Contents

[Business Considerations 1](#_Toc504036370)

[Technology Considerations 2](#_Toc504036371)

[Permissioned Versus Public Permissionless Blockchain 2](#_Toc504036372)

[Determining which blockchain is right for you 2](#_Toc504036373)

[Digital Assets and DLT Requirements 3](#_Toc504036374)

[Specific Considerations of Blockchain supporting Digital Assets 3](#_Toc504036375)

[Deployment Models 4](#_Toc504036376)

[Open source versus Proprietary Blockchain Platforms 4](#_Toc504036377)

[How to Buy 4](#_Toc504036378)

[Appendix 6](#_Toc504036379)

[Appendix A - Blockchain Types 6](#_Toc504036380)

[Public or Permissionless Blockchain networks 6](#_Toc504036381)

[Public Permissioned Blockchain (a.k.a. permissioned Permissionless blockchain) 7](#_Toc504036382)

[Private Permissioned Blockchain 8](#_Toc504036383)

[Appendix B – Deployment Models and Common Usecases 10](#_Toc504036384)

[Deployment Models 10](#_Toc504036385)

[Use cases of Blockchain Digital Assets 11](#_Toc504036386)

[Appendix C – Open Source vs. Proprietary Blockchain 13](#_Toc504036387)

[Appendix D- Blockchain as a service (BaaS) 18](#_Toc504036388)

[Appendix E – Platform Resource Requirements 20](#_Toc504036389)

[Foundation Blockchain Platform - 20](#_Toc504036390)

[Ethereum Development Ecosystem 20](#_Toc504036391)

[Angular 22](#_Toc504036392)

[WebPack 23](#_Toc504036393)

[Swarm 23](#_Toc504036394)

[IPFS 23](#_Toc504036395)

[Appendix F – Blockchain Technology Criteria 24](#_Toc504036396)

[Blockchain Scalability and Volume 24](#_Toc504036397)

[Speed and Latency 24](#_Toc504036398)

[Security and immutability 24](#_Toc504036399)

[Storage and structural needs 25](#_Toc504036400)

[Operational Considerations, tools. Monitoring 26](#_Toc504036401)

[Performance Considerations – Benchmarks 28](#_Toc504036402)

[Blockchain Notaries 29](#_Toc504036403)

[Blockchain Network 30](#_Toc504036404)

[User Authentication and Authorization 30](#_Toc504036405)

[Asset Issuance 30](#_Toc504036406)

# Business Considerations

The business problem and requirements must be clearly identified to ensure the appropriate blockchain platform is selected. Blockchain is a cross-cutting and transformative technology but it is not the solution to every problem. Make sure to review and understand the description of the platform to ensure it matches the need. Look past the jargon to understand the specific differences noted below:

