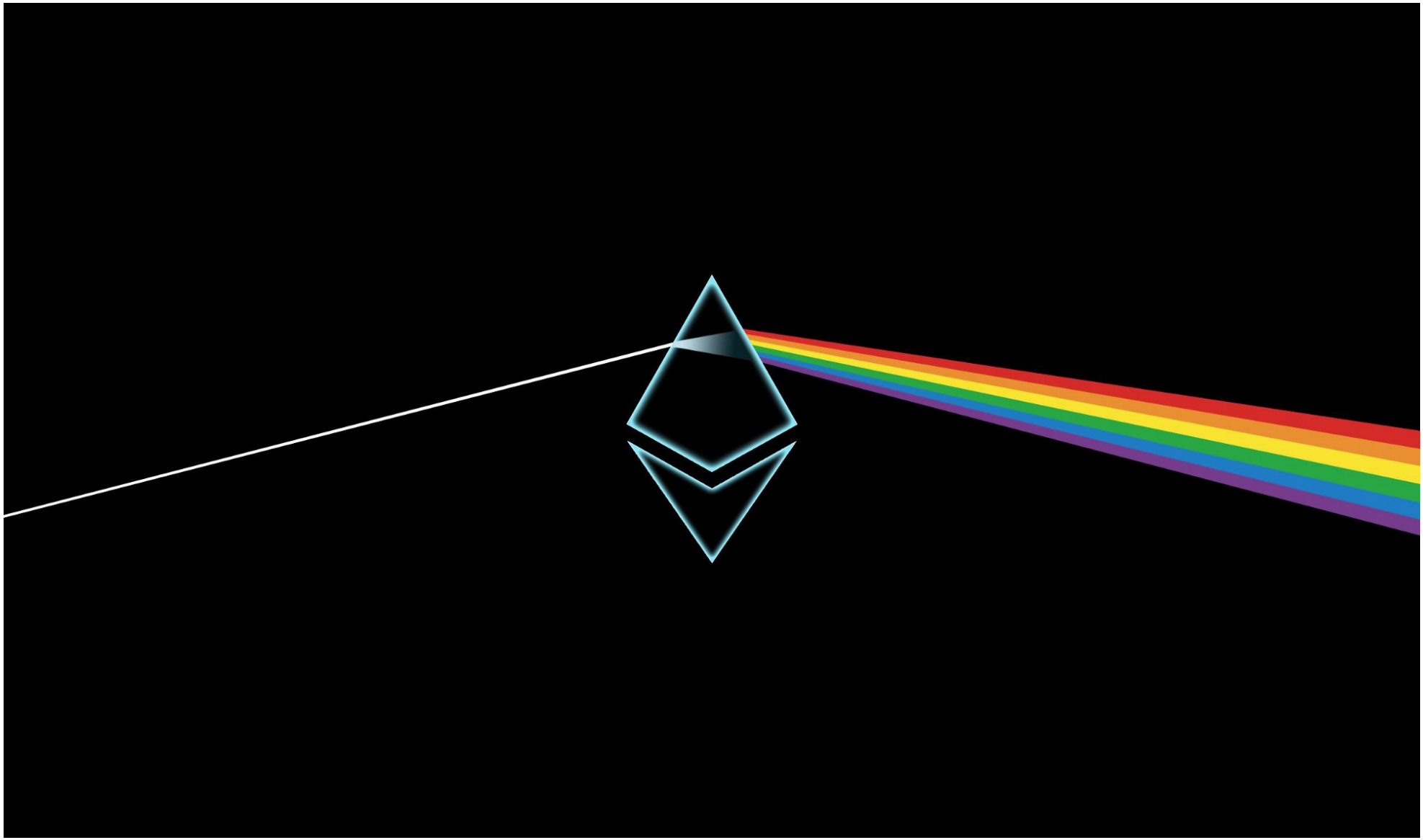
jon choi Follow

Oct 22 · 25 min read

Ethereum Casper 101

tl;dr Casper will implement proof of stake in Ethereum. We begin with a review on why proof of stake matters and continue with its strengths & weaknesses. This post aims to provide a broad overview of Casper and clarify some of the confusion with respect to the two protocol design efforts related to Casper. The two proposed implementations share the same core design principle: **applying cryptoeconomic mechanism design to secure the network while managing challenges regarding liveness, safety and synchrony assumptions**. This post is also an overview of the progress so far and the challenges that lie ahead. Most importantly for fellow newcomers, the post identifies & defines key concepts and ties together various helpful resources under one context. The overarching intention is to make Casper and proof of stake more approachable to everyone in the community.

Enjoy and please don't hesitate to reach out with questions, corrections or feedback (email, twitter).



