



ConsenSys [Follow](#)

A blockchain venture production studio building decentralized applications on Ethereum. Go to [www.consensys.net](http://www.consensys.net) and subscribe to our newsletter.

Apr 5 · 23 min read

## Blockchain vs. Distributed Ledger Technologies

*Part 1 of a comparative analysis between Ethereum, Hyperledger Fabric and R3 Corda*



by Brent Xu—Protocol Business Architect at ConsenSys

The Ethereum blockchain maintains both similarities and differences when compared to distributed ledger technologies like Hyperledger Fabric or R3 Corda. In making well founded assessments of blockchain and distributed ledger platforms and the value they bring to enterprise, it’s useful to categorize platforms based on their core functionality and characteristics. As blockchains were derived from tenets of cryptography and data configurations, certain functionality can be replicated in a coordinated database system, while other functionality is only feasible in a true blockchain environment.

In this article, we’ll assess the foundational business functionalities for the main enterprise facing platforms including Ethereum, Hyperledger Fabric and R3 Corda in terms of where the software acquires its influence and how the system is overall optimized, whether through traditional distributed systems or a contemporary blockchain basis.

Hyperledger Fabric	R3 Corda	Ethereum
Distributed Ledger Technology	Distributed Ledger Technology	Blockchain

Figure 1: Demarcation of Underlying Technology

**In particular, we’ll focus on three key areas of functionality:**

- Data coordination—How information and trust within a system is better distributed and allocated among stakeholders
- Cryptoeconomic internal incentive layers—How is a system architected so that different stakeholders and users are motivated based on economic incentives to ensure functionality of a system eg. Game Theory and mechanism design.
- Integration into the digital commoditization of assets—How the systems can integrate into a digital goods economy. In some nominal characterizations this is known as the token economy

# Main goals of blockchain: what do businesses want to achieve with this technology?

Blockchains such as Ethereum have similar goals to their distributed ledger counterparts. Identifying the goal of what businesses hope to achieve using blockchain technology can be a challenging approach, because like the internet in the 1990's, businesses did not yet know how to conceptualize use of the powerful tool. Similarly today, it is known that blockchain technology is capable of instantiating various functions, though how to architect those functions into a business solution requires further insights and assessments of the underlying capabilities.

The three main axes explored—processing and coordination of data, trusted and immutable records, and digitization of assets—are broad enough to encapsulate the primary usability of a blockchain while allowing for further extrapolation of those functions into business scenarios. By discussing these three aspects, it is possible to reveal the meaning behind why a business entity would want to use the technology.

## Efficient processing and coordination of information

If improved distributed system design or database coordination is the sole purpose of a protocol or platform, then perhaps a blockchain is not necessarily what is needed. Blockchain platforms have traditionally promoted the concepts of better data coordination and distributed consensus mechanisms in which data is facilitated and transferred through a technology platform. While useful, a significant portion of these desired functional traits can be obtained through better coordination of a central database or improved distributed systems design. In this investigation, it is necessary to determine the extent to which platforms and protocols are trying to optimize existing data coordination functionality versus implementing new blockchain functionality. Blockchains are designed for more than just advanced data coordination.

## Immutable/trusted record of the products and transactions

The original thesis around why we need blockchains revolved around the concept of digitizing trust. A theme promoted by Andrew Keys of ConsenSys is that “As the internet resulted in a digitization of information, blockchains result in the digitization of trust and agreements.” This meaningful thesis embodies the ethos of what blockchains hope to achieve while also paving the way for a further path. The additional variable would be the digitization of value. When value is attached to the trust that is implemented into a system, then certain alignment structures and incentive mechanisms will influence and incentivize proper behaviors within a system, resulting in a robust platform.

It is often the case that immutability is used synonymously with trust when designing a system *ie* because the system is immutable, it is trusted that bad things will not go unpunished. Though in our platform protocol assessment, it is important to also assess the mechanisms behind how a trusted system is implemented to ensure a business model that can be beneficial to users of a platform (further exploration through cryptoeconomics).

