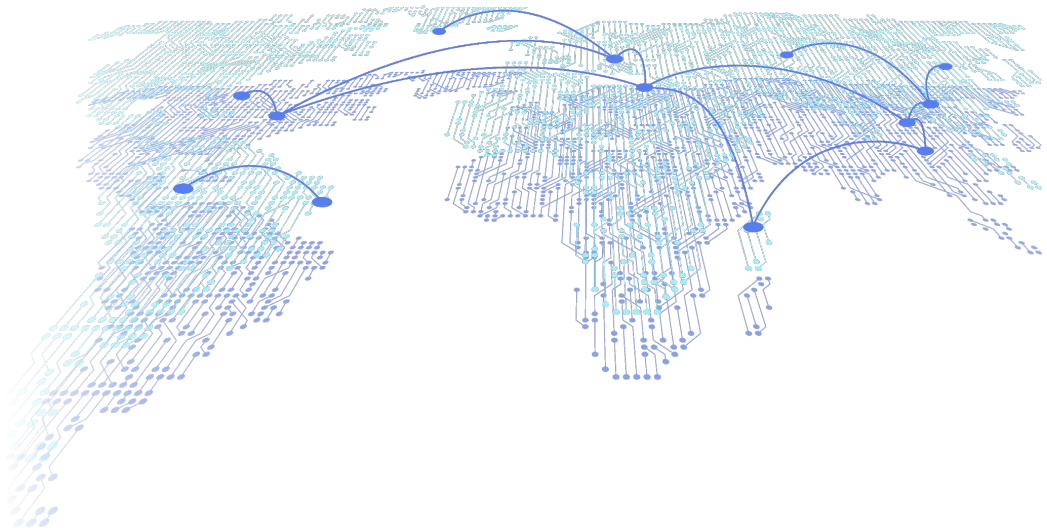
Digital Asset

Blockchain Network Evaluation for Regulated Finance

Authored by Bernhard Elsner
Chief Product Officer, Digital Asset

TABLE OF CONTENTS



Summary

In the context of blockchain use by regulated financial institutions for tokenization or related use cases, there are many properties of the network that matter and play together to make it a viable platform for regulated finance. In the discourse about blockchain for enterprise, these properties often get bundled in binary (private vs public) or one-dimensional (private permissioned, public permissioned, public permissionless) categories. Such simple, broad categories are useful for efficient discussions in which no more nuance is needed. But they are too broad and vague to enable an informed evaluation and comparison of blockchain systems and networks for their applicability to regulated use cases.

This doc lays out a set of independent characteristics each of which matters to such an evaluation. It also offers an evaluation of a range of popular blockchain networks against those characteristics. The characteristics are:

Network Access: Public or Private

Independent Security: Delegated, Trailing, Full

Privacy: Transparent, Sharded, Transaction Level, Sub-transaction level

Infrastructure Independence: Monolith, Hub and Spoke, Network of Networks, Single Operator

Independent Control: DIY, Logical, Consensus, Interactive

Composability: None, Single-Provider, Messaging, Full

Finality: Probabilistic, Slow Deterministic, Instant Deterministic

Native Token Required: Everyone, Validators, None

The above list may well be incomplete, but there is no question that all of the above matter independently of each other for a regulated enterprise to be able to use a network and derive value from it.

The usual terms above (private, public, public permissionless, private permissioned, public permissioned) have a loose association with at least some of the characteristics above.

Term	Network Access	Indep. Security	Privacy	Infra. Indep.	Indep. Control	Composability	Finality	Native Token
Public/Public Permissionless	Public	Trailing	Transparent	Monolith	DIY or Logical	Full	Probabilistic or Slow	Everyone
Public Permissioned	Public	Full						
Private/Private Permissioned	Private	Full		Full Control	Consensus	Single-Provider at best	Instant	None

The Private/Permissioned row here is evaluated from the point of view of the enterprise wanting to tokenize - the application provider. They get to participate in consensus and exert full control, etc. But they cannot compose with applications by other application providers deploying in an equivalent way.

The rhetoric in the current discourse goes roughly like this: Public Permissionless is untenable because the lack of Privacy, Infrastructure Independence and Independent Controls present a risk that the regulators are rightly asking institutions to manage through punitive capital requirements. Private Permissioned networks can be fine to use, but only deliver partial value from blockchain as they don't offer a ready made venue and no blockchain level interoperability between different application providers. Thus, the right kind of network for a regulated enterprise is a "public permissioned" one.

