance of each vary vastly among them.
5. Documentation data may become significantly large in volume; this is especially true when hi-resolution images, videos, sounds and 3D scanings are part of the documentation. Importing the master documentation in the Hyperledger is both impractical and of limited value. An assimilated fingerprint of the document (a hashed SHA version of the document) is proposed to be used instead.
6. Yet, for the Hyperledger to be of practical use to third parties, a minimal set of reference information to the initial artifact has to be contained to the Blocks in plain, (un-hashed) text.

Considerations 1, 2 and 4 indicate PoS BFT as the most appropriate consensus model (see Table 1). Considerations 3 and 4 stress the need for functionality agreement among all members; this is efficiently compromised by existing Smart-contracts frameworks, (i.e. the Hyperledger Fabric, Ethereum, etc.). Considerations 5 and 6 pinpoint the need for customized software agents to bridge current documentations DBMS's to the Blockchain.

In addition, for a Blockchain architecture to be suitable for the domain of discourse:

- It has to be distributed; cultural heritage being public in notion and essence, its' documentation should be decentralized as well, to address the “single point of failure” issues and distribute trust and liability among all involving actors.
- It has to be fair; no member should be able to gain control of the process over the rest, in spite of his computational power or stake, especially without the explicit approval of the rest.
- It has to be documentation method-and-content agnostic; minimal id data of the imported tokens, (e.g. “Title” or “Id”) are desired but not mandatory, and shall exist solely to facilitate third parties' quests in the future. The documenting authority shall fully preserve content and rights and prove these undoubtedly via the chain.

4 Proposed Architecture

4.1 Structuring the Content

Cultural artifacts documentation, irrespective of their actual storage format (flat file, E-R/OO DBMS, etc.) may, in the generic case be represented as structured XML [1, 2] (Fig. 3).

```

<CRM Entity (E1)>
- <Number (E60)>: #32142
- <Temporal Entity (E2)>
...
- <Beginning of existence (E63)>:
  - <Time primitive(E59)>: "130 BC"
...
- <Persistent Item (77)>
...
- <Physical Man-Made Thing (E24)>:
  "Aphrodite of Milos"
...
- <Place (E53)>: "Milos Island, Greece"
  - <Spatial Coordinate(E47)>: "36° 44' 35"
    N, 24° 25' 28' W"
...
  
```

Fig. 3. CIDOC XML documentation excerpt example.

The volume of the documentation being considerably high, block chaining the master documentation becomes evidently impractical; instead, a signed, assimilated fingerprint is used. The fingerprint bears and verifies the validity of the original, which may by no means be altered over time without being noticed [18] (Fig. 4).

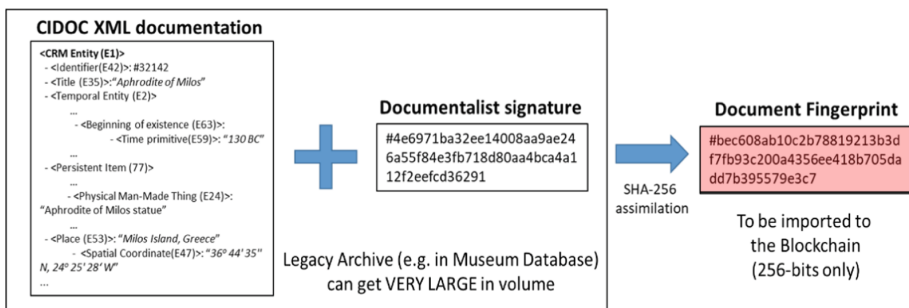


Fig. 4. Fingerprinting the Cultural Heritage artifact documentation.

The Blockchain will act as an immune distributed ledger of cultural heritage tokens’ documentation. Each items’ *Identifier* and *Title* (CIDOC E42 and E35 respectively) is also proposed to be added to the Block un-hashed, to allow for back-reference by third clients (Fig. 5).

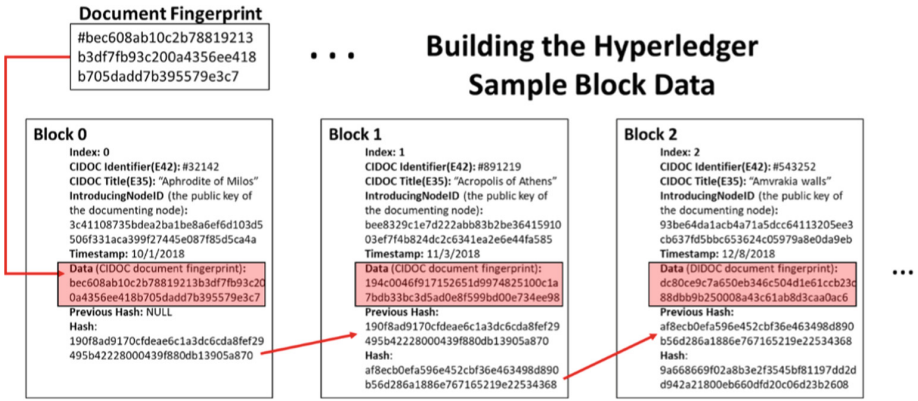


Fig. 5. Building the chain: putting essential data to blocks.