Outline

Introduction

Casper is hardly a new project as it dates back to 2014, so fortunately it has a lot of related information online. However, despite the wealth of information out there, there's no easy jumping-off point for a beginner to ramp up and start engaging with the project. That's why I'm writing this post.

While other posts, videos and papers focus on the specification, implementation and verification of Casper. This post focuses on the various guiding design principles of Casper, how it differs from competing approaches, why you should care about it, and how you can contribute to this project.

In addition, Casper has maintained an exceptionally open and collaborative culture among Ethereum researchers, developers and community members. I hope this post can continue that tradition by introducing you to Casper and convincing you why it's important to Ethereum as well as the larger public blockchain ecosystem.

To summarize, this post is:

- A quick and broad introduction to PoS and Casper.
- Discussion on why Casper matters, its design principles and its challenges.
- A list of key resources and terms to get you started with PoS and Casper.

This is **not**:

- A full history of Casper.
- Specification of Casper.
- Implementation details of Casper.
- Formal verification of Casper.

If this is review for you, a list of resources are linked throughout the post.

With that, I hope you enjoy this beginner-friendly introduction to Casper: Ethereum's proof of stake research and implementation.

Proof of Stake

(If you're already familiar, please feel free to skip ahead to the next section: "What is Casper?")

Proof of Stake (PoS) is a category of consensus algorithms for public blockchains that depend on a validator's economic stake in the network.

In proof of work (PoW) based public blockchains (*e.g.* Bitcoin and the current implementation of Ethereum), the algorithm rewards participants who solve cryptographic puzzles in order to validate transactions and create new blocks (*i.e.* mining). In PoS-based public blockchains (*e.g.* Ethereum's upcoming Casper implementation), a set of validators take turns proposing and voting on the next block, and the weight of each validator's vote depends on the size of its deposit (*i.e.* stake). Validators are rewarded for their service to the network, but the stake also acts as an economic disincentive for bad actors.

Significant advantages of PoS include **security, reduced risk of centralization, and energy efficiency**.

Explicit Economic Security

In PoW, your downside is capped at how much energy cost and hardware depreciation you incur and therefore has an implicit cost that adjusts dynamically (via 10 min block time target). While PoS has its own challenges (discussed later in the post), one of its major strengths is the flexibility to explicitly design the penalties of Byzantine behavior (*i.e.* not following the protocol). This gives the protocol designer increased control over dialing in the "shape" of the asymmetric risk & reward profile of various actions within the network. One proxy for security is the cost of doing damage to the utility and correctness of the

network, and therefore the ability to have explicit penalties (perhaps at levels that are more draconian than PoW) can increase the security of the network (*i.e.* economic security).

On a related note, Vitalik further argues that PoS has better recovery properties. In PoW, there's an issue of "spawn camp" attacks that can render a blockchain unusable. In PoS, the network can delete the attacker's stake and prevent repeated attacks. The economic analysis further clarifies this concept. The marginal cost of repeated attack is the same as the first round in PoS. Whereas in PoW, the main marginal cost for another round during a 51% attack is the power cost (the incremental hardware depreciation & physical space cost is minimal for repeated attacks). To drive the point home, let's paraphrase Vlad Zamfir, the cost profile of a repeated 51% attack in PoS is as if "your ASIC farm burned down" with each additional round.

Mitigation of Centralization

Proof of Stake mitigates the impact of economies of scale in making consensus. In proof of work, we can already see that the network trusts a relatively concentrated set of mining pools (*e.g.* AntPool) to secure the network. Larger proof of work mining pools can lower the unit cost of their technical and operational infrastructure (datacenter costs, power costs, personnel) by both (1) amortizing a fixed cost over a larger operation and (2) having bargaining power by operating as a larger entity.

This means that two sets of mining pools with equal economic cost, one may be able to achieve a higher hash rate and have more influence in the network, dollar for dollar. For example, 10,000 miners that each spend $1/min ($87.6M/yr) may have less hashing power than one

mining pool that spends $10,000/min (despite also spending $87.6M/yr). (Further work: quantifying the benefits to centralization in PoW mining would be fascinating. *i.e.* is it 1bps, 1% or multiples in hashing power per dollar invested?)

However, in proof of stake, a dollar is a dollar. The benefit here is that you can't pool together to make a dollar worth more. Nor can you develop or buy application-specific integrated circuits (ASICs) to have an advantage technologically. So, PoS intends to mitigate the regressive distribution of PoW mining rewards and move directionally towards proportional distribution. (Going beyond proportional to progressive distribution will require mature decentralized reputation/identity management services).

Energy Efficiency

Proof of work relies on wasting resources to secure the network. Bitcoin currently uses over 20 TWh per year of power, which is close to the power consumption by the entire country of Ecuador. In order for Bitcoin to have broader adoption and operate at Visa's scale, it would have to waste as much power as a much larger country. Extrapolating it's current pace, one can see why proof of work may not be a sustainable path forward.

