SMITH+CROWN

ORIGINAL RESEARCH

Blockchain Tech Series

PART 1: ETHEREUM, EOS & HYPERLEDGER

This memo aims to provide a technical overview of the most popular blockchain platforms by exploring their core components and examining how each of them interact to enable a fully functioning blockchain platform.

Overview

'Blockchain.' While many now associate the word with cryptoassets, such as Bitcoin, Ethereum or Ripple, the underlying blockchain technology enabling these cryptoassets largely remains a mystery to mainstream audiences. Despite this general lack of familiarity, experts in certain fields such as IT are beginning to acknowledge the potential for blockchain technology to have great impact on their industry—given the right use case and implementation, blockchains can enhance security, improve data coordination, lower transaction costs, increase trust and enable transparency. Large enterprises have started embedding the technology into their operations: HSBC used blockchain to transact $250 billion of foreign exchange transactions in 2018; a consortium of leading Indian insurers are using blockchain to enable cross-company data sharing; the Department of Homeland Security is exploring blockchain for anti-forgery solutions.

This piece examines the core underlying technology, to help understand what makes leading blockchain platforms (Ethereum, EOS, Hyperledger) so promising by demystifying some blockchain fundamentals, including:

Consensus

How the entire network comes to agreement

Governance

Who's responsible for upkeep of the system

Architecture

Major components that make up the system

Scalability

Throughput of the system

Security

As applied to the system and user

Finality

How long it takes for the system to finalize a transaction

Longevity

How long the system has been operating

Developer Friendliness

The experience of building on this system

Key Builders

Highlighting important enterprises, communities, etc.

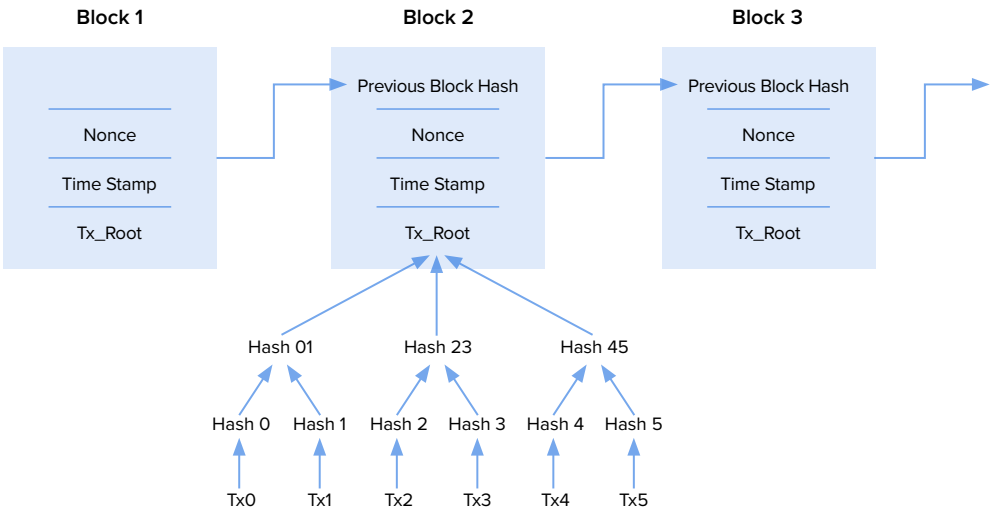
Unpacking and comparing these variables across several leading blockchain platforms will help readers better understand how each project's technology functions, the relative development of respective current applications, and the implications on future project trajectory.

What is Blockchain Technology?

In the simplest of terms, blockchain technology is an immutable ledger that records transactions into blocks. A collection of transactions forms a block and the interlinking of blocks forms a blockchain. Once a transaction is placed in a block, the block cannot be reversed to change the contents of a previous transaction—unless the security of the system is compromised. A user can only append new details to a previous transaction—all previous transactions are still contained in the ledger's immutable history. Diagram 1 shows the contents of each block and how blocks are interlinked to one another to form a chain.

Diagram 1

Blockchain Formation



From 2008 to 2013, most blockchains were designed to be a single purpose ledger that transferred assets between accounts. Bitcoin, for example, was designed expressly to facilitate the transfer of 'Bitcoin' between two parties. Early projects emerging in 2014 started exploring other possibilities, such as data storage (Sia) and computation (Ethereum), which sparked the beginning of multi-use blockchain technology. As of April of 2019, the most popular multi-purpose public and private blockchains as measured by integrations are Ethereum, EOS and Hyperledger. These blockchains are particularly useful to examine first because their widespread influence is seen in other blockchain designs.

