Ethereum Blog

technical

Introducing Casper “the Friendly Ghost”

Vlad Zamfir

Posted by Vlad Zamfir on August 1st, 2015.

Hi everyone – Vlad here. I’ve been working on the analysis and specification of “proof-of-stake” blockchain architecture since September 2014. While Vitalik and I haven’t agreed on all of the details of the spec, we do have consensus on many properties of the proof-of-stake protocol that will likely be implemented for the Serenity release! It is called Casper “the friendly ghost” because it is an adaptation of some of the principles of the GHOST (Greedy Heaviest-Observed Sub-Tree) protocol for proof-of-work consensus to proof-of-stake. This blog post (my first one!) shares properties that are likely to be true of Casper’s implementation in the Serenity release. Formal verification and simulation of Casper’s properties is under way, and will be published eventually – in the meantime, please enjoy this high-level, informal discussion! :)

LATEST POSTS

The History of Casper – Chapter 2

07th December, 2016

The History of Casper — Chapter 1

06th December, 2016

Security-deposit based security and authentication

Casper is a security-deposit based economic consensus protocol. This means that nodes, so called “bonded validators”, have to place a security deposit (an action we call “bonding”) in order to serve the consensus by producing blocks. The protocol’s direct control of these security deposits is the primary way in which Casper affects the tors. Specifically, if a validator produces anything that Casper considers "invalid", their deposit are forfeited along with the privilege of participating in the consensus process. The use of security deposits addresses the "nothing at stake" problem; that behaving badly is not expensive. There is something at stake, and bonded validators who misbehave in an objectively verifiable manner *will* lose it.

Very notably, a validator's signature is only economically meaningful so long as that validator *currently* has a deposit. This means that clients can only rely on signatures from validators that they know are *currently* bonded. Therefore, when clients receive and authenticate the state of the consensus, *their authentication chain ends in the list of currently-bonded validators*. In proof-of-work consensus, on the other hand, the authentication chain ends in the genesis block – as long as you know the genesis block you can authenticate the consensus. Here, as long as you know the set of currently-bonded validators, you can authenticate the consensus. A client who does not know the list of currently bonded validators must authenticate this list out-of-band. This restriction on the way in which the consensus is authenticated solves the "long range attack" problem by requiring that everyone authenticate the consensus against current information.

The validator list changes over time as validators place deposits, lose their deposits, unbond, and get unbonded. Therefore, if clients are offline for too long, their validator list will no longer be current enough to authenticate the consensus. In the case that they are online sufficiently often to observe the validator set rotating, however, clients are able to securely update their validator list. Even in this case, clients must begin with an up-to-date list of currently-bonded validators, and therefore they must authenticate this list out-of-band *at least* once.

Vlad Zamfir

LATEST POSTS

The History of Casper – Chapter 2
07th December, 2016

The History of Casper — Chapter 1
06th December, 2016

uthentication only necessarily once” property is what Vitalik calls *weak subjectivity*. In this context information is said to be “objective” if it can be verified in a protocol-defined manner, while it is “subjective” if it must be authenticated via extra-protocol means. In weakly subjective consensus protocols, *the fork-choice rule is stateful*, and clients must initialize (and possibly sometimes renew) the information t k-choice rule uses to authenticate the consensus. In our case, this entails iden currently bonded validators (or, more probably a cryptographic hash of the validator list).

Gambling on Consensus

Casper makes validators bet a large part of their security deposits on how the consensus process will turn out. Moreover, the consensus process “turns out” in the manner in which they bet: validators are made to bet their deposits on how they expect everyone else to be betting their deposits. If they bet correctly, they earn their deposit back with transaction fees and possibly token issuance upon it – if on the other hand they do not quickly agree, they re-earn less of their deposit. Therefore through iterated rounds of betting validator bets converge.

Moreover, if validators change their bets too dramatically, for example by voting with a high probability on one block after voting with a very high probability on another, then they are severely punished. This guarantees that validators bet with very high probabilities only when they are confident that the other validators will also produce high probability bets. Through this mechanism we guarantee that their bets never converge to a second value after converging upon a first, as long as there there is sufficient validator participation.