There is a resulting tendency for networks to position themselves as "public permissioned" and thus insinuate their suitability for regulated finance. But as the above table indicates through empty cells on six of the eight characteristics treated in this document, the term has no broadly accepted meaning yet and can therefore be applied to networks that range from indeed being suitable for regulated finance all the way down to networks that differ from a classic public permissionless networks only in some nuance.

The evaluation in this document attempts to show the important differences between networks at that next level of detail. The evaluation is summarized in this table.

Network	Network Access	Indep. Security	Privacy	Infra. Indep.	Indep. Control	Composability	Finality	Native Token
Public / Public Permissionless								
Bitcoin	Public	Trailing	Transparent	Monolith	DIY	Full*	Probabilistic	Everyone
Ethereum	Public	Trailing	Transparent	Monolith	DIY	Full	Slow	Everyone
BNB	Public	Trailing*	Transparent	Monolith	DIY	Full	Probabilistic	Everyone
Solana	Public	Trailing	Transparent	Monolith	Logical	Full	Probabilistic	Everyone
Ripple/XRP	Public	Trailing*	Transparent	Monolith	DIY	Full	Probabilistic	Everyone
Cardano	Public	Trailing	Transparent	Monolith	DIY	Full	Slow	Everyone
Avalanance (C-Chain)	Public	Trailing	Transparent	Monolith	DIY	Full	Instant (Claimed)	Everyone
Polygon PoS (Public MainNet)	Public	Trailing*	Transparent	Monolith	DIY	Full	Probabilistic	Everyone
Internet Computer	Public	None	Transparent to operators	Special*	Logical	Messaging	Instant	Everyone
Stellar	Public	Trailing	Transparent	Monolith	Logical	Full	Instant	Everyone
Polkadot	Public	Trailing	Transparent	Hub and Spoke	Logical	Messaging	Instant	Everyone
Hedera Hashgraph	Public	None*	Transparent	Monolith	DIY	Full	Instant	Everyone
Aptos	Public	Trailing	Transparent	Monolith	Logical*	Full	Instant	Everyone
Algorand	Public	Trailing	Transparent	Monolith	Logical*	Full	Instant	Everyone

Network	Network Access	Indep. Security	Privacy	Infra. Indep.	Indep. Control	Composability	Finality	Native Token
Public Permissioned - Hub and Spoke								
Avalanche Private Subnet	Public	Full	Sharded	Hub and Spoke	Consensus	Messaging	Instant	Validator
Cosmos private appchain	Public	Full	Sharded	Hub and Spoke	Consensus	Messaging	Instant	Validator
Provenance zone	Public	Full	Sharded	Hub and Spoke	Consensus	Messaging	Instant	Validator
Public Permissioned - Public L2s/Rollups								
Arbitrum One	Public	Trailing	Transparent	Monolith	DIY	Full	Slow*	Everyone
Optimism	Public	Trailing	Transparent	Monolith	DIY	Full	Slow*	Everyone
Polygon zkEVM	Public	Trailing	Transparent	Monolith	DIY	Full	Slow	Everyone
Public Permissioned - Private L2s/Rollups								
Arbitrum Orbit AnyTrust	Public	Full	Sharded	Hub and Spoke	Consensus	Messaging	Slow	Validator
Private OP Stack Based	Public	Full	Sharded	Hub and Spoke	Consensus	Messaging	Slow	Validator
Polygon CDK Validium	Public	Full	Sharded	Hub and Spoke	DIY	Messaging	Slow	Validator
Public Permissioned - Other								
Canton Network	Public*	Full	Sub-transaction Privacy	Network of Networks	Consensus	Full	Instant	None
Private / Private Permissioned								
EEA (Besu, Quorum) Single Group	Private	Full	Transparent	Single Operator	Consensus	Single-Provider	Instant	None
EEA (Besu, Quorum) Group Per App	Private	Full	Sharded	Single Operator	Consensus	None	Instant	None
Fabric Single Channel	Private	Full	Transparent	Single Operator	Consensus	Single-Provider	Instant	None
Fabric Channel Per App	Private	Full	Sharded	Single Operator	Consensus	None	Instant	None
Corda Validating Notary	Private	Full	Transaction*	Single Operator	Interactive	Single-Provider	Instant	None
Corda Nonvalidating Notary	Private	Full	Transaction*	Single Operator	Interactive	Single-Provider	Instant	None
CometBFT/Tendermint /Cosmos SDK Private Network	Private	Full	Transparent	Single Operator	Consensus	Single-Provider	Instant	None
Daml/Canton Private Networks	Private	Full	Sub-transaction Privacy	Single Operator	Consensus	Single-Provider	Instant	None

*See detailed evaluation

Blockchain/Network Characterization

This document is about the properties and suitability of networks and their backing technology for use cases like tokenization in regulated finance. In that spirit, "network" here means the "venue" for that token, the place where it is made available to users and other applications. This evaluation is not going to be overly precise about what the boundary of a "network" is, but rather focuses on the implications for the token as a result of using a particular venue.