## Digitalization of assets

The digitization of goods and assets is considered a primary goal for most blockchain or distributed ledger platforms. If businesses are looking for a digitization of assets, a distributed ledger or coordination of a database is able to offer some capabilities though much consideration should be put into the accessibility of these digital goods. Because coordinated databases are essentially centrally run or distributed among a group or subgroups of counterparties via a legacy software paradigm, the levels of digitization may be limited based on the freedom that is afforded by the digitizing platform. While the concept of digitizing goods sounds like a simple process, the different incentivisation dynamics and economic reasonings around how goods such as real estate, human attention or even electricity are digitized requires significant consideration into what type of platform would be responsible for the digitization as certain vendor platforms do exhibit degrees of “vendor lock-in” and reliance on a centrally managed platform in various instances.

Records and registries like titling systems and supply chains are also feasible via a distributed ledger system though their level of interaction with an economic incentive layer is fairly limited if reliant on a closed proprietary system, and a proliferation of those assets into a digital ecosystem or marketplace would be severely stunted if based on closed rails. A free market system that fully utilizes the various facets that the open market is able to provide is necessary to facilitate true digital goods in a constantly developing digital ecosystem.

## Assessment of database coordination characteristics

### Database Coordination: Characteristics

While in depth analysis has been performed on the functionalities of these platforms in terms of characteristics like immutability, security, scalability, manageability, and performance, much more can be ascertained through understanding the foundations upon which the architectures are built.

Many tools have been invented and implemented for proper data coordination within distributed systems. One example would be heavy emphasis on tools like Hadoop and the various ensembles in this ecosystem including Spark, Hive, and Zookeeper. A reliance on these products show a heavy integration of distributed system tools and protocols. Further parallels can be shown in protocols such as Tendermint, a BPFT consensus engine being designed with similar functionalities as tools like Apache Zookeeper. Internally there has also been research along the lines of [event sourcing databases](#) which can replicate several functionalities desired from a coordinated data sharing system.

Through assessing tools like Apache Kafka and how the data streaming service is able to achieve significant levels of throughput in an enterprise setting, we can demarcate the functional differences between a blockchain and a distributed ledger based on varying levels of dependency on these database coordination and optimization tools in

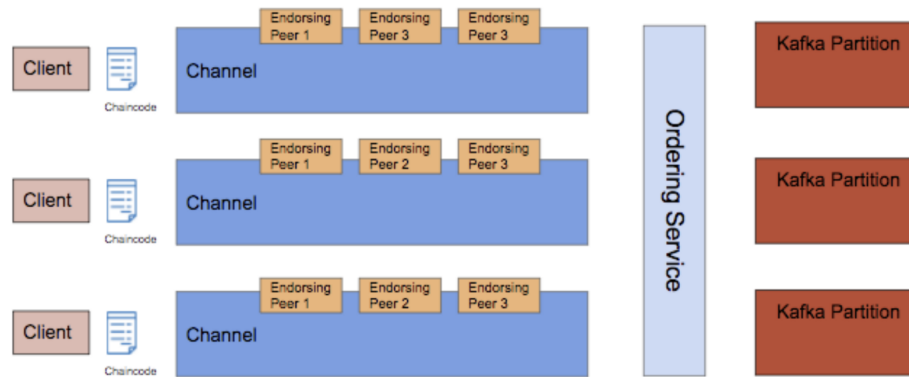
terms of the foundational concepts. Implementations of Ethereum including Plasma are utilizing tools like MapReduce to augment certain mapping functionalities on top of a UTXO and account based model while also reducing components into merkle proofs, though it is important to realize that the base layer of the protocol is still reliant upon Ethereum as the root blockchain. By decomposing these details, further insight can be obtained on how best to assess technological characteristics of these software platforms.

## Data Coordination: Platform Comparison

### IBM Fabric

Through a deep dive into [Fabric architecture](#), it can be determined that the platform has created an intricate development environment focused on allowing superior throughput based on detailed configurations of the software architecture for optimal performance in a distributed systems environment. The movement of chaincode between the client and a network of distributed endorsing peer nodes along with the transaction mechanisms and transfer of receipts that satisfy endorsement policies is effective in the closed system, while the gossip protocol that propagates transactions within private channels allows for the coordination of large datasets. While the infrastructure is robust and capable, additional consideration should be put into the thought process of how the architecture was designed to allow multilateral coordination structures where there may eventually be a factorial of channels involved in a network which can be difficult to manage.

