Consensus

Throughout this book we have talked about "consensus rules”—the rules that everyone must agree to for the system to operate in a decentralized, yet deterministic, manner. In computer science, the term *consensus* predates blockchains and is related to the broader problem of synchronizing state in distributed systems, such that different participants in a distributed system all (eventually) agree on a single system-wide state. This is called "reaching consensus."

When it comes to the core function of decentralized record keeping and verification, it can become problematic to rely on trust alone to ensure that information derived from state updates is correct. This rather general challenge is particularly pronounced in decentralized networks because there is no central entity to decide what is true. The lack of a central decision-making entity is one of the main attractions of blockchain platforms, because of the resulting capacity to resist censorship and the lack of dependence on authority for permission to access information. However, these benefits come at a cost: without a trusted arbitrator, any disagreements, deceptions, or differences need to be reconciled using other means. Consensus algorithms are the mechanism used to reconcile security and decentralization.

In blockchains, consensus is a critical property of the system. Simply put, there is money at stake! So, in the context of blockchains, *consensus* is about being able to arrive at a common state, while maintaining decentralization. In other words, consensus is intended to produce a system of *strict rules without rulers*. There is no one person, organization, or group "in charge”; rather, power and control are diffused across a broad network of participants, whose self-interest is served by following the rules and behaving honestly.

The ability to come to consensus across a distributed network, under adversarial conditions, without centralizing control is the core principle of all open public blockchains. To address this challenge and maintain the valued property of decentralization, the community continues to experiment with different models of consensus. This chapter explores these consensus models and their expected impact on smart contract blockchains such as Ethereum.

NOTE	While consensus algorithms are an important part of how blockchains work, they operate at a foundational layer, far below the abstraction of smart contracts. In other words, most of the details of consensus are hidden from the writers of smart contracts. You don’t need to know how they work to use Ethereum, any more than you need to know how routing works to use the internet.

Consensus via Proof of Work

The creator of the original blockchain, Bitcoin, invented a *consensus algorithm* called *proof of work* (PoW). Arguably, PoW is the most important invention underpinning Bitcoin. The colloquial term for PoW is "mining," which creates a misunderstanding about the primary purpose of consensus. Often people assume that the purpose of mining is the creation of new currency, since the purpose of real-world mining is the extraction of precious metals or other resources. Rather, the real purpose of mining (and all other consensus models) is to *secure the blockchain*, while keeping control over the system decentralized and diffused across as many participants as possible. The reward of newly minted currency is an incentive to those who contribute to the security of the

system: a means to an end. In that sense, the reward is the means and decentralized security is the end. In PoW consensus there is also a corresponding "punishment," which is the cost of energy required to participate in mining. If participants do not follow the rules and earn the reward, they risk the funds they have already spent on electricity to mine. Thus, PoW consensus is a careful balance of risk and reward that drives participants to behave honestly out of self-interest.

Ethereum is currently a PoW blockchain, in that it uses a PoW algorithm with the same basic incentive system for the same basic goal: securing the blockchain while decentralizing control. Ethereum's PoW algorithm is slightly different than Bitcoin's and is called *Ethash*. We will examine the function and design characteristics of the algorithm in Ethash: Ethereum's Proof-of-Work Algorithm.

Ethereum also considers switching to *ProgPOW*, a more ASIC-resistant PoW algorithm, which is still under development at the time of writing.

Consensus via Proof of Stake (PoS)

Historically, proof of work was not the first consensus algorithm proposed. Preceding the introduction of proof of work, many researchers had proposed variations of consensus algorithms based on financial stake, now called *proof of stake* (PoS). In some respects, proof of work was invented as an alternative to proof of stake. Following the success of Bitcoin, many blockchains have emulated proof of work. Yet the explosion of research into consensus algorithms has also resurrected proof of stake, significantly advancing the state of the technology. From the beginning, Ethereum's founders were hoping to eventually migrate its consensus algorithm to proof of stake. In fact, there is a deliberate handicap on Ethereum's proof of work called the *difficulty bomb*, intended to gradually make proof-of-work mining of Ethereum more and more difficult, thereby forcing the transition to proof of stake.

At the time of publication of this book, Ethereum is still using proof of work, but the ongoing research toward a proof-of-stake alternative is nearing completion. Ethereum's planned PoS algorithm is called *Casper*. The introduction of Casper as a replacement for Ethash has been postponed several times over the past two years, necessitating interventions to defuse the difficulty bomb and postpone its forced obsolescence of proof of work.

In general, a PoS algorithm works as follows. The blockchain keeps track of a set of validators, and anyone who holds the blockchain's base cryptocurrency (in Ethereum's case, ether) can become a validator by sending a special type of transaction that locks up their ether into a deposit. The validators take turns proposing and voting on the next valid block, and the weight of each validator's vote depends on the size of its deposit (i.e., stake). Importantly, a validator risks losing their deposit, i.e., "being slashed", if the block they staked it on is rejected by the majority of validators. Conversely, validators earn a small reward, proportional to their deposited stake, for every block that is accepted by the majority. Thus, PoS forces validators to act honestly and follow the consensus rules, by a system of reward and punishment. The major difference between PoS and PoW is that the punishment in PoS is intrinsic to the blockchain (e.g., loss of staked ether), whereas in PoW the punishment is extrinsic (e.g., loss of funds spent on electricity).