Ethereum

Pioneering the concept of 'smart contracts', which enable multi-purpose computing capabilities, the Ethereum project was founded to add functionality beyond a single purpose ledger. Prior to Ethereum's development, Bitcoin's technology could only store and record Bitcoin transactions, as it was meant for one specific use. By way of background, a smart contract is an executable code that is triggered to run when specific, pre-programmed conditions are met. Inputs to a smart contract can consist of several conditions but outputs will almost always result in a transfer of value (assets) between accounts or in the storage of data onto the chain. Ethereum's design principles enable a generalized blockchain architecture through a bare minimum feature set which ensures that everyone is free to access the network. In many aspects, Ethereum's approach is similar to the Linux Kernel design, whereby the core software is simple and bare-minimum, and developers build their systems by adding their own specialty features.

> Ethereum's approach is similar to the Linux Kernel design, whereby the core software is simple and bare-minimum, and developers build their systems by adding their own specialty features.

Consensus

Ethereum currently uses Proof-of-Work (PoW) consensus to secure its chain, much like Bitcoin. Miners use computer or graphics card's computational capacities (hash power) to solve a predetermined puzzle set by system. Each puzzle is set so to take mining rigs roughly fifteen seconds to solve. Miners who solve the puzzle get to create a block (and receive compensation as a result), with the miner being responsible for placing any pending transactions into the block. Hence, transactions that pay higher fees (paid with its native token Ether) are preferred by miners over ones with lower fees, irrespective of how long a transaction has been pending.

Blockchain implementations face a common set of tradeoffs known in the industry as the scalability trilemma: an architecture must pick two among decentralization, scalability and security. A network is considered decentralized if anyone can join it, scalable if it can process a sufficiently high number of transactions, and secure if it can prevent attacks. Etherum's current proof-of-work consensus prioritizes decentralization and security at the expense of scalability; as of April of 2019, it can only process 7-9 tractions per second (further explained in scalability section).

After a block is filled with transactions, it is sent to other miners, who confirm the transactions in the block. Once a block is confirmed individually by at least 51% of the network, it is considered confirmed and added to the blockchain. If, by random chance, two miners solve the puzzle at the same time—effectively creating a block of their own—the block that propagates first through 51% of network miners will be added to the network. All other blocks are called 'uncle blocks' and transactions contained within them are voided. This method for confirmation allows for situations where a double-majority splits the network, each group choosing to add its own block to the system. In such cases, the chain splits into two, commonly known as a 'fork.'