### Figure 2: Hyperledger Fabric Architecture



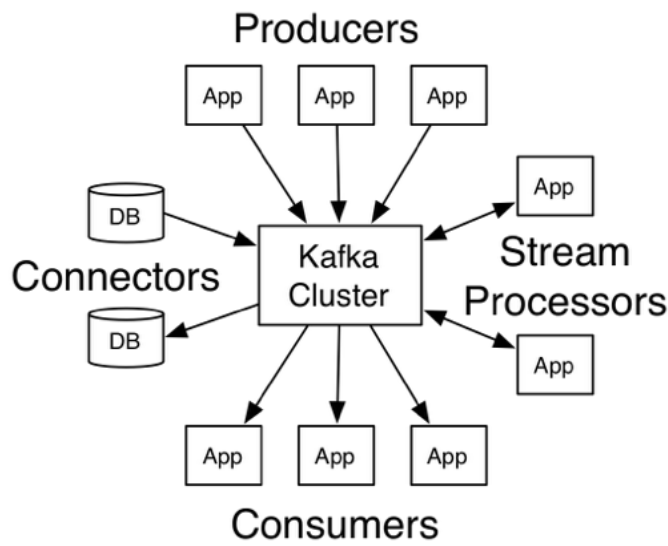
This figure demonstrates some of the architectural configuration of Fabric and how components are organized into a system designed for advanced information processing and maximum transaction throughput.

The main idea is that channels provide opportunities for moving transactions along within the platform. In looking at the architecture, the function of ordering service nodes (OSNs) serve to record transactions in the Apache Kafka ordering service. In the data streaming ecosystem, Kafka is a powerful tool with capabilities of appending various forms of transactions into separate Kafka clusters and eventually partitions.

In this setup, data is able to be distributed across clusters to formulate a distributed storage platform that can record the data structures that are sometimes referred to as “blocks” or blobs within the Fabric definition of “state” in the context of their key/value store configuration. A conceptualization to acknowledge within this software framework is that all of the participants and data structures within this ecosystem are native in that they function primarily alongside other users within this software ecosystem.

### Figure 3: Apache Kafka

Fabric does in fact employ a ledger-type substructure that deploys certain hash linked data stores, though it should be recognized that the configuration of the hashes does not follow the original architectural design affiliated with a blockchain system derived from Bitcoin or Ethereum. While data blobs are batched and undergo *deliver* events to eventually create a hash link of the transactions, it must be understood



Source: Apache Kafka

that this process does not necessarily transition the data into a modification of the system's state. Rather, the blocks are configured in a way that the information is stored in a database type structure with different instances of hashes.

In the Fabric ecosystem, *deliver* events are called *blocks* while chaincode goes through *deploy* events to eventually secure the data within the *chainpartitions* of the ordering service structure. The configuration of the data-structures and modules of this system are able to allow transaction throughput that would be expected of

the distributed database architecture, though it should be acknowledged, that asset-code coordination is still a challenge that has not been completely solved within the Fabric ecosystem as assets and value do not necessarily have a digital representation that can be coordinated within the ledger.

## R3 Corda

R3 Corda is built on an environment that does not claim to be a blockchain, but rather a decentralized database that utilizes various forms of structural reconfiguration toward building a system that would primarily be used by banks and other institutions for their processes. The platform borrows heavily from the UTXO model used in bitcoin transactions where state is defined by a series of inputs and outputs and the varying reconfigurations of the inputs can dictate the state of the output.

The R3 Corda architectural framework relies upon a nodal structure that is reliant on submodules called notaries that help maintain the validity of a network similar to validator structures in other platforms that abstract the function of consensus. Nodes are accompanied by relational databases that are appended within the data-structures allowing for



querying using SQL. Transaction communication is restricted within subprotocols called flows.

These flows are comparable to the channel architecture that is seen in IBM Fabric where only individual parties privy to the transactions are able to access the information. Classes undergo transformations that result in state machines called fibers or co-routines. The architecture relies on flows communicating with sub-flows and interacting with libraries of flows that have predefined functions within the confines of the platform. Additionally there is a self contained identity layer within Corda that allows varying degrees of access control within the overall network.

While R3 Corda has openly stated that it does not intend to be a blockchain, it should be taken into consideration that the reconfiguration of the concept of a distributed database to a decentralized database does rely fairly significantly upon traditional database systems. While the system is architected around novel data-structures and different compositions of how a distributed system is organized, the platform does have data allocation in mind and does find various ways to optimize the functions of a data distribution system. One thing to keep in mind is that because the system is limited to certain facets of data coordination in the confines of a specific architecture, integration into actual blockchain systems has been sacrificed as modularity and interoperability were not implemented for the original design.