4.2 Forming the Contract

Let the universe of all cultural heritage artifacts $\{1, N\}$, and the universe of all documented artifacts $\{1, M\}$, ($M, N \in \mathbb{N}$). The Documentation may be modeled as a one-way function $\{1, M\} \xrightarrow{D_A} \{1, N\}$ carried out by member $i \in \{1, K\}$, where K is the number of active members in the Blockchain network.

The following contract is proposed:

- Each document has a specific Document Value of 1 DV, in a way analogous to Ethereum Wei [25]¹.
- The first member to put records in the Blockchain is assigned a stake value of TDV_M (*Total Documents Value*) representing the value of all documents in the chain².
- Voting is needed only to add a new member, and to re-validate memberships.
- The voting power of member i is $\frac{DV_i}{TDV_M}$.
- A voting takes place among the members for a new-comer to join. Anyone (physical or legal entity) can join the network with the consent of the members holding $\frac{2}{3} TDV$ votes. This consent is achieved by a special “voting” round among the members, upon the request of a newcomer to put a block in the chain.

¹ The valued asset is the single documentation. All documents are considered of having the same value.

² Each document corresponds to a single Block. The voting power of each member is proportional to the number of Blocks it has proposed and have been successfully imported to the Hyperledger.

- All members are considered *trusted trustees* and *consent* to add a Block to the chain upon the *request* of any other member.
- A special round of “re-validation” between the members is executed, should any member tend to reach the critical limit ($1/3$ of the votes) i.e. $DV_i \leq \frac{1}{3}TDV_M - l, \forall i \in [1, K]$, where $l (> 0)$ is the “safety” limit, preventing a member to acquire potential control of the chain without the explicit consent of the rest. The member questioned each time does not participate in the voting.
- The “re-validation” cycle also takes place if a member is marked as “*unavailable*”³.
- Upon exclusion, the stakes of the rejected member DV_r are assigned to the rest proportionally to their stakes (i.e. $DV_r * \frac{DV_i}{TDV_M}$).
- Each member i is chosen to add a block to the chain explicitly, with probability $\frac{DV_i}{TDV_M}$ in a pseudorandom way. Upon unavailability, another node is chosen following the same rule.

The notion of “rewarding” in this type of Hyperledger is unnecessary. Increasing the length of the chain over time, ($M \rightarrow N$) the real value of it is expected to increase proportionally; voting power is introduced as a “rewarding” mechanism on the long term⁴.

5 Challenges

The proposed Blockchain architecture presents attractive features in terms of fairness, efficiency, computational load and response time.

New Blocks are imported at $O(1)$ cost, while new members are added at the cost of $O(K)$. Periodic re-validation of the memberships requires $O((K - 1)^2)$, distributed over the network.

Even though recent PoS implementations suggest they easily overwhelm the criteria set, choosing the most suitable is a major challenge. *Hyperledger Fabric* [26] and *Ethereum* [14] are the most prominent, yet their attributes remains to be proved in practice. The proposed architecture is subject to further testing and development.

6 Conclusion

This study illustrates the prospective of introducing Hyperledger to the vast repository of Cultural Heritage documentation tokens. It reveals the potential, identifies the core features, and sets the requirements for a universal Hyperledger in the domain of discourse. It proposes a solid architecture and elaborates an initial performance evaluation.

³ Has failed to add a block to the chain, even if he was chosen to, in more than $n\%$ of the calls.

⁴ It also safeguards the smooth operation of the chain, since the largest members are expected to present higher availability.

Blockchain shall not resolve disputes by itself. The validity of the documentation shall always rely on the validity of the “documentalist”, yet the community is empowered with the means to constantly re-evaluate the trustworthiness of its’ members.

Introducing Blockchain in Cultural Heritage documentation shall provide a price-less, universal, immune time sequence of the Cultural Heritage documentation records (i.e. ISO 21127:2014), safeguarding the validity of the data, the value of the tokens, and facilitating in the most effective way the resolution of ownership and valorization issues for the generations to come.