* **Governance:** Determine the nature of who controls and governs the software platform. For example, open-source platforms such as Ethereum and Hyperledger Fabric are governed by their developer communities via non-profit foundations whereas Corda is managed by a corporate consortium called R3. The governance model could affect the support available.
* **Mode of Operation**: Blockchain infrastructure can be operated as “permissioned” or “permission-less”. Government typically requires the permissioned mode for security reasons but there could be use cases for connecting to the public permission-less chains.
* **Transaction Costs**: Apps deployed on the Ethereum public blockchain incur transaction costs based on computational resources consumed. Private blockchain implementations do not have this requirement but require their own supporting infrastructure such as cloud servers.
* **Consensus**: Blockchains must reconcile transactions to maintain a single version of truth. Ethereum uses a Proof of Work (PoW) algorithm which ensures a high level of immutability and transparency whereas other blockchains may have more fine-grained approaches that offer better performance and privacy.
* **Smart Contract**: The “smart contract” refers to the blockchain’s ability to store and execute computer programs. Each solution supports a different set of programming languages, so consideration should be given to in-house programming expertise. For example, Hyperledger supports Java and offers a composer tool that allows organizations to develop Smart Contracts without writing much code while Ethereum uses its own Solidity language. Corda also expands on the “smart contract” concept by supporting the incorporation of legal prose as well as programming code.
* **Currency**: Although initial government use cases are likely to be focus on non-currency digital assets such as contracts or land deeds, multiple forms of crypto-currencies may need to be supported in the future. Support for this varies by blockchain with Ethereum having Ether built in, Ethereum and Hyperledger providing the ability to create other cryptocurrencies, and Corda providing little support for currency functionality overall.
* **Chain Data Requirements**: There are multiple different ways that data can be stored on the blockchain with implications for security and performance. For example, you could store an entire set of contract documents in a block or just a hash value of the location of those documents somewhere else. One approach may be more secure but the other could offer higher performance.
* **Operational Requirements**: Will the system operate in a SaaS model or within the agency’s private cloud? Where are all the participating nodes going to reside? What interface requirements have to be met? What is the response time requirements for the blockchain? Other considerations include reliability and uptime in addition to the standard operational requirements for any software solution.
* **Security Requirements**: A key factor in choosing any blockchain solution is to ensure its ability to obtain an Authority to Operate (ATO). A blockchain platform operating in an already FedRAMP authorized environment might be attractive as it makes getting an ATO much simpler. However, government must consider the potential of the peer nodes – the connection points to the blockchain -- residing outside the standard system boundary.
* **Private Cloud vs. SaaS**: The choice of deploying blockchain on a private cloud versus using blockchain as a service model is governed by the following factors:
* Sensitivity of Data – Data sensitivity and security is paramount for government. The blockchain vendor deploying their platform over a private cloud could provide private distributed cloud storage of data for maintaining data sensitivity, as opposed to an Ethereum or Bitcoin like data storage where anyone can get access to the data if the credentials are presented to the controlling node. Data sharing among entities would be controlled by the blockchain provider based on permission rules defined in smart contracts.
* Deployment Model – Implementation of consortium, semi-private or private blockchains requires 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 by implementing their own custom protocols, e.g. permissions, type of consensus protocol, size of consensus notary blocks, and number of blocks required to verify transactions etc.
* System Maintenance – Will you use your own DevOps resources and managing the SLA governance or leave that to the blockchain service provider?
* Initial Costs – Setting up an environment to test and research blockchain is not a trivial undertaking. A private cloud is more expensive upfront but allows the organization to better control the resource usage across users. A service offering is simpler but less configurable.
* **(Please refer Appendix D for more details)**.

# Technology Considerations

### 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 mainly 3 different types of blockchains.

* Public blockchain networks
* Permissioned blockchain networks
* Private blockchain networks

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.

In addition to an introduction to the types of blockchains in the ACT-IAC Blockchain **Primer** for the Government document, **Appendix A** provides further details about these blockchain networks.

## Determining which blockchain is right for you

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

* Product Purpose fit for requirements
* Blockchain Scalability and Volume
* Performance of the Blockchain, especially speed and latency (review benchmarks)
* Security and immutability
* Storage and structural needs
* Operational Considerations
* Technology Resource Requirements (**Appendix E**)
* Internal Resource requirements
* Off chain data management

There may be exceptions depending on project, and it is possible to use a different type of blockchain to reach project goal. Please refer to **Appendix F** for further explanation of the technical criteria.

## Digital Assets and 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.

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

## Specific Considerations of Blockchain supporting Digital Assets

There are some additional considerations that have to be taken into account when using Blockchain technology on a Digital Asset, and are listed below (additional details in **Appendix F**):

* Blockchain Specification
* **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.
* Blockchain Notaries
* Blockchain Network
* User Authentication and Authorization
* Asset Issuance

## Deployment Models

There are various deployment models for Blockchains in the market. Government should consider which model suits them the best. Listed below are the various models. Please refer to **Appendix B** for details.

* Separate Blockchains for Assets
* Colored Coin Protocols
* Multi-Asset Blockchains
* Smart Contracts

## Open source versus Proprietary Blockchain Platforms

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 an area of consideration, as open source blockchain platforms don’t do well traditionally in these areas as well. Details for Opensource / Proprietary models in **Appendix C)**

Blockchain-as-a-Service (BaaS) (details in **Appendix D)** is an up and coming model that combines the benefits of an open source platform with the other benefits that are taken for granted with proprietary solutions.

