**Infosys**

1. **In your recent project who all are stake holders in your network?**

My Project is related to crowdfunding and has three stakeholder

* Manager - He / she is an idea person who will create spending request.
* Vendor - He / she will receive money from the campaign for specific purpose.
* Contributor- He / she will contribute for a campaign and vote for every spending request.

1. **In your current project what you are trying to achieve in Blockchain?**
2. **We could have easily achieved this (current project) using web services, why did you use Blockchain?**
3. **Your project is built on which Blockchain framework and why did you choose this?**
4. **Apart from transparency what all things blockchain brings?**
5. **In your project which cryptocurrency are you using. How payment is done?**
6. **For which client you are doing this project?**
7. **Many countries have banned ICO, what is your take on this?**
8. **When you use Ethereum, did you consider scalability? What are the challenges you faced?**
9. **What is the difference between proof of work (POW) & proof of stake (POS)?**

Proof of Stake (POS) and Proof of Work (POW) are algorithms for reaching consensus on the blockchain. Anyone can create a block; while we only want an unique chain, so we want a way to decide which block we should trust.

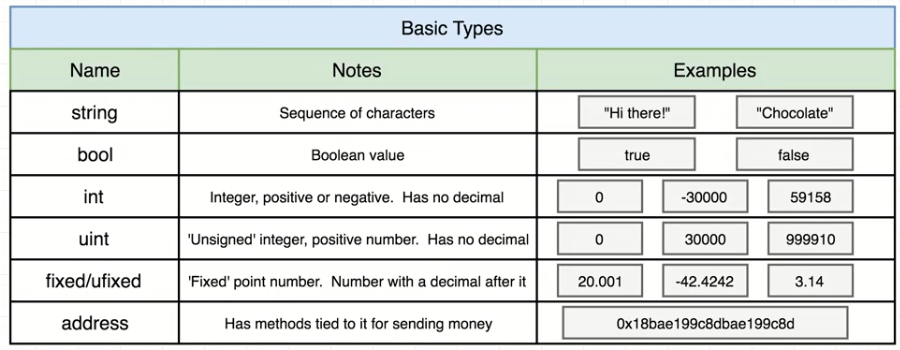
The goal of a consensus algorithm in a public blockchain network is to let many different users agree on the current state of the blockchain even though they don’t trust each other or any central authority.

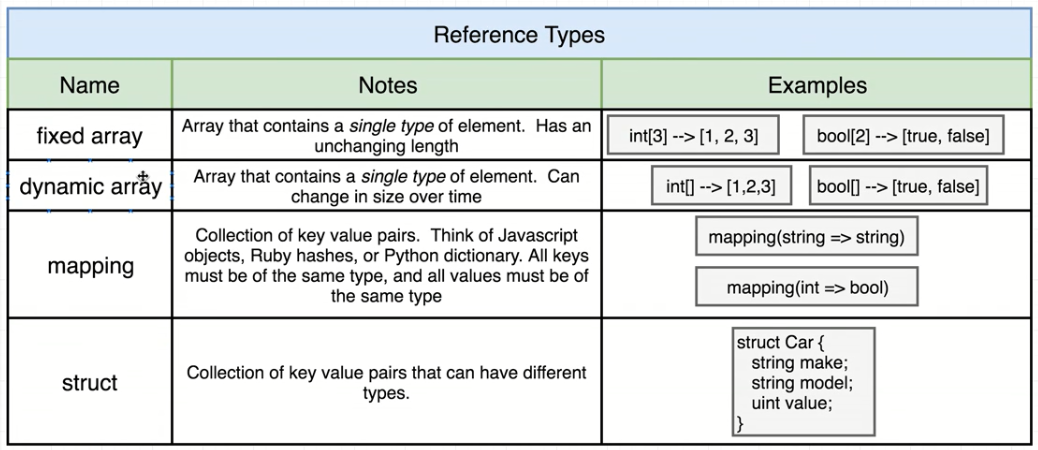
Proof of work has the nice property that you can use Bayes’ Theorem and the laws of Thermodynamics to prove that a given block has indeed required a certain amount of work to be mined. That way, users can simply pick the longest valid chain with the highest amount of work as the correct chain. This implies that Proof of Work is extremely inefficient in term of energy, and therefore also very expensive which incentivize miners to centralize the hashing power.

