1. **Permissioned Versus Public Permissionless Blockchain**

A blockchain is a data structure that makes it possible to create a digital ledger of data and share it among a network of independent parties. There are many different types of blockchains.

**Public blockchains**: Public blockchains, such as Bitcoin, are large distributed networks that are run through a native token. They’re open for anyone to participate at any level and have open-source code that their community maintains.

**Permissioned blockchains**: Permissioned blockchains, such as Ripple, control roles that individuals can play within the network. They’re still large and distributed systems that use a native token. Their core code may or may not be open source.

**Private blockchains**: Private blockchains such as hijro tend to be smaller and do not utilize a token. Their membership is closely controlled. These types of blockchains are favored by consortiums that have trusted members and trade confidential information.

All three types of blockchains use cryptography to allow each participant on any given network to manage the ledger in a secure way without the need for a central authority to enforce the rules. The removal of central authority from database structure is one of the most important and powerful aspects of blockchains.

**Public Blockchain networks** are large and decentralized, anyone can participate within them at any level — this includes things like running a full node, mining cryptocurrency, trading tokens, or publishing entries. They tend to be more secure and immutable than private or permissioned networks. They’re often slower and more expensive to use. They’re secured with a cryptocurrency and have limited storage capacity.

A “public block chain” e.g. Bitcoin is open to anyone at any time. No one needs permission. In Bitcoin, there’s no one who would even give permission, because the network is a collection of thousands of users who don’t know each other and weren’t even involved in Bitcoin’s creation. Many cryptocurrencies that have followed Bitcoin adopted this feature – people can remain anonymous or pseudonymous to be a miner, a staker, a node, or an entrepreneur.

This openness comes with risks that are part of operating in an environment where you don’t know and can’t trust other users.

* It can be used for anything, including scams, money laundering, and buying contraband.
* You need financial incentives to keep validators maintaining the blockchain while also keeping it economically infeasible to gain control of the network,
* Introducing changes is hard, because 51% of the network needs to agree—and it could fork,

This ultimately makes public/permissionless blockchains less viable option for enterprise blockchains.

Permissionless blockchains are much more disruptive and difficult to fit into existing legal and business frameworks.

**Permissioned networks** are viewable to the public, but participation is controlled. Many of them utilize a cryptocurrency, but they can have a lower cost for applications that are built on top of them. This feature makes it easier to scale project and increase transaction volume. Permissioned networks can be very fast with low latency and have higher storage capacity over public networks.

Permissioned blockchains address many of the risks created by public blockchains. Many are based on the fact that people in the network can be trusted to some degree.

* You can blacklist users, such as terrorists and scammers
* Blockchain protocols can be optimized for specific uses that the consortia members agree on (such as cross-border payments)
* The cost of verifying transactions is generally lower because you have trusted validators instead of a distributed community of miners/stakers.
* Updating the protocol is much easier: users who don’t agree lose permission
* Nodes are well known and not at risk of going down
* You can more easily control things like privacy

Ultimately, many see permissioned blockchains as the future that banks, regulators, and other established players will embrace. They provides more control and can be adapted to existing regulations and business processes.

**Private networks** are also a type of permissioned blockchain network that is shared between trusted parties and may not be viewable to the public. They’re very fast and may have no latency. They also have a low cost to run and can be built fast.

Most private networks do not utilize a cryptocurrency and do not have the same immutability and security of decentralized networks. Storage capacity may be unlimited.

Most private blockchains are actually private consortia blockchains: they aren’t controlled by one entity but rather consortia of entities who give others permission to use it. Everyone who operates on it has been ‘vetted’ to some degree.

Public blockchains could be either permissionless (e.g., existing cryptocurrencies) or permissioned

(e.g., the federated sidechains concept whereas a private blockchain could only be permissioned.

Public permissioned blockchains eliminate possibility of Sybil attacks, thus in principle providing a greater degree of scalability and flexibility compared to permissionless blockchain designs.

The properties of public blockchains – easy entry and exit, openness, transparency, built-in precautions

for operation in untrusted environment – could be beneficial for their adoption for decentralized

applications. Thus, public blockchains could create ubiquitous infrastructure for the Internet of Value

(IoV), with digital assets being one of its core parts

.

In contrast, private blockchains could retain reliance on trusted third parties for basic operations, thus limiting their innovative potential.

**Determining which blockchain is right for you**

The following criteria should be considered while evaluating a blockchain for a given project

* Scale and volume
* Speed and latency
* Security and immutability
* Storage and structural needs

The following table lists the common use cases that are suited for each type of blockchain

|  |  |
| --- | --- |
| **Primary Purpose** | **Type of Blockchain** |
| Move value between untrusted parties | Public |
| Move value between trusted parties | Private |
| Trade value between unlike things | Permissioned |
| Trade value of the same thing | Public |
| Create decentralized organization | Public or permissioned |
| Create decentralized contract | Public or permissioned |
| Trade securitized assets | Public or permissioned |
| Build identity for people or things | Public |
| Publish for public record keeping | Public |
| Publish for private recordkeeping | Public or permissioned |
| Perform auditing of records or systems | Public or permissioned |
| Publish land title data | Public |
| Trade digital money or assets | Public or permissioned |
| Create systems for Internet of things (IoT) security | Public |
| Build system security | Public |

There may be exceptions depending on project, and it is possible to use a different type of blockchain to reach project goal.

1. **Digital Assets & DLT Requirements**

Digital Asset is a floating claim of a certain service or goods ensured by the asset issuer, which is not linked to a particular account, and is governed using computer technologies and the Internet, including asset issuance, claim of ownership, and transfer.

Digital asset management has numerous use cases, including:

• Shares and financial securities

• Smart property

• Tie to a fiat currency

• Local community money

• Coupons

• Digital collectibles

• Access and subscription to certain resources.

Digital Asset could leverage security properties of blockchain as

• Immutability

• Impossibility of counterfeit

• Disintermediation and ease of transfer

• Transaction Verification

• Possible to link a digital or physical asset to the blockchain in an irrevocable manner

• Transparency and ease of auditing

• No overhead related to transaction processing

• Network effect brought by the unified infrastructure for multiple types of tokens

Blockchain based distributed ledger technology (DLT) provide an alternative to centralized digital asset management system by providing

**Distributed Transaction processing** – Transaction processed in a decentralized manner by geographically distributed nodes of the network. Moreover, defining the rules for transaction processing (i.e., defining valid transactions) could be split from the processing.

**Asset Issuance -** In the most general case, this could be performed by any user of the blockchain network.

**Securing user’s funds:** This could be performed by third parties using custodial or non-custodial wallets.

**Identities of services** (and optionally **customers**): This could be established by building public key infrastructure based on a blockchain

**Application development:** This does not require cooperation with blockchain maintainers

**Constituents of Blockchain DLT ledgers to support asset management**

1. **Blockchain Specification –**

Specification regulates the way data is transmitted among nodes in the supporting blockchain network

and how the state of the blockchain is derived locally based on the received data, i.e., semantics of

transactions. The specification includes:

• **Transaction logic**: Valid transactions with regard to the present system state; the rules how transactions transform the system state, etc.

• **Immutability logic**: How transactions are grouped into blocks, and how block headers are secured

• **Consensus logic**: How nodes agree upon the state of the system; how blockchain forks are resolved, etc.

• **Network logic**: How transactions, blocks and other data are transmitted among network nodes,

etc.

Fully automated transaction processing could be achieved by embedding rules for transaction processing (e.g., transaction finality rules and auditability requirements) into the blockchain specification.