While Bitcoin may serve an important social function that may indeed exceed its financial costs and environmental externalities (*i.e.* Nick Szabo's social scalability argument), proponents of PoS believe it is possible to replicate the incentive mechanisms of a PoW blockchain without wasting as much energy (by orders of magnitude). Alternatively, some may argue there exists a price of negative

externalities where even the benefits of social scalability might be outweighed by the environmental costs.

While the exact answer is hard to ascertain *a priori*, I believe the crypto ecosystem as a whole has the responsibility to explore all promising consensus mechanisms to weigh their pros/cons and feasibility. (For example, it's great that other projects are testing the benefits and feasibility of various forms of proof of storage, among others.)

Taking a step back, it's also worth noting the two kinds of costs to PoW issuance. There are *internalized costs*, which are paid by miners and passed on to currency holders. Then there are *externalized costs*, such as the environmental costs and subsidies from governments (most likely in the form of cheaper electricity). In PoS, there are low consensus costs (no power & hardware costs), which allows for low issuance. As the network matures, it may even allow for negative issuance (net of burned tx fees as well as any penalties and slashing), which can have a price-stabilizing effect.

Therefore, not only is lower energy consumption better for the environment but it also enables simpler mechanism design. That is because lower energy consumption allows for substituting **realized costs** (power and depreciation costs are nonreversible) with **potential economic value-at-loss** (i.e. believable risk of unrealized costs) in order to secure the network**.** This is a key underlying hypothesis of PoS: **both realized costs and prospect of losing money can incentivize participants to secure the network.** Therefore–while difficult–it is possible (and thus preferable) to secure public blockchains via loss aversion, which can decrease the public costs and deadweight losses in a system.

Proof of Stake: Summary

That sums up the main benefits of PoS. While Casper provides specific benefits to Ethereum (discussed below), a significant portion of its importance are the general benefits of PoS: **explicit economic security, mitigation of centralization and energy efficiency**.

So, now that we've discussed PoS, let's dive into Casper.

Further Reading on PoS

- Proof of Stake FAQ in Ethereum wiki
- Proof of Stake Design Philsophy by Vitalik Buterin
- Critique of PoS by Tuur Demeester (and comments 1, 2, 3)
- Tendermint Whitepaper by Jae Kwon
- Cryptocurrencies without Proof of Work by Iddo Bentov et al. (Slides)
- Bitcoin Wiki on Proof of Stake
- Select examples: Tendermint, Polkadot, Peercoin.

A Tale of Two Caspers

In simple terms, Casper is Ethereum's Proof of Stake work stream.

Casper is not a specific implementation but a family of two main projects under active research by the Ethereum team. Informally, there's "Vitalik's Casper" aka **Casper FFG** and "Vlad's Casper" aka **Casper CBC** (*explained below*). This subtlety isn't clear until you start

diving deep into Casper materials online, which can be very confusing to the "uninitiated." (In fact, that is a main motivator for this post). While independent implementations, they have the same goal in mind: moving Ethereum over to proof of stake.

(Despite a surprisingly common impression that Ethereum is already implementing PoS, it is still a PoW chain (using ethash). While being memory hard and more ASIC-resistant than Bitcoin, Ethereum is a PoW chain nonetheless and has the same drawbacks with respect to energy efficiency.

So with that, let's briefly discuss the two Caspers.

FFG vs CBC

Caveat: both of these projects will present more detailed papers and proof of concepts the weeks following Devcon3, but here's a quick preview of their approaches.

Casper the Friendly Finality Gadget ("FFG")—aka "Vitalik's Casper"—is a hybrid PoW/PoS consensus mechanism, which is the immediate candidate for Ethereum's first bridge to proof of stake. More specifically, FFG implements a proof of stake mechanism as an overlay on top of a proof of work chain (such as Ethereum's ethash PoW chain). Simply, the blockchain would grow every block with the familiar ethash PoW algorithm, but every 50 blocks is a PoS "checkpoint" where finality is assessed via a network of validators.

Casper the Friendly GHOST: Correct-by-Construction ("CBC")—aka "Vlad's Casper"—differs in approach from traditional protocol design: (1) the protocol is partially specified in the beginning and (2)

and the rest of the protocol is derived in way that is proven to satisfy the desired/requisite properties (typically protocol is fully defined then tested to satisfy the said properties). In this case, one way to derive the full protocol is to implement an estimate safety oracle ("an ideal adversary"), which either raises exceptions of a fault of a justified estimate or enumerates the potential future faulty estimates. More specifically, Vlad's work focuses on designing protocols where one can extend local views of a node's estimate of safety to achieve consensus safety.