We'll use the term application provider here for an entity that wants to issue a token, or provide some service on a network. User will refer to an entity that wants to participate in that application at the blockchain level, for example by holding a token using their own cryptographic identity.

Network Access

This is the classic Public/Private dichotomy. Access and connecting here are to be understood at the blockchain level - a user reading the raw blockchain data, controlling their own cryptographic identity, and signing their own transactions.

Network Access Level	Definition
Private *Example: Hyperledger Fabric*	There is a governing body of the network that maintains a whitelist of other entities that can connect to and access the network. Accessing the network may require KYC activities with the governing body.
Public *Example: Ethereum*	Anyone can access and connect to the network, often anonymously and in a censorship proof way. The governing body of the network may blacklist identities that broke network rules but enforcement of blacklists is conducted by network participants.

Public vs Private is less a property of the technology than how it's deployed. For example, one might take Bitcoin Core and start a new network from a new genesis block, but use IP-level protections to only allow a select set of nodes to connect, thus making the network private.

From a regulatory and enterprise point of view, this characteristic is unimportant. The internet is a public network and yet it is perfectly suitable as an infrastructure to run regulated business.

From an enterprise security point of view, public and private networks are enforced by different mechanisms, yet both can be secured.

From a business perspective, using a public venue is advantageous. Private networks mean closed audiences of possible users and small venues for assets and services. Having to get users to join a new network to access an application adds a lot of friction to business development. And unless the application provider is also the governing body of the network, providing access to potential users isn't even under their control.

Independent Security

This describes the degree to which the application provider provides their own security by validating the transactions that pertain to their application and thus safekeep the records for which they have legal obligations.

Consensus Participation Level	Definition
None *Example: Hedera Hashgraph*	A limited set of entities perform all validation and extend the blockchain (e.g. PoA validators). The network's governing body or process controls who is in this group. Typically the application provider is not in this group.
Trailing Verification *Example: Ethereum*	The provider can independently verify the blockchain or at least that part of the blockchain that matters after the fact. This is usually done by connecting a "Full Node". But the blockchain is either extended by a closed group controlled by a governing body and the application provider is not in this group, or the consensus algorithm is of a nature where the application provider only rarely and randomly gets to participate in consensus. This level allows the app provider to detect foul play, but not to intervene.
Full *Example: Avalanche Private Subnet*	The application provider participates in the blockchain's consensus at least as it applies to their application's transactions. Thus they can independently ensure that no invalid transactions get appended.

Regulated Entities have legal obligations to safekeep financial records and be able to keep their services on those records available. That means they must validate transactions pertaining to those records, and ideally be able to stop any invalid transactions from ever being recorded.

Privacy/Read Permissioning

Privacy is permission to read data from the blockchain. We'll use ownership records as an example.

Privacy Level	Definition
Transparent *Example: Bitcoin*	Every transaction on the network is fully visible to every user of the network. At best you can obfuscate using pseudonymisation.
Sharded *Example: Cosmos Private Appchain*	Private subnets, rollups, or similar constructs allow for corners of the network that only some users can read from. But within such a shard or rollup, there's full transparency, meaning all users connected to a shard can read all transaction data within that shard.

Privacy Level	Definition
Transaction Level *Example: Corda*	Transactions are only visible to a small set of entities that have some sort of involvement in that transaction. This makes it possible to keep one user's transactions hidden from another user on the same network. But it can leak information between applications when they are composed. Eg in a DvP, the payment provider finds out about the delivery leg.
Sub-transaction Level *Example: Canton*	Privacy can be maintained within a single atomic transaction, meaning applications can be atomically composed without leaking sensitive data. For example, in a DvP, the payment provider learns only about the payment leg, not about the delivery leg.

For the most traditional financial instruments, privacy between users is a must. Thus transaction level privacy is needed at a minimum to enable tokenization. But if composing leaks data between applications, a network with sub-transaction privacy is needed to allow a network of applications to grow via the kind of smart contract composition that powers DeFi.

Infrastructure Independence

Infrastructure Independence is about how isolated the application provider is from network wide governance, availability, scalability, lifecycling or similar issues. Ie how much *infrastructure risk* they face.