A public blockchain facilitate compliance for next-generation financial services without directly implementing service-specific compliance, such as obligatory customer identification, on the blockchain-wide level. Public specification aligns with the overall spirit of blockchains as consensus-driven systems.

1. **Blockchain Notaries**

An asset issuer using blockchain infrastructure is not generally required to process transactions or to write data to the blockchain – this task could be delegated to blockchain notaries.

Notaries could be either known entities (in permissioned blockchains), or any users satisfying technical capabilities imposed by a blockchain consensus algorithm (in permissionless blockchains).

Permissioned blockchains could be more beneficial for financial institutions in the short term because of the flexibility of the blockchain specification and increased compliance. On the other hand, permissionless blockchains could prove more attractive for consumer-to-consumer markets and IoT applications because of inherent trustlessness and permissionless entry and exit.

Blockchain notaries get revenue incentives by keeping blockchain safe e.g. by running services in top of it.

For Chain protocol, the Consensus Notary pools e.g. RAFT , BFT-SMaRT etc. provide a uniqueness service by operating consensus over uniqueness by nodes operated by a set of distrusting entities.

A notary consensus pool could differ by the protocol configuration, and by their size (number of notary nodes in the pool), and their location (for a given pool, notary node location could be in any geographic location).

The size of notary consensus pool determines the performance (TPS) of a blockchain as it directly impact the consensus delay in verifying the transactions.

1. **Blockchain Network**

A public blockchain network provides three security modes for constituent nodes:

• **Full verification nodes** that verify and store every transaction circulating in the network. This security mode could be used by blockchain notaries, regulators, auditors, analytical services and dedicated “blockchain as a service” providers.

• **Simplified payment verification (SPV) nodes**, which would be used by a vast majority of end users, as this security mode requires little computational resources and memory space.

• **Partial verification nodes** made possible with the help of segregated witness and fraud proofs

These nodes could verify a small percentage of transactions (e.g., 1%), while contributing to the overall security of the blockchain network. Partial verification nodes could be operated by service providers on the blockchain.

In the case of a blockchain with restricted read access, the architecture of the blockchain network would be determined by transaction processors. For example, transaction processors could operate full nodes, and all other users could be provided to concerning transactions either through SPV network nodes or through equivalent web application interfaces. Thus, blockchains with restricted access could be less scalable or reliable because of uneven distribution of transaction processing.

There is an important distinction between SPV nodes and web API access to blockchain data. While

SPV nodes do not increase the security of the blockchain network, their use together with the publicly

available chain of block headers could provide uniqueness of blockchains and immutability of data as any change would not be accepted by SPV nodes. Alternatively, same thing could be achieved by proof of work consensus. In the case that access to the blockchain is provided via web APIs without disclosing the blockchain structure, reliably proving uniqueness and immutability becomes more difficult. Even if the regulator or an auditor would have complete access to the blockchain (e.g., by operating a full verification node), data provided to the regulator could differ from data served via API as a result of an eclipse attack performed by colluding blockchain notaries.

1. **User Authentication and Authorization**

User authorization in blockchains is performed using public key cryptography. In the simplest case, blockchain-based assets are bearer assets; i.e., the ownership of an asset is determined by the knowledge of a private key. Two-factor authentication or other security measures comparable to those of centralized e-money systems could be implemented by using dedicated wallet services.

Security properties of public key cryptography could be boosted by the use of specialized *hardware wallets* for signing transactions.

In order to maintain user privacy, blockchain users could utilize hierarchical deterministic wallets and the pay-to-contract protocol.

In the case of more complex transaction models, e.g. for smart contracts, zero-knowledge proofs and secure multi-party computations could be used in order to execute contracts while not disclosing data to any of computers.

1. **Asset Issuance**

As asset issuance is a special type of transactions, the identity of the issuer could be determined according to the general user identification rules (using the blockchain-based PKI or other techniques).

A regulatory body could explicitly acknowledge asset issuance by co-signing the corresponding transaction together with the issuer, or by granting the issuer a special kind of the digital certificate.

An asset could be marked as ***locked***; meaning the assets of the same type cannot be issued in

the future by anyone, including the initial issuer. This type of assets is useful, e.g., for creating

non-dilutable shares

• An asset could be marked as ***divisible***to several decimal places (cf. with Bitcoin, which is divisible

to 8 decimal places)

• An asset could be made ***non-transferable***in order to limit secondary market (e.g., due to regulation requirements)

• Additional metadata could be associated with the asset, either directly or in the form of a hash commitment. In the second case, off-chain data could be retrieved with the help of distributed

hash tables, e.g., implemented using IFSC or Bit Torrent protocol. Metadata could be useful, e.g., in implementing event tickets

**Deployment Models**

1. **Separate Blockchains for Assets**

Each digital asset or a set of assets maintained by the same issuer could potentially have its own blockchain, either permissionless or permissioned.

(Merged mining allows securing multiple blockchains with the same computational resources. However merged mining in a permissionless environment could be unsafe, as an attacker with enough hash rate could deliberately mine empty blocks or otherwise disrupt transaction processing.) . A permissioned blockchain could be more resilient to attacks, but it would still have a single point of failure in the form of a single transaction processor.

From the auditing and regulating points of view, properties of an issuer-managed blockchain could

be similar to existing asset management systems. The cost of operating an issuer-specific blockchain (either on-site or using a PaaS) could be comparable to traditional asset management systems because of the need to develop end user applications (such as wallet services with secure authentication), accounting tools, etc. Additionally, using separate blockchains could complicate the development of third-party applications and diminish the network effect by requiring additional tools to interact with other digital assets.

1. **Colored Coin Protocols**

Colored coin protocols share the user authentication model with the underlying blockchain. However, because the validity of colored coin transactions is not checked by the blockchain network, colored coin protocols lack efficient payment verification methods Colored coin protocols using the Bitcoin blockchain include ChromaWay, Open Assets and Colored Coins Protocol

1. **Metacoins**

A metacoin system is a colored coin protocol coupled with a middleware layer in the form of dedicated servers, which verify colored coin transactions. A metacoin system could provide automated order matching for trading asset pairs, dividend payments, and so on. Metacoin systems may utilize a dedicated cryptocurrency as a means of payment for provided services

Metacoin systems on top of the Bitcoin blockchain include OmniLayer, Counterparty and CoinSpark.

1. **Multi-Asset Blockchains**

Multiple assets can be natively supported by a blockchain. Compared to other deployment models,

multi-asset blockchains have more space-efficient proofs of ownership, as simplified payment verification

could be utilized for all natively supported blockchain assets. On the other hand, known mechanisms of sharing blockchain security (merged mining and blockchain anchoring) pose security risks in permissionless context. The federated governance model puts the greater responsibility on the blockchain maintainers. As the maintainers can effectively determine the state of the blockchain, they could be legally obliged to be able to reverse transactions, freeze funds, etc. by the regulatory bodies.

A multi-asset blockchain could be integrated into existing blockchain infrastructure by using sidechain technology ;

Smart property represents the ownership of real-world objects with the help of blockchain data. For example, a blockchain-enabled car would operate only if the driver holds the blockchain-based ownership token.

1. **Smart Contracts**

User-defined assets could be represented with the help of a smart contract on a smart contract blockchain e.g. ethereum blockchain. The contract could store the mapping of the addresses of current holders of the asset to the corresponding balances . These balances could be updated with the help of messages sent to the contract encoding asset transfer or issuance. The contract could use the conventional authorization scheme of the underlying blockchain in order to check transfer and issuance permissions, or could specify new rules for asset transactions.