# How to Buy

When procuring blockchain or distributed ledger technology, agencies should consider agile acquisition methods. Agile acquisition advocates rolling out capabilities in smaller chunks, more frequently. It also deemphasizes extensive up-front capabilities planning. Instead, developers put capabilities into action as soon as possible, then modify and adapt them as needed. Agile acquisition allows programs to be more responsive to changes in operations, technology, and budgets. It also offers more opportunities for collaborating with users and other stakeholders to deliver priority capabilities rapidly. Implementing agile development practices often requires changes in an agency's policies, processes, and culture. But the rewards are ample.

Traditional procurement methods for waterfall software implementations lack the flexibility to take advantage of the benefits of time, schedule, and cost that agile software development methods bring to the Federal government. For this reason, the acquisition workforce needs to make its processes agile, using innovative and creative solutions to procure IT services while maintaining compliance with the FAR and Federal law. For more information on agile acquisitions, please see [FAI’s Agile Acquisition 101](https://www.fai.gov/media_library/items/show/81).

The table below provides a list of possible acquisition vehicles that can be used to procure blockchain:

|  |  |
| --- | --- |
| Acquisition Type | Information |
| Stand-Alone Acquisition | Using the procedures identified in [FAR Part 12 Acquisition of Commercial Items](https://www.acquisition.gov/far/html/FARTOCP12.html), [FAR Part 13 Simplified Acquisition Procedures](https://www.acquisition.gov/far/html/FARTOCP13.html), or [FAR Part 15 Contracting by Negotiation](https://www.acquisition.gov/far/html/FARTOCP15.html).  Competition can be categorized in three ways, in accordance with [FAR Part 6, Competition Requirements](https://www.acquisition.gov/far/html/FARTOCP06.html):   1. Full and Open Competition: All responsible sources can compete. 2. Full and Open Competition “After Exclusion of Sources”: This type of acquisition is reserved for competition among a specific type of business concern. 3. Other Than Full and Open Competition: Sources are limited based on one of the seven reasons listed at FAR 6.302. This generally requires a justification and approval (J&A) in accordance with FAR 6.303 and 6.304. |
| Use of SBA’s 8(a) Program | Allows procuring agencies to make quick, direct awards for procurements up to $4 million in value.  Benefits:   * May select the awardee through market research or capabilities briefings and award directly to the firm without further competition * A sole source justification and authority (J&A) is not required for contracts under $4 million * Awards are not protestable.   Options:   * SBA’s [TechFAR Hub memo](https://techfarhub.cio.gov/assets/files/8aSBA_USDSMemo.pdf) in conjunction with US Digital Service highlights the use of the 8(a) Program for digital services and clarifies that procuring agencies which award digital services requirements through the 8(a) Program need not request release from 8(a) competition when awarding digital service developmental iterations or add-on services. The memo identifies each iteration or add-on is a distinctly new project and should be treated as a new requirement for purposes of 8(a) release requirements and permits agencies to award contracts for additional development utilizing different acquisition strategies (to include non-8(a) strategies) without requesting release from 8(a) competition. * [GSA’s 8(a) STARS II](https://www.gsa.gov/technology/technology-purchasing-programs/governmentwide-acquisition-contracts/8a-stars-ii-governmentwide-acquisition-contract-gwac) government wide acquisition contract (GWAC) leverages SBA’s 8(a) Program authority 8(a) allowing directed task orders up to $4 Million, including options. Orders more than $4 Million must be competed among the industry partners in your chosen constellation and functional area. |
| Leverage of Interagency Vehicles | * [GSA FAS Schedule 70 Contracts](https://www.gsa.gov/about-us/organization/federal-acquisition-service/office-of-information-technology-category): Long-term government wide contracts with commercial companies to provide access to commercial products and services at volume discount pricing. GSA Schedules are Indefinite Delivery, Indefinite Quantity contracts that provide for an indefinite quantity of supplies and services during a fixed period of time. The contract has a five-year base period with (3) fiver year options resulting in a potential 20 year contract. [IT Schedule 70](https://www.gsa.gov/technology/technology-purchasing-programs/it-schedule-70/buy-from-it-schedule-70) is the government’s largest IT contract vehicle that delivers