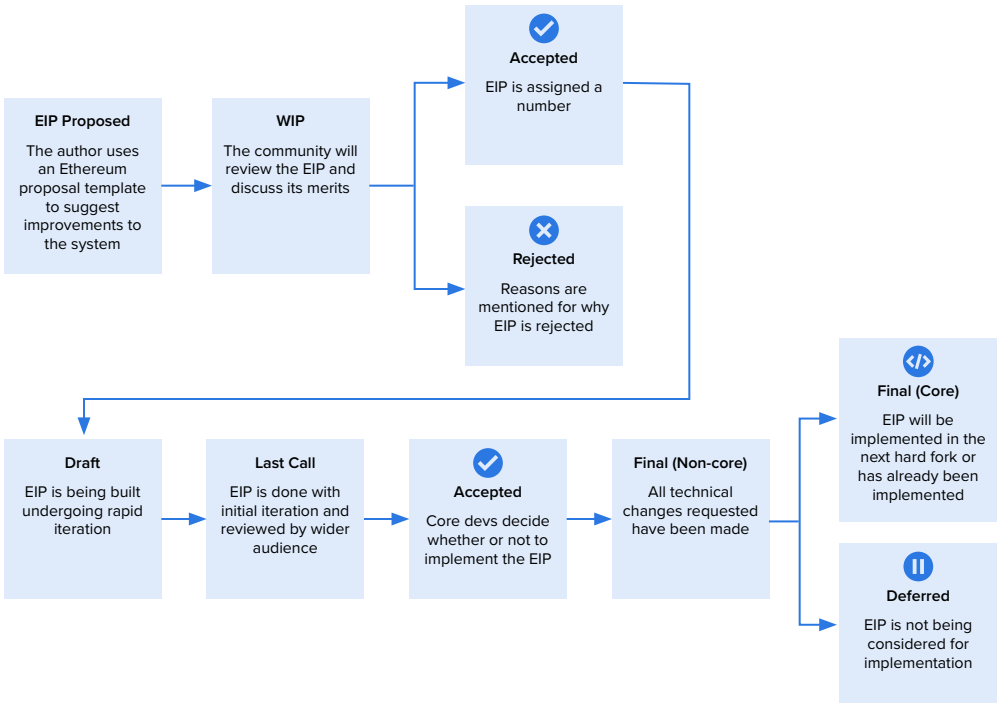
Governance

In a PoW system such as Ethereum, governance is usually handled by the miners, who bring computing power to the chain. There is no formal or on-chain governance structure, miners can simply choose to fork the chain in any case of ideological disagreements. The DAO attack, where a malicious actor stole funds from a development fund, precipitated one such event. Miners conducted an ad-hoc vote, with the majority of miners choosing to 'reverse' the transaction by rolling back the chain to its previous state (i.e. choosing that fork). The miners that disagreed with the majority decided to continue running the old chain, which is called Ethereum Classic, and the majority ended up contributing their hash power to the new chain, which retained its original name: Ethereum. In theory, miners choose to contribute their hash power to the chain most likely to have a higher price, which would increase the effective value of their mining reward.

Miners are primarily responsible for maintaining the system, and updating the chain to add any new features. System upgrades are handled through an EIP (Ethereum Improvement Proposal) process, where developers submit a EIP request, which is then debated and developed before being implemented. Once the core developers support a request, the EIP is added for the next scheduled hardfork. Diagram 2 shows the detailed flow of the EIP process. In practice, Ethereum's core foundation has a lot of say over the direction of the Ethereum ecosystem; one of its founders, Vitalik Buterin, has a significant following in the mining community. Any work or updates endorsed by Vitalik will likely be adopted into the Ethereum ecosystem. Whether such an influence has negative repercussions in the decentralized ecosystem is widely debated, and is perhaps an aspect to watch out for as the ecosystem matures.

Diagram 2

EIP Proposal Process



Architecture

There are three core components that make up the Ethereum architecture: clients, miners and the EVM (Ethereum Virtual Machine). To interact with the core blockchain, there has to be an Ethereum client setup in order to talk to the EVM. There are currently 15+ clients available — all meant to serve the same purpose of translating high level code to ethereum bytecode for the EVM to understand. In Ethereum, transactions are used to store data and change the state of the system — any change to the core blockchain takes the form of a transaction. A transaction is usually in the form of a smart contract, which instructs the EVM of what to do. Once a user creates a smart contract, they send it along to the client (either hosted by the user or a trusted third party), which adds it to mem (memory) pool. When the miners create a new block, they pick transactions from the mem pool and, once the block is confirmed that state of system is changed.

Scalability

Ethereum can currently process roughly 7-9 transactions per second. In comparison, a traditional AWS instance can handle upwards of a million transactions per second. During times of congestion—when the chain is being asked to process more transactions than it is able to—transactions are prioritized by fee amount and many low-fee everyday transactions are delayed. Ethereum's scalability is a well-known and frequently lamented limitation and a variety of upgrades are currently being worked on. These include 'Layer 2' solutions that move computation off the main chain and a major network-wide upgrade that will break the entire chain into a

series of smaller networks called 'shards.' The specifics of these upgrades will be discussed in greater detail in forthcoming papers. Suffice to say, lack of scalability is a prominent criticism of Ethereum and, in response, many companies are building on private instances of the public ethereum codebase; also many dApp developers are either looking to implement Layer 2 solutions or simply launching projects that do not need high transaction throughput.

Security

Ethereum's security can be split into 2 three parts: core chain security and application security. With respect to core chain security, as a general rule of thumb, a chain becomes more secure as its computational (hash) power increases. This is because it becomes extremely hard for any one party to acquire 51% of compute power of the chain. In the event that one miner has 51% of compute power in the chain, they can compromise the security by reversing the transactions in the previous block, essentially allowing them to double spend; a double spend is where the same set of coins are used in multiple transactions.

Application security concerns two types of attacks: bugs that cause annoying or catastrophic failure and intentional attacks. Bugs in application code, primarily caused by a lack of security audits prior to an apps' launch, can lead to potential attacks. Intentional attacks occur when an attacker actively looks to exploit the code with a malicious intent. A famous example of this is the DAO attack, where an attacker found a bug in the application code and started draining the fund—the core ethereum chain that processes transactions and enforces cryptography had no security problems. With the emergence of security audit firms, application security attacks are drastically reduced. A potential contributing factor to both of these types of attacks is the programming language that Ethereum uses, Solidity. Solidity's 'loose' syntax is designed to make development easier but has also led to an increased likelihood of making mistakes.