Network Architecture	Infrastructure Independence Level	Definition
Decentralized Monolith *Example: Ethereum Mainnet, Polygon PoS Mainnet*	**Low**	The network is a shared resource as a whole. It acts like a decentralized mainframe where compute and network resources are competed for by users and applications. The rules of the network apply globally. The network clogs up or goes down globally. The software and protocol evolve globally. Throughput is a global limit.
Hub and Spoke *Example: Polkadot, Polygon CDK Validium*	**Medium**	Subnets or L2s provide some scalability, lifecycling and governance independence from the network as a whole. But typically if the MainNet is down, the subnets/L2s are down as well.

Network Architecture	Infrastructure Independence Level	Definition
Network of Networks *Example: Canton*	**High**	Like the internet, the network as a whole is a loose mesh of subnets that optionally connect to each other, but can run fully independently from each other. An application operator's core services are not affected by developments in the wider network. Only connectivity between applications running on different subnetworks may be dependent on infrastructure and services not controlled by the application operators themselves.
Single Operator *Example: Hyperledger Fabric*	**Full**	The application provider has full control over all involved infrastructure.

This is the flip side of the independent security property. A regulated financial institution must participate in the consensus related to the records they administer. But they also must not critically depend on other, potentially unknown entities, to also behave and allow them to advance with valid transactions. They (or trusted delegates like a cloud provider) need as much control over the infrastructure on which they run their core services as they can get, and they need as much isolation from other applications and developments on the network that they can get.

Full control is attractive purely from the control point of view, but it limits the network to a single application provider. The difference between “Network of Networks” and “Full Control” is that of the internet vs an intranet.

Independent Control

Independent control refers to the ability of the application provider to keep unilateral control over application access and permissions, to lifecycle application logic, and ultimately to respond to unforeseen circumstances like regulator interventions.

Means of Control	Independent Control Level	Definition
DIY *Example: Avalanche*	**Low**	The application operator has only those controls that they encode in the original smart contracts. Techniques and patterns like proxy contracts have to be used to allow changes to logic. Hard forks in the sense of abandoning smart contracts and creating amended copies are a last resort if unforeseen controls are needed.

Means of Control	Independent Control Level	Definition
Logical Control *Example: Solana*	**Medium**	The network's smart contract model has some notion of "ownership" of a smart contract which allows the owner(s) to upgrade the logic and thus assert some controls according to the rules of the network. However, the owner does not necessarily take part in consensus on actions on their contracts, so they are ultimately at the mercy of others to enforce the rules they have set and let them amend those rules.
Consensus Control *Example: Cosmos Private Appchain*	**High**	Each smart contract has a notion of who owns it. The owner(s) of a smart contract run the consensus protocol for transactions regarding their contracts. The owner(s) of a contract therefore enforce the rules they set themselves, and can intervene by jointly manipulating contract logic and data at will outside the protocol and network rules.
Interactive Control *Example: Corda*	**Medium***	Owners not only take part in consensus, but as part of consensus, the node involved in consensus can run non-deterministic interactive actions (eg calls to off-ledger data sources) to decide whether to approve a transaction or not. In other words, all parties to a contract must agree to changes to that contract, but the business logic upon which they agree or disagree need not be defined a priority.

Regulated financial enterprises need to be able to intervene in the system according to unforeseen circumstances, for example a sanction, a default, a court ruling, or simply a regulatory change. Whether the means available with Logical Control are sufficient is hard to predict. Nothing less than Consensus Control can prepare an application provider for all eventualities and thus be considered safe from a regulatory standpoint.

*Interactive control sounds like more control than Consensus control at the surface. But it has the downside that it diminishes the power of smart contracts. Anyone involved in consensus can independently decide to reject a transaction beyond the restrictions imposed by the smart contracts. The smart contracts are reduced to filtering out completely unacceptable transactions like double spends. It is no longer possible to encode positive rules like "any user can transfer the token to any other user".

Composability

Composability is the ability to transactionally (also called atomically) perform actions between two applications. This is where the definition of "network" above starts to matter. In the context of this evaluation, composability characterizes the ability for two applications that are deployed in the described way to compose with each other. The standard example is a DvP between two tokenized assets operated by two providers.

Composability Level	Definition
None *Example: Hyperledger Fabric with one channel per app*	There are no atomic transactions between two applications deployed in this way.
Single-Provider *Example: Hyperledger Fabric with one channel for all apps*	There are atomic transactions only between two applications run by the same application operator.
Messaging *Example: Polkadot*	Applications across the network can message each other or make non-transactional remote smart contract calls. But these calls are not guaranteed to execute atomically. Constructing DvPs requires Hashed Time Lock Contract style constructs or trusted intermediaries. There is no general smart contract composability.
Full *Example: Ethereum*	Any smart contract can call any other smart contract and have the actions take effect all at once or not at all.

