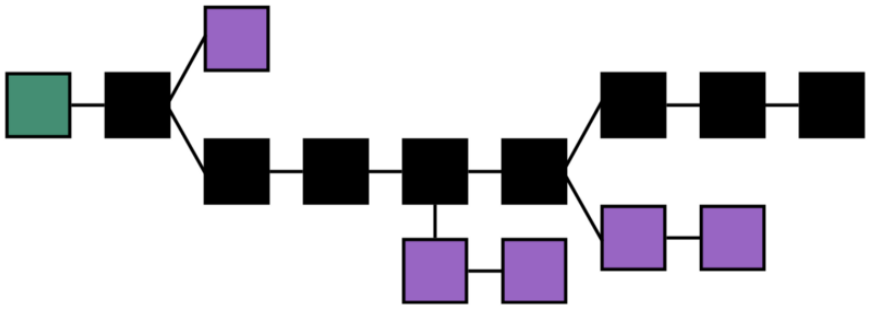
Chris Hammerschmidt Following
Scientist and Risk Taker · math, informatics and their applications, in particular ML and blockchain · aspi...
Jan 27 · 9 min read

Consensus in Blockchain Systems. In Short.

Blockchain technologies top the lists of 2017's hot trends. Many companies already back their products with blockchain technologies. Competitors use different approaches to blockchain technologies, emphasizing different aspects and pitching them as features to their customers. In this post I will provide an overview of the role of one of those particular aspects of blockchain technologies—"**consensus**".

Blockchains are diverse and can be approached from a variety of perspectives. For the sake of this article, I would like to define a blockchain as a **public**, **decentralized** database, that keeps public records in an append-only fashion. Moreover, once added into the database (the blockchain) a record cannot be modified and it is very

difficult to falsify entries. This last feature is called **persistence**. When an entry in the database (the blockchain) needs to be updated, a new record must be appended to the existing information. Finally, each of records can be viewed by any member of the public, allowing for any person to individually verify the authenticity of each transaction recorded for any single entry in the database (the blockchain). This transparency means that blockchains are **auditable**.



CC-BY-3 Theymos from Bitcoin wiki vectorization: The main chain (black) consists of the longest series of blocks from the first (genesis) block (green) to the current block. Orphan blocks (purple) exist outside of the main chain.

But why bother with blockchains over traditional databases anyway? Blockchains become immediately appealing as soon as a database needs to be decentralized. An organization looking to avoid putting too much emphasis on a single potential point of failure and to create a generally more robust system for their information might find a blockchain database more appealing than a traditional one. A distributed database cannot be hacked, manipulated, or otherwise disrupted the way a database build on one single operator can be. In addition, a traditional centralized database requires a user-controlled access system. That is to say, it requires a system directly operated by known and trustworthy individuals (whether that is a known person, organization, computer, or any other familiar operating unit). A blockchain, on the other hand, is operated by unknown and untrusted parties (that is to say, you cannot know if it is an individual person, and organization, a computer operating automatically, or whatever else—let alone know them well enough to trust their decisions and actions implicitly).

The lack of trust inherent in the blockchain system is particularly noteworthy to our topic of ‘consensus’. Because any entity, individual, or party can submit information to the blockchain (that is to say, try to add information to the database), it is necessary for the distributed operators of the blockchain to evaluate and agree on all addenda before they are permanently incorporated into the blockchain (the database). Because we cannot be sure of the author’s trustworthiness,

it is vital that all new information must be reviewed and confirmed before being accepted. This review results in the 'consensus' I am examining here.

There are four main methods of finding consensus in a blockchain (and all distributed systems, for that matter): the practical byzantine fault tolerance algorithm (PBFT), the proof-of-work algorithm(PoW) ,the proof-of-stake algorithm (PoS), and the delegated proof-of-stake algorithm (DPoS).

The Practical Byzantine Fault Tolerance Algorithm (PBFT)was designed as a solution to a problem presented in the form of an allegory (source):

> *Imagine that several divisions of the Byzantine army are camped outside an enemy city, each division commanded by its own general. The generals can communicate with one another only by messenger. After observing the enemy, they must decide upon a common plan of action. However, some of the generals may be traitors, trying to prevent the loyal generals from reaching agreement. The generals must decide on when to attack the city, but they need a strong majority of their army to attack at the same time. The generals must have an algorithm to guarantee that (a) all loyal generals decide upon the same plan of action, and (b) a small number of traitors cannot cause the loyal generals to adopt a bad plan. The loyal generals will all do what the algorithm says they should, but the traitors may do anything they wish. The algorithm must guarantee condition (a) regardless of what the traitors do. The loyal generals should not only reach agreement, but should agree upon a reasonable plan.*

To clarify the allegory for our purposes: the 'generals' in the story are the parties participating in the distributed network running the blockchain (database) in question. The messengers they are sending back and forth are the means of communication across the network on which the blockchain is running. The collective goal of the "loyal generals" is to decide whether or not to accept a piece of information submitted to the blockchain (database) as valid or not. A valid piece of information would be, in our allegory, a correct opportunity to decide in favor of attack. Loyal generals, for their part, are faithful blockchain participants, who are interested in ensuring the integrity of the blockchain (database) and therefore ensuring that only correct information is accepted. The trecherous generals, on the other hand, would be any party seeking to falsify information on the blockchain (the database). Their potential motives are myriad—it could be an individual seeking to spend a BitCoin that she does not actually own

or another person who wants to get out of contractual obligations as outlined in a smart contract he already signed and submitted.

