**7. How is the technology underlying Ethereum similar to and different from the technology underlying Bitcoin?**

To better understand it’s important to look at the technology side by side:

|  |  |
| --- | --- |
| **Ethereum** | **Bitcoin** |
| * Ethereum blocks are mined in seconds (usually 10-17 seconds on average). This allows for faster transaction times and Ethereum does it by using the “Ghost” Protocol | * Bitcoin transactions are confirmed in around 10mins on average |
| * Ethereum releases the same amount of Ether each year * Ethereum rewards miners with 3 Ether( current reward) for each block mined * The amount of Ether earnable through mining is capped at 18 million per year, so there will always be new Ether entering circulation (higher spending value) | * Bitcoin block rewards halve every 4 years * Bitcoin is currently valued at 12.5 Bitcoins, meaning the total supply of Bitcoins will eventually reach 21 million and stop (it has higher holding than spending value) |
| * Ethereum has a different method for costing transactions depending on their computational complexity, bandwidth use and storage needs which is measured by “Gas” and is limited per block, meaning, higher gas transactions would take up a larger chunk of the block | * Bitcoin transactions compete equally with each other and are limited by block size |
| * Ethereum has its own Turing complete internal code - a Turing-complete code means that given enough computing power and enough time, anything can be calculated | * Bitcoin uses a stack-based language and as compared to Ethereum this is not the case |
| * Ethereum discourages centralized pool mining through its Ghost protocol rewarding stale blocks | * There is no advantage to being in a pool in terms of bitcoin block propagation |
| * Ethereum uses a memory hard hashing algorithm called Ethash that mitigates against the use of ASICS and encourages decentralized mining by individuals using their GPU’s | * Bitcoin doesn’t have any such algorithm |

**Our View** – Ethereum is different for all the above reasons and it can be put in simple analogies what bitcoin and Ethereum can be understood as:

* + Bitcoin – is like Western Union
  + Ethereum – is like a World Payment System, that can be automated, personalized, but can never be censored

**8. Does the Ethereum Network face scalability challenges? If so, please describe such challenges and any potential solutions. What analyses or data sources could be used to assess concerns regarding the scalability of the underlying Ethereum Network, and in particular, concerns about the network’s ability to support the growth and adoption of additional smart contracts?**

As with any other blockchain network, scalability is a concern. It’s a trade-off between security and scalability on the blockchain networks. Security on these public blockchains, that handle value transfers, becomes paramount and the algorithms and techniques used to achieve security prevents the blockchain from scaling up.

Ethereum uses a combination of technical tricks and incentives to ensure that they accurately record who owns what without a central authority. This is done, by the current design, by each full node in Ethereum processing every single transaction in the network. This ensures that the network is highly secure but at the cost of scalability.

The problem is that it becomes difficult to maintain the balance between increasing the users and adding more users to the chain. This is because Ethereum depends on a network of ‘nodes’, each of which stores the entire Ethereum transaction history and the current ‘state’ of account balances, contracts and storage.

This is obviously a cumbersome task, especially since the total number of transactions is increasing approximately every 10–12 seconds with each new block. The gas limit is an estimate that regulates the blocksize and every transaction requires a certain amount of gas.

The worry is that, if developers raise the size of each block to fit more transactions, the data that a node will need to store will grow larger – effectively kicking people off the network. If each node grows large enough, only a few large companies will have the resources to run them. In other words, decentralization and scalability are currently at odds, but developers are looking for ways around this.

Potential solutions for solving the scalability challenge:

* **Sharding** - The main Idea behind sharding is that to break down the network into smaller groups (shard) without sacrificing security and decentralization and achieve unlimited scalability. However, sharding doesn’t work with Proof of Work (PoW) Consensus mechanism but with the release of “Casper”, where Ethereum would be implementing a new algorithm – Proof of Stake (PoS), sharding would potentially solve the scaling issue.
* **Off-chaining** - A possibly more ambitious capacity-expanding technology borrows from bitcoin’s Lightning Network, a proposed top-layer to the blockchain that mirrors how the multi-layered internet works. According to this vision, most transactions will be made on off-chain micropayment channels, lifting the burden from the underlying blockchain. The reason that this works, in theory, is that either party can kick the transaction back to the blockchain anytime they want, giving both parties the ability to end the interaction. With this add-on, Ethereum’s computational limit doesn’t need to increase too much, and the hope is that it will still be reasonable for regular Ethereum enthusiasts to run a full node. Two possible options proposed are:
  + *Plasma*
  + *State channels*
* **Raiden** - Raiden is an Ethereum’s version of the [Lightning Network](https://lightning.network/#intro) (Bitcoin) designed to handle micro payment at a fraction of the cost. The network will allow for secure transfer of value off-chain using ETH and any ERC20 compliant token. It’s like set of channels that then allow two parties to exchange unlimited transactions at a fraction of cost and in near real-time.

**9. Has a proof of stake consensus mechanism been tested or validated at scale? If so, what lessons or insights can be learned from the experience?**

**10. Relative to a proof of work consensus mechanism does proof of stake have particular vulnerabilities, challenges, or features that make it prone to manipulation? In responding consider, for example, that under a proof of stake consensus mechanism, the chance of validating a block may be proportional to staked wealth.**

**\*Note – the below response applies to questions 9 & 10\***

The proof of stake consensus mechanism requires the user to show ownership of a certain number of cryptocurrency units (32, which is called the Stake).