**Use cases**

1. **Complex Financial Assets**

Digital assets could represent publicly traded financial assets (e.g., securities). These assets require a high level of security, are heavily regulated and used in business-to-business contexts, therefore requiring

Permissioned blockchains, at least in the short term

Permissionless blockchains could be useful for novel financial services, such as crowdfunding.

1. **Smart Property**

The ownership could be transferred using a transaction with an input bearing the Token

Smart property assets would have slow transaction velocity and would require security before scalability. Therefore, smart property could plausibly be implemented with the help of dedicated ownership protocols on top of highly secure public blockchains, which do not necessarily support the concept of smart property natively.

1. **Electronic Money**

Digital assets could represent e-money, such as alternative currencies (e.g., local currencies or in-game

currencies) or claims of fiat money. Electronic money pegged to real-world currencies generally have

high transaction velocity; therefore, they would require scalable, high-throughput infrastructure provided

by multi-asset blockchains. Currencies with lower transaction velocity (e.g., local currencies)

could use multi-asset blockchains, colored coin protocols or metacoins.

1. **Business to Consumer Assets**

Digital assets could be used to represent discount, coupons, vouchers, gift cards, loyalty points etc. The assets would be issued by a merchant and transferred to buyers during purchases; the merchant would define a transparent set of rules of how assets can be redeemed for goods. A large retailer could issue multiple types of tokens and track their distribution and ownership, which would be useful for analyzing

the customer base. Compared to existing implementations, blockchain infrastructure would provide

a built-in secondary market for assets (although asset transfer could be restricted with the help of issuance metadata).

1. **Digital Subscription**

Digital assets could be used to monetize access to digital resources, such as stream content

Because of the transparency of blockchains, the content provider could easily check when the user’s token was issued and whether it is still valid. The provider could issue multiple types of tokens that correspond to various levels of access (read/write, or read-only), or to the access to specific resources or types of resources.

Similar to digital subscription, non-transferable digital assets could be useful for role-based authentication.

1. **Digital Democracy**

Digital asset coins can be used to implement voting by sending tokens to the one of several Designated addresses While the existing digital asset systems are not secure enough to hold government elections,

they can be used for voting among shareholders or in contests; in the latter case, voting process is

easily monetized.

Permissionless or loosely regulated permissioned blockchains are expected to play a significant role in emerging IoT and consumer-to-consumer markets.

Multi asset blockchain and smart contract blockchains come as a viable alternative for business to consumer and consumer to consumer digital asset issuance.

A permissionless blockchain is suitable for on chain assets (virtual bearer assets) whereas in a permissioned permission less blockchain , a bearer asset becomes a registered asset and blockchain maintainers have a greater transparency and control on assets transfer across users compared to a permission less blockchain.

A permissioned blockchain is more suitable for off-chain assets (e.g. fiat, securities or titles).

1. **Open source versus Proprietary Blockchain Platforms**

The open source [PT-BSC (Blockchain Security Controls)](https://github.com/primechain/blockchain-security-controls) defines a blockchain as a peer-to-peer network which timestamps records by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work.

A blockchain can be permissioned, permission-less or hybrid.

On the other hand, a distributed ledger (DLT) is defined as a peer-to-peer network, which uses a defined consensus mechanism to prevent modification of an ordered series of time-stamped records. Consensus mechanisms include Proof of stake, Federated Byzantine Agreement etc.

Some of the popular open source Blockchain/DLT systems are

**BigChainDB**

BigchainDB is an open source system that “*starts with a big data distributed database and then adds blockchain characteristics — decentralized control, immutability and the transfer of digital assets*”.

BigchainDB seeks to attain performance of 1 million writes per second throughput, storing petabytes of data, and sub-second latency.

**BigchainDB key features include:**

* Each write is recorded on the blockchain database without the need for Merkle Trees or sidechains.
* Support for custom assets, transactions, permissions and transparency.
* Federation Consensus Model (federation of voting nodes).
* Supports public and private networks.
* Has no native currency — any asset, token or currency can be issued.
* Set permissions at transaction level.
* It is open source.

**Consensus mechanism:** Federation of nodes with voting permissions

**Useful links:**

* BigchainDB official website: <https://www.bigchaindb.com/>
* BigchainDB whitepaper: <https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf>
* BigchainDB roadmap: <https://github.com/bigchaindb/org/blob/master/ROADMAP.md>

**Chain Core**

Chain Core is a blockchain platform for issuing and transferring financial assets on a permissioned blockchain infrastructure. Chain Core runs on the open-source Chain Protocol. Chain Core Developer Edition is free while the Chain Core Enterprise Edition is a commercial product.

The creation, control and transfer of assets are decentralised among participants on Chain blockchain networks. The operation of the network is governed by a federation — a designated set of entities. The assets on Chain blockchain networks include currencies, securities, derivatives, gift cards, and loyalty points.

**Chain core key features include:**

* Native digital assets — currencies, securities etc.
* Role-based permissions for operating, accessing, and participating in a network.
* Support for multi-signature accounts.
* Federated consensus.
* Support for smart contracts.
* Transaction privacy.

**Consensus mechanism:** Federated consensus

**Links**

* Chain Core official website: [https://chain.com](https://chain.com/)
* Chain Core whitepaper: <https://chain.com/docs/protocol/papers/whitepaper>

**Corda**

Corda is an open-source distributed ledger platform with pluggable consensus — “it supports multiple consensus providers employing different algorithms on the same network”.

Corda is probably the only distributed ledger platform with pluggable consensus.

**Corda’s key features include:**

* No global broadcasting of data across the network.
* Pluggable consensus.
* Querying with SQL, join to external databases, bulk imports.

**Consensus mechanism:** Pluggable consensus

**Useful links:**

Corda official website: [https://www.corda.net](https://www.corda.net/)

Corda whitepaper: <https://docs.corda.net/_static/corda-technical-whitepaper.pdf>

**Domus Tower Blockchain**

Domus Tower Blockchain is an interesting solution that has been designed for regulated environments such as securities trading where participants know each other and can independently decide whom to trust.

According to its whitepaper, Domus Tower Blockchain has been “benchmarked at ingesting over 1 million transactions per second on hardware costing less than $50 per hour on Amazon’s Web Services with the potential to scale to greater than 10 million transactions per second”.

Data storage is contained in a Merkle directional acyclic graph (MerkleDAG) and nodes on this graph are referred to as “blocks”. The data transmitted to the blockchain is digitally signed and verified before it is written to a block.

**Domus Tower Blockchain’s key features include:**

* Creation of linked blockchains where the assets of an account on one blockchain must match the liabilities on the account of another blockchain.
* Capability of recording a high rate of transactions in a scalable manner.
* Recording of double-entry balance sheet that tracks credits and debits.

**Consensus mechanism**: Any agent that has write access to a blockchain has 100% authority to write transactions to that chain. Authority is centralized under this model.

**Useful links:**

* Domus Tower Blockchain official website <http://domustower.com/>
* Domus Tower Blockchain whitepaper <http://domustower.com/domus-tower-blockchain-latest.pdf>

**Ethereum**

Ethereum is a decentralized platform that runs smart contracts on a custom built blockchain.

**Ethereum’s key features include**:

* Ethereum Wallet — which facilitates holding crypto-assets as well as writing, deploying and using smart contracts.
* Creation of crypto-currencies
* Creation of democratic autonomous organizations (DAOs)
* Command line tools built in Go, C++, Python, Java etc.