federal, state, and local customer agencies the tools and expertise to shorten procurement cycles, ensure compliance, and acquire the best value for innovative technology, products, services, and solutions. Schedule 70 includes the [FAStlane](https://www.gsa.gov/technology/technology-purchasing-programs/it-schedule-70/sell-through-it-schedule-70/making-it-easier-fast-lane) and [Startup Springboard](https://www.gsa.gov/technology/technology-purchasing-programs/it-schedule-70/sell-through-it-schedule-70/making-it-easier-it-schedule-70-startup-springboard) programs that are part of the Making It Easier initiatives. * [8(a) STARS II Government wide Acquisition Contract (GWAC)](https://www.gsa.gov/technology/technology-purchasing-programs/governmentwide-acquisition-contracts/8a-stars-ii-governmentwide-acquisition-contract-gwac): offers access to highly qualified, certified 8(a) small disadvantaged businesses. The contract has a $10 Billion program ceiling with a five-year base period and one five-year option. * [Alliant and Alliant Small Business Government wide Acquisition Contracts (GWAC)](https://www.gsa.gov/technology/technology-purchasing-programs/governmentwide-acquisition-contracts/alliant-governmentwide-acquisition-contract-gwac): represent the next generation GWAC vehicles for comprehensive information technology solutions through customizable hardware, software, and services solutions purchased as a total package. * [VETS 2 Government wide Acquisition Contract (GWAC)](https://www.gsa.gov/technology/technology-purchasing-programs/governmentwide-acquisition-contracts/vets-2-governmentwide-acquisition-contract-gwac): the successor to the VETS GWAC, VETS 2 is a service-disabled, veteran-owned small business set-aside that provides access to customized IT solutions from a diverse pool of industry partners. * [NIH Chief Information Officer Solutions and Partners 3 (CIO-SP3)](https://nitaac.nih.gov/services/cio-sp3) and [CIO-SP3 Small Business Government wide Acquisition Contract (GWAC)](https://nitaac.nih.gov/services/cio-sp3-small-business): NIH's three GWACs for information technology procurement. CIO-SP3, CIO-SP3 Small Business, and CIO-CS can be used by any federal civilian or DoD agencies to acquire information technology products, services, and solutions. |
| Other Transactional Authority | [Other Transaction Agreements (OTAs)](http://www.transform.af.mil/Portals/18/documents/OSA/OTA_Brief.pdf?ver=2015-09-15-073050-867): generally do not follow a standard format or include terms and conditions required in traditional mechanisms, such as contracts or grants. Meant to help meet project requirements and mission needs. The statutory authorities for most agencies include some limitations on the use of their agreements, although the extent and type of limitations vary.  Benefits:   * OTAs make it easier to work with nontraditional partners such as start-up companies) * Flexible and streamlined method for procurement which can reduce the time and cost of delivery of technological advancements while improving capabilities * Ability to address industry concerns regarding intellectual property and cost accounting provisions that would otherwise need to be included when using traditional mechanisms * Opportunity to tailor some terms and conditions of agreements as needed when working through the agile development of a capability * Not considered procurement contracts, grants, or cooperative agreements even though considered legally binding instruments so they are not subject to the Federal Acquisition Regulation, the Competition in Contracting Act, the Truth in Negotiations Act, or many other federal contracting regulations   Note: Consult with your agency’s legal counsel to determine which laws and regulations are still applicable and which provisions for intellectual property are suitable. |
| NTIS Joint Venture Program | NTIS' basic authority to operate a permanent clearinghouse of scientific and technical information is codified as chapter 23 of Title 15 of the United States Code (15 U.S.C. 1151-1157). This chapter also established NTIS' authority to charge fees for its products and services and to recover all costs through such fees "to the extent feasible." |

# Appendix

## Appendix A - Blockchain Types

### Public or Permissionless Blockchain networks

**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.

Bitcoin and Ethereum are the most prominent examples of Permissionless blockchains, which are public and decentralized. Participants can leave and join the blockchain’s network at any time. No central authority or trusted third party manages who is allowed to join the network, or bans illegitimate users from connecting to the network. Anyone can read the chain, make legitimate changes or write a new block into the chain. Thus the number of readers, writers and untrusted writers are very high.

