State of the Art and Challenges Facing Consensus Protocols on Blockchain

Nutthakorn Chalaemwongwan
Information Science and Technology
Mahanakorn University of Technology
Bangkok, Thailand
nutthakorn.c@monsterconnect.co.th

Werasak Kurutach
Information Science and Technology
Mahanakorn University of Technology
Bangkok, Thailand
werasak@mut.ac.th

***Abstract*—Nowadays, the blockchain is a favorite platform, for instance, for use with cryptocurrency, smart contracts, IoT, and so forth. Blockchains are distributed ledgers that enable parties who do not trust each other to maintain states. The parties agree on the existence, values, and histories of the states. The blockchain applies the consensus protocol to verify the block which is distributing the network node. Consensus has many practices such as, for example, Byzantine general problem, Proof of work, and Proof of stake. However, there have not been numerous papers that have undertaken analysis with regard to various aspects, and which have incorporated the adoption summary as appropriate to the application. In this paper, we have provided a technical verification review with regard to the consensus algorithm taken from previous research. The audience will receive a consensus protocol, and the algorithm typically analyzes the application to match which platform is appropriate by viewing the node identity, energy saving and the tolerated power of the adversary, the data model, language, execution, application, and examples. Finally, we finish by presenting several research directions with regard to the consensus protocol.**

Keywords: blockchain; consensus; distributed ledger; distributed consensus; consensus algorithms

I. INTRODUCTION

A ledger is a documentary framework which includes an ordered record of transactions. For instance, a ledger will record transactions between several commercial banks, or goods exchanged with known parties. An aspect of blockchains is that the ledger is replicated entirely on almost all the nodes. Moreover, transactions are assembled into blocks, then chained to one another. Therefore, the distributed ledger is a replicated attach only data framework. A blockchain begins with the initial states, while the ledger records all the history of the revision procedure.

A technique encouraging distributed ledgers could classify them in terms of three aspects. (Account, Asset, and Coin). Initially, the application created on the top of the ledger establishes the information entirely in such a way as to keep within the ledger A cryptocurrency application will use to create a user account similar to that found in conventional banking. The intention is that it is simple to build and apply a lower level model such as an example table or a critical value as mentioned. Second, the functional system might have one or higher ledgers which can link to one another. A large enterprise, for instance, may have multiple ledgers, one for every department such as engineering, sales, supply chain, payroll, and so forth. Finally, ledger possession can fluctuate significantly from being exposed entirely to the public to being totally managed by one participant. Bitcoin, for instance, was presented and, as a result, makes people think that it is costly to determine who will update the public ledger. Parity [1] predefines some owners who adjust the public ledger simply by adding their signature to the blocks.

The information contained in the ledger demonstrates the history of transactions up to the current time through the blockchain. Updates to the ledger should be consented to by every participant. Specifically, several participants need to agree with a consensus. Observing which it is may not be the situation found in countless real-world applications such as a fiat digital currency, by which a single entity (e.g., a bank or a government) decides on the updates.

The importance of a blockchain strategy is actually being able to deal with those nodes that do not trust working together, indicating which variables might respond in Byzantine formality. The consensus protocol consequently needs to endure Byzantine problems. The research involving a survey on consensus was comprehensive, and there are numerous versions of previously recommended protocols that have been manufactured for blockchains [2]. These could be categorized in terms of their combined range. Overwhelming consists of solely computation-based protocols in which the users utilize proof of computation to find a node that arbitrarily determines the next operation. Bitcoin's Proof of work (PoW) is a simple example of the consensus. The other severe are solely communication-based protocols for which nodes have matching votes and run through several rounds of communication to achieve consensus. All protocols, Practical Byzantine Fault Tolerance (PBFT) [3] being the best illustration, are applied in particular configurations because they take authenticated nodes. In the middle, all opposites we find mixed consensus protocols that intend to increase the efficiency of PBFT and PoW. As an example, Proof of Elapsed Time removes the need for costly mining at PoW with the use of profiting reliable devices. The other illustration is

Proof of Authority [4] which helps PBFT by pre-finding trusted nodes in which users vote on their own in order to achieve consensus. Likewise, Stellar [5] and Ripple [6] augment PBFT by achieving consensus in smaller networks.