Taking a step back, FFG focuses more on a multi-step transition to introducing PoS for the Ethereum network. This prepares for an iterative implementation that increases the role of PoS in the network over time. (PoS will claim a smaller portion of the rewards to start). In contrast, CBC focuses on formal methods that derive safety proofs "by construction" from first principles. Albeit confusing, the different approaches to solving this problem created two distinct work streams that complement each other well. The final form of Casper will likely draw from learnings from both FFG and CBC.

Next steps

While substantial progress was achieved, many of the details—both at the higher mechanism design level and at the lower implementation level—remain to be finalized. This is openly acknowledged by both Vitalik and Vlad, and this can be an invitation for more of the community to engage to move the conversation forward.

In sum, both research projects are very active, and more updates should be available at Devcon3 in November. The role of this overview does not include going into more detail, but please feel free to dive into

more implementation and design details in the links below (This doc will likely be updated or followed up once FFG and CBC papers are released).

Links for FFG

- Casper FFG Basics Paper by Vitalik Buterin and Virgil Griffith (Draft)
- Casper Github Repo (FFG)
- Karl Floersch's FFG Implementation on pyethereum

Further Reading on FFG

- Simple explanation on Reddit by Vitalik Buterin. August 2017.
- Minimal Slashing Conditions by Vitalik Buterin. March 2017.
- Safety Under Dynamic Validator Sets by Vitalik Buterin. March 2017.
- Slasher by Vitalik Buterin. January 2014.
- Weak Subjectivity by Vitalik Buterin. November 2014.
- On Settlement Finality by Vitalik Buterin. May 2016.
- Triangle of Harm by Vitalik Buterin. July 2017.
- Casper Economic Incentives Overview by Karl Floersch. Sept 2017.
- Formal Methods on Another Casper by Yoichi Hirai.

- Github. ethereum/casper.

Links for CBC

- *Will be updated after Devcon3*

Further Reading on CBC

- Casper CBC CESC deck and video by Vlad Zamfir. September 2017.
- Casper CBC EDCON deck by Vlad Zamfir. (Video part 1, part 2) February 2017.
- Devcon2 CBC Casper Deck and video by Vlad Zamfir. October 2016.
- 2016.12.06 History of Casper: Chapter 1 by Vlad Zamfir—How Vlad started working on Ethereum and PoS. Slasher and deposits (Mar 2013—Sep 2014)
- 2016.12.07 History of Casper: Chapter 2 by Vlad Zamfir—Nothing-at-stake. Bribing attack. Long range attack. Game theory and security research. (Fall 2014)
- 2016.12.11 History of Casper: Chapter 3 by Vlad Zamfir—Finality. Synchronicity Assumptions. Slashing Conditions. Tendermint. (Sep ’14—Dec ’14)
- 2016.12.12 History of Casper: Chapter 4 by Vlad Zamfir—Cooperative Game Theory. Oligopolies. How Casper is different than other PoS consensus designs. (Dec’14—Jan ’15)

- 2016.12.30 History of Casper: Chapter 5 by Vlad Zamfir—Censorship. Defining decentralization. Availability vs Consistency. Friendly GHOST (Feb ’15—March ’15)
- *History of Casper series is to be continued…*

Why Casper Matters

Now that we have decoded what this mysterious Casper project is. Let’s synthesize what we learned about PoS and Casper to understand why this matters.

In simple terms:

1. Decentralization (*covered in PoS*)
2. Energy efficiency (*covered in PoS*)
3. Explicit Economic Security (*covered in PoS*)
4. Scaling Ethereum
5. Gentle Transition from PoW

Because of Proof of Stake

The first three points are covered in the Proof of Stake section. However, it is worth mentioning that at ~$28B, Ethereum is the second largest cryptocurrency and represents ~18% of the total market cap of the space. So any incremental decentralization and gains in energy efficiency can have a non-trivial impact today and a very significant impact in the future.

As a review, (1) PoS has less available economies of scale because —"dollar for dollar"—a miner/validator can't achieve outsized influence on the network. In PoW, a large mining pool might get more hashing power per dollar than an individual miner, whereas in PoS, a dollar is a dollar, which will likely mitigate the centralizing forces in mining.

And (2) whereas PoW relies on energy wastage for network security, PoS relies on the potential of losing a deposit for network security. The challenge then becomes (3) how we mimic (and enhance) the positive features of PoW and mitigate the weaknesses of PoS with economic mechanism design.

Because of Scaling

Next, let's talk about something new: (4) scaling. The key to understanding this is two-fold: (a) Casper is about establishing explicit finality (as opposed to probabilistic finality) and (b) explicit finality enables maintaining network security while scaling via sharding.