Vlad Zamfir

LATEST POSTS

The History of Casper – Chapter 2
07th December, 2016

The History of Casper — Chapter 1
06th December, 2016

ensus is also a betting scheme: miners bet that their block will be part of the heaviest chain; if they eventually prove to be correct, they receive tokens – whereas if they prove to be incorrect, they incur electricity costs without compensation. Consensus is secured as long as all miners are betting their hashing power on the same chain, making it the blockchain with the most work (*as a direct result of and as pre their coordinated betting*). The economic cost of these proof-of-work bets ac ly in the number of confirmations (generations of descendant blocks), while, in Casper, validators can coordinate placing exponentially growing portions of their security deposits against blocks, thereby achieving maximum security very quickly.

By-height Consensus

Validators bet independently on blocks at every height (i.e. block number) by assigning it a probability and publishing it as a bet. Through iterative betting, the validators elect exactly one block at every height, and this process determines the order in which transactions are executed. Notably, if a validator ever places bets with probabilities summing to more than 100% at a time for a given height, or if any are less than 0%, or if they bet with more than 0% on an invalid block, then Casper forfeits their security deposit.

Transaction Finality

When every member of a supermajority of bonded validators (a set of validators who meet a protocol-defined threshold somewhere between 67% and 90% of bonds) bets on a block with a very high (say, > 99.9%) probability, the fork-choice rule never accepts a fork where this block does not win, and we say that the block is *final*. Additionally, when a client sees that every block lower than some height H is final, then the client will never

Vlad Zamfir

LATEST POSTS

The History of Casper – Chapter 2
07th December, 2016

The History of Casper — Chapter 1
06th December, 2016

has a different application state at height **H – 1** than the one that results from the execution of transactions in these finalized blocks. In this eventuality, we say that this state is finalized.

There are therefore two relevant kinds of transaction finality: the finality of t

the transaction will be executed at a particular height (which is from finality

and therefore priority over all future blocks *at that height*), and the finality of the consensus state after that transaction's execution (which requires finality of its block and of unique blocks at all lower heights).

Censorship Resistance

One of the largest risks to consensus protocols is the formation of coalitions that aim to maximize the profits of their members at the expense of non-members. If Casper's validators' revenues are to be made up primarily of transaction fees, for example, a majority coalition could censor the remaining nodes in order to earn an increased share of transaction fees. Additionally, an attacker could bribe nodes to exclude transactions affecting particular addresses – and so long as a majority of nodes are rational, they can censor the blocks created by nodes who include these transactions.

To resist attacks conducted by majority coalitions, Casper regards the consensus process as a cooperative game and ensures that each node is most profitable if they are in a coalition made up of 100% of the consensus nodes (at least as long as they are incentivized primarily by in-protocol rewards). If **p**% of the validators are participating in the consensus game, then they earn **f**(**p**) ≤ **p**% of the revenues they would earn if 100% of the validators were participating, for some increasing function **f**.

Vlad Zamfir

LATEST POSTS

The History of Casper – Chapter 2
07th December, 2016

The History of Casper – Chapter 1
06th December, 2016

Casper punishes validators for not creating blocks in a protocol-prescribed order. The protocol is aware of deviations from this order, and withholds transaction fees and deposits from validators accordingly. Additionally, the revenue made from betting correctly on blocks is linear (or superlinear) in the number of validators who are participating in at that height of the consensus game.

Will there be more transactions per second?

Most probably, yes, although this is due to the economics of Casper rather than due to its blockchain architecture. However, Casper's blockchain does allow for faster block times than is possible with proof-of-work consensus.

Validators will likely be earning only transaction fees, so they have a direct incentive to increase the gas limit, if their validation server can handle the load. However, validators also have reduced returns from causing other, slower validators to fall out of sync, so they will allow the gas limit to rise only in a manner that is tolerable by the other validators. Miners investing in hardware primarily purchase more mining rigs, while validators investing in hardware primarily upgrade their servers so they can process more transactions per second. Miners also have an incentive to reinvest in more powerful transaction processing, but this incentive is much weaker than their incentive to purchase mining power.

Security-deposit-based proof-of-stake is very light-client friendly relative to proof-of-work. Specifically, light clients do not need to download block headers to have full security in authenticating the consensus, or to have full economic assurances of valid transaction execution. This means that a lot of consensus overhead affects only the validators, but

Vlad Zamfir

LATEST POSTS

The History of Casper – Chapter 2
07th December, 2016

The History of Casper — Chapter 1
06th December, 2016

, and it allows for lower latency without causing light clients to lose the ability to authenticate the consensus.