Finality

Finality is considered to be the time it takes for users to trust a transaction or the time it takes for the transaction to be considered final and irreplaceable. The core Ethereum chain processes a block of transactions every 10 - 15 seconds, but the industry generally requires a certain number of successive blocks as 'confirmations.' It roughly takes around five to six minutes to confirm (finalize) a transaction once users submit it to the chain (assuming that the transaction is paying fees that are on par with others in the ecosystem). Yet, the security measures taken by implementers can vary these confirmation times greatly. For example, Coinbase requires 35 confirmations that, on average, take 7 minutes vs. 12 confirmations on KuCoin, which takes about 4 minutes. The general rule of thumb is that more confirmations increases the chance making the transaction irreversible, with a diminishing rate

of return after a 25 confirmation threshold. Applications currently being built on Ethereum account for this delay in one of two ways. First, transactions requiring instant confirmations are processed on the side chain and are periodically dropped into the main chain to maintain proof. Loom's DAppChain enables such optimization, where each dApp has its own chain and computational power can be scaled up to meet the application's demands. Second, some applications are built accounting for this delay; for example, Crypto Kitties, a popular Ethereum-based game, addressed the problem by employing a breeding period that accounts for longer confirmation times, to set user expectations and ensure that the game is still enjoyable.

Longevity

Ethereum officially launched in July of 2015 and, since then, has gone through three major updates. The first major update occured in March of 2016, dubbed Homestead, which added more efficiency to transaction processing, security updates and gas pricing. The next update, named Byzantium took effect in 2017 which reduced the EVM's complexity and provided more flexibility for smart contracts. Lastly, the latest Constantinople update in Feb. 2019 makes it 10x cheaper to run smart contracts on Ethereum.

Developer Friendliness

Due to Ethereum's simple design philosophy, the developer usually bears the responsibility for creating the tools needed to develop the app. In the past, this usually translated into poor design, extended development times, and apps riddled with bugs. As a result, third-party companies like Consensys stepped in and started creating developer tools so that developers could concentrate on an app's business logic. Frameworks such as Truffle started to create common design principles that prompted new thinking about how smart contracts should be structured. This drastically cut down the development time involved in launching new contracts. It also greatly enhanced application's security, as security audit firms can parse through the frameworks much more thoroughly, given their deep understanding of how the framework functions.

Key Builders

Developers may build on the Ethereum chain in two ways: by building on the public chain or by developing on a private chain. Most enterprise companies choose to build on their own private instances of Ethereum. The Enterprise Ethereum Alliance (EEA) was formed in March 2017, with thirty founding members to discuss and address the needs for the enterprise by designing specifications for large scale implementations. Today, it has over 150 members that participate in creating these open standards. Other private instances of Ethereum include JP Morgan Chase, which developed a private Ethereum blockchain meant for the financial services and the foundation technology for JPM coin.

The public chain has also seen notable developments. Companies that choose to develop on the public chain typically do so to increase trust and transparency between customers. For example, AXA, an insurance group based in Japan, was one of the first to launch a public blockchain project. AXA aims to be completely transparent in how the payouts are handled; when a plane is delayed by over two hours, the company automatically paid its policyholders. Currently, there are 2500+ dApps built on the Ethereum public blockchain.

EOS

EOS, like Ethereum, is a smart contract platform which introduced a new architecture that promised greater scalability in exchange for partial decentralization. Partial decentralization represents an attempt to avoid both complete decentralization (where no one party has complete control of the system's state) and centralized system access controls (where a set of administrators have complete control over the system), both of which have downsides. EOS strikes this balance between complete centralization and decentralization where a group of 21 Block producers are responsible for maintaining the state of the system and token holders vote on the selection of Block producers. EOS is designed by Block One, which raised an estimated $4 billion in a year-long Initial Coin Offering just for the development of EOS. Examining EOS' components clarifies how the project is meaningfully different from other blockchains.

> To allow developers to focus on core app development efforts, EOS has adopted a design philosophy that abstracts the bulk of cryptography and application security design efforts.

Consensus

EOS uses Delegated Proof of Stake (DPoS) consensus algorithm to maintain its chain. A set of 21 block producers are chosen by continuous voting to form blocks for a round (roughly two minutes). Each round consists of 252 blocks, where a block producer is chosen to produce 12 blocks each. Each EOS token holder must lock (stake) their tokens to vote for a block producer and each EOS account can vote up to 30 block producers at once. Votes cast by ballots held over a week are weighed less, losing roughly 1% voting power, and unused ballots expire after two years. A new block is produced every half second, whereby any pending transactions in the RAM are added to the network. Once a block is created, it is sent to other block producers for confirmation. Blocks receiving 15/21 votes are deemed as confirmed on the blockchain. In EOS, a collection of block producers has the authority to rollback a transaction, which strongly distinguishes the project from other systems where such an action is not possible.