In PoW chains, finality is implicit (just as "skin in the game" is implicit via power wastage). The implicit nature of finality in PoW chains is apparent when you examine how transactions are finalized in real use cases. Depending on the magnitude and importance of the payment, you wait for additional number of block confirmations (number of block times since a transaction appears in the longest chain). For example, for a coffee, you might be okay with a few confirmations, but for buying a car, you may need more than the average number of confirmations.

In contrast, Casper provides a concept of explicit finality. For example, Casper FFG begins to overlay finality on top of the PoW chain. So the underlying chain continues to have an implicit way of determining how final a transaction is. However, Casper FFG provides explicit finality after about 2.5 "epoch times." (Each epoch time being a sequence of 50 PoW blocks. A checkpoint is the last block in an epoch. It is first "justified" then "finalized" by a validator set. More details available in the paper linked above and potentially a future post.)

At that point, with certain Byzantine Fault Tolerance assumptions, we can be sure that either our assumptions have been violated or that the checkpoint is final. Since we are also aware of the validator set *a priori* (which can also be dynamic), bad actors are penalized through fault attribution.

So how does this relate to sharding & scalability? Having this explicit finality allows more flexibility in how much (more accurately, how little) each node in the network has to do. Having more regular explicit finality allows further exploration of questions such as: what if not every node had to hold all of the state or all of the transactions? what if not every node had to validate every transaction? These questions of having heterogenous "responsibilities" of nodes in public blockchains are tackled by a workstream around *blockchain sharding*.

So to go back to the point, **if we are to explore each node in the network "doing less" or "knowing less," it is hugely beneficial to consider only the past few epochs of finality rather than the entire probabilistic chain since the genesis block**. Therefore, at this epoch time interval, finality doesn't actually help with confirming a simple transaction, since transactions clear in number of confirmations fewer

than the epoch time. Instead, finality will enable public blockchains to scale beyond the current ~10 transactions per second to larger orders of magnitude.

Because Ethereum is worth ~$28B

The final point is (5) a gentle transition from PoW. Here's the context for newcomers such as myself. Ethereum's explicit goals to move to PoS predates the significant increase in the value of ether this year. The plan is to start with a hybrid PoS overlay on top of the ethash PoW chain and to ramp up gradually towards "pure" PoS. Given the significant increase in ETH network value, this gradual transition to PoS is a prudent strategy to prevent potential value destruction while transitioning a significant piece of the underlying Ethereum infrastructure.

Further Reading

- Sharding FAQ, Ethereum wiki
- On Slow and Fast Block Times by Vitalik

Design Principles

This is a collection of design principles found in various posts by Vlad and Vitalik. Currently, the guiding design principles for Casper are scattered around various resources. Hope reading them in one go provides more clarity on the overall design principles.

1. **Economics to design behavior.** Explicit economic mechanism design can achieve implicit economic incentives found in other

social contracts (*i.e.* consensus protocols like proof of work). If you like analogies, search "big games" in History of Casper part 3.

2. **Maximize cost of attack.** For example, the amount of damage that an attacker can do to a protocol's utility should be bound by some "griefing factor." In order to do $100 of damage, it shouldn't cost $0.01. It should hopefully cost in the order of $100. In other words, we want to minimize the "attack multiple" of each dollar used for attacking the protocol (more on this coming in another post).

3. **Public cost-benefit, not just private.** Protocol economics should account for social (*i.e.* "public") cost & benefit (negative and positive externalities) as we begin scaling public blockchains. Energy cost, environmental impact and wealth distribution are some notable examples.

4. **Prevent economies of scale.** Centralization weakens the main value proposition of public blockchains. Preventing economies of scale prevents those centralizing factors and build more secure blockchains.

5. **Network security is derived from "skin in the game."** Simple yet worth reiterating. The more you have to lose, the more we can trust you as a validator. Whereas burning energy secures proof of work chains, "putting up economic value-at-loss" secures proof of stake chains.

6. **Design for oligopolies.** Cooperative game theory is the name of the game as a protocol won't be able to fully mitigate the inherent centralizing forces (*e.g.* economies of scale) in a network. This means analyzing all edge cases that involve self-interested cartel

behavior. Notably, a protocol should disincentivize cartels from bullying non-cartel validators (*i.e.* be "friendly")

7. **Accountable safety.** Design that makes it possible to attribute faults to bad actors as much as possible. Casper relies on the ability to slash attributable Byzantine behavior.

8. **Plausible liveness**. Design that doesn't allow attackers to block the chain from continuing to propose and vote on checkpoints/blocks. This is where Casper differs from other implementations such as Tendermint, which will "lock up" if safety isn't achieved synchronously.

9. **Minimal synchronicity assumptions.** To allow for liveness and "unblock" the chain growing, Casper has minimal synchronicity assumptions. In fact, we expect nodes to log on as infrequently as every couple of months.