**Consensus mechanism**: Ethash, a proof of work algorithm

**Useful links:**

* Ethereum’s official website: <https://ethereum.org/>
* Ethereum whitepaper: <https://github.com/ethereum/wiki/wiki/White-Paper>

**HydraChain**

HydraChain is an Ethereum extension for creating Permissioned Distributed Ledgers for private and consortium chains.

**HydraChain’s key features include:**

* Full Compatibility with the Ethereum Protocol
* Accountable Validators
* Instant finality of blocks and no forks or reverts.
* Support for sub-second block times.
* New blocks are only created in the presence of pending transactions.
* Infrastructure for developing smart contracts in Python.
* Customizability of transaction fees, gas limits, genesis allocation, block time etc.
* Open Source

**Consensus mechanism**: Byzantine fault tolerant consensus protocol

**Useful links**:

HydraChain official site: <https://github.com/HydraChain/hydrachain>

**Hyperledger Fabric**

Hyperledger Fabric supports the use of one or more networks, each managing different Assets, Agreements and Transactions between different sets of Member nodes.

**Hyperledger Fabric’s key features include:**

* Query and update ledger using key-based lookups, range queries, and composite key queries.
* Read-only history queries.
* Transactions contain signatures of every endorsing peer and are submitted to ordering service
* Peers validate transactions against endorsement policies and enforce the policies
* A channel’s ledger contains a configuration block defining policies, access control lists, and other pertinent information
* Channel’s allow crypto materials to be derived from different certificate authorities

**Consensus mechanism**: Consensus is ultimately achieved when the order and results of a block’s transactions have met the explicit policy criteria checks.

**Useful links**:

Hyperledger Fabric’s githib page: <https://github.com/hyperledger/fabric>

**Multichain**

Multichain is an open-source blockchain platform, based on bitcoin’s blockchain, for multi-asset financial transactions.

**Multichain’s key features include:**

* Native multi-currency support.
* Atomic two- or multi-way exchanges of assets between participants.
* Permission management.
* Rapid deployment.
* Multiple networks can simultaneously be on a single server.
* Per-network custom parameter (permitted transaction types, confirmation times, minimum quantities, transaction rate and size limits).
* Data streams.

**Consensus mechanism**: Distributed consensus between identified block validators. This is similar to Practical Byzantine Fault Tolerance) with one validator per block, working in a round-robin type of fashion.

**Useful links**:

* Multichain official site: <http://www.multichain.com/developers/>
* Multichain whitepaper: <http://www.multichain.com/white-paper/>

Most of the commercial (proprietary) Blockchains provide platform as a service (PaaS) or Blockchain as a service (BaaS) offering

Few commercial blockchain platforms include

**Dragon Chain**

A turnkey proprietary blockchain and smart contract platform

The use cases of Dragon Chain include Identity systems, ticketing, distributed storage, processing and computing.

Dragonchain smart contracts run in a trusted context such that sensitive business data and business logic are not exposed to the network.

There are also multiple types of smart contracts in Dragonchain.

* Transaction smart contract – captures business logic for transaction approve/deny
* Broadcast receipt smart contract – allows user to execute code when transactions reach specific level of consensus.
* Subscription smart contract – allows user to execute code against subscribed transactions/data feed from another node.
* Cron/scheduled smart contract – allows user to schedule recurring or timed execution of business logic.
* Library smart contract – allows a node to expose or use reusable utility smart contracts.

The DragonChain platform utilizes serverless architecture to enable to enable simple and powerful scaling, and allows developers to code in Python, Java, Node, or C+. The platform is also currency agnostic, so users of the DragonChain platform can build applications with or without a currency, or even with multiple currencies. There will also be an incubator and a marketplace for new projects built on the DragonChain platform.

DragonChain tokens (also called as Dragons) can be used for access to any part of the DragonChain platform, such as spinning up a node, accessing advanced smart contract libraries, access to incubated projects, and early/discounted access to incubated project tokens.

The user can control level of decentralization of business nodes.

Some of the benefits of Dragon proprietary blockchain over other blockchains are

* Blockchain expertise not required
* Ease of integration
* Currency Agnostic
* Interoperability
* Protection of Business Data
* Short fixed length blocks
* Simple Architecture

**Chain Core (Enterprise)**

Chain core is enterprise grade blockchain infrastructure platform for building financial services

* Enables institutions to issue and transfer financial assets on permissioned blockchain network.
* This can be conceived as a novel type of ledger that is shared across entities and enables electronic records to behave like transferable financial instruments, eliminating many of the complex messaging-based systems that are typically involved in clearing, reconciliation, and settlement.
* Designed for currencies, securities, and other issued financial instruments
* Role-based permissions for operating, accessing, and participating in a network
* A perfectly auditable record of transaction activity that cannot be forged or altered
* Native integration with hardware security modules, multi-signature support, best-in-class cryptographic primitives, and an auditable, open source stack
* Transaction privacy
* Federated consensus designed for immediate transaction confirmation with absolute finality
* Throughput to meet market-scale applications and server architecture designed for high availability
* Assets definitions, compliance data, and arbitrary annotations are included directly in the transaction structure

**Blockchain as a service (BaaS) on Microsoft Azure**

Azure Blockchain service applies tried-and-true digital-signature technology to create transactions which reduce fraud and establish trust and accountability.

Azure blockchain as a service Key features are

* Cryptograhically Secure , Shared Distributed Ledger
* [Blockchain as a Service (BaaS)](https://azure.microsoft.com/en-us/solutions/blockchain/) by Microsoft Azure claims to provide a rapid, low-cost, low-risk, and fail-fast platform for organizations to collaborate together by experimenting with new business processes – backed by a cloud platform with the largest compliance portfolio in the industry.
* As an open, flexible, and scalable platform, Microsoft Azure makes it ridiculously easy to spin up the blockchain of your choice, including leading platforms such as Ethereum, Quorum (EEA), Hyperledger Fabric, R3 Corda and Chain Core that address specific business and technical requirements for security, performance, and operational processes.
* They additionally claim that their intelligent services, such as Cortana Intelligence, are able to provide unique data management and analysis capabilities unlike any other platform offering.
* Azure BaaS Data and AI platform provides unique off-chain data-management and analysis capabilities that no other platform offers.
* Azure provides a rapid, low-cost, low-risk and fail-fast platform for organisations to collaborate on by experimenting with new business processes—and it is all backed by a cloud platform with the largest compliance portfolio in the industry.
* The BaaS offerings could help companies who don't want to build out new infrastructure or try to find in-house developers, which are in hot demand.

There is a plenty of choice available in open source as well as propriety Blockchain platform and service. The particular choice by a startup or enterprise organization depends on

* Scale of Investment and cost considerations
* Scalability and Throughput requirements
* Consensus mechanism
* Type of digital asset holding
* Specific use cases e.g. financial or non-financial
* In-house Development expertise
* Type of Blockchain network – permissioned , permissionless , private or public
* Access control
* Off chain data management
* Security considerations

Different Open source blockchain platforms are suitable options in implementing different consensus protocol mechanism, block chain network types or specific use cases. Good option when implementing blockchains with more censorship resistant use cases.

The use of open source blockchains would reduce the investment cost in building blockchain services however organizations may need to manage the security, scalability and throughput considerations in their own custom ways. Interoperability and Ease of integration is also not much good in open source blockchain platforms.

The enterprise Blockchains offer enterprise platform capabilities as well as blockchain as a service (BaaS) which provide enterprise scale performance and benchmark proved scalability and throughput. BaaS helps organizations in lowering their cost of investment in procuring hardware and software components for implementing a blockchain. The data security and privacy concerns are well met with almost all proprietary blockchain types However proprietary blockchains implement specific protocols and are designed to build specific industry segment requirements e.g. financial, banking , capital market or mortgages.