In the next section, we provide related work associated with consensus protocol. Part III explains consensus protocol comparisons which are used in Section IV for analysis purposes. Section V offers a conclusion with regard to the survey of consensus protocols, and section VI deals with future research directions.

II. Related Work

A. CAP Theorem

CAP Theorem [7] is a system that distributes multiple nodes that handle common data, and shows which distributed systems can completely satisfy the following three properties: Consistency (C) is a status by which each node has the latest data at the same time. Availability (A) is the status where any failure of a specific node does not affect other nodes. Partition tolerance (P) is the status by which nodes can continue operations, even with a failure in the network.

The blockchain is one of the distributed systems which can satisfy both availability (A) and partition tolerance (P) but cannot satisfy the condition of consistency (C). Instead, as long as the separation of the network is time-limited, the system is regarded as maintaining the ultimate consistency. Ultimate consistency is the strategy that aims to discover if it is acceptable if the texture is eventually preserved, even with some time lags.

B. Byzantine Generals Problem (BGP)

Blockchain has fixed the issue that permits the whole network to validate transactions to ensure that double spending is detected. Whether a participant confirms the presence and authenticity of the transaction or not, Bob must accept it. The approach resembles the so-called BGP [8][9].

This is related to the challenge of tolerating faults in a distributed environment. With the BGP in mind, the question of false identities arises by which an adversary could mount a Sybil attack. Like Alice, the user might create multiple instances, each confirming the transaction. Despite the reality, this is a double spend. Bob will trust these, then accept the transaction.

C. Practical Byzantine Fault Tolerance (PBFT)

PBFT [10] is an algorithm for solving a Byzantine Fault resulting from a failure in building a consensus caused by the BGP. It was considered to be a serious challenge to put the algorithm to practical use due to the enormous amount of calculation required.

Nevertheless, the entire amount of nodes need be known, and the maximum amount of illegal nodes needs to be set. These requirements make it hard to employ this algorithm with regard to public systems. PBF is currently used in Ripple and Stellar.

D. Proof of Work (PoW)

PoW [11] refers to a mechanism to confirm a person's innocence by having them do a sure work which is troublesome but straightforward. Once they have done this work, it can be early verified.

For example, the PoW algorithm entitled Hashcash can be adapted for sending emails. A specific hash calculation is necessary each time an email is sent, thereby excluding spammers.

Bitcoin uses PoW to produce a mechanism which prevents the misrepresentation of data and duplicate payments without the need for a central authority, and which can preserve the system against attacks by malicious users.

E. Proof of Stake (PoS)

PoS [12] is an alternative mechanism for PoW which selects the next mining node based on their holding of the native digital currency of the blockchain network.

For example, the miners need to prove the ownership of a certain amount of peer coin currency in order to mine blocks.

PoW [13] mining is very costly. The procedure is especially power intense. PoS is significantly projected to minimize the expense of mining. PoS preserves a single branch, and yet transforms the puzzle's trouble to be inversely relative to the miner's stake in the network. A stake is ultimately a secured membership alongside a particular stability, addressing the miner's dedication to maintaining the network. This enables the feature which comes back the stake. Subsequently, a miner could produce the latest block by resolving the puzzle.

F. Proof of Activity (PoA)

PoS has many more restrictions that ultimately depend on the coinage which could involve keeping coins in a coin stake transaction to oneself.

Coins sent to standard transactions assigning coins to rest further ruin the coinage, but are not included in the PoS. However, the main weakness is that coinage collects whenever the node does not link to the network. The node is sufficient whenever nodes appear online sometimes and delay for their incentive, exclusively to go offline again subsequently.

The way it works does result in bursts of incentive distributions compared to the situation wherein nodes continuously continue online [14]. Nevertheless, the stakeholders probably do accept the outcome. An insufficient amount of online nodes, however, facilitates attacks.

G. Proof of Publication (PoP)

A time-stamping service provides timestamps with regard to digital records which firmly maintains monitoring of the

creation and modification time of the record. There are a number of various time stamping techniques, including PKI based centralized services. These records and timestamps are hashed and secured through a private key server. Nonetheless, the servers could backdate records perfectly while hashing and signing past timestamps [15].