10. **Decentralized things should be able to regenerate**. A protocol is decentralized only if it can fully recover from the permanent removal of all but one of its nodes. Availability, not just consistency (Again, Tendermint is susceptible to getting "blocked" and cannot regenerate; the validator set for each branch can keep getting smaller per Matthew Wampler-Doty and Vlad's observation)

11. **Disincentivize censorship.** The major tradeoff is that validators have a new attack vector by deliberately going offline. However, cartel censorship is the greater evil here. Choosing the right relative cost of censorship vs. rewards and other penalties (as a % of deposits) will be the key to getting this right.

Further Reading

- PoS Design Philosophy
- History of Casper (links above)

Challenges

Broader list of Ethereum challenges are available here.

Challenges for Proof of Stake

Nothing-at-stake problem—If there's a fork in the chain, the optimal strategy for any validator is to validate on every chain, so that the validator gets their reward regardless of the outcome of the fork.

Long range attack—Same mechanism as 51% attack (make a longer chain that rewrites the ledger in the attacker's favor), but instead of starting the attack 6 blocks back, go much further back in the chain's history (*i.e.* 60,000 blocks). This is a problem for PoS since it there's no proof of work required to rewrite a very long chain.

These two main challenges are solved via ideas from slasher (and its improved variations). The main points are that (1) validators are known, which allow for fault attribution at a validator level and (2) by having "slashing conditions" that strongly disincentivize certain actions, it is possible to mitigate these issues. Again, this example is crucial in understanding the Casper team's view on consensus algorithm design: we can leverage economic mechanism design to a secure distributed system.

Criticisms of Proof of Stake

Adverse selection—Given the potentially draconian penalties, many average or risk-averse "candidate validators" may stay away from participating as a validator. Then, one may argue people with more to gain by "gaming the system" are more likely to join as a validator. More broadly, one may claim that—on average—a good actor may never have a better ROI than a bad actor.

Response:
This falls under future work and an area of great focus for the research team: cryptoeconomics. As the parametrization of the mechanism is progresses, the team will iterate on optimizing the constants the balance the risk reward tradeoffs as well as their proportion to the size of the deposits and the action of others (Byzantine behavior).

It's worth mentioning that this problem is also existent (albeit to a lesser degree given the nature of proof of work) in Bitcoin.

"The rich get richer"—Another common concern when people hear "a consensus algorithm based on how much money you're staking" is that this may exacerbate wealth inequality within the crypto ecosystem as well as more broadly in the global economy.

Response:
The main takeaway here should be that Proof of Stake is considerably more egalitarian (*i.e.* gives less benefit to having more capital) than the incumbent Proof of Work based algorithm of Bitcoin.

As discussed in the Proof of Stake overview above, **PoS mitigates economies of scale**, which diminishes the centralizing force among miners. Again, **in PoS, a dollar is a dollar**.

Therefore, against the fair intuition that PoS can exacerbate wealth inequality, it is in fact a non-trivial improvement on the status quo.

Quick aside: In order to have diseconomies of scale or progressive wealth distributions in PoS (one more degree of counteracting wealth inequality), I posit that it is necessary to have mature and reliable identity or reputation systems. Otherwise, larger pools of money will have "sybil behavior" that spreads their wealth over many "less wealthy" false identities that will collectively capture the benefits of a progressive reward system. However, this challenge is out of scope for Casper and will be faced further down the road.

Questions/Concerns around Casper

"It's confusing to have multiple Caspers."
Sorry about that! This post aims to lessen the cognitive dissonance around that fact. But to review, Casper is Ethereum's family of PoS research and implementations. These work streams will likely converge, but the nature of protocol research requires forking sometimes to explore various approaches until deciding on the best way forward. Things tend to get more complex before they get simpler 🙂

"How does Casper differ from Tendermint?"
The simplified answer here is that Casper focuses on liveness (availability) and can accept less immediate safety (correctness). While Tendermint is a great project, its downside is that the chain will stall if a checkpoint doesn't have 2/3 of the votes. That is one of the reasons why Ethereum is working on Casper rather than working off of Tendermint.

To quote Vlad Zamfir:

> *Tendermint favours consistency over availability, Casper favours availability over consistency (see the CAP theorem).*

> *Tendermint doesn't punish online validators for potentially censoring potentially-actually-just-offline validators.*

For further reading on this topic: Hudson Jameson's explanation this quote, reddit discussion featuring Vitalik & Jae Kwon, and the Tendermint whitepaper.

" 😱You're going to change the engine of a live $28B network?"
Yes, that is very ambitious and daunting. However, moving to PoS was the intent since the early days, and one of the guiding principles of the project. The community members were very much aware of Ethereum's plans to switch to PoS (*i.e.* refer to Ethereum Ice Age—the PoW chain difficult adjustment to encourage migration to PoS—which is commonly known in the ecosystem).