Recovery from netsplits

Casper is able to recover from network partitions because transactions in n blocks can be reverted. After a partition reconnects, Casper executes transact n blocks that received bets on the partition with higher validator participation. In this manner, nodes from either side of the partition agree on the state of the consensus after a reconnection and before validators are able to replace their bets. Validato ts converge to finalize the blocks in the partition that had more validator participation, with very high probability. Casper will very likely process the losing transactions from losing blocks after the ones from winning blocks, although it is still to be decided whether validators will have to include these transactions in new blocks, or if Casper will execute them in their original order, himself.

Recovery from mass crash-failure

Casper is able to recover from the crash-failure of all but one node. Bonded validators can always produce and place bets on blocks on their own, although they always make higher returns by coordinating on the production of blocks with a larger set of validators. In any case, a validator makes higher returns from producing blocks than from not producing blocks at all. Additionally, bonded validators who appear to be offline for too long will be unbonded, and new bonders subsequently will be allowed to join the validation set. Casper can thereby potentially recover precisely the security guarantees it had before the mass crash-failure.

Vlad Zamfir

LATEST POSTS

The History of Casper – Chapter 2
07th December, 2016

The History of Casper — Chapter 1
06th December, 2016

What is Casper, in non-economic terms?

Casper is an eventually-consistent blockchain-based consensus protocol. It favours availability over consistency (see the CAP theorem). It is always available, and consistent whenever possible. It is robust to unpredictable message delivery times because nodes come to consensus via re-organization of transactions, after delayed messages are eventually received. It has an eventual fault tolerance of 50%, in the sense that a fork created by >50% correct nodes scores higher than any fork created by the remaining potentially-faulty validators. Notably, though, clients cannot be certain that any given fork created with 51% participation won't be reverted because they cannot know whether some of these nodes are Byzantine. Clients therefore only consider a block as finalized if it has the participation of a supermajority of validators (or bonded stake).

What is it like to be a bonded validator?

As a bonded validator, you will need to securely sign blocks and place bets on the consensus process. If you have a very large deposit, you will probably have a handful of servers in a custom multisig arrangement for validation, to minimize the chance of your server misbehaving or being hacked. This will require experimentation and technical expertise.

The validator should be kept online as reliably and as much as possible, for it to maximize its profitability (or for otherwise it will be unprofitable). It will be very advisable to buy DDoS protection. Additionally, your profitability will depend on the performance and availability of the other bonded validators. This means that there is risk that you cannot directly mitigate, yourself. You could lose money even if other nodes don't perform well – but you will lose more money yet if you don't participate at all, after bonding. However,

Vlad Zamfir

LATEST POSTS

The History of Casper – Chapter 2
07th December, 2016

The History of Casper — Chapter 1
06th December, 2016

often means higher average profitability – especially if the risk is perceived but the costly event never occurs.

What is it like to be an application or a user?

Applications and their users benefit a lot from the change from proof-of-wo[illegible]s to Casper. Lower latency significantly improves the user's experience. In normal conditions transactions finalize very quickly. In the event of network partitions, on the other hand, transactions are still executed, but the fact that they can potentially still be reverted is reported clearly to the application and end-user. The application developer therefore still needs to deal with the possibility of forking, as they do in proof-of-work, but the consensus protocol itself provides them with a clear measure of what it would take for any given transaction to be reverted.

When can we hear more?

Stay tuned! We'll be sure to let you know more of Casper's specification over the next months, as we come to consensus on the protocol's details. In addition, you can look forward to seeing simulations, informal and formal specification, formal verification, and implementations of Casper! But please, be patient: R&D can take an unpredictable amount of time! :)

Twitter   Facebook

Vlad Zamfir

LATEST POSTS

The History of Casper – Chapter 2
07th December, 2016

The History of Casper — Chapter 1
06th December, 2016

Comments

Tymat

Posted at 9:45 pm August 1, 2015.

Great work Vlad et al! Can't wait to see the results of the simulations.

Reply

STAndrews

Posted at 10:24 pm August 1, 2015.

Thanks for the update Vlad! Looking forward to future posts!

Reply

Greg Slepak

Posted at 3:39 am August 2, 2015.

Awesome writeup! Very excited to see if this really feasible. I posed a mini-review plus some concerns here: https://www.reddit.com/r/ethereum/comments/3ff8g5/introducing_casper_the_friendly_ghost/ctogcce

Great work Vlad!