More better in implementing trusted and permissioned networks with more censorship, role based access permissions on blockchain.

1. **Operational Considerations, tools. Monitoring**

The Blockchain protocol defines three functional roles an entity can play on a blockchain network

* Asset Issuers – define and issue digital assets
* Account Managers –Custody and transfer assets
* Observers – receive blocks and view blockchain data but do not create transactions

Corporations, brands, merchants and governments can act as asset issuers.

Custodians and Banks can transform into account managers on a blockchain network

Meanwhile regulators and risk managers can reinvent their roles with real time insight and perfectly auditable records.

Any entity running a blockchain network can participate in one or multiple of these roles.

The firm that launches a blockchain network in market, is called as operator of that blockchain. Exchanges, brokers, payments networks or government agencies are examples of entities that adopt the responsibility of network operators.

Network operators perform following four functions on a network

* Determine who can participate in the network
* Gather signed transactions from participants
* Generate and sign blocks of these valid transactions
* Distribute blocks to participants

A block is valid when it is signed by a quorum of block signers in a process called federated consensus.

All members of the network know the identities of block signers and accept blocks only if they have been approved by a threshold number of block signers.

Each network participant can also cryptographically validate the whole chain of transactions. This consensus process ensures that competing transactions are resolved and guarantees that transactions are final.

In order to operate or participate in a blockchain network, an entity runs a node in the network.

Implementation can be based on open source blockchain protocol or using proprietary blockchain platform or services. The nodes in the permissioned network are designed to run in enterprise IT environments. In case of public permissionless blockchain implementations, complete blockchain node chain is deployed on participant machines in decentralized manner and each machine act as node connected with other nodes to form Internet Of Value.

Any blockchain network operator manages following layers of implementation

* Storage Layer – stores global blockchain data as well as local account data
* Services Layer - services layer is on top of storage layer that allows creation of assets and transactions.
* Communication layer – consists of API that connects applications and link nodes together
* SDK layer – allows developers to create applications on top of stack

The major considerations in operating a blockchain network are security, performance throughput and scalability.

* Proper network management and rotation of key material is required to secure digital assets.
* Industry standard Hardware security modules (HSM) technology should be integrated with blockchain network to secure the blockchain nodes and transaction signing. Multi-signature accounts using independent HSMs could further increase security.
* The deployment model of blockchain network should be based on vertical scaling of server resources as well as scaling blockchain horizontally over several servers, deployed across many data centers. The linear scale out strategy for increasing scalability by addition of more hardware would provide a very scalability of blockchain network with a high availability.
* The communication and service layer should follow stateless architecture so that high availability could be simply achieved by simple addition of active redundant servers.
* All new features should undergo rigorous performance testing and optimization to ensure optimized resource utilization and high throughput.
* The high availability of storage layer should be achieved with a combination of synchronous and asynchronous replication together with a simple failover scheme.
* The application requests should be load balanced across the communication and services layer and data should be replicated and sharded across storage layer to achieve high performance and high availability.
* Blockchain platforms are not just data management platforms but need to be integrated with enterprise integration adaptors and identity management platforms to provide specialized Dapps based functionalities built upon blockchain network. So there is a need to implement EAI patterns based interoperability standards in designing middleware for blockchain applications.
* Tool based Monitoring of blockchain network is important to monitor blockchain transaction activities and detect t any suspicious or maligning activities.
* Using blockchain explorers (Etherscan.io, Etherchain.org, Digix, Augur etc.) for quickly checking transaction or a specific smart contract activity is ok. But it gets tricky when you want to do real monitoring on the long run as :
* there is no control on what is scanned or what information
* the service is not local, so you are at risk any moment the service is not available
* since these explorers take the task of monitoring and reporting activity about the whole blockchain you will end up with some restrictions. Etherscan, for the example doesn't process requests that return more than 10,000 transactions.
* The solution is to create a local tool that can run on blockchain network or server that will monitor specific addresses you specify and return the whole activity they conduct.
* High availability and disaster recovery planning should be provisioned to ensure the critical service availability during network failure.
* Blockchain operator should maintain a robust and up-to-date and internal knowledgebase repository and resources with right skillset to manager blockchain network and operations effectively.

1. **Performance Considerations – Benchmarks**

Most of the Blockchain vendors claim performance of blockchain network in terms of TPS (transactions per second).

e.g. According to the claim of fabric, 100,000 TPS is the aim to achieve if there are about 15 validating nodes running in proximity in Hyperledger fabric blockchain network.

However as per the results of past performance stress tests done in fabric environment using the simplest example of running chaincode on 4 peer nodes running on different servers in close proximity , query performance for each peer could reach 7000 QPS per second, while the simple invoke performance for each peer was only 700 TPS. (Hardware environment: Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz 64G DRAM 1T SATA Disk)

Blockchain throughput is linearly scalable by addition of more peer nodes, however even if the throughput could be linear scalable, the peak TPS of current system would be only 10,000 on P2P networking of 15 nodes, which is only 1/10 of the claim made by fabric and this is due to the fact that the overhead of PBFT consensus would grow exponentially with the increasing number of nodes offsetting the linear scaling factor of blockchain throughput with addition of more nodes.

The TPS performance of blockchain is largely impacted by following three factors

* Total Number of nodes in blockchain network as blockchain scalability linearly scales with addition of more nodes in P2P network
* Total number of nodes involved in consensus to validate a transaction as consensus delay exponentially grows with more number of nodes participating in consensus validation.
* Type of consensus protocol used in validation of transaction by verifier nodes.

The TPS performance claim made by any blockchain vendor should be assessed in terms of following supporting metrics -

* How is transaction defined for this claim?
* How is throughput defined for this claim, considering what were the start and stop points for measuring TPS?
* What number of peer nodes are meant to receive a copy of the transaction, AND are involved in validating the transaction, e.g., is the platform broadcasting, is it limiting interaction between parties?
* What consensus model is used to validate the transactions? What protocol? Is it probabilistic? What is the consensus delay?
* What data store is used?
* What peer communication protocol is used?
* What were the software and hardware environment conditions for the test?
* What use case(s) did the test workload represent?
* What testing tools were used to complete the test?
* May the results be replicated by other parties?
* What are the performances SLA for the target network?

The Consensus delay is the most impacting factor in determining the performance of a blockchain network. For example, it’s evident that scaling the number of nodes in a broadcast network using a probabilistic consensus protocol such as Proof-of-Work presents an enormous scaling barrier.

This has motivated a number of platform builders, including R3 to consider “performance & scalability” in their platform designs. For example, Corda limits the consensus interaction to only the parties involved in a particular transaction, along with the consensus pool needed to verify uniqueness, and validate the contract if requested. Other platforms, e.g., Hyperledger Fabric V1.0 have also taken a bespoke approach to minimizing transaction sharing. Of course, the primary reason for restricting transaction sharing is “privacy” under the principle, “the best way to keep a secret is to not share it.” However, this policy does also provide ancillary performance benefits. Some might debate the loss of network resiliency in such a restrictive model.

Some of the blockchain network vendors e.g. Corda delegates the task of validating transactions to pool of selected notes (Consensus Notary pools) These Notary pools provide a uniqueness service by operating consensus over uniqueness by nodes operated by a set of distrusting entities.

A notary consensus pool could differ by the protocol configuration, and by their size (number of notary nodes in the pool), and their location (for a given pool, notary node location could be in any geographic location) which may impact the performance of a blockchain network.