DeFi on networks like Ethereum illustrates the power of Full composability. On the flip side, traditional financial systems illustrate how hard it is to achieve application interoperability with messaging. To build a new financial network consisting of multiple applications by multiple operators, network wide Transactional RPC is a must.

Finality

A transaction on a blockchain is considered final once a user that sees that transaction can be certain that that transaction is committed and will be seen by other users as committed forever.

Finality Level	Definition
Probabilistic *Example: Solana*	There is no point where a reversal of the transaction according to the consensus algorithm becomes impossible. The probability of a reversal merely gets lower - usually exponentially lower - as the chain extends beyond the transaction in question. This is typically the case for consensus algorithms that prioritize availability over consistency during partitions.

Finality Level	Definition
Slow Deterministic *Example: Ethereum*	Some networks are probabilistic in the short run, but have epochs or other checkpoints that effectively serve as markers of deterministic finality for everything that came before.
Instant Deterministic *Example: Hedera Hashgraph*	The consensus algorithm does not allow for reorderings or forks. As soon as a user sees a transaction as committed, it's final. This is typically the case for consensus algorithms that prioritize consistency over availability in case of partitions.

All three models are workable even for regulated financial enterprises, but probabilistic and slow deterministic finality are harder to integrate in off-ledger financial systems. The blockchain network is always optimistic, meaning transactions can build on each other immediately after commits. But off-ledger systems need to be pessimistic and deal with the possibility of reorderings, including ones that reverse a previously committed transaction. Often that means that a risk threshold needs to be chosen, and off ledger integrations only work against transactions considered final against that threshold. This slows systems down considerably, and adds considerable complexity. This complexity is even higher for workflows across two such probabilistic systems like intermediary free cross-network swaps using hashed time lock contracts, or similar.

Native Token Required

This is about whether some sort of native token needs to be transacted by the regulated enterprise or users to use the network.

Native Token Requirement Level	Definition
Everyone *Example: Aptos*	All users must transact with a native token to use the network.
Validators *Example: Avalanche Private Subnet*	At least all validators (ie those that participate in consensus) must handle a native token.
None *Example: Quorum*	There is no native token, or It is not necessary to handle it even to participate in the network even at the consensus layer.

Most regulated financial institutions cannot hold cryptocurrency and similar tokens on their balance sheet. As argued under independent security and independent control, they likely do need to participate in consensus. So only networks with no native token requirement are truly viable.

Blockchain Evaluation

This section looks at a large set of the most popular blockchain networks and technologies and evaluates them against the characteristics above. Many networks have a lot in common so the focus is always on what makes a particular network *different* if anything.

Public Permissionless

All public permissionless networks have a lot in common with each other and with the one that started it all: Bitcoin. As such, we'll look at the characteristics for Bitcoin in some detail, and then merely describe differences for the other ones.

Bitcoin

Network Access

Anyone can generate a Bitcoin address and interact with the network via wallet software and many publicly accessible full nodes. Bitcoin is the archetype of a public network.

Independent Security

Not only can anyone use Bitcoin, anyone can spin up a full node (usually Bitcoin Core) and start validating the blockchain independently as well as extending the blockchain through Proof of Work (PoW) mining. It remains one of the most decentralized networks in existence. However, the chances of an application provider mining the block containing their transactions are diminishingly small. In almost all cases they can only validate after the fact, so validation is Trailing.

Privacy

The entire Bitcoin blockchain is replicated in full to every full node, and thus potentially to any user. It is entirely transparent. Every user can read every holding and transaction ever.

Infrastructure Independence

The Bitcoin blockchain is a good example of a monolith. It has static block size and self-calibrates to fairly constant block times, which means throughput is globally constant. All blockchain resources are thus a shared good that users bid on using gas fees. A user or app provider deploying on Bitcoin has no infrastructure independence at all.

Independent Control

Bitcoin is usually not considered a smart contract platform, but it is in fact possible to build simple applications on Bitcoin like token issuances. Such applications use Bitcoin Script as a form of smart contract. As a user of the network it is very difficult to ensure that one is involved in consensus for one's own transactions. Control over applications is instead maintained at a logical level by building in "escape hatches", i.e., app providers hard-code signing keys that can manipulate application state and script at will. It's "Do it yourself" logical control.