In a Permissionless setup, the number of nodes is expected to be large, and these nodes are anonymous and untrusted since any node is allowed to join the network. Most of the DMMS (dynamic membership multi-party signature) validators join Permissionless blockchain as a DMMS digital signature to sign blockheads is formed by group of signers of no fixed size. Bitcoin’s blockheaders are DMMSes because their proof of work has a property that anyone can contribute without enrolment required and their contribution is weighted by computational power rather than one threshold signature contribution per party, which allows anonymous membership without risk of sybil attack (where party joins many times and has disproportional input into the signature).

Completely decentralized as the permissions to read and write data onto the Blockchain are shared equally by all the connected users, who come to a consensus before any data is stored on the database. Censorship resistant with anonymous consensus based on a completely trust-less system where no user is given special privileges on any decision

Consensus mechanism to prove the transactional verifiability is mainly based on proof of work (PoW). Nodes have to prove that they have expended significant amount of energy as Proof-of-Work (PoW) towards solving a hard cryptographic puzzle. Proof of work consensus assumes that amount of validators (miners) is unknown and validators are anonymous and have no reputation. In order to vote, proof of work needs to be presented, which requires hard computations. A consensus is reached if parties that control majority (usually 51% of computational power) agree. Highly Immutable blocks as any tempering requires >51% participant node consensus in a very large public blockchain which is almost impossible.

Some permissionless blockchains also support PoS (proof of stake) based consensus. For Example Ethereum (in future), NXT, Peercoin. PoS (proof of stake) consensus assumes that the validator of the next block is chosen in a deterministic way. The chance that a validator is chosen depends on its stake and a validator loses its stake if it commits an attack. The consensus is reached if the parties that control majority of wealth agrees. High public verifiability with each state change validated by verifiers, e.g. miners on bitcoin’s or ethereum’s  blockchains. Any observer, or reader, on the other hand, can verify that the blockchain’s state has changed as per the protocol and eventually, all readers will have the same version of the blockchain.

## Public Permissioned Blockchain (a.k.a. permissioned Permissionless blockchain)

A permissioned Blockchain requires only pre-selected parties to validate transactions. Permissioned blockchains have been proposed to authorize only a confined group of users to participate in the blockchain network. A central authority (consortium) determines and gives right to the predefined peers to write, read , monitor or audit the transactions on blockchain with a public verifiability of content is desired.

Permissioned Blockchains are operated by known entities (known as consortium entities) such as stakeholders of a given industry with immutability and efficiency is preferred over anonymity and transparency.

The participants require some means of identifying each other while not necessarily fully trusting each other.

In the world of business, permissioned blockchain systems often come across critical requirements (from a practical and regulatory perspective) for transactional security and privacy of business logic that is put on a shared ledger. In addition, commonly enterprise-purposed permissioned ledgers need to meet certain performance and scalability standards and/or comply with different cryptographic standards and practices, ultimately calling for modularity of crypto components.

In a permissioned blockchain, organization determines who may act as transaction validator on their network, a blockchain developer may choose to make the system of record available for everyone to read, but they may not wish to allow anyone to be a node, serving the network’s security, transaction verification or mining.

With permissioned blockchains, this may or may not involve 'proof of work' or some other system requirement from the nodes.  There may not need PoW but broad understanding of consensus at transaction level among peer nodes that allows multiple approaches, Generally permissioned blockchain algorithms are based on BFT (Byzantine Fault Tolerance) consensus type.

BFT consensus type assumes that amount of validators are known upfront. Validators know each other and adding or removing a validator require approval of the rest.