**Figure 4: R3 Corda Workflow**

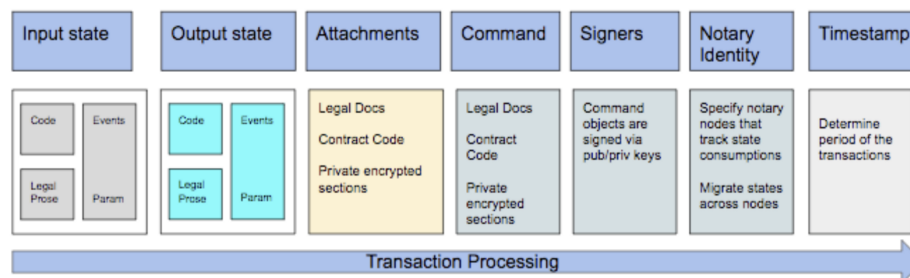


Figure details: the workflow of transactions within Corda and how Input states and Output states are moved through the system and how documentation is appended into the workflow process.

## Ethereum

The Ethereum ecosystem is built from a combination of private blockchain and public blockchain ecosystems. The public chain does not have anywhere near the throughput and data processing capabilities as described in the data coordination context so should not be assessed based on those capabilities. When assessing this aspect of Ethereum, it makes the most sense to synthesize the different nuances of the network topology of private instantiations of Ethereum.

The Ethereum Yellow Paper adamantly decrees a set of specifications on what constitutes Ethereum as well as the technical particularization of the code base. Due to this strict adherence to the blueprints of this protocol, forks of Ethereum as well as consortium implementations do resemble the original substrate upon which the technology is built. In fact the same specifications are continuous whether in a proof of work, proof of authority, or proof of stake implementation because the protocols are considered progeny of the same Ethereum Virtual Machine (EVM) specifications.

Modified architectures still specify alignment with the original EVM. Key changes in platforms like Quorum include an alteration of the consensus mechanism, modification of global state roots to accommodate private and public states, alterations of State Patricia Merkle tries, and additional modules to handle private transactions. The architecture allows this software to maintain the lineage and data structures from the original Ethereum configuration while also offering increased transaction throughput via the alterations. In addition to the improved data transaction optimization that Quorum provides, the capability of coordinating and integrating with public Ethereum environments via tools like Plasma, Truebit, and Cosmos provide additional extensibility to the protocol.

Through technical evaluations of tools like Plasma and formats of obtaining consensus in Casper, it is apparent that database management tools like MapReduce and Abstract Rewrite Systems will be implemented in Ethereum. In Plasma, MapReduce is an integral part of assembling the coordination of an account based system and a bitmap-UTXO commitment structure of a multichain setup.

The orchestrated transaction processing paradigm using the interplay between rootchains, plasma chains, and child chains through a combination of *fraud-proof* mechanism designs and *fidelity bond* incentive structures help satisfy dynamics between the *block-withholding* and *mass withdrawal* surfaces. It also allows for further cryptoeconomic structures to be filled using mechanisms from systems like Casper or Truebit for mirroring concepts used in erasure coding in terms of the *data availability* problem that is prevalent in the space. For a multichain architecture, Ethereum would be able to combine the database coordination and throughput capabilities of a distributed database system with the public chain compatible capabilities of an actual blockchain.

## Database Coordination: Conclusion

A viable conclusion regarding the database coordination spectrum of capabilities would be that IBM has superior database management toolsets due to a reliance on traditional database and distributed systems software architecture, based on the overall monolithic design and substantially resource intensive process that went into building Fabric. R3 Corda is still further defining its capabilities, while offering several coordination services to banks and financial institutions in a private reconfiguration of nuances from the bitcoin protocol. Ethereum, while designed for public chain compatibility, does not have the raw database processing capabilities of IBM Fabric, though it does have certain coordination schematics within the context of scalability for enterprise use cases that Fabric does not have.

Private instantiations of Ethereum and complementary clients are able to act as the architectural building blocks upon which larger systems can be built, based on modular design that adheres to a comparatively unix-based philosophy. The Ethereum related code bases are designed to rival the transaction throughput capabilities of the database platforms like Fabric while allowing functionality that is nonexistent in both Corda and Fabric, though complementary relationships can also be explored across platforms. The main differentiating factors may be further elucidated from assessments of the subsequent factors.