Composability

Two tokens on the Bitcoin network can be composed into a DvP pretty easily. This is a property all monolithic blockchains share: Full smart contract composability.

Finality

Bitcoin is the original Proof of Work consensus algorithm which is famously probabilistic. The general recommendation is to wait for a block depth of about 6 (meaning ~60 minutes) before considering a transaction final. Some recommend as much as 13 blocks. At that point, the argument goes, probabilities get so low that you can stop caring, but that does make some assumptions on distribution of compute power and the financial incentives involved.

Native Token Required

Every transaction requires the payment of transaction fees in BTC so everyone involved must handle cryptocurrency.

Ethereum

Finality

Ethereum works almost exactly like Bitcoin with respect to the characterization here, except that it switched to a "Proof of Stake" (PoS) consensus algorithm in 2023. It now has 12 minute long epochs after which a transaction can be considered fully final. So it's a good example of "slow deterministic" finality.

BNB

Independent Security

BNB is an Ethereum clone running on the Clique consensus algorithm, which is in the family of Proof of Authority (PoA) algorithms with probabilistic finality. The pool of entities that validate is size limited and controlled by a centralized governing body. They generally assign 40 of the 56 available slots to the top stakers, but this is not encoded in the protocol so there is no real way for application providers to participate in consensus at all. But they can still validate in a Trailing fashion by running full Ethereum nodes.

Solana

Independent Control

Solana has a notion of a smart contract owner that is by default the address that deployed the smart contract. That owner can upgrade the smart contract at will so there is some logical control out of the box.

Ripple/XRP

Independent Security

The XRP network is nominally open to anyone for participation in consensus. However, each node keeps a "Unique Node List" (UNL) which lists the other nodes whose votes are taken into account for consensus. The XRP Ledger Foundation and Ripple distribute the "default UNL" (dUNL) so effectively they control who can participate in practice. Thus as for BNB, there is no consensus participation at all, but providers can still do Trailing validation.

Cardano

Cardano is not notably different from Ethereum from the point of view of this evaluation.

Avalanche (C-Chain)

Finality

Avalanche claims that their Proof of Stake consensus has instant finality. Otherwise the C-Chain's functioning is identical to Ethereum from the point of view of this evaluation.

Polygon PoS

Independent Security

Polygon PoS is an early Ethereum clone with its own "Proof of Stake" algorithm. However, staking is not enough to become a validator that participates in consensus. The list of validators has a size limit and is controlled by Polygon. Polygon PoS, like BNB and XRP, therefore only offers Trailing validation with no consensus participation.

Finality

Just for clarity as the above comparison is with Ethereum, Polygon PoS does not have slow deterministic finality, but classic probabilistic finality.

Internet Computer

Network Access

Network access is entirely through API gateways run by the group that runs the network. They allow public access, but the network is far from censorship proof which is a usual argument for public networks. Since the group does allow anonymous public access, this evaluation still lists the Internet Computer as Public.

Decentralization

The Internet Computer is run by an invite-only group of data centers. Users have no direct access to the smart contract or blockchain data so can't independently verify. Thus unusually for a public chain, decentralization is only at "Consortium" level.

Privacy

Thanks to the lack of decentralization, and user lack of access to smart contracts and blockchain data, there is privacy between users and applications. However the entities running the network have full transparency. While the presence of some privacy may be a positive for a regulated enterprise, the asymmetry in information access is a big negative. Effectively every datacenter processing sensitive transaction data would have to be trusted to the degree of a cloud data center that a traditional application runs through.

Infrastructure Isolation

The Internet Computer is internally highly sharded and parallelized. The technology promises to make it possible to run fairly isolated subnets. What's possible in practice today is unclear. The reliance of centralized API gateways means there is significant reliance on third parties for availability and access.

Independent Control

Like Solana, smart contracts (called Canisters) have a notion of ownership and upgrading. There is inbuilt logical control.

Composability

Internet computer smart contracts (Canisters) do not have traditional transactional composability at all. Rather, in each "block" Canisters consume and emit messages to and from other Canisters. There are strong guarantees on the messages including exactly once delivery, but there is no atomicity.

Stellar

Independent Controls

Like Solana and the Internet Computer, Stellar's WASM based "Soroban" smart contracts have an inbuilt mechanism for code upgrades, thus it has "Logical" level controls.

Polkadot

Infrastructure Independence

Polkadot is one of the original "Hub and Spoke" models. A set of PoS validators runs a "Relay Chain". All business logic happens on "Parachains" that are isolated from each other in terms of blockspace and thus throughput. However, the relay validators validate all parachains so the infrastructure independence here is only with respect to scalability and block space consumption.