To tackle this issue, there have been developed so-called linked timestamps. Every time stamp certificate contains a previous hash block. This assures a complete order of records even in the case of incorrect clocks. Moreover, it offers hardened fake certificates which are not feasible when it comes to retroactively connecting records in a linked chain of time stamps.

H. Proof of Burn (PoB)

PoB [16] is a very well-known and viable tool for migration. Natural protocol changes such as invalidating a previous transaction may incorporate a soft fork but commonly continue backward compatible Former clients might agree to accept transactions that are regarded as being incorrect via new applications releases.

To implement the changes, most of the miners should upgrade. Miners might use computing power by creating blocks which might be rejected by other nodes. Alternatively, miners handle to oppose resistant upgrades providing the most declines.

I. Proof of Retrievability (PoR)

POR [17] is similar to the proof of burn but increases the ability to provide provable commitment protocols by considering bandwidth and retrievability that can be used to repay expenses as it proposes to decentralized file storage.

J. Proof of Elapsed Time (PoET)

PoET [18] [19] replaces PoW as a more efficient protocol. Exclusively, PoET operates in a protected enclave. This is initiated by generating a block and input, creating a random duration time. Later, this could be used to build certificates, signifying how far duration maintains transferred when the timer begins.

A node which PoET produces a few times could append the block, once the timer expires. For example, the node could connect their PoET certificate to the block, additionally providing that time was time shorter than what created with any other node the block is approved.

K. Proof of Capacity (PoC)

PoC [20] [21] uses reliable equipment to decrease the number of replicas required to tolerate failure. The N-node network can now tolerate up to N=2 adversarial nodes, as opposed to N=3 adversarial nodes in the original PBFT.

L. Proof of Importance (PoI)

PoI [7] uses a method that clusters nodes through transaction graph analysis, utilizing the transaction quantities and the balances of individual nodes as indicators, determining the importance of each node and designating the priority using hash computations to more significant nodes.

Clustering is expected to make it feasible to detect nodes which are possibly attempting to commit illegal transactions.

M. Proof of Ownership (PoO)

PoO [22] is considered with regard to various goods and rights in terms of receiving services (ownership, right to use, and so forth). It is not limited to value-related information.

N. Proof of Space (PoSp)

PoSp [23] is recovering decentralization in mining node available and limiting the wastage of computational work in the system.

O. Delegated Proof of Stake (DPOS)

DPOS [24] is a fast, excellent, decentralized, and convenient consensus model. DPOS utilizes the energy to stakeholder endorsement voting, in order to solve consensus problems in a reasonable and fashionable manner. All node free activities involve block timing and different transaction sizes. They can be updated through a chosen delegate. Collection block manufacturers enable transactions to be verified in around 1 second. Potentially, above all, the consensus protocol was created to protect all individuals with regard to undesirable regulatory disturbances.

P. Ripple

Ripple [6] is real-time gross settlement system (RTGS) relating to currency transfers and remittance networks. It is usually known as Ripple Transaction Protocol (RTXP). Ripple was created after the spread of open source Internet protocols. Ripple enables security, is fast, and offers free worldwide monetary transactions with no chargebacks. This aids tokens that represent adjudication, cryptocurrency, commodities or a range of different benefits. Most importantly, Ripple was structured distributed public ledger of the kind that utilizes consensus procedures which permit payments, exchanges, and remittances.

Q. Tendermint

Tendermint [25] consists of applications involving the secure and regular replication of an application on multiple nodes. We secure which Tendermint operates 33.34% of nodes fail which all available nodes status the same transaction record and computes the same state. Secure and constant replication was an underlying issue on distributed networks. This technic performs an essential function in terms of fault tolerance.

III. COMPARISON

In this section, we have compared blockchain consensus protocols concerning a collection of fundamental blockchain characteristics. These types of characteristics incorporate node identity, energy saving, tolerated power of adversary, data model, language, execution, application, example and power of

the adversary, network synchrony assumptions, and, last but not least, the existence of correctness proofs of protocols underlying blockchain.

This set of properties is probably not exhaustive, but we believe it is representative for comparing all consensus protocols. In the rest of this section, we propose Table I. in more detail.