Blockchain’s performance is further determined by the number of transactions in a block (block size) and the time between blocks (dwell time). Playing with parameters by increasing block size, or decreasing transaction size or dwell time can provide a significant one-time boost and optimize blockchain for today’s network,

The performance of a blockchain application is also determined by the architecture of storage, services and interoperability layer and the capacity of the hardware used and the network used to connect the peer nodes.

The mining volume is an additional constraint for Ethereum, as serialising mining as Bitcoin does limits the number of computations per block. Sharding an Ethereum chain might improve its performance as it would enable smart contracts to be processed in parallel

Open source Hyperledger “Caliper” project should be used to conduct performance benchmarking on a given blockchain network, before deciding on a particular vendor choice. The tool is designed for Hyperledger but is platform agnostic and can be used with any other blockchain network.

1. **Product Resource requirement**

Following development tools/environments are needed to develop blockchain Dapps.

**Foundation Blockchain Platform -**

There are several existing networks on the likes of Bitcoin, Ethereum or Hyperledger that can be used to build Dapps. Ethereum and Bitcoin are both decentralized; public chains that are open source, while Hyperledger is private and also open source.

Bitcoin may not be a good choice to build Dapps on as it was originally designed for peer-to-peer transactions and not for building smart contracts.

**Ethereum Development Ecosystem –**

**Solidity**

It’s an object-oriented language that developers can use for writing smart contracts. The best part of Solidity is that it can be used across all platforms – making it the number one choice for many developers to use. It’s a lot like JavaScript and way more robust than other languages. Along with Solidity, developers might want to use Solc, the compiler for Solidity. At the moment, Solidity is the language that’s getting the most support and has the best documentation.

**Serpent**

Before the dawn of Solidity, Serpent was the reigning language for building Dapps. Something like how bricks replaced stone to build massive structures. Serpent though is still being used in many places to build Dapps and it has great real-time garbage collection.

**Geth**

* Geth is the official client software provided by the Ethereum Foundation.
* It is written in the Go programming language.
* Components of Geth:
* Client Daemon
* Geth console
* Mist Browser

**Geth Client Daemon**

* Client Daemon connects to other nodes and communicates with them to keep it’s copy  of the blockchain up to date.
* Has the ability to mine blocks and add transactions to the blockchain.
* Validates the transactions in the block and also executes the transactions.
* It also acts as a server by exposing APIs you can interact with through RPC.

**Geth Console**

* Geth console is a command line tool which lets you connect to your running node.
* Performs various actions like
* create and manage accounts
* query the blockchain
* sign and submit transactions to the blockchain

**Geth Mist Browser**

* Mist Browser is a desktop application used to communicate with the blockchain.
* It can be considered as GUI for the geth console since it supports the actions performed through Geth console

**Parity**

* Parity is an Ethereum client that s integrated directly into your web browser providing access to all the features of the Ethereum network including dApps
* Parity is a full node wallet, which means that  you store the blockchain on your computer
* It allows you to:
* access the basic Ether and token wallet functions
* create and manage your Ethereum accounts
* manage your Ether and any Ethereum tokens
* create and register your own tokens etc.
* Parity has a number of features that erfect for deployment in private or consortium setting.
* Fast transaction processing
* Proof-of-Authority consensus engines
* Privacy and control features
* Variety of deployment solutions
* Ability to augment features

**Remix**

* Remix is an IDE for the smart contract programming language Solidity and has an integrated debugger and testing environment.
* With Remix you can
* Develop smart contracts (remix integrates a solidity editor).
* Debug a smart contract’s execution.
* Access the state and properties of already deployed smart contract.
* Debug already committed transaction.
* Analyze solidity code to reduce coding mistakes and to enforce best practices.
* Together with Mist (or any tool which inject web3), Remix can be used to test and debug a dApp.

**MetaMask**

* MetaMask is a chrome plugin used to interact with the Ethereum node.
* Allows you to run Ethereum dApps right in your browser without running a full Ethereum node
* MetaMask includes a secure identity vault, providing a user interface to manage your identities on different sites and sign blockchain transactions.

**Embark**

* Embark is a framework for dApps that handles compiling, deploying, and interfacing with the contracts.
* Embark integrates with EVM, Decentralized Storages (IPFS), and Decentralized communication platforms (Whisper and Orbit).
* Swarm is supported for deployment.
* It allows blockchain developers to develop and deploy dApps easily, or even build a serverless HTML5 application that uses decentralized technology. It equips developers with tools to create new smart contracts which can be made available in JavaScript code

**Features of Embark framework**

* **Decentralized Storage (IPFS)**
* Easily store & retrieve data on the dApps through EmbarkJS including uploading and retrieving files.
* Deploy the full application to IPFS or Swarm.
* **Decentralized Communication platforms (Whisper, Orbit)**
* Easily sends/receives messages through channels in P2P network through Whisper or Orbit.
* **Web Technologies**
* Integrate with any web technology
* Use any build pipeline or tool you wish

**Truffle**

* Truffle is another Framework for dApps
* Features of Truffle
* Built-in smart contract compilation, linking, deployment and binary management.
* Automated contract testing and rapid development
* Configurable build pipeline with support for custom build processes.
* Scriptable deployment & migrations framework.
* Network management for deploying to many public & private networks.
* Interactive console for direct contract communication.
* Instant rebuilding of assets during development.
* External script runner that executes scripts within a Truffle environment.

**Angular**

* Angular is a platform that makes it easy to build applications with the Web, Mobile and Desktop.
* Angular combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges.
* It can be used for developing the front end pages for the dApps

**WebPack**

* WebPack is an open-source JavaScript module bundler.
* WebPack takes modules with dependencies and generates static assets representing those modules.
* Bundles ES Modules, CommonJS and AMD modules
* Can create a single bundle or multiple chunks that are asynchronously loaded at runtime reducing the load time
* Dependencies are resolved during compilation, reducing the runtime size.
* Loaders can preprocess files while compiling, e.g. Typescript to JavaScript, Handlebars strings to compiled functions, images to Base64, etc.
* Highly modular plugin system to do whatever the application requires.

**Swarm**

* Swarm is a distributed storage platform and content distribution service, a native base layer service of the Ethereum web 3 stack
* Designed to deeply integrate with the devp2p multiprotocol network layer of Ethereum as well as with the Ethereum blockchain for domain name resolution, service payments and content availability insurance
* Provides a sufficiently decentralized and redundant store of Ethereum’s public record, in particular to store and distribute dApps code and data as well as block chain data
* Offers peer-to-peer storage and serving solution

**IPFS**

* IPFS (Inter Planetary File System) is a protocol designed to create a permanent and decentralized method of storing and sharing files.
* is a distributed file system that seeks to connect all computing devices with the same system of files.
* is a content-addressable, peer-to-peer hypermedia distribution protocol.
* defines a content-addressed file system
* coordinates content delivery
* combines Kademlia + BitTorrent + Git
* is a web that can be used to view files accessible via HTTP at http://ipfs.io/<path> like the web
* uses cryptographic-hash content addressing
* is P2P
* has a name service: IPNS, an SFS inspired name system
* Work in progress to integrate domain naming service with IPFS URL to give meaningful URLs.