Privacy

Just for clarity and to contrast with other Hub and Spoke models, since all parachains are validated by the relay chain validators, and Polkadot is fully decentralized (any user can participate in consensus), all parachains are transparent.

Independent Controls

Polkadot's consensus has two steps. Validators of a given parachain first validate transactions and then submit a "Proof of Validation" to the relay chain. The relay chain validators then re-validate. Since an application provider can thus participate in consensus for every transaction on their application, it could be argued that they thus have "Consensus" level controls. However, this only provides an emergency stop button. It does not allow for ad-hoc overruling of the network rules in the case of completely unforeseen circumstances. The anonymous and disinterested relay chain validators would detect and block any such operation. Thus a hard fork to a new smart contract or parachain would be the only recourse. Therefore this evaluation classifies Polkadot as only providing "Logical" controls.

Composability

Between parachains, Polkadot offers messaging, but not atomic smart contract calls.

Hedera Hashgraph

Independent Security

Hedera Hashgraph currently does not allow anyone to run a node. Validation is delegated to a closed group so there is no independent security here. Hedera is planning decentralization of the MainNet in the future.

Finality

Hedera claims fast deterministic finality.

Aptos

Finality

Like Avalanche (C-Chain), Aptos claims very fast deterministic finality.

Independent Controls

Aptos uses the Move smart contract language originally designed for Facebook's Libra system. It has this in common with the Sui blockchain. The current state of independent controls of assets and smart contracts in Move are not entirely clear to the author of this document as the last full analysis of the language was performed in 2019. At the time, upgrading features were discussed as "future work". Being a Monolith, the best case scenario is Logical control so that's what this evaluation assumes.

Algorand

Finality

Algorand claims fast deterministic finality.

Independent Controls

Algorand has its own smart contract language and runtime (AVM) that the author is not familiar with. As with Aptos and the Move language, Algorand is a Monolith, the best case scenario is Logical control, and that's what this evaluation assumes.

Public Permissioned - Hub and Spoke

Avalanche Private Subnet
Avalanche private subnets will serve as the reference Public Permissioned Hub and Spoke networks like Bitcoin served for Public Permissionless networks.

Network Access
As the Infrastructure Control section below discusses, every Avalanche private subnet is firmly part of the Avalanche network, which is public.

Decentralization
Each subnet runs its own Avalanche consensus with a configurable set of validators. Effectively it's Avalanche "Proof of Authority". Thus the important user - the application provider - can participate in consensus, giving this deployment model User Consensus level.

Privacy
Subnets can be properly private in the sense that only authorized users can read from the blockchain. However the subnet is fully transparent for those that can read the blockchain. Thus this deployment model gives sharded privacy.

Infrastructure Isolation
Every subnet validator also must act as a primary chain validator. Subnet configuration is kept on the primary chain. This model gives scalability and resource isolation, but keeps dependency on governance and availability.

Independent Control
The subnet owners run consensus themselves and can thus exert a great degree of control down to the consensus level.

Composability
As with Polkadot and ICP, two applications on two different subnets only connect via non-transactional messaging.

Finality
As with Avalanche (C-Chain), the Avalanche consensus used on subchains claims instant finality.

Native Token Required
To validate an Avalanche subnet, the validators also need to be P-Chain validators. That requires staking of AVAX and therefore handling of cryptocurrency.

Cosmos Private Appchain
Cosmos may be the hardest network to evaluate on this entire list, for the boundaries of the Cosmos Network are unclear. One viewpoint is to say that the Cosmos Network is the collection of all blockchains built using Cosmos SDK. But a Cosmos SDK blockchain is by default a private PoA network built on CometBFT (formerly Tendermint). This setup is evaluated further below.

Connections between such blockchains could theoretically be made point to point using the Inter Blockchain Communication (IBC) protocol to send messages from one chain to the other and build bridges. So another viewpoint is that it's a "network of networks".

But in practice, the Cosmos network is firmly centered on the Cosmos Hub and the Cosmos Hub takes care of a lot of cross-subnet communication. So this evaluation looks at a deployment where the application provider runs their own (possibly private) subnet, but is also connected into the Cosmos Hub. With this model in mind, Cosmos Subnets are equivalent to Avalanche Subnets in the eyes of this evaluation.

Provenance Zone
Provenance is an extension of Cosmos SDK and arguably part of the Cosmos Network. The extensions are aimed at tokenization fit for regulated enterprise. Zones correspond to Cosmos app-chains, and offer some higher-level functions across zones built on IBC. However, at the level of this evaluation, its properties are exactly the same as those of Cosmos Private Appchains.