IV. ANALYSIS

A. *Consensus algorithms comparison*

Various consensus algorithms have different strengths and drawbacks. Table I. brings an assessment around various consensus algorithms, and we use the properties considering following.

Algorithm	Node Identity	Energy Saving	Tolerated power of adversary	Data model	Language	Execution	Application	Example
CAP [7]	Private	No	50% Rate should be	Key-Value	SQL, Python, Java, Go	SQL, No SQL	Database, Big Data, Storage	MongoDB, Cassandra, MS Azure Storage
BGP [8,9]	Private	Yes	33.3% Fail	Key-Value	Any	N/A	General Applications	General
PBFT [10]	Private	Yes	33.3% Faulty Replicas	Key-Value	Golang, Java	Dockers	General Applications	Hyperledger
PoW [11]	Public	No	25% Computing Power	Transaction-Based, Account-Based	Golang, C++, Solidity, Serpent, LLL	Native, EVM	Crypto-currency, General Applications	Bitcoin, Litecoin, ZCash, Ethereum
PoS [12,13]	Public	Partial	50% Stake	Account-Based	Michaleson	Native	Michaleson Applications	Peercoin, Tezos, Tendermint
PoA [14]	Public	Partial	50% of online stake	Account-Based	Solidity, Java, Python	EVM, Dockers	Crypto-currency	Parity
PoP [15]	Private	No	25% Computing Power	Transaction-Based, Account-Based	Golang, C++, Solidity, Serpent, LLL	Native, EVM	Crypto-currency, General Applications	Bitcoin
PoB [16]	Public	No	25% Computing Power	Transaction-Based, Account-Based	Golang, C++, Solidity, Serpent, LLL	Native, EVM	Crypto-currency, General Applications	Slimcoin
PoR [17]	Public	Yes	25% Computing Power	Transaction-Based, Account-Based	Golang, C++, Solidity, Serpent, LLL	Native, EVM	Outsource Storage	File Share
PoET [18,19]	Public	Yes	Unknown	Key-Value	Python	Native	General Applications	Sawtooth Lake
PoC [20,21]	Public	No	Unknown	Key-Value	Unknown	Unknown	Outsource Storage	File Share
PoI [7]	Public, Private	Yes	Unknown	Transaction-Based, Account-Based	Java	NEM	Blockchain Platform	XEM
PoO [22]	Public, Private	No	Unknown	Account-Based	Any	C#	Data Certification	Stamper
PoSpc [23]	Public	No	25% Computing Power	Transaction-Based, Account-Based	Golang, C++, Solidity, Serpent, LLL	Native, EVM	Mail, Reputation Services, Storage	Anti-spam
DPOS [24]	Public	Partial	< 51% Validators	Transaction-Based, Account-Based	No Scripting	Native	Decentralized Exchange	Bitshares
Ripple [25]	Public	Yes	< 20% Faulty Nodes in UNL	Account-Based	Java, Go, C++	Node.js, NPM	Digital Assets, Payment	Ripple
Tendermint [26]	Private	Yes	< 33.3% Byzantine Voting Power	Account-Based	Solidity	EVM	General Applications	Tendermint, Monax

Table I. Comparison of blockchain consensus protocols for a set of essential blockchain properties.

1) *Node Identity:* needs to understand the identity of every miner to select a main in all circular network. However, Tendermint, and PoP need to understand the validators in order to be able to pick a proposer in every round. With regard to PoW, PoS, DPOS, and Ripple, nodes might be associated with the network.