This should go without saying, but the team will be rolling out the change in stages with the test network. Also, the initial implementation is a hybrid where the PoS elements have less of an economic impact—and therefore impact on security—than in the eventual "pure" PoS consensus.

"In practice, transactions are cleared in 10 blocks. Why does finality matter over a 50 block epoch?"
We covered this above, but let's go over this once again since it's important.

First, Casper enables **sharding**.

This was very unclear to me in the beginning, but that's because we need to take a step back from Casper for a moment. Ethereum has many goals, but one of them is to provide a scalable blockchain solution, both technically and environmentally. Ethereum is building for a world where cryptocurrencies have a larger place in the global economy and grow by many orders of magnitude. In that vision, Casper aims to prevent the wasted energy of PoW mining but we still need to scale Ethereum technically. That project is largely encompassed under sharding.

Today, every node in the network does everything. Sharding explores various ways to decrease the average responsibility of each node. The details are out of scope of this post, but an example might be to pose the question: "are there ways to create a new mechanism, where only small subset of nodes verifies each transaction?"

The takeaway is that the periodic finality provided by Casper will mitigate lower security related to implementing sharding.

Second, explicit finality allows for a consistency-favoring blockchain. To quote Vitalik:

> *In a PoW chain, if there is, for example, a geth/parity consensus split, then both chains keep growing, and exchanges running one client or the other risk finalizing deposits on a bad chain. But if exchanges wait for Casper finality, then in a 50/50 split it's likely *neither* chain will finalize. This increases the safety of the platform, as in extreme circumstances it*

> *“defaults toward finalizing nothing” rather than finalizing *the wrong thing**.

In sum, contrary to intuition, explicit finality matters less for transaction clearing and more for blockchain scalability and safety.

Future Work

Execution

- Finalizing the design of FFG and CBC.
- Implementing proof of concept code.
- Deploying into test nets.

”The One Casper”

Thinking about about the final state of Casper PoS. How to leverage concepts from both FFG and CBC to converge on a final state that is compelling, secure and elegant.

Parameter Optimization

For a given mechanism, optimize parameters and constants for intended incentivization. Iterate on mechanism design with findings.

Community Education

Writing more about the thought process beyond various research work streams to keep engaging with Ethereum, PoS and broader crypto communities.

Potential Future Posts

- History of Casper Part 6
- Deep dive into how Casper enables sharding
- Deep dive into how and why Casper is different from competing designs
- Non-monetary measures of “stake” and “skin in the game” (*i.e.* $10,000 is worth more to average person than billionaire)
- Griefing Factor Analysis 2.0

Glossary

We covered a lot of concepts in this post, and you will run into some other common ones as you dive into the maze of Chrome tabs that will inevitably ensue. Here’s a rough summary of the most helpful and common concepts you need to grok. Hope it helps!

Proof of Stake—a category of consensus algorithms for public blockchains that depend on a validator’s economic stake in the network.

Casper—Ethereum’s proof of stake research and projects.

Finality—Once an operation in a system completes, the system doesn’t allow for the operation to be reverted (Vitalik on settlement finality). Context: in proof of work, finality is probabilistic and implicit. Casper is designing mechanisms that explicitly enforce finality.

Fork Choice Rule—A fork choice rule is a function, evaluated by the client, that takes as input the set of blocks and other messages that have been produced, and outputs to the client what the "canonical chain" is.

Slashing Conditions—Set of rules that, if violated, penalize the validator.

Sybil Attack—an attack wherein a reputation system is subverted by forging identities in peer-to-peer networks.

3 E's of Sybil Resistance—1. Entry Cost 2. Existence Cost 3. Exit Penalty. (coined by Dominic Williams).

Nothing-at-stake problem—A proof of stake implementation challenge that refers to the inherent lack of downside for validating both chains in the case of a fork. It is a well-known problem of proof of stake that is considered solvable. For example, refer to Slasher.

Bribe attack—Attacker uses a bribe to change the Nash Equilibrium of a validator's game theory framework to undermine the security of the protocol. (More context in History of Capser pt 2)

Long range attack—Same mechanism as 51% attack (make a longer chain that "rewrote" the ledger in the attacker's favor), but instead of starting the attack 6 blocks back, go much further back in the chain's history (think 60,000 blocks).

DAG—"Directed Acyclic Graph". A finite directed graph with no directed cycles. (ETH Stack Exchange).

GHOST—"Greedy Heaviest Observed Subtree." It's a chain selection rule with the objective of fast confirmation times while limiting compromises in security or decentralization. (Original Paper, ETH GHOST)

Synchronicity—Refers to the timing assumptions around the messages (*i.e.* synchronous, partially synchronous and asynchronous).