Governance

EOS also has a governance aspect built into the chain. EOS intends for decision making to operate similarly to a democracy, whereby each EOS token holder votes for a Block Producer and BP's make decisions on critical system upgrades and maintenance. EOS also differs in ideology where the "intent of code is law",

as opposed to other blockchains where 'code is law' is maintained as a guiding philosophy to be interpreted literally. Each EOS developer publishes a Ricardian contract, where they state the intent of the code so that the end user is always protected from any resulting bugs (the block producers have the authority to 'reverse' transactions), this protection isn't a feature in Ethereum.

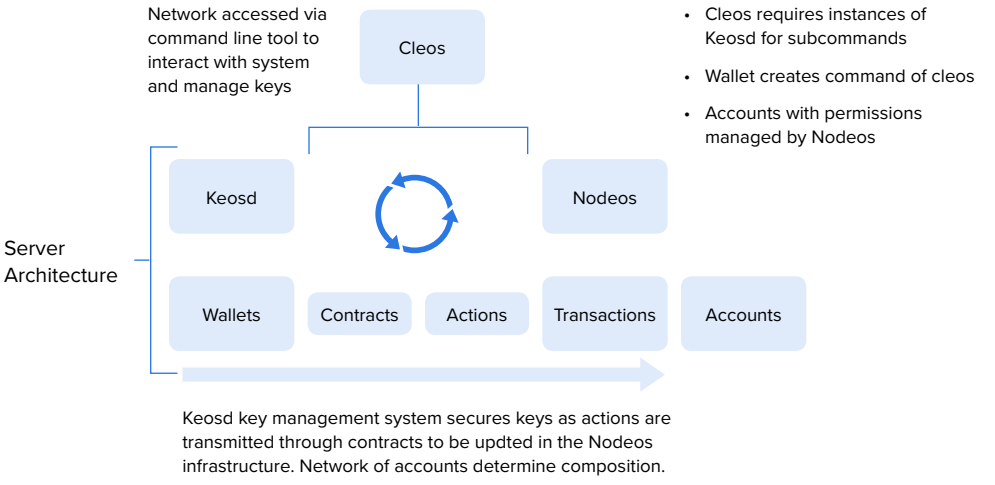
Architecture

The core EOS architecture consists of three different components: Cleos for client interaction, Nodeos for data storage and, and Keosd for storing the private keys that act as the authority source to interact with the Nodeos (DBMS). The primary difference in EOS relative to most public chains is how the accounts are handled: it has a permission-based architecture similar to that of a traditional databases, where Block Producers can elevate or block access to user accounts. Block Producers in EOS can be thought of as system administrators, since, given such permissions, they could roll back, or block a transaction—yet they cannot change a transaction's contents. Corporations, being aware of these implications, generally opt to have private implementations of EOS, where a single organization or a set of trusted actors serve as system admins (Block Producers). These three components control all aspects of the system starting with wallets (where EOS tokens are held), contracts and actions (where logic is deployed), transactions (where state changes are recorded) and finally leading to accounts (which is a history of all tokens, contracts and transactions). Diagram 3 shows the interaction with all three components and the different modules of the system.

Diagram 3

EOS Architecture Overview

Adapted from: Xu, Brent, et al. "EOS: An Architectural, Performance, and Economic Analysis."



There are three main categories of resources on the EOS chain: Bandwidth, CPU and RAM. EOS token holdings correlate to the amount of network allocation, for example 1% of EOS holdings grants 1% of network Bandwidth, and CPU. While bandwidth and CPU can be used in accordance with holdings, RAM cannot and requires staking of EOS tokens. RAM staking has a 2% fee attached to it, a percent each when buying

and selling, which is meant to discourage RAM market arbitrageurs. There are no transaction fees in the network but app developers are expected to hold enough network capacity (EOS tokens) for the app to be operational. This a stark difference from other public chains where the end user is expected to pay the computational costs.

Scalability

Since Block Producers know which blocks they are producing beforehand, the system has an inherent efficiency in transaction routing and confirmation. EOS is thought to support up over 1000 transactions per second, although 70 tx/second is the better evidenced rate as of April of 2019. As the blockchain grows, block producers can collectively vote to increase system capacity, which further improves system scalability. A limiting factor in EOS is the amount of RAM used in the system, primarily because the data will persist forever in RAM (until deleted) whereas CPU and Bandwidth is only used sporadically. EOS currently has 66gb RAM limit, which means data intensive applications will not be able to run on the blockchain.