2) *Energy Saving:* Inside PoW, miners hash the block header continuously to accomplish the preferred worth. As a result, the amount of electrical energy needed to process the work is immense. With regard to PoS and DPOS, miners are regularly required to hash the block header to search for benefits, but the work generally decreases as the search place is produced to be restricted. Regarding PBFT, Ripple, and Tendermint, there is no mining in terms of consensus strategy. Therefore it saves power dramatically.
3) *Tolerated power of adversary:* Commonly 51% of hash power is understood while the threshold to one to acquire handle of the network. Nevertheless, the selfish mining strategy in PoW practices may assist miners to achieve more earnings while using only 25% of the hashing power. PBFT and Tendermint are manufactured to manage up to 33.34 defective nodes. Ripple is shown to maintain correctness when the proportion of faulty nodes in a unique node list are lower than 20%.
4) *Data model:* a data model is a transaction that focuses on assets. All systems require particular configurations, with numerous organizations being able to spin up a network to exchange assets with each other. These organizations are the ledger owners. It is typical for them to have more than one ledger involving all of them. Stellar, Ripple and IOTA issue their own tokens assets and provide their ledgers as a method of exchange, or as a platform for micropayments. IOTA, specifically, enables zero fee micropayments via its tokens, which creates worthwhile ledgers for transfers among IoT equipment. The ledgers in these systems adopt account-based data models. One ledger exists per open policy.
5) *Execution:* A smart contract pertains to a computation performed once a transaction has taken place. A smart contract could be regarded as a stored procedure invoked on a deal. The inputs, outputs, and status impacted by the execution of a smart contract obtain consent on all node.
6) *Application:* Going beyond cryptocurrency and asset management, some ledgers support running general, user-defined computations (or original contracts). Ethereum and its derivatives, namely Hydrachain, Quorum, Monax, Parity, and Definity let users write arbitrary business logic executed on top of the ledger. For example, Ethereum contracts range from simple crowdfunding campaigns to those involving investment funds like the DAO. Definity offers a specific kind of agreement in terms of governance agreements that enforce real-world rules upon Ethereum-like blockchains. Hyperledger and its close cousin, Sawtooth Lake, likewise support running Turing-complete code. They offer a key-value data model, with which the applications can create and update key-value tuples on the blockchain.
7) *Example:* Bitcoin is founded on PoW although Peercoin is a new peer-to-peer PoS cryptocurrency. Furthermore, Hyperledger Fabric uses PBFT to achieve consensus. Bitshares, a smart contract platform, explores DPOS as their consensus algorithm. Ripple implements the Ripple protocol, while Tendermint makes use of the Tendermint protocol.

PBFT and Tendermint are permissions protocols. Node identities are expected to be known for the whole network so that they might be used in a commercial mode rather than a public one. PoW and PoS are suitable for public blockchains. Consortium or private blockchains might have a preference for PBFT, Tendermint, DPOS, or Ripple.

B. Advances consensus algorithms

A successful consensus algorithm suggests performance, security, and convenience. Recently, a good deal of effort has been put into improving consensus algorithms in the blockchain. New consensus algorithms have been developed with the aim of solving some specific dilemmas found with regard to the blockchain.

The primary approach of PeerCensus is to decouple block production and transaction confirmation, to guarantee which consensus efficiency can be dramatically increased. Irrespective of this, the author has suggested a new consensus approach to ensure that a block produces a relatively consistent speed. Note that high block generation levels compromise Bitcoin's security. Hence the Greedy Heaviest Observed SubTree (GHOST) chain selection principle has been proposed to solve this issue. In contrast to the most prolonged branch scheme, GHOST weights the branches, and miners may choose to follow the best one.

GHOST introduced a new consensus algorithm for peer-to-peer blockchain systems in which anybody who offers non-interactive proofs of retrievability for the past state snapshots is given consent to generate the block. During these a protocol, miners have to gather old block headers instead of full blocks exclusively.

V. CONCLUSION

Blockchain has shown its potential for transforming the traditional industry with its essential properties in the form of

decentralization, persistency, anonymity, and auditability. In this research, we offer a complete overview of the blockchain. We first give an overview of blockchain technologies including blockchain architecture, and the critical characteristics of the blockchain. We then discuss the standard consensus algorithms used in the blockchain. We analyze and compare these protocols in different respects. Furthermore, we list some challenges and problems that would hinder blockchain development, and summarize some of the existing approaches for solving these issues. Some possible future directions are also proposed. Nowadays blockchain-based applications are springing up, and we plan to conduct an in-depth investigation of blockchain-based applications in the future.

VI. Research Directions

Future work will undoubtedly be compelling and worthwhile. Specifically, putting many protocols in the appropriate spot using the value to their efficiency against their particular to-node scalability, node security, and node privacy, involves a reasonable level of research. Nevertheless, this represents an immediate open challenge which requires researchers to improve their comprehension to facilitate future blockchain scalability and security enhancements. Additionally, there are plenty of prospective opportunities involving synergies between consensus, involving incorporating protocol strategies whenever it comes to improving the adversarial and network models.