Although some degree of decentralization is maintained in their structure, the participants have the power to grant read/write permissions to other participants, leading to the ‘Partially Decentralized’ design of Permissioned Blockchains. The transactions are quick to verify in a Permissioned Blockchain as there are a handful of verifiers, with the transaction fee miniscule thus increasing the overall efficiency of transactions. The Permissioned Blockchains maintain the privacy of a user’s data, without consolidating power with a single organization.

Hyperledger offers open source pluggable architecture based implementation of permissioned blockchain with flexibility for users to configure the consensus module that meets their needs.

### Private Permissioned Blockchain

Permission to write data onto the blockchain is controlled by a single or (group consortium of) organization entities which is highly trusted by all other users. Useful when no public readability of content is desired. This organization may/may not allow users to have access to read the data, as public readability might not be necessary in most cases. In some situations, the organization might want the public to audit the data. Limited/restricted read permissions also provide a greater level of privacy to the users, a feature not available in Public Blockchains.

The organization in control has the power to change the rules of a Private Blockchain and may also decline transactions based on their established rules and regulations.

In a Private Blockchain, the transactions are quicker as they can be verified be a small number of devices. Thus, the users pay lesser amounts of transaction fees since the number of people verifying the transaction is fewer than in a Public Blockchain. The devices are very well connected and any faults can be fixed by human intervention, which can be easily approved by the users since the users trust the single organization in control of the Blockchain. Like public permissioned blockchain, private blockchains have access control layer built into protocol.

With permissioned blockchains, this may or may not involve 'proof of work' or some other system requirement from the nodes.  There may not need PoW but broad understanding of consensus at transaction level among peer nodes that allows multiple approaches, Generally permissioned blockchain algorithms are based on BFT (Byzantine Fault Tolerance) consensus type. BFT consensus type assumes that amount of validators are known upfront. Validators know each other and adding or removing a validator require approval of the rest.

The transactions are quick to verify in a private Blockchain as there are a handful of verifiers, with the transaction fee miniscule thus increasing the overall efficiency of transactions. The private blockchains maintain the privacy of a user’s data, without consolidating power with a single organization.

Hyperledger offers open source pluggable architecture based implementation of private permissioned blockchain with flexibility for users to configure the consensus module that meets their needs.

## Appendix B – Deployment Models and Common Usecases

### Deployment Models

#### 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.

#### 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.

#### 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.

#### 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.

#### 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 of Blockchain Digital Assets

#### 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 crowd funding.

#### 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.

#### 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.

#### 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).

#### 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.

#### 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).

## Appendix C – Open Source vs. Proprietary Blockchain

#### Definition of Blockchain vs. Distributed Ledger Technology

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.

#### Popular open source/proprietary Blockchain/DLT systems

|  |  |  |
| --- | --- | --- |
| **BlockchainType** | **Blockchain** | **Features** |
| Open source | BigChainDB | * 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”. * 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 |
| Open source | Chain code | * 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. * 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 |
| Open Source | Ethereum | * Ethereum is a decentralized platform that runs smart contracts on a custom built blockchain. * 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 |
| Open Source | 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. |
| Open Source | 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. * No global broadcasting of data across the network. * Pluggable consensus. * Querying with SQL, join to external databases, bulk imports. * Consensus mechanism: Pluggable consensus |
| Open Source | Multichain | * Multichain is an open-source blockchain platform, based on bitcoin’s blockchain, for multi-asset financial transactions. * 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. |
| Proprietary | 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. * Based on serverless architecture to enable simple and powerful scaling and allows development in various coding languages (Python, Java, Node or C++) * Currency agnostic platfiorm, applications can be built with or without currency or even with multiple currencies. * There is incubator or marketplace available on this platform for new applications/projects developed on 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 |
| Proprietary | 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 |

## Appendix D- 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.

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.

**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.

## Appendix E – Platform Resource Requirements

### 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 DApps.

**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 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.
  + A distributed file system that seeks to connect all computing devices with the same system of files.
  + 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.

## Appendix F – Blockchain Technology Criteria

### Blockchain Scalability and Volume

The scalability of a particular blockchain network type determines the maximum transaction throughput (number of transactions processed per second) and the maximum volume of transactions that can be processed within a reasonable performance criteria, with a growing blockchain.  
The scalability of a blockchain is impacted by the volume of transactions processed on blockchain, size limit of a block, size of entire blockchain, number of verifying nodes to provide consensus, time taken to process a transaction, high processing fee to be paid for transactions processing etc.