Ethash: Ethereum’s Proof-of-Work Algorithm

Ethash is the Ethereum PoW algorithm. It uses an evolution of the Dagger–Hashimoto algorithm, which is a combination of Vitalik Buterin’s Dagger algorithm and Thaddeus Dryja’s Hashimoto algorithm. Ethash is dependent on the generation and analysis of a large dataset, known as a *directed acyclic graph* (or, more simply, “the DAG”). The DAG had an initial size of about 1 GB and will continue to slowly and linearly grow in size, being updated once every epoch (30,000 blocks, or roughly 125 hours).

The purpose of the DAG is to make the Ethash PoW algorithm dependent on maintaining a large, frequently accessed data structure. This in turn is intended to make Ethash "ASIC resistant," which means that it is more difficult to make *application-specific integrated circuits* (ASIC) mining equipment that is orders of magnitude faster than a fast *graphics processing unit* (GPU). Ethereum’s founders wanted to avoid centralization in PoW mining, where those with access to specialized silicon fabrication factories and big budgets could dominate the mining infrastructure and undermine the security of the consensus algorithm.

Use of consumer-level GPUs for carrying out the PoW on the Ethereum network means that more people around the world can participate in the mining process. The more independent miners there are the more decentralized the mining power is, which means we can avoid a situation like in Bitcoin, where much of the mining power is concentrated in the hands of a few large industrial mining operations. The downside of the use of GPUs for mining is that it precipitated a worldwide shortage GPUs in 2017, causing their price to skyrocket and an outcry from gamers. This led to purchase restrictions at retailers, limiting buyers to one or two GPUs per customer.

Until recently, the threat of ASIC miners on the Ethereum network was largely nonexistent. Using ASICs for Ethereum requires the design, manufacture, and distribution of highly customized hardware. Producing them requires considerable investment of time and money. The Ethereum developers' long-expressed plans to move to a PoS consensus algorithm likely kept ASIC suppliers away from targeting the Ethereum network for a long time. As soon as Ethereum moves to PoS, ASICs designed for the PoW algorithm will be rendered useless—that is, unless miners can use them to mine other cryptocurrencies instead. The latter possibility is now a reality with a range of other Ethash-based consensus coins available, such as PIRL and Ubiq, and Ethereum Classic has pledged to remain a PoW blockchain for the foreseeable future. This means that we will likely see ASIC mining begin to become a force on the Ethereum network while is it still operating on PoW consensus.

Casper: Ethereum’s Proof-of-Stake Algorithm

Casper is the proposed name for Ethereum’s PoS consensus algorithm. It is still under active research and development and is not implemented on the Ethereum blockchain at the time of publication of this book. Casper is being developed in two competing "flavors":

- Casper FFG: "The Friendly Finality Gadget"
- Casper CBC: "The Friendly GHOST/Correct-by-Construction"

Initially, Casper FFG was proposed as a hybrid PoW/PoS algorithm to be implemented as a transition to a more permanent "pure PoS" algorithm. But in June 2018, Vitalik Buterin, who was

leading the research work on Casper FFG, decided to "scrap" the hybrid model in favor of a pure PoS algorithm. Now, Casper FFG and Casper CBC are both being developed in parallel. As Vitalik explains:

> The main tradeoff between FFG and CBC is that CBC seems to have nicer theoretical properties, but FFG seems to be easier to implement.

More information about Casper's history, ongoing research and future plans can be found at the following links:

- Ethereum Casper (Proof of Stake)
- History of Casper, Part 1
- History of Casper, Part 2
- History of Casper, Part 3
- History of Casper, Part 4
- History of Casper, Part 5

Principles of Consensus

The principles and assumptions of consensus algorithms can be more clearly understood by asking a few key questions:

- Who can change the past, and how? (This is also known as *immutability*.)
- Who can change the future, and how? (This is also known as *finality*.)
- What is the cost to make such changes?
- How decentralized is the power to make such changes?
- Who will know if something has changed, and how will they know?

Consensus algorithms are evolving rapidly, attempting to answer these questions in increasingly innovative ways.

Controversy and Competition

At this point you might be wondering: Why do we need so many different consensus algorithms? Which one works better? The answer to the latter question is at the center of the most exciting area of research in distributed systems of the past decade. It all boils down to what you consider "better"—which in the context of computer science is about assumptions, goals, and the unavoidable trade-offs.

It is likely that no algorithm can optimize across all dimensions of the problem of decentralized consensus. When someone suggests that one consensus algorithm is "better" than the others, you should start asking questions that clarify: Better at what? Immutability, finality, decentralization, cost? There is no clear answer to these questions, at least not yet. Furthermore, the design of consensus algorithms is at the center of a multi-billion-dollar industry and generates enormous

controversy and heated arguments. In the end, there might not be a "correct" answer, just as there might be different answers for different applications.

The entire blockchain industry is one giant experiment where these questions will be tested under adversarial conditions, with enormous monetary value at stake. In the end, history will answer the controversy.

Conclusions

Ethereum’s consensus algorithm is still in flux at the time of completion of this book. In a future edition, we will likely add more detail about Casper and other related technologies as these mature and are deployed on Ethereum. This chapter represents the end of our journey, completing *Mastering Ethereum*. Additional reference material follows in the appendixes. Thank you for reading this book, and congratulations on reaching the end!