Security

The core chain security in EOS, is maintained as long as 15 out of the 21 block producers are not compromised. To allow developers to focus on core app development efforts, EOS has adopted a design philosophy that abstracts the bulk of cryptography and application security design efforts. This design philosophy leads to a more secure application layer, where communications between the app and the network are handled securely. Applications still run the risk of having bugs but Block Producers can rectify the situation if they find any malicious code by rolling back the transactions.

Finality

As mentioned previously, since the block producers in the system are known and fixed at 21, confirmation is achieved relatively quickly. Current EOS architecture has a 330 block delay, which means that it takes roughly 165 seconds to make the block/transaction irreversible. In reality however, confirmation is nearly instant as blocks are confirmed within seconds and are just queued up to be deemed as final. This is in contrast to 'finality' in Ethereum, which is judged by block height and not coded into the protocol a transaction several blocks deep is considered 'more' final than a transaction one block deep. In the case of rollbacks in EOS, a counter transaction is issued to the original and it takes the same amount of time to make it irreversible.

Longevity

EOS officially launched its network in June of 2018 after a delay due to its software vulnerabilities. After launch, the network faced another issue with its block producers where it was estimated that a group of block producers were colluding with one

another, which seriously compromised the chain's security. Block One, which has a vested 10 year stake in EOS, is primarily responsible for releasing any updates to the ecosystem. Most of the development in late 2018 and early 2019 focused on updating the core system framework and security.

Developer Friendliness

EOS has a philosophy of supporting a robust ecosystem where developers have to just worry about the code. This makes EOS a very developer friendly ecosystem as they provide the development toolkits needed for developers to get started quickly. In a lot of ways, EOS can be compared to many Cloud Service Providers, where they provide tools to get started on their ecosystem. For example, EOS provides developers with its own smart contract toolkit, API for developers and command line tools—thus significantly cutting down the development time.

Key Builders

EOS, like Ethereum, follows a similar path of public and private implementations. Companies like Strong Block and EOS PRO have been working on enterprise implementations of EOS primarily meant to offer better data protection and security. On the public chains, EOS development fund encourages startups to build on chain and, for the most part, EOS apps have been cloned version of Ethereum apps. Block One, the development company for EOS, claims that there are at least 260 projects being built on the platform, some of which are notable projects like Bancor porting over from Ethereum.

Hyperledger

Hyperledger is the brainchild of the Linux Foundation and is intended as a scalable blockchain platform for enterprise use. Hyperledger's system is meant to be used for permissioned or private systems where each participant has to be added by an administrator ('Membership Service Provider'). Hyperledger's permissioned architecture allows for a two-tiered data access mechanism: users are first given network credentials and are subsequently granted access to specific portions of the network.

> Hyperledger's permissioned architecture allows for a two-tiered data access mechanism: users are first given network credentials and are subsequently granted access to specific portions of the network.

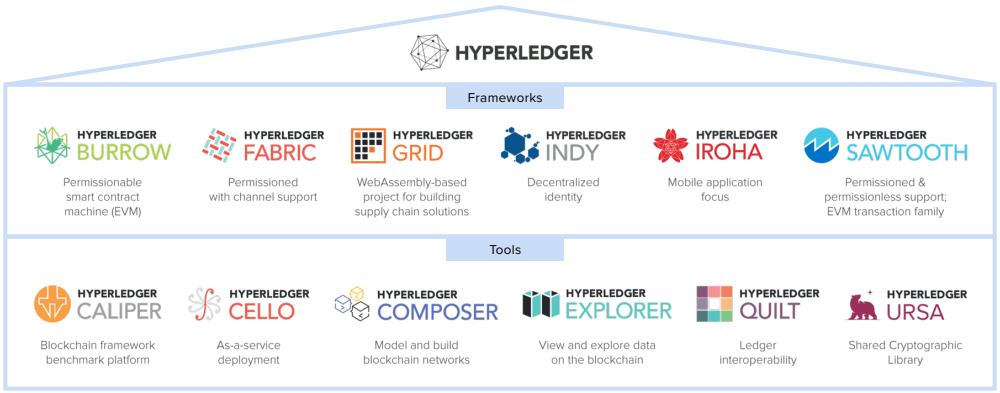
Hyperledger, unlike Ethereum and EOS, is not a codebase of a network but a consortium of several different Hyperledger frameworks, each suited for a specific purpose:

- **Hyperledger Burrow** is an Ethereum-based Smart Contract Machine that executes Ethereum based contracts in a permissioned architecture
- **Hyperledger Fabric** is IBM's general purpose blockchain compute platform similar to that of Ethereum
- **Hyperledger Indy** is a decentralized identity protocol that utilizes on-chain key and access management solutions across the whole enterprise
- **Hyperledger Iroha** is the smart contract platform meant to be used on mobile applications, it uses gossip protocol where each Iroha peer can act as a validating node
- **Hyperledger Grid** is framework for supply chain solutions
- **Hyperledger Sawtooth** is also a general purpose blockchain platform that was originally started by Intel and requires an Intel chip and SGX environment

Collectively, these six frameworks make up the Hyperledger universe, each suited for a different purpose. Along with these frameworks Hyperledger also has six different tools that help developers create robust applications that meets all enterprise needs. Diagram 4 shows different frameworks and tools available for the Hyperledger Ecosystem. Hyperledger Fabric is the project's most popular general computing platform, as measured by presumed implementations, so is worth focusing on.