The creator of a new block is chosen in a pseudo-random way, depending on the user’s wealth, also defined as ‘stake’. In the proof of stake system, blocks are said to be ‘forged’ or ‘minted’, not mined.

In order to validate transactions and create blocks, a forger must first put their own coins at ‘stake’. Think of this as their holdings being held in an escrow account: if they validate a fraudulent transaction, they lose their holdings, as well as their rights to participate as a forger in the future. Once the forger puts their stake up, they can partake in the forging process, and because they have staked their own money, they are, in theory, now incentivized to validate the right transactions.

The problem that Proof of Stake attempts to solve is related to energy consumption.

**Potential Problems with the Proof of Stake consensus mechanism:**

* The ”**nothing at stake**” problem is grounded in the fact that voting on a particular version of a proof of stake blockchain requires no resources and therefore has no opportunity cost
  + Unlike Proof of Work, where miners must choose which chain to point their mining power at, to the exclusion of other chains, miners can stake their coins on every version of a Proof of Stake blockchain that exists
  + Since there’s no opportunity cost to mining on a particular chain, the miners have ‘nothing at stake’. So rational miners should simply mine on every competing branch they see, so as to maximize the amount of mining returns they get.

Vitalik gives two ways of addressing the nothing at stake problem in his essay, [”On Stake”](https://blog.ethereum.org/2014/07/05/stake/):

1. Introduce into the protocol a way to penalize those who ”equivocated” on a given block, i.e., voted on two different versions of it.
2. Introduce into the protocol a way to penalize those who voted on the wrong block, regardless of whether or not they double-voted.

* **Long Range Attacks** –
* In Proof of Stake chains, in the early stages there will be a relatively small group of miners with coins staked. As more and more users join the chain and obtain the underlying asset, the pool of miners, i.e., the users who have staked coins, becomes larger
* However, after the fact, the original, small group of miners can get together and decide to go back and ‘revive’ that early version of the chain, and since in the ensuing stages they would be the only ones who could mine blocks, they would soon hold a large share of the assets on that chain
* And since there isn’t a limit on the growth-rate of Proof of Stake chains, only how long it takes each chosen miner to mine the next block, these chains can suddenly become extremely long

Most clients, for example Casper and NXT, address this in a roundabout way, by requiring, in their protocols, that only blocks with a certain range of prior blocks (720 in NXT’s case) can be disputed, while the rest are a part of the “main chain”. However, this simply changes the scope of the problem. Under this protocol, nodes will have undefined behavior when they:

1. Come back online after more than the amount of time in the ‘window’ given by the client
2. Come online for the first time

These two cases lead to something called ‘Weak Subjectivity’ (all proof of stake blockchains have this issue).

* Peercoin, for example, simply gets around it by broadcasting the hash of the “legitimate” chain on a daily basis.
* NXT simply ignores the problem, saying in their wiki that since clients automatically reject any changes more than 720 blocks in the past, they are not susceptible to long-range attacks. However, they do [acknowledge](http://nxtwiki.org/wiki/Whitepaper:Nxt#Nxt.E2.80.99s_Proof_of_Stake_Model) the problem’s existence.
* Vitalik [acknowledges](https://blog.ethereum.org/2014/11/25/proof-stake-learned-love-weak-subjectivity/) that CASPER will need to depend on trusted nodes to broadcast the correct block hash.

**11. There are reports of disagreements within the Ether community over the proposed transition to a proof of stake consensus model. Could this transition from a proof of work to a proof of stake verification process result in a fragmented or diminished Ether market if the disagreements are not resolved?**

This question has a simple fact based answer posted already on Github. We’ve researched extensively and don’t have a point of view which would contribute.

**12. What capability does the Ethereum Network have to support the continued development and increasing use of smart contracts?**

Developers could program a truly decentralized application (DApp) on the Ethereum blockchain using smart contracts – a simple program that would automatically execute an action as soon as a specific criterion was fulfilled.

There are plenty of great projects already demonstrating how dapps can improve user experience. For instance, ENS ([Ethereum Name Service](https://ens.domains/)) is a solution that allows users to turn their hexadecimal wallet address into a unique domain name instead.

Projects like these encourage ideas that might lead to developers making Dapps to appeal to the man on the street.

The open sourced Ethereum clients being developed out there for different programming languages makes the network more decentralized, robust, and secure.

“Constantinople”, a big upgrade to the Ethereum chain, features small, yet highly technical, Ethereum improvements to network efficiency and fee structure, as well as, upgrades that pave the way to Ethereum’s hotly anticipated scaling roadmap.

The five EIPs set to be released in Constantinople are:

* EIP 145: A technical upgrade written by two Ethereum developers, Alex Beregszaszi and Pawel Bylica, EIP 145 details a more efficient method of information processing on Ethereum known as bitwise shifting.
* EIP 1052: Authored by core developer Nick Johnson and Bylica, EIP 1052 offers a means of optimizing large-scale code execution on Ethereum.
* EIP 1283: Based on EIP 1087, which was written by Johnson, this proposal mainly benefits smart contract developers by introducing a more equitable pricing method for changes made to data storage.
* EIP 1014: Created by the founder of Ethereum himself, Vitalik Buterin, the purpose of this upgrade is to better facilitate a certain type of scaling solution based upon state channels and “off-chain” transactions.
* EIP 1234: Championed by Afri Schoedon, release manager for major Ethereum client Parity, this upgrade is the most contentious of the batch, reducing block mining reward issuance from 3 ETH down to 2 ETH, as well as, delaying the difficulty bomb for a period of 12 months.