## Cryptoeconomic Configurations of Blockchain Platforms

A cryptoeconomic subsystem inside of a software platform requires various configurations of mechanism design and game theory that would exist to incentivize actors to behave in the most optimal manner that is both beneficial to their own self interest as well as the interest of the ecosystem. A core tenet that distinguishes blockchain ecosystems from distributed ledger-designed database systems is the capability to use mechanism design as an economic incentive layer that ensures proper allocation of trust and cooperation to make a system behave in a way that is conducive towards instantiating decentralized consensus among users as well as security. The main goal of these systems which rely on a “reverse game theory” design is to create a dominant strategy within a subsystem that results in an incentivized equilibrium structure that further strengthens the overall integrity of the whole system.

## Examples of Cryptoeconomic Mechanism Design

### Plasma & Truebit

Plasma was designed to bring scalability and multichain capabilities to the Ethereum network. By providing the catalyst upon which multiple blockchains of the Ethereum lineage can communicate with each other, Plasma acts as a viable bridge between private blockchain and public

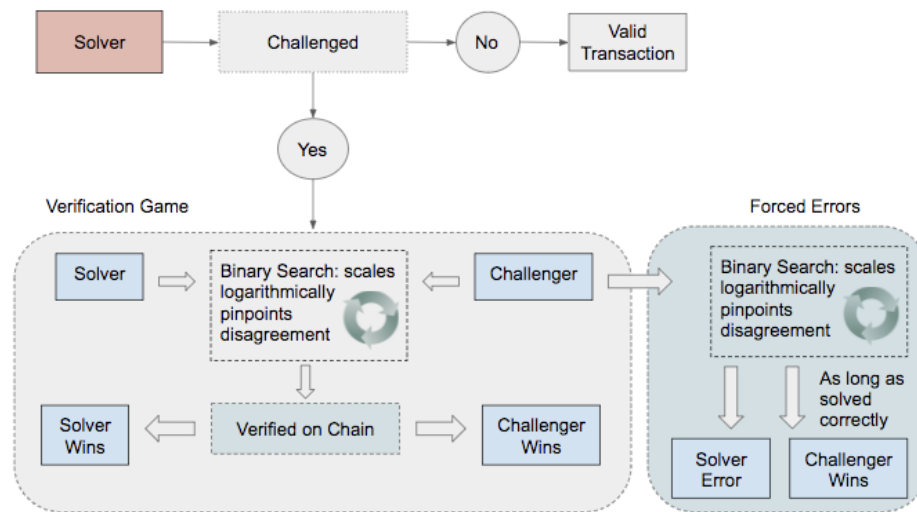
blockchain networks. From further analysis, it is apparent that Plasma offers both scalability and availability to the Ethereum network.

Though to understand the effectiveness of plasma, it is important to understand the mechanics upon which Plasma was designed. A significant amount of the interoperability is achieved through what are known as *fraud proofs*. By configuring blockchains so that derived sub-blockchains (or child blockchains) can still reliably validate transactions, based on computations from the MapReduce functions, scalability can be achieved with minimized trust.

A mechanism was designed around Plasma to allow for what are called *Mass Exists* when faulty chains are discovered. These situations pertaining to faulty actions are related to inconsistencies in *data availability* and *block withholding attacks*. By allowing a mechanism where nefarious activity can be punished through alternating configurations of the interrelated chains, the ecosystem hopes to instantiate a cohesive equilibrium for how entities interact.

Plasma shares quite a bit of influence from a heavily cryptoeconomic incentive structure focused platform called Truebit which was designed to increase the offchain computational capabilities of the Ethereum network. By architecting the Truebit system around a verification game in which *Solvers* of the overall consensus mechanism can be challenged by *Verifiers* which obtain a reward if they identify a nefarious counterparty, an internal cryptoeconomic “checks and balances” of the system is created to incentive a dominant strategy of behaving fairly. As Plasma through the influence of TrueBit is focused on creating a multichain interoperability network, the internal enforcement of the system is paramount toward achieving information and consensus fidelity.

As seen in the figure below, the cryptoeconomic game involved in Truebit and derived into Plasma includes counterbalancing interactions between solvers and challengers to verify the correctness of computation towards eventually being verified on chain. Challengers are incentivized to constantly challenge due to forced errors that guarantee a payout if solved correctly.