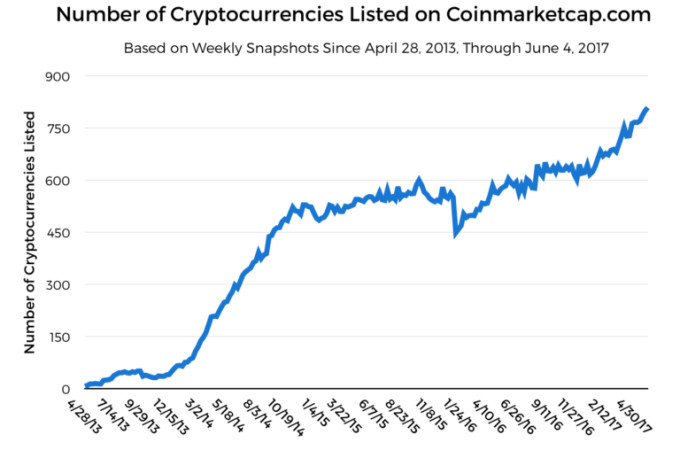
**Internal Knowledgebase requirement**

Blockchain development requires internal knowledgebase of following skills

* Distributed Application Development
* Web App and Mobile Development
* Middleware (web services , middleware platforms and tools)
* Knowledge of Object Oriented Programming languages (Java , C#,C++, Go)
* System Management teams (Data center, Security and Network management)
* Architects with expertise in development of enterprise scale architectures.
* Domain experts with domain knowledge of specific industries (e.g. financial, Banking, Supply Chain Management, Health Science etc.)

1. **Product purpose vs. Requirement Purpose**

For most of the history of blockchain-based currencies and assets, the story has been all about Bitcoin. At a market capitalization of around $40 billion, it remains the [most valuable cryptocurrency](https://about.crunchbase.com/news/morning-report-cryptos-take-14b-breather-following-highs/). But with the rise of a new ‘chain on the [Ethereum](https://ethereum.org/), and new ways to fund the development of new crypto-platforms with ICOs, the narrative is shifting somewhat to the [entire cryptographic asset class](https://about.crunchbase.com/news/morning-report-54-cryptocurrencies-worth-10m/).



From the chart above it shows in roughly the past 12 months, the number of cryptocurrencies listed on [**CoinMarketCap.com**](http://coinmarketcap.com/), a main reference site for digital asset developers and speculators alike, has increased significantly.

The open-source nature of most cryptocurrency systems means that it’s trivially easy to make copies of the software (or “fork” its code), make some modifications to the protocol and release it as a new, wholly separate system.

However after the price spike of Bitcoin in 2013, many new crptoassets came in vogue because it was getting very costly to mine on Bitcoin with costly graphic card hardware.

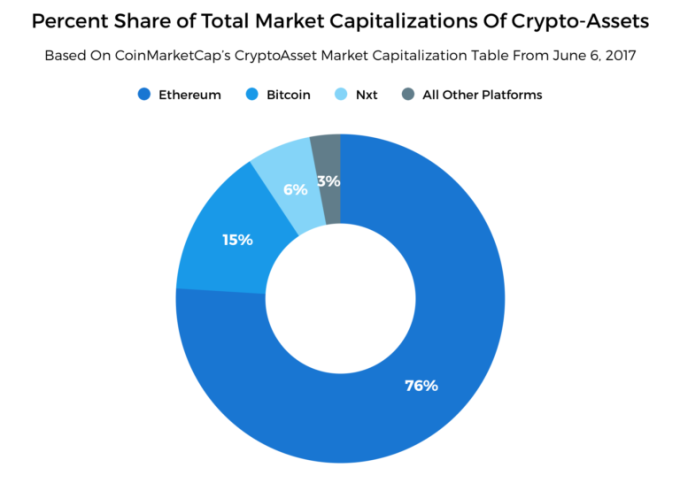
More ever with the increased use of blockchain assets and crypto currencies in real world applications as well as non-financial use cases , the value of et ethereum blockchain Increased rapidly over Bitcoin blockchain network.

Bitcoin is a relatively bare-bones blockchain system that requires layers of protocols to be built on top of it to make it a usable platform for utilities like smart contracts. Platforms like [Counterparty](https://counterparty.io/) and [Omni](http://www.omnilayer.org/) are both built on the Bitcoin blockchain and have sprouted their own collection of digital assets and services that ride on top of them. The basic scripting language of Bitcoin is not Turing complete so developers cannot script complex contracts to meet the functional requirements of complex crypto assets and non-financial use cases.

Ethereum, on the other hand, was launched with its own scripting language baked in, making it possible to build complex smart contracts, decentralized autonomous organizations (DAOs), decentralized autonomous apps (DApps) and even other cryptocurrencies with relative ease.

* This ease of development, combined with the rising price of Ether and a desire by early stakeholders to re-invest in the Ethereum ecosystem, has made Ethereum the platform of choice for crypto-asset entrepreneurs. The ethereum scripting language is Turing complete and its smart contract concept enables to develop blockchain use cases for real world scenarios which is much more costlier to implement in Bitcoin network.

From the chart below, it is evident that ethereum has a total 76% share compared to Bitcoin having just 15% share of total market capitalization of crypto assets.



**Source :- CoinMarketCap**

The product and requirements purpose has demanded rise of Ethereum blockchain compared to Bitcoin network for the following differences in two networks

**Ethereum Versus Bitcoin**

|  |  |  |
| --- | --- | --- |
| **Points to be considered** | **Bitcoin** | **Ethereum** |
| Application Difference | * Bitcoin is equivalent of Gold coinsin Digital industry. * Its application was mainly in the monetary transactions. | * Ethereum is like a world computer. * Its application not only ends with money, but it extends to all fields. |
| The speed of Transactions | * Transaction speed is Low, because bitcoin has a block time of 10 minutes. | * Transaction speed is High, because Ethereum has a block time of 10 to 12 seconds. |
| Scripting Language Difference | * Basic scripting language (C++) , not Turing complete | * Ethereum is written in Turing- complete language which consists of C++, Python, Java, Javascript, Haskell, Rust and Go. |
| Type of Blockchain | * Bitcoin has a blockchain of PoW (Proof-of-work) which now consists of 1 Megabyte blocks. * Blocks are mined in every 10 minutes on an average with the SHA-256 hashing. * Bitcoin blockchain can process 3 transactions at a time. | * Ethereum also has a blockchain of PoW (Proof-of-work) but trying to shift to Proof-of-Stake blockchain. * Ethereum blockchain has blocks of different sizes * Ethereum blockchain can process around 25 transactions per second. |
| HASH RATE Difference | * Bitcoin hash rate peak value is 1,803,059,256 GH/s which is higher than Ethereum. | * Ethereum hash rate peak value is  3,010 GH/s (3 TeraHash) which is lower than that of Bitcoins |
| Scalability | * The Design of bitcoin is not scalable. * Bitcoin is trying to improve their method of transaction | * The Design of Ethereum is Scalable. * Ethereum uses a block gas limit for the cost of transactions, therefore it can scale infinitely |
| Mining Difference | * Bitcoin mining is done with high-cost ASICs whose hardware cost almost $2000 and cannot be used for any other purposes besides mining. * The reward of mining will become half in every 4 years and the current reward is 25 Bitcoins per block | * Ethereum is profitable to mine on high-end GPUs basically considering their low power cost. * These graphical cards can be used for playing online and other efficient games. * For mining each block, the reward is 5 Ether. |
| How Protocols work | * Bitcoins use serial numbers for each coin to make them unique and identifiable. | * In Ethereum, the account is a 20-byte address. * External accounts which are controlled by private keys and contract accounts, Contract Accounts controlled by contract codes. |
| Smart contracts | * There is no such method available for bitcoin. | * The most important advantage of Ethereum over bitcoin is the availability of smart contracts. * Ethereum allows the users to create their own digital tokens for different assets, virtual assets, proof of membership, etc. which are called the smart contracts |
| The cost of transactions | * The cost of Transaction is less but the method is different. * Bitcoin depends on the block size and they compete with each other | * The cost of Transaction is less but the method is different. * In Ethereum, it is called gas and depends on many factors like the bandwidth usage, complexity or storage needs. |

Bitcoin is used for developing crypto currencies because of its high Hash rate compared to ethereum. The hash rate of Bitcoin network represents the measures *how much power Bitcoin* network is consuming to be continuously functional. By continuously functional I mean how much hash power is it consuming to generate/find blocks at the normal mean time of 10 minutes and thus represents the mining difficulty.