Liveness—"Availability." Protocol-following nodes eventually decide on a value. Opposite would be a network state that is blocked from deciding on a value (i.e. Tendermint without 2/3 votes at a given depth)

Safety—"Correctness." Protocol-following nodes decide on the same value. Another intuitive proxy is whether two conflicting blocks can be committed.

FLP Impossibility Theorem—"It's impossible to have a live, safe and asynchronous network" (formally proven as well).

Accountable Faults—Faults that can be attributed to a specific validator or specific set of validators.

Byzantine Faults / Byzantine Behavior—Any fault presenting different symptoms to different observers. Non-protocol following behavior.

Byzantine Failure—Loss of a system service due to Byzantine faults in a system that requires consensus.

Byzantine Fault Tolerance ("BFT")—Ability for a system to tolerate Byzantine faults. 1/3 Byzantine fault threshold in asynchronous networks. 1/2 in synchronous networks. (BFT consensus algorithms include Paxos, PBFT as well as newer Casper and Tendermint).

Nakamoto Consensus—PoW-based bitcoin-style consensus building. Also, Nakamoto-style Consensus exists, which would be chain-based PoS as opposed to BFT-based PoS.

Tendermint—Proof of Stake implementation that focuses on consistency. Never forks with less than 1/3 malicious actors, but downside is that the chain can stall if a chain lacks 2/3 of the validator votes.

Validator—An entity that validates checkpoints/blocks of a blockchain for rewards. A miner analog in PoS.

Validator Set—Set of validators for a given chain at any given time.

Checkpoint—In FFG, it is a block spaced in regular intervals (*i.e.* every 50 blocks) where the PoS validation mechanism is overlaid on top of the underlying PoW chain (*e.g.* Ethereum with ethash)

Epoch—In FFG, it is a 50 block period where a validator can vote on the finality of its final block (*i.e.* checkpoint). PoW miners mine blocks and PoS validators validate checkpoints every epoch.

Dynamic Validator Sets—The idea that a blockchain can have changing set of validators throughout a period. Treatment of this is a huge breakthrough in BFT-style consensus algorithms. Tendermint was first notable breakthrough. Casper is also working on this actively.

Equivocation—The act of a validator sending two messages that conflict with each other (more specific definition on slide 28 of this deck).

Dunkles—Mechanism that includes data from non-dominant block into the dominant block. This provides better incentive mechanisms and notably helps alleviate the nothing-at-stake problem (link).

Proposal Mechanism—Mechanism by which a validator in the set will suggest a block to assess for justification or finalization.

Justification —In FFG for example, 2/3 of a validator set voting on a fresh checkpoint that a checkpoint is the accurate record. This is the intermediary step for a finalized checkpoint.

Finalization—In FFG for example, 2/3 of a validator set voting on a justified checkpoint that a checkpoint is the accurate record. The completion of this step gives a checkpoint finality.

State Transition System—A system that maintains a given state (*e.g.* set of transactions or accounts) and its mutations over time (*i.e.* transition). Bitcoin, Ethereum, and other public blockchains can be considered state transition systems.

Protocol Utility Function—"…a formula that tells us how well the protocol is doing, that should ideally be calculable from inside the blockchain. In the case of a proof of work chain, this could be the percentage of all blocks produced that are in the main chain. In Casper, protocol utility is zero for a perfect execution where every epoch is finalized and no safety failures ever take place, with some penalty for every epoch that is not finalized, and a very large penalty for every

safety failure. If a protocol utility function can be formalized, then penalties for faults can be set as close to the loss of protocol utility resulting from those faults as possible." (from Triangle of Harm)

Conclusion

OK, this was a lot of information for a "101," but now you have all the basics covered and the context to dive into any Casper conversation. My intention in writing this post was (1) to provide an overarching thread of context across the plethora of information out there on Casper, (2) solidify my own understanding of Casper as I dive into cryptoeconomic research for Ethereum scaling (Casper, sharding, gas pricing), (3) to raise awareness and encourage conversation about PoS, Casper and Ethereum and (4) to convince talented mathematicians, economists, computer scientists and developers to start diving into the problems we are trying to solve.

If you want to dive deeper, dive into the resources linked throughout the post, consider contributing to the future work, and talk to your math, CS, economics, game theory, and distributed systems friends (in and out of crypto) about Ethereum and Casper.

If you've made it this far, thank you for reading. Hope this helped and please reach out with questions, concerns or comments via email or twitter. We would love to hear your feedback!

. . .

Special thanks to Vitalik Buterin, Vlad Zamfir, Virgil Griffith and Karl Floersch for discussion and review. All errors remain my own.