Proof of Stake on the other hand, isn’t about mining it’s about validating. In effect blocks still need to be created by someone, and who gets to create the next block depends on the specific Proof of Stake algorithm, but the selection process must have some kind of randomness, or at least distribute voting shares. POS is greener because you’re not intensively making computing calculations. You have to essentially lock up your coins/tokens to mine or process transactions.

In summary. POW is a process to validate transactions with a lot of computational expensive operations where in POS the transactions are validated by voting or beting with you money, your “stake” on the right blockchain.

1. **Beauty of consensus algorithm is democracy, but in POS anyone can act as a miner and solve most of the puzzles. Can’t you think it will spoil the democracy.**
2. **Transparency is achieved by sharing the information, In this case how you scale the system? because I have to keep purchasing hard disk as long as the network grows. So many transactions are happening in the network, my memory space are always up. Pulling the existing record will always a time consuming for me. What is the mitigation for this?**
3. **What are the smart contracts you have written?**
4. **What is the core structure of smart contract? How it will start?**
5. **What are the different datatypes being available in Solidity?**





1. **Which all tools are used for developing Ethereum Smart contract?**

**Truffle**: A development environment, testing framework and asset pipeline for Ethereum. In other words, it helps you develop smart contracts, publish them, and test them, among other things.

**Ganache**: It was called TestRPC before. What Ganache does is simple, it creates a virtual Ethereum blockchain, and it generates some fake accounts that we will use during development.

**Mist**: It’s a browser for decentralized web apps. It seeks to be the equivalent of Chrome or Firefox, but for Dapps.

**Ethereum wallet**: It’s a version of Mist, but only opens one single dapp, the Ethereum Wallet. Mist and Ethereum Wallet are just UI fronts. And we need a core that will connect us to an Ethereum blockchain

**Geth**: Is the core application on your computer that will connect you to a blockchain. It can also start a new one, create contract, mine ether etc.

1. **While writing a contract, how do you estimate the gas? Are you doing manual calculation or any tool you are using?**
2. **For basic transfer or a transaction how do you measure the gas required?**
3. **How you are generating bytecode for contract?**
4. **How you are integrating your Ethereum code with UI?**

Web3 is a JavaScript API that wraps RPC calls to help us interact with a blockchain in an easy way.

1. **What command are you using for compiling the contract?**
2. **What is truffle & geth. How to compile using these tools?**

Truffle: A development environment, testing framework and asset pipeline for Ethereum. In other words, it helps you develop smart contracts, publish them, and test them, among other things.

*truffle compile  
truffle migrate --network development*

*Compile* will compile our Solidity code to bytecode (the code that the Ethereum Virtual Machine (EVM) understands), in our case, Ganache emulates the EVM.

*Migrate* will deploy the code to the blockchain, in our case, the blockchain could be found in the network “development” we set earlier in the “truffle-config.js” file.

Geth: Is the core application on your computer that will connect you to a blockchain. It can also start a new one, create contract, mine ether etc.

1. **What is the difference between Hyperledger fabric & sawtooth? What consensus algorithms they are using?**
2. **Even Fabric support growing network why we have to go for sawtooth?**
3. **How do you deploy chaincode in fabric?**
4. **Which version of Fabric you have used?**
5. **Which Ethereum version you have used?**
6. **What coding issues you faced in Solidity?**
7. **Once you deploy contract, Is it accessible to all participant in the network?**
8. **Access is given to everybody or there is no restriction for the contract?**
9. **If you want to modify the contract, how do you update the contract?**

Contracts deployed on a blockchain are immutable. This means that the address and the code of that Smart Contract can’t be modified since it’s permanently written on the blockchain.

However there are ways to use a new contract instead of that one, which is similar to “updating” a contract. One way to update the code is to create an intermediary Smart Contract that will hold the address of the active Smart Contract. So all the calls and transactions will be redirected to the active version with the function delegatecall. That way, you’ll be using the same contract address but that contract will execute a different Smart Contract code in the end. From an “updated” contract.

Note that there are important security risks when delegating calls and that you have to be extra cautious when designing upgradable smart contracts.

Another way is to simply extract all the information from the old contract and insert it into a new version. Then update the address that your users will see.

1. **How to get access of new deployed contract?**
2. **You have modified the contract, how others will come to know that you have updated the contract.**
3. **What is Forking?**