Various computer scientists have outline a number of potential solutions to the Byzantine generals problem from the allegory. The **practical byzantine fault tolerance algorithm (PBFT)**, which is used to establish consensus in blockchain systems, is only one of those potential solutions. Three examples of blockchains that rely on the PBFT for conses are Hyperledger, Stellar, and Ripple. Very roughly and without explaining the whole algorithm (which would take a multiple page research paper), what the PBFT does is as follows: Each 'general' maintains an internal state (ongoing specific information or status). When a 'general' receives a message, they use the message in conjunction with their internal state to run a computation or operation. This computation in turn tells that individual 'general' what to think about the message in question. Then, after reaching his individual decision about the new message, that 'general' shares that decision with all the other 'generals' in the system. A consensus decision is determined based on the total decisions submitted by all generals.

Among other considerations, this method of establishing consensus requires less effort than other methods. However, it comes at the cost of anonymity on the system.

The best-known method of reaching consensus on a blockchain is the **proof-of-work (PoW)** scheme, which is used by Bitcoin. In contrast to the solution in the PBFT, PoW does not require all parties on the network (all nodes) to submit their individual conclusions in order for a consensus to be reached. Rather, PoW is a system which uses a 'hash function' to create conditions under which a single participant is permitted to announce their conclusions about the submitted information, and those conclusions can then be independently verified by all other system participants. False conclusions are prevented by the parameters of the hash function, which ensure that false information will fail to compute in an acceptable way. (For more information, follow me here, or subscribe down below.) In the Bitcoin system specifically, the participant who publicly verified the information on behalf of the network is in turn rewarded for its participation (which is costly in real terms of energy cost and computing resources) with newly created ('mined') Bitcoins. Therefore this process of searching for valid 'hashes' (solutions to the 'hash function' created by the message input), is known as 'mining'. Incentivizing participation in the network ensures broad participation,

which in turn ensures a more robust network and a safer blockchain (database).

This Bitcoin-style PoW scheme allows for easy, broad participation, which in turn ensures greater network stability with minimal requirements on each participant, allowing participants to remain for example anonymous.

A third variation on establishing consensus on blockchain systems is extremely similar to the aforementioned PoW system, however participation in the consensus-building process is restricted to parties identified as having a legitimate stake in the blockchain (e.g. individuals who own bitcoins or entities who have smart contracts saved in the same blockchain database). This third variation, called **proof-of-state (PoS) algorithms** replace the hash function calculation with a simple digital signature which proves ownership of the stake. The network selects an individual to approve new messages (that is to say, confirm the validity of new information submitted to the databse) based on their proportional stake in the network. In other words, instead of any individual attempting to calculate a value in order to be chosen to establish a consensus point, the network itself runs a lottery to decide who will announce the results, and system participants are exclusively and automatically entered into that lottery in direct proportion to their total stake in the network. As in the PoW system run by Bitcoin, the PoS system run by organizations such as Peercoin also provides an incentive to participation, which ensures broadest possible network participation and therefore the most robust network security possible. In the Peercoin system, the chosen party is rewarded with a new Peercoin in a process called ‘minting’ (rather than BitCoin’s ‘mining’).

This system, however, by rewarding those who already are most deeply involved in the network inherently creates an increasingly centralized system. This is inimical to a truly robust network. Therefore proponents of PoS systems have put forward a number of various modifications to help ensure the base for their networks remain as broad (and therefore secure) as possible.

The final method of establishing consensus is perhaps the most centralized, however it counteracts the large stake holder power that PoS systems entail and enables the system to work much more quickly. This method, called a delegated **proof-of-stake (DPoS) system** works along the same lines as the PoS system, except that individuals choose an overarching entity to represent their portion of

stake in the system. So imagine, each individual decides if entity 1, 2, or 3 (these could be, for example, computer servers, and are called 'delegate nodes' within a DPoS system) will 'represent' his or her individual stake in the system. This allows individuals with smaller stakes to team up to magnify their representation, thereby creating a mechanism to help balance out the power of large stake holders. This comes at the cost, however of greater network centralization. Bitshares is one company that employs a DPoS system.

While these four systems for establishing consensus are currently the most dominant, the field is still wide open to innovation. As blockchain systems continue to gain in popularity, they will also continue to grow in scale and complexity. Which of these four consensus building systems (if any) is best equipped to handle this ongoing expansion remains to be seen. Currently, companies choose a system for their product that best meets their (or their customer's) needs for speed, efficiency, and security.

It is important to note, these systems differ not only in the details of the formation of their respective consensus-building communities, but importantly they differ in how they would handle potential attacks. This is, in fact, one of the clearest distinguishing features between the four consensus-building systems: the potential size of an attack on the system that could be easily managed. To return to the Byzantine general analogy I opened with, that is to say, each of these four system can cope with a different proportion of 'treacherous generals' before it begins to make bad choices.

The systems that don't use proof-of-work are also often called **virtual mining** systems because they don't have a mining activity. It is currently unknown whether or not having to solve a work problem has a real security advantage over the virtual mining systems.

If you liked the article, follow me and recommend the article. If you want to read more about blockchain and related technologies, consider filling out my survey and subscribe on my mailing list down below. Let me know what topics you want to hear more about—on a non-technical, digestible level.

. . .

Sign up to get updates on my favorite topics, blockchain and AI.

yourname@example.com

Sign up