Diagram 4

Hyperledger Ecosystem
Source: "Hyperledger – Open Source Blockchain Technologies." Hyperledger, The Linux Foundation.
https://www.hyperledger.org/



Consensus

In Hyperledger Fabric, consensus is highly customizable according to an organization's needs. Current consensus algorithms include Apache Kafka, Honey Badger BFT, Simplified Byzantine Fault Tolerance (SBFT), and BFT Smart. Each specific consensus algorithm is a trade off between speed, scalability and confirmation (finality); the trade-off being that only algorithms can only realize two of the three properties at the same time. For example, an algorithm like Kafka has high speed throughput and confirmation but is only moderately scalable. One of the problems with Kafka is that the system is not in consensus if there is just one malicious or faulty node. In comparison, some of the BFT algorithms offer instant finality but, as more nodes join the network, it takes more time to reach consensus in the network. Depending on the specific application use cases, the network administrator will choose the appropriate consensus mechanism. In some cases, custom consensus algorithms can be created if none of the current algorithms match application specific requirements.

Governance

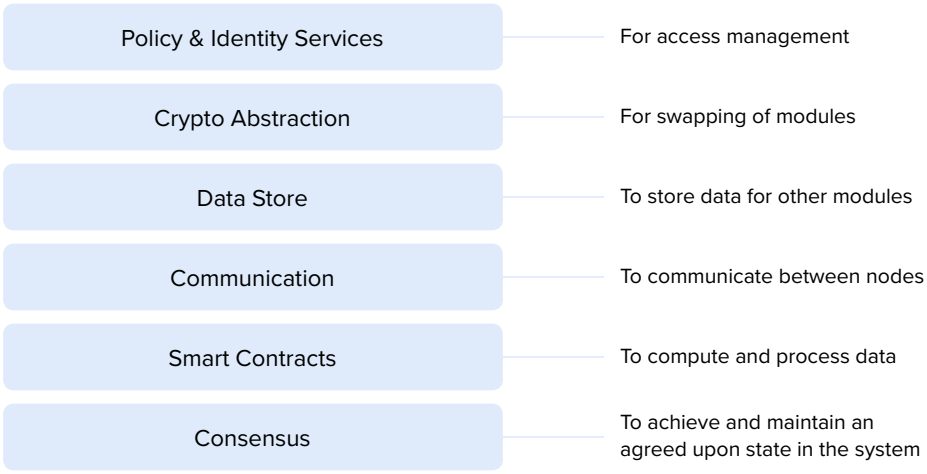
Since all of Hyperledger Fabric implementations are going to be on a private or permissioned chain, the governance burden falls on the system administrators themselves. Each organization designs their own governance process according to the needs of applications or commonly adapted software governance practices. Hyperledger, like the Linux Foundation, follows an open governance model. The community elects developers from an active pool of participants to serve on the Technical Steering Committee (TSC) and the TSC has the ultimate authority on all technical decisions.

Architecture

Hyperledger consists of different architectures across all its projects with a design philosophy for each of its different components, thus enabling each module to be interoperable with the whole system. The Hyperledger ecosystem is composed of several different layers (as laid out in diagram 5):

Diagram 5

Hyperledger Architecture



For the consensus layer, the core algorithm has to satisfy two requirements: safety (all nodes should provide the same output given the same input) and liveness (a live node in the network should receive all transactions). For smart contracts, the architecture has to authenticate any given contract and produce consistent outputs overtime. Other components of the system should at a minimum be interoperable and provide a safe execution environment.

Scalability

Hyperledger's scalability depends on the ordering service (consensus algorithm) used. Apache Kafka is estimated to have rechead 2 million writes per second, although in practice it is estimated to handle about 280,000 transactions per second on AWS Instances. BFT based ordering service, on the other hand, can only handle 10,000 transactions per second.