**Figure 5: Cryptoeconomic Design**

## Ethereum Casper Proof of Stake

An example of cryptoeconomic incentive layers can also be seen in Ethereum’s transition to a proof of stake consensus mechanism via implementations of Casper. While proof of work has its own internalized game theoretical incentive structure to dissuade participants from commandeering the network, the transition to proof of stake has even further internal structures for disincentivizing participants from *equivocating* or trying to create alternative instances of the blockchain when encountering forks. The staking protocol creates a Byzantine Fault Tolerant environment where Ether would be bonded into the consensus mechanism. What this means is that individuals would be bound by a *fidelity bond* to behave honorably within the system.

If an attacker were planning to *equivocate* or try to assume control within the consensus mechanism, various protocols pertaining to “slasher algorithms” would destroy the Ether holdings or bonds of the attacker, hence punishing them for their nefarious actions. In the mechanism design behind the punishments, the amount of Ether destroyed is consistently programmed to be proportioned to the amount that an attacker wished to gain in which the equilibrium achieved is one where the attacker would never want to compromise the system in the first place.

## Cosmos and Tendermint

Cosmos is also building an ecosystem that relies on the Tendermint consensus mechanism that relies heavily upon Byzantine Fault Tolerance algorithms. The platform depends on validators that have similar roles as miners in the bitcoin network. The validators have staking tokens called Atoms which are used to secure the network via a proof of stake mechanism that relies upon the trust generated by the bonded validators. The interplay between the players in the ecosystem is also indicative of a game theoretical structure where validators can lose their tokens or the tokens delegated to them if discovered to be violating the protocol. Due to this bonded deposit design of stakeholders within this system, the consensus mechanism allows for an incentivization mechanism that secures the network. This security design allows for the proper functioning of the Application Blockchain Interface (ABCI), the Inter-Blockchain Communication protocol (IBC) as well as the varying interactions between the Cosmos hub and zones.

## R3 Corda & IBM Fabric

An important note to recognize is that R3 Corda and Hyperledger Fabric do not have these cryptoeconomic incentive layers instantiated within their software architectures. Due to the fact that the software architectures are foundationally designed based on distributed database focused paradigms, they were not originally designed for the incorporation of native cryptocurrency layers within the overall framework. And because of this inherent difference in design of the software, they are not yet calibrated to be able to take part in multi-chain ecosystems where there is interoperability and coordination with a multitude of blockchains. Because the systems are structured with maximum throughput in mind, architectural layouts for an interoperable network topology with blockchains including the public blockchain mainnets were overlooked based on the initial construction of these systems.

## Why is cryptoeconomic mechanism design necessary?

One may ask why a cryptoeconomic infrastructure layer is necessary in software design. What this paradigm creates is a new layer of trust and immutability that can exist within a computing environment without relying on a centralized entity. We have been building software in a particular client server and database architecture for decades. Companies like IBM, Intel, and Oracle have already perfected this model along with the systems and subsystems that were created subsequent to the initial creation, and these models are still being used within distributed system architectures as well as newly labeled distributed ledger systems. Though these systems are still centralized in various aspects, whether through a central entity or a cartel-like consortium structure in which incentives are aligned based on inherent reliance on a centralized entity as opposed to a genuine incentive structure to ensure the proper functioning of the system.

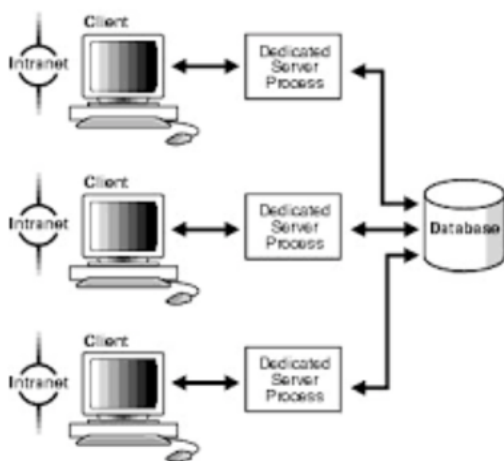


Figure 6: Client Server Model

A decentralized system allows viable alternative pathways toward achieving certain goals within a software environment. The main tradeoffs that are highlighted within this interchange would be trust vs execution. Because a large centralized system is better trusted, it is considered to be capable of better execution. Though what blockchain systems hope to instill are the characteristics of a system in which trust and value can be reallocated

without reliance on a large centralized entity.