Reply

Vlad Zamfir

LATEST POSTS

The History of Casper – Chapter 2
07th December, 2016

The History of Casper — Chapter 1
06th December, 2016

ıre

Posted at 4:57 am August 4, 2015.

Isn’t the whole point of a blockchain to be able to verify incrementally that each block is a valid transition from previous block? If you need information that isn’t contained in the blockchain then this isn’t a blockchain

If the state of the consensus process is recorded in the blockchain including earmarking of deposits, forfeiture, bonding and unbounding as well as stake holders votes, there is no need to trust external sources as the state of blockchain until block N will be sufficient to know who are the legit validators at block N+1. That’s what Bitshares’ DPOS is doing and it works fine.

Reply

tomtruitt

Posted at 5:05 am August 10, 2015.

I’m confused isn’t everything discussed above stored in the blockhain? it sounded to me that each block is verified, only instead of needing to start to verify from the genesis block one only needs to verify from the last list of bonded miners… ahhh and there it is… the only way to know that list is accurate would be to start at the beginning or rely on a trusted source… or am I still missing something? sorry trying to keep up…

Reply

tomtruitt

Posted at 5:09 am August 10, 2015.

is there any comparison mentioning how DPOS stacks up against POST?

Reply

Vlad Zamfir

LATEST POSTS

The History of Casper – Chapter 2
07th December, 2016

The History of Casper – Chapter 1
06th December, 2016

Brian Coverstone

Posted at 3:14 am August 21, 2016.

It's still a blockchain. One of the definitions of the blockchain i [illegible] ll participants will switch to the LONGEST blockchain. So if a l [illegible] 51% or more resources than everyone else, they can essentia [illegible] t the blockchain to their own choosing and re-mine previous blocks in the chain.

I believe this is a way to prevent that from happening so that everyone will not just automatically accept a longer chain, potentially undermining past transactions, just because it was introduced.

Reply

Alexey

Posted at 2:15 am August 5, 2015.

Thank you for the writeup Vlad! What is the motivation behind penalising the validators for not being able to guess the "correct" fork? I though initially that they will only be penalised for voting on more than one block of the same height. Now they will have to somehow pre-agree off-chain about what they'll all vote, so that they "cleanly" vote for the single fork. And this "off-chain" element can easily turn into some private cartel of people who run validators. New validators, without having access to off-chain pre-agreement, might be at disadvantage. IMHO such penalty would complicate the validation protocol too

Reply

Vlad Zamfir

LATEST POSTS

The History of Casper – Chapter 2

07th December, 2016

The History of Casper — Chapter 1

06th December, 2016

Ariel Gabizon

Posted at 8:52 pm June 13, 2016.

” Therefore, when clients receive and authenticate the state of the consensus, their authentication chain ends in the list of currently-bonded validators. In proof-of-work consensus, on the other hand, the authentication [illegible]n the genesis block –” .. but where is this list of currently-bonded va[illegible]nd how is consensus reached on the content of this list?

Reply

Ariel Gabizon

Posted at 9:01 pm June 13, 2016.

“This “out-of-band authentication only necessarily once” property is what Vitalik calls weak subjectivity. ”

Why only once? Doesn’t this list of bonded validators change all the time?

Reply

Leave a Reply

Add your comment here...

You may use these HTML tags and attributes: `<a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote cite=""> <cite> <code>`

Vlad Zamfir

LATEST POSTS

The History of Casper - Chapter 2

07th December, 2016

The History of Casper — Chapter 1

06th December, 2016

```
"> <em> <i> <q cite=""> <s> <strike> <strong>
```

Name (Required)

Email (Required)

Website URL (Required)

Post Comment

Vlad Zamfir

Recent Posts

- Introduction of the Light Client for DApp developers
- December Roundup
- Security alert [12/19/2016]: Ethereum.org Forums Database Compromised
- Swarm alpha public pilot and the basics of Swarm
- The History of Casper – Chapter 2

Recent Comments

- Christian Reitwiessner on zkSNARKs in a nutshell
- ARIEL GABIZON on zkSNARKs in a nutshell
- Martin Köppelmann on Uncle Rate and Transaction Fee Analysis
- Vitalik Buterin on Uncle Rate and Transaction Fee Analysis
- Martin Köppelmann on Uncle Rate and Transaction Fee Analysis

LATEST POSTS

The History of Casper – Chapter 2

07th December 2016

The History of Casper — Chapter 1

06th December 2016