Security

As Hyperledger blockchains are private or permissioned, they usually sit behind a company firewall. This is crucially important, as the chain's core security aspects are usually not as strong as those of public chains. Since there is an inherent tradeoff between security and scalability, most companies, by design, choose more relaxed on-chain security. For example, in Apache Kafka, compromising just one node will effectively bring the blockchain to a halt, so while companies tend to keep the security of the chain relaxed, they typically protect individual nodes with traditional security measures, such as software and hardware firewalls.

Finality

Depending on the consensus algorithm used, the confirmation could be instant or it could take a couple minutes for a transaction to be considered final. Most instances are designed to be around specific app use cases and hence adjust for this speed when designing the overall experience.

Developer Friendliness

Hyperledger Fabric is a well-funded project by the Linux Foundation with numerous enterprises contributing to the development of the core project. Hence, most implementations ship out with pre-built templates and documentation. IBM, as of April of 2019, reported that 159 engineers are actively working on Hyperledger Fabric and over 3000 developers are using the project to build their apps, so ample resources can be found about common errors and potential workarounds. Though these open source developer tools are still in the incubation stage, IBM's Cloud services team has built proprietary tools on which enterprises can deploy their solutions.

Key Builders

Hyperledger is the most used blockchain in the enterprise, thanks in part to IBM's early marketing efforts, IBM claims that over 250 organizations are part of the Hyperledger development or actively implementing Hyperledger. Companies developing on Hyperledger belong to sectors like Finance, healthcare, IoT, and supply chain.

Conclusion

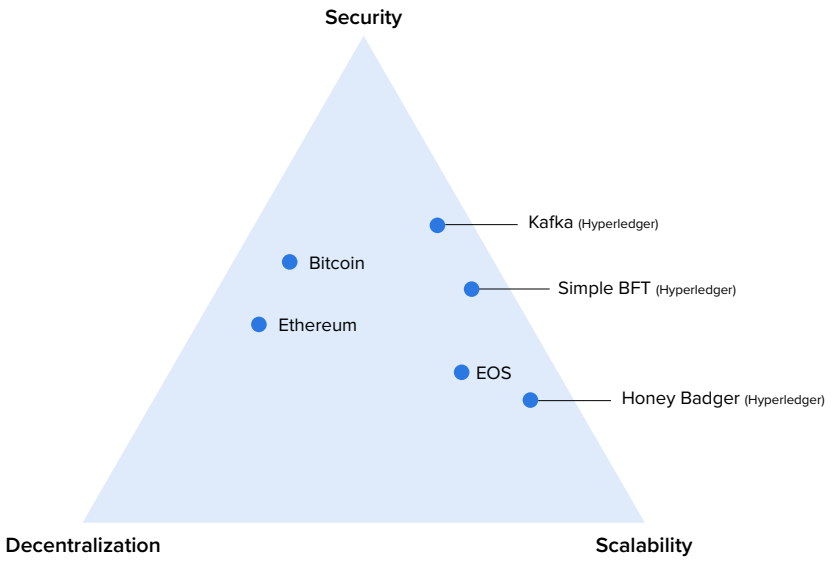
Exploring the range of leading blockchains reveals how each platform has its own general philosophy and serves as a unique incarnation of what it means to be a decentralized system. On one end of the spectrum lies a completely decentralized architecture, where no one party controls the state of the system, and on the other lays a centralized system, where one party effectively controls the system. Each philosophy comes with its own trade-offs; a choice must be made among decentralization, scalability, and security. Companies or developers considering integrating or building on the above discussed platforms now should understand platforms' potential trade offs better, so they can more aptly choose platforms most appropriate for their use cases in light of the considerations of decentralization, scalability, and security. Diagram 6 shows where each of the platforms fall under these considerations. Those preparing for an implementation should also be aware of the propensity for significant development costs—hiring the necessary talent—operational and maintenance costs, and how future on-chain prospects might impact their intended application. Potential implementers of the system should also be aware of clear advantages—enhanced trust and transparency—when implementing on blockchain.

Diagram 6

Trade-offs

Popular blockchain platforms are plotted with respect to the 3 trade-offs:

- Decentralization (number of actors in the system)
- Scalability (transaction throughput)
- Security (potential attack vectors)



Current web technologies have been in development for over two decades; they are well-tested, and continuously iterated upon to deal with the challenges of the day. Blockchain, in comparison, is relatively nascent and its technology will likely have to go through several iterations before it is as robust as Web 2.0 technology. Both Web 2.0 and Web 3.0 technologies will likely coexist in the foreseeable future. Given the technology constraints, Smith + Crown sees certain applications as more apt than others at this stage. Future memos will further delve into various aspects of each platform, discussing optimal use-cases, considerations for implementation, and likely development trajectories.