One idea that is championed within certain facets of system design is that in order to optimize a system, it is necessary to also suboptimize the subsystems. What this means is that the coordination of a system must be orchestrated and architected so that internal subsystems also have a stake or incentivization mechanism within the overall larger ecosystem, to further achieve cooperative goals. By creating a cryptoeconomic game theoretic approach towards this optimization of the overall environment, a confluence of both computer science and economic models can be



created allowing for the creation of new software architectures that can be envisioned within the digital economy.

Based on this visioning of a digital economy it should be recognized that the use of a combination of private blockchains and public blockchains that can interoperate is what will create a viable digital ecosystem where various layers of commerce and business relationships can emerge, and develop beyond what is possible in legacy technological configurations.

## Integration into a Blockchain Token Economy

For the purpose of this investigation, it is necessary to define the concept of tokenization. The concept borrows from the notion that businesses or entities are able to create fungible or non fungible representations of various forms of assets, commodities and services based on certain digital standards that currently exist in our ecosystem.

While the token economy is still developing, its important to distinguish that the first wave of products will initially have various failures and flaws that require time and iteration to perfect. Even though the tokenization of assets, financial products, energy and digital attention are all viable business models, the exact dynamic that they are implemented upon require additional layers of functionality and access that will only be improved upon with time. A successful token economy will be the resultant artifact created from significant developments and discoveries that are being made in game theoretical mechanism design and blockchain innovations.

As described in [Josh Stark's](#) article on cryptoeconomics, the tokens that exhibit the strongest signs of usability are evaluated on whether they form a necessary component within the economics and game theory design of the overall business. If a business can digitize or tokenize various facets of its ecosystem, the lines of products that can be created expand exponentially beyond our traditional means of exchanging physical goods, financial assets, commodities, or technological services.

By creating the digital medium upon which tokenized assets can come to fruition, significant developments can evolve from the new ecosystem.

In viewing the ecosystem of blockchain tools, it is apparent that Ethereum is in fact the substrate upon which the token economy can be built upon. And if the token economy model is able to incorporate the functions of private blockchains, scalability solutions and privacy tools like ZK-Snarks, the overall tokenization of digital assets will overshadow the current capabilities upon which our economic models are limited to due to inherent restrictions in organizational feasibilities.

## Achieving Business Goals of Blockchain

In order to achieve the mentioned business goals of the blockchain, we must assess the various avenues that need to be serviced. In an overview of the chart detailing capabilities of the mentioned models, Ethereum is able to service the Distributed Database Coordination scenario as well as the additional functions while R3 Corda and IBM Fabric have not yet chosen to touch upon those layers of functionality.

In the context of business use-cases, we overlay the different functionalities discussed on top of real world business scenarios to better understand the capabilities of the platforms.

**Figure 7: Summary of Capabilities**

	<b>IBM Fabric</b>	<b>R3 Corda</b>	<b>Ethereum</b>
Distributed Database Coordination	Yes	Yes	Yes
Cryptoeconomic Incentive layer	No	No	Yes
Token Economy	No	No	Yes

## Efficient allocation of information

Functionally speaking, the products are similarly matched from the viewpoint of database coordination and utilization of distributed systems. R3 Corda, IBM Fabric, and enterprise versions of Ethereum do in fact have distributed information allocation features that can facilitate the allocation of information via different layers of access control and consortium configurations of governance. While each platform is different in terms of its software architectural configuration, each one is able to execute the necessary performance on efficient information allocation and coordination.

## Trusted immutable information

Immutability has been used somewhat as a synonymous concept to trust in the context of a lot of these technologies. In assessing the immutability characteristics it must be understood that within an ecosystem that utilizes a Apache based data streaming tools like Kafka, there are inherent capabilities that allow the read/write access to data. Therefore the immutability aspects of IBM Fabric are somewhat limited due to some of the choices made in the system design.

For R3 Corda's UTXO model based system, the aspect of immutability is preserved differently within the overall confines of the system. Due to the overall distributed ledger design of their system, they have established certain facets of trust that can be demonstrated throughout the platform.