A public permissionless blockchain can have unlimited number of participants to join in network and perform read and write transactions without any censorship resistance and thus the blockchain scalability is majorly constrained due to very large volume of transactions and big blockchain size.

Bitcoin blockchains are much less scaled compared to ethereum network as there is a built in hard limit of 1 megabyte per block (10 minutes to mine a new block) compared to ethereum which takes maximum 20 seconds. Furthermore, there is a cost to performing certain actions on the public Bitcoin or Ethereum networks. BTC spending per transaction is high in bitcoin compared to ‘gas price’ per transaction processing required in ethereum.

Permissioned blockchains have comparatively much lesser blockchain size as participation is controlled and consensus is done using identified selected blockchain notary nodes. The consensus delay is much lower than that in public blockchains. The scalability of a blockchain grows linearly with addition of more hardware. Thus permissioned blockchains can better scale up by using more storage and addition of peer nodes in P2P network.

### Speed and Latency

The blockchain speed is the transaction throughput (maximum number of transactions per second) which is determined by the block size and the consensus delay. As public blockchains have larger number of verifying nodes that are involved in verifying the transactions, the consensus delay is much higher compared to permissioned networks where consensus is achieved by lessor number of verifying nodes( blockchain notaries or identified incentivizing nodes) and has much less latency due to consensus delay and thus high speed. Most of the permissioned networks implement Byzantine tolerance consensus protocol which does not require consensus from every participating node and provide high transaction processing speed.

The speed of transaction processing on a particular blockchain is further determined by any hard limit set on maximum block size e.g. bitcoin has a hard limit of 1 megabyte on a block set which causes a latency of 10 minutes compared to 10-20 seconds of latency in ethereum network.

### Security and immutability

A public blockchain network is completely open and anyone can join and participate in the network. The blockchain network typically has an incentivizing mechanism to encourage more participants to join the network. Bitcoin is one of the largest public blockchain networks in production today.

One of the drawbacks of a public blockchain is the substantial amount of computational power that is necessary to maintain a distributed ledger at a large scale. More specifically, to achieve consensus, each node in a network must solve a complex, resource-intensive cryptographic problem called a proof of work to ensure all are in sync and thus immutability is very high. The disadvantage is the openness of public blockchain, which implies little to no privacy for transactions and only supports a weak notion of security. Both of these are important considerations for enterprise use cases of blockchain.

A private blockchain network requires an invitation and must be validated by either the network starter or by a set of rules put in place by the network starter. Businesses who set up a private blockchain, will generally set up a *permissioned* network. This places restrictions on who is allowed to participate in the network, and only in certain transactions. Participants need to obtain an invitation or *permission* to join. The access control mechanism could vary: existing participants could decide future entrants; a regulatory authority could issue licenses for participation; or a consortium could make the decisions instead. Once an entity has joined the network, it will play a role in maintaining the blockchain in a decentralized manner. Thus immutability of DLT in permissioned blockchains is largely controlled by set of access permission rules and the industry level protocols to achieve consensus with known number of verifying nodes and also most of the participants are trusted users only.

Data privacy is better managed by defining read and writes level access permissions for each users in permissioned blockchains.

### Storage and structural needs

The permissionless blockchain has always untrusted participants and to maintain immutability of transactions stored in blockchain, consensus is required by all participant nodes involved in the blockchain thus the distributed ledgers are shared with all complete blockchain blocks downloaded in a decentralized manner at all participant nodes with greater computational power needed to validate the transactions. Thus blockchain size is too large and more number of transactions are processed. Thus the scalability of permissionless blockchain is managed by adding more storage and processing servers in permissionless blockchains compared to permissioned blockchains where participation is controlled and the consensus size is less as all participant nodes are not required to validate transactions to ensure immutability of distributed ledger but only selected nodes does the transaction validation.

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.

### 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.

### 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 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.

### 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.

### 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.

### 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.

### 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