With a higher Hash rate which means more mining difficulty and thus greater data security and a low transaction cost, makes Bitcoin network favorable for creating new crypto currencies and exchanges.

However , with a basic scripting language which is not Turning complete, low rate of block generation (transaction speed) and lack of support of smart contracts , the Bitcoin is not used for developing non-financial and other financial; use cases where volume and transaction speed is high. Here , ethereum is a preferred network because of high rates of block generation and support of smart contracts.

The requirement. Purpose further determines the type of blockchain network – e.g. for developing permissioned private network, generally open source Hyperledger fabric is a good choice over ethereum blockchain which is suitable for building permissionless public or private networks.

For financial use cases in private network. Multichain which is a variant of Bitcoin is an option.

**Multichain** is another variant of Bitcoin network as it is based on reference implementation of Bitcoin

**What is Multichain**

* An off­ the­ shelf platform for the creation and deployment of private blockchains, either within or between organizations
* Derived from Bitcoin core
* Provides Privacy and easy-to-use package that suits the needs of financial institutions
* Supports Windows, Linux and Mac servers and provides a simple API and command­ line interface
* Ensures that the activities on the blockchain are visible only to the chosen participants
* Controls which transactions are to be permitted
* Enables Mining in a secured manner without Proof of Work and the cost associated with it

**Multichain versus Bitcoin**

* Multichain is a fork of Bitcoin Core, the reference implementation for the bitcoin network.
* Multichain is a private blockchain unlike Bitcoin core which is public
* Multichain provides configurable parameters, much higher performance, a richer set of APIs, multiple native asset support and data streams which Bitcoin core does not
* Multichain allows storing assets and exchange them in a "native way" to do it whereas Bitcoin core is purely a cryptocurrency

**Difference between Hyperledger and Ethereum Blockchain**

.

|  |  |  |
| --- | --- | --- |
| **Characteristics** | **Ethereum** | **Hyperledger** |
| Description of platform | * Generic Blockchain Platform | * Modular Blockchain Platform |
| Governance | * Ethereum developers | * Linux Foundation |
| Mode of Operation | * Permissionless – public or private | * Permissioned private |
| Consensus | * Mining based on Proof of Work (PoW) * Ledger level | * Broad understanding of consensus that allows multiple approaches. * Transaction level |
| Smart Contracts | * Smart contract code (e.g. Solidity) | * Smart Contract code e.g. (Go, Java) |
| Currency | * Ether * Tokens via smart contract | * None * Currency and Tokens via chaincode |

1. **Private Cloud vs. SaaS**

The choice of deploying blockchain on a private cloud versus using blockchain as a service model is governed by following factors

* Sensitivity of data – Data sensitivity and security is paramount for enterprise data for financial institutions and governments. The blockchain vendor deploying their platform over private clouds could provide private distributed cloud storage of data for maintaining data sensitivity. Data sharing among different entities will be controlled by blockchain provider , based on permission rules defined in smart contracts.
* Single source of truth of data required - This is more useful in cases where single source of truth of data is required and blockchain vendor need to provide a self-verifying, tamper-proof chain (sharded, encypted and distributed) of data that can be viewed and shared – openly or via permissions – and so provides a single version of the truth that is distributed between all users without maintaining multiple copies of same data across different users.
* Deployment model of blockchains – Implementation of consortium , semi-private or private blockchains require a greater level of permissioned control over blockchain resources by blockchain providers. With blockchains deployed in their own private clouds, the blockchain providers can better control the resource configurations in a much optimized way by implementing their own custom protocols e.g. permissions, type of consensus protocol, size of consensus notary blocks, number of blocks required to verify transactions etc.
* Maintenance of blockchain network using their own DevOps resources and managing the SLA governance is the direct responsibility of blockchain provider.
* Cost investment is high in maintaining own private blockchain networks but the roganizations can better control the resource usage across users.
* Setting up an environment to test and research blockchain is not a trivial undertaking. One of the big benefits of cloud is the ability to stand up, deploy, test, and break down environments. No large hardware investments are needed, nor any capital investment. The cost involved is only during the time the environments are up and being used.
* The peer nodes will be deployed on the data center servers in private cloud forming the P2P network with other peer nodes.

**Blockchain as a service (BaaS)**

Many of the leaders in the cloud space have seen the potential benefits of offering Blockchain as a Service (BaaS) to their customers and have started providing some level of BaaS capabilities.

As enterprises look to deploy distributed ledgers, the industry's largest IT providers have launched blockchain-as-a-service (BaaS), offering a way to test the nascent technology without the cost or risk of deploying it in-house.

The BaaS offerings could help companies who don't want to build out new infrastructure or try to find in-house developers, which are in hot demand.

**Microsoft(Azure)** –Microsoft  [partnered with ConsenSys](https://azure.microsoft.com/en-us/blog/ethereum-blockchain-as-a-service-now-on-azure/) to provide the [Ethereum](https://www.ethereum.org/) Blockchain as a Service (EBaaS) in their Azure environment. Offering the service will allow “customers and partners to play, learn, and fail fast at a low cost in a ready-made dev/test/production environment.

[Blockchain as a Service (BaaS)](https://azure.microsoft.com/en-us/solutions/blockchain/) by Microsoft Azure claims to provide a rapid, low-cost, low-risk, and fail-fast platform for organizations to collaborate together by experimenting with new business processes – backed by a cloud platform with the largest compliance portfolio in the industry.

As an open, flexible, and scalable platform, Microsoft Azure makes it ridiculously easy to spin up the blockchain of your choice, including leading platforms such as Ethereum, Quorum (EEA), Hyperledger Fabric, R3 Corda and Chain Core that address specific business and technical requirements for security, performance, and operational processes.

They additionally claim that their intelligent services, such as Cortana Intelligence, are able to provide unique data management and analysis capabilities unlike any other platform offering.

**IBM (BlueMix)** –, IBM  [is offering Blockchain as a Service](https://www-03.ibm.com/press/us/en/pressrelease/49029.wss) using the [Hyperledger](https://www.hyperledger.org/). The IBM release stated, “Using IBM’s new blockchain services available on Bluemix, developers can access fully integrated DevOps tools for creating, deploying, running, and monitoring blockchain applications on the IBM Cloud.”

**Amazon (AWS)** - Amazon, [in a collaboration](http://www.forbes.com/sites/laurashin/2016/05/02/amazon-steps-up-blockchain-commitment-web-services-partners-with-digital-currency-group/) with the [Digital Currency Group](http://dcg.co/), one of the largest investors in blockchain firms. Is providing Blockchain as a Service to members of DCG’s portfolio so they “can work in a secure environment with clients who include financial institutions, insurance companies, and enterprise technology companies.

BaaS (Blockchain as a service) option provides users a low cost option to use blockchain services offered by IT vendors. Here the enterprise need to connect with the peer nodes deployed over IT vendor cloud and leveraging blockchain service offerings by It vendors, to form their enterprise P2P chain network.