The layers of trust and immutability established within an Ethereum context are all conceptualized within a subprotocol of public blockchain derived state roots from Patricia Merkle Tries. Due to this preservation of core software paradigms within the ecosystem and a viable connection to the public chain, the Ethereum blockchain and related derivations of Ethereum are able to fully substantiate immutability. Trust gained from this immutability can eventually be attached to a new value system as assets begin to undergo digitization.

## Digitization of assets

It should be recognized that IBM Fabric is in fact able to create digital assets in the nominal sense as the digitization of an asset is derived from a registry of the product into a digital format. Though the digitization of an asset on Fabric would result in an asset that can only function on systems that use Fabric. This would be equivalent of if an email client was created to only be able to send emails back and forth with people who use the exact same email client, unlike what exists in our current world where a multitude of email clients can all interoperate together.

R3 Corda has similar inconsistencies in that users of R3's platform would be restricted from interacting with other platforms beyond R3 within their overall landscape creating a bit of vendor lock in. Because R3 Corda is focused on mainly bank clientele, it may make sense to have a separate banking software, though it should be noted that users of the platform will be restricted to banking relationships with only the institutions using R3 Corda, and will not be able to seamlessly interoperate with the ecosystem of counterparties that do not use the vendor platform.

Because Ethereum is meant to act as an underlying protocol similar to HTTP or TCP/IP in web services, there is no conception of "vendor lock in" to just one builder of Ethereum applications. The trust that can be established via the different facets of the Ethereum blockchain allow for the digitization of global assets that can occur within a new economic system unlike what is currently available. If referring back to the email example, the Ethereum protocol can be perceived as analogous to IMAP or POP3 as universal protocols for accessing email.

Ethereum and Ethereum derived protocols are able to act as a blockchain infrastructure upon which companies can build digital assets. Similar to how every company was able to create a website in the late 90's using HTML for the scaffolding of the web page, every company will be able to create digital economies for their services and products using Ethereum smart contracts that can create tokens which will be accessible by a broader network.

## The Road Ahead

In order to have a robust enough platform that can interact with public

markets, the system must be able to satisfy business requirements that allow for efficient processing of data, additional layers of trust allocation, and an ability to represent assets in a developing digital economy. It is apparent that all three platforms aim to achieve similar goals though through different avenues in terms of technological advancements and utilization of technical configurations.

In the road ahead, we must consider where we see economic business models evolving in this developing ecosystem, and it is apparent that Ethereum based platforms have an advantage on the true integration into a digital economy, though have apparent weaknesses in some of the data transaction throughput functions that IBM Fabric and R3 Corda can excel in. As different blockchain and distributed ledger platforms are iterated upon and transcend beyond the capabilities that exist in our current technological *zeitgeist*, decisions around which platform to use to build upon will fall heavily upon the direction of the use cases in our ecosystem, and I see different types of use cases layered upon each other.

This document does not aim to say one platform is overall better than another platform, rather aims to stipulate that the platforms are inherently different from each other. Ethereum has certain functionality that distributed ledgers like Fabric and Corda do not have while Fabric and Corda have performance capabilities that Ethereum currently is not able to achieve to the same extent.

In order to truly achieve the level of interaction and scalability that is desired by our existing systems, a protocol must be built and designed with all interactions in mind, similar to how the internet was first designed. Ethereum as a protocol, is able to act as the foundational technology stack that services a broad enough ecosystem to encompass the necessary factors in an economic environment, though keep in mind, the platform is currently incomplete and could also benefit from some of the capabilities inherent in the DLT counterparts.

While the road ahead will include technologies that have not yet been perfected, protocols should be examined on how closely they will eventually replicate the degrees of functionality that we hope to see in the next generation of the internet and sometimes the the most apparent solution is not to focus on only one technology.

## Sign up for the Consensys Newsletter

Join the decentralized future

Sign up

## Useful Links:

[Ethereum Whitepaper](#)

[Grid+ on Plasma](#)

[Truebit Whitepaper](#)

[Plasma Whitepaper](#)

[Public & Permissioned Blockchains](#)

[Data availability & Erasure coding](#)

. . .

*Disclaimer: The views expressed by the author above do not necessarily represent the views of Consensys AG. ConsenSys is a decentralized community with [ConsenSys Media](#) being a platform for members to freely*

*express their diverse ideas and perspectives. To learn more about ConsenSys and Ethereum, [please visit our website.](#)*