References

1. International council of museums, International committee for documentation official site, definition of the CIDOC Conceptual Reference Model. <http://www.cidoc-crm.org/>. Accessed 12 May 2018
2. ISO standards official homepage - ISO 21127:2014 A reference ontology for the interchange of cultural heritage information. <https://www.iso.org/standard/57832.html>. Accessed 28 Aug 2018
3. Web archives homepage, Hal Finney Reusable Proofs of Work. <https://web.archive.org/web/20071222072154/http://rpow.net/>. Accessed 28 Mar 2010
4. Hackernoon tech BlogSpot, Decentralized Objective Consensus without Proof-of-Work, C.V. Alkan. <https://hackernoon.com/decentralized-objective-consensus-without-proof-of-work-a983a0489f0a>. Accessed 5 Feb 2017
5. Archaeology Data Service UK official site. <http://archaeologydataservice.ac.uk/>. Accessed 14 Aug 2018
6. Proof of stake vs Proof of work, Github Software Archive. <https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQs#what-is-proof-of-stake>. Accessed 10 June 2018
7. Miller, A., LaViola Jr., J.J.: Anonymous Byzantine consensus from moderately-hard puzzles: a model for Bitcoin. Technical report, University of Central Florida, CS-TR-14-01, Florida (2014)
8. Proof of work vs proof of stake Turner Schumann, Hackernoon tech blogspot. <https://hackernoon.com/consensus-mechanisms-explained-pow-vs-pos-89951c66ae10>. Accessed 5 Apr 2018
9. Dwork, C., Naor, M.: Pricing via processing or combatting junk mail. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 139–147. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-48071-4_10
10. Jakobsson, M., Juels, A.: Proofs of work and bread pudding protocols (Extended abstract). In: Preneel, B. (ed.) Secure Information Networks. IFIP AICT, vol. 23, pp. 258–272. Springer, Boston, MA (1999). https://doi.org/10.1007/978-0-387-35568-9_18
11. Bitcoin official homepage, A Peer-to-Peer Electronic Cash System white paper. <https://bitcoin.org/bitcoin.pdf>. Accessed 14 Aug 2018
12. Cai, J., Lipton, R., Sedgewick, R., Yao, A.: Towards uncheatable benchmarks. In: IEEE Structures, pp. 2–11 (1993)
13. Ethereum official blog, Vitalik Buterin p-epsilon attack. <https://blog.ethereum.org/2015/01/28/p-epsilon-attack/>. Accessed 10 Aug 2018
14. Ethereum’s Casper protocol explained in simple terms. <https://www.finder.com/ethereum-casper>. Accessed 12 July 2018

15. University of Amsterdam on-line courses, Smart contracts definition (Nick Szabo 1994). <http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html>. Accessed 4 July 2018
16. The Ethereum official homepage. <https://www.ethereum.org/>. Buterin, Vitalik. “Ethereum Whitepaper”. Accessed 1 June 2017
17. IBM Research – Zurich official homepage, Christian Cachin “Architecture of the Hyperledger Blockchain Fabric”. https://www.zurich.ibm.com/dclcl/papers/cachin_dccl.pdf. Accessed 1 Oct 2016
18. Vukolić, M.: The quest for scalable blockchain fabric: proof-of-work vs. BFT replication. In: Camenisch, J., Kesdoğan, D. (eds.) iNetSec 2015. LNCS, vol. 9591, pp. 112–125. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39028-4_9
19. Goldreich, O.: Foundations of Cryptography I: Basic Tools. Cambridge University Press, Cambridge (2001). ISBN 978-0-511-54689-1
20. Schneider, F.B.: Implementing fault-tolerant services using the state machine approach: a tutorial. ACM Comput. Surv. **22**(4), 299–319 (1990)
21. Lamport, L., Shostak, R., Pease, M.: The Byzantine generals problem. ACM Trans. Program. Lang. Syst. **4**, 382–401 (1982)
22. Medium Blogspot Vlad Zamfir “Simple model of an internal PoW attacker”. https://medium.com/@Vlad_Zamfir/simple-model-of-an-internal-pow-attacker-1a713cf00672. Accessed 2 May 2017
23. Eyal, I., Sirer, E.G.: Majority is not enough: Bitcoin mining is vulnerable. In: Christin, N., Safavi-Naini, R. (eds.) FC 2014. LNCS, vol. 8437, pp. 436–454. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45472-5_28
24. Dwork, C., Lynch, N., Stockmeyer, L.: Consensus in the presence of partial synchrony. J. ACM **35**, 288–323 (1988)
25. Wood, G.: Ethereum: A Secure Decentralised Generalised Transaction Ledger Byzantium Version e94ebda - 2018-06-05. The Ethereum yellow paper homepage. <https://ethereum.github.io/yellowpaper/paper.pdf>. Accessed 8 July 2018
26. The Hyperledger Fabric platform official homepage. <https://www.hyperledger.org/projects/fabric>. Accessed 14 Aug 2018