References

[1] "Parity - fast, light, and robust Ethereum client," https://github.com/paritytech/parity, 2016

[2] M. Vukolic, "The quest for scalable blockchain fabric: proof-of work vs. but replication," in Open Problems in Network Security - iNetSec, 2015.

[3] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in Proceedings of the 3rd USENIX Symposium on Operating Systems Design and Implementation (OSDI), New Orleans, Louisiana, USA, 1999, pp. 173–186.

[4] "Proof of authority chains," https://github.com/paritytech/parity/wiki/Proof-of-Authority-Chains., 2016

[5] David Mazieres, "The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus," February 25, 2016.

[6] Armknecht F., Karame G.O., Mandal A., Youssef F., Zenner E. "Ripple: Overview and Outlook," Trust and Trustworthy Computing. Trust 2015, Lecture Notes in Computer Science, vol 9229. Springer

[7] Nomura Research Institute, "Survey on Blockchain Technologies and Related Services FY2015 Report", This report is the result of the survey contracted by Japan's Ministry of Economy, Trade, and Industry (METI), March 2016

[8] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," ACM Transactions on Programming Languages and Systems (TOPLAS), vol. 4, no. 3, 1982.

[9] N. Szabo, "Advances in distributed security," 2003.

[10] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in Proceedings of the 3rd USENIX Symposium on Operating Systems Design and Implementation (OSDI), New Orleans, Louisiana, USA, 1999, pp. 173–186.

[11] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008

[12] N. T. Courtois, "On the longest chain rule and programmed self-destruction of cryptocurrencies," Computing Research Repository, Tech. Rep. abs/1405.0534, 2014.

[13] J. A. Kroll, I. C. Davey, and E. W. Felten, "The economics of bitcoin mining, or bitcoin in the presence of adversaries," in WEIS '13: Proceedings of the 12th Workshop on the Economics of Information Security, June 2013.

[14] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending bitcoin's proof of work via proof of stake," in NetEcon '14: Proceedings of the 9th Workshop on the Economics of Networks, Systems, and Computation, June 2014.

[15] J. Clark and A. Essex, "Commission: Carbon dating commitments with bitcoin," in FC '12: Proceedings of the 16th International Conference on Financial Cryptography and Data Security, Mar. 2012.

[16] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Tim´on, and P. Wuille, "Enabling blockchain innovations with pegged sidechains," Tech. Rep., Oct. 2014.

[17] A. Miller, A. Juels, E. Shi, B. Parno, and J. Katz, "Permacoin: Repurposing bitcoin work for data preservation," in SP '14: Proceedings of the 35th IEEE Symposium on Security and Privacy, May 2014.

[18] "Sawtooth lake," Intel Corporation, https://github.com/hyperledger/sawtooth-core, 2016

[19] "Proof of elapsed time (poet)," https://intelledger.github.io/introduction.html#proof-of-elapsed-time-poet., 2014

[20] Ittay Eyal, Adem Efe Gencer, Emin Gun Sirer, and Robbert van Renesse, "Bitcoin-ng: A scalable blockchain protocol," arXiv preprint arXiv:1510.02037, 2015.

[21] Marko Vukolic, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," In Proceedings of the IFIP WG 11.4 Workshop iNetSec 2015. 2015.

[22] Zhang Y, Wen J. "An IoT electric business model based on the protocol of bitcoin," 18th International conference on intelligence in next-generation networks (ICIN); 2015. p. 184–91. doi:http://dx.doi.org/10.1109/ICIN.2015. 7073830.

[23] S. Park, K. Pietrzak, A. Kwon, J. Alwen, G. Fuchsbauer, and P. Gazi, "Space mini: A cryptocurrency based on proofs of space," IACR Cryptology ePrint Archive, vol. 2015, p. 528, 2015.

[24] D Larimer, "Delegated Proof-of-Stake (DPOS)," Bitshare whitepaper, 2014

[25] E Buchman, "Tendermint: Byzantine Fault Tolerance in the Age of Blockchains," 2016