Public Permissioned - Public L2s/Rollups

Public L2s and rollups are scalability solutions for the big public networks, most prominently Ethereum. There are two flavors of such rollups: Optimistic and Zero Knowledge, which compete first and foremost on their security properties which come into this evaluation only in the Finality characteristic. Any differentiation in the space is on the nuanced degree of decentralization on the range from Consortium to User Consensus (which nobody has reached yet). L2Beat tracks this in some detail.

The astute reader will notice that these networks share more with Public Permissionless than Public Permissioned. They are only categorized under Public Permissioned here because they are sometimes floated as a "solution" for regulated enterprise in line with the Public Permissionless category.

Arbitrum One
Arbitrum One is the leader in the public L2 space according to "Total Value Locked" (TVL) so will serve as the reference for this category. It is an optimistic rollup, but as mentioned above, that's for interest only and doesn't play into this analysis.

Network Access
Arbitrum one is fully public.

Privacy
The rollup is replicated in full to every node, and thus potentially to any user. It is entirely transparent. Every user can read every holding and transaction ever.

Infrastructure Independence
As a user of Arbitrum One, one has some resource independence from Ethereum MainNet. That's what makes it an Ethereum scaling solution. But users have full dependence on Arbitrum One which acts as a new monolith.

Independent Control
Arbitrum One uses Solidity smart contracts so compared to Ethereum all that changes is that Ethereum validators are replaced by Arbitrum validators. Arguably, it could even be said that dependency is now on both sets of validators.

Composability
Two tokens on Arbitrum One can transact with each other just like they can on Ethereum MainNet. But there is no transactional interoperability between Arbitrum One and Ethereum MainNet, only messaging and custodial asset bridges run by the validators of Arbitrum One.

Finality
Finality for optimistic rollups is a nuanced topic. In theory, a transaction could be reversed until its dispute time delay (DTD) is over. That's 7 days. So while it's still "Slow Deterministic", the slow is at a different level to Ethereum MainNet.

Native Token Required
Every transaction requires the payment of transaction fees in (bridged) ETH so everyone involved must handle cryptocurrency.

Optimism
Equivalent to Arbitrum for the purposes of this evaluation.

Polygon zkEVM
Finality
Being a ZK-Rollup, finality doesn't have to wait for a dispute timeout on Polygon zkEVM. A transaction can be considered final once the proofs are final on Ethereum MainNet, which means finality is borrowed almost one to one from the MainNet.

Public Permissioned - Private L2s/Rollups

The idea behind private L2s/Rollups is to use the same stack as the public rollups, but to deploy it in such a way that only a select group of users can access the rollup.

The three examples evaluated here correspond to the three public versions.

Arbitrum One -> Arbitrum Orbit AnyTrust*
Private OP Stack Based -> Optimism
Polygon zkEVM -> Polygon CDK Validium

* Arbitrum would make the correspondence not with Arbitrum One, but Arbitrum Nova. The difference is only about whether data availability is delegated to the Ethereum MainNet or to the rollup validators. This makes no difference in this evaluation.

Arbitrum Orbit AnyTrust
Arbitrum Orbit AnyTrust will serve as the reference for this category.

Network Access
As considered as part of a wider network consisting of Ethereum MainNet and all rollups on top, network access is still Public.

Independent Security
In this setup the app provider can validate their own rollup offering Full independent security.

Privacy
The rollup is replicated in full to every participant in the rollup, and thus potentially to any user of the application. But it is private against other Ethereum users. Thus Privacy is Sharded.

Infrastructure Independence
There is some limited availability and governance dependency on Ethereum MainNet.

Independent Control
Since the validators of the rollup can also control the fraud proofs, too, they can arguably override everything if need be. Thus the level is Consensus controls.

Composability
Between private rollups, there is only non transactional messaging.

Finality
The validators can control dispute time delay (DTD) and possibly fraud proofs in general so in practice finality is finality of the rollup headers on the MainNet. This means Slow Deterministic finality is borrowed from Ethereum MainNet.
Native Token Required
The Validators have to write rollups to the MainNet so must handle ETH.

Private OP Stack Based
Equivalent to Arbitrum Orbit AnyTrust in this evaluation.

Polygon CDK Validium
Independent Controls
Unlike with optimistic rollups, the validators of a zkRollup cannot easily “override” the blockchain protocol. Thus independent controls are reduced to an emergency brake similar to Polkadot. But Polygon CDK is EVM based, which means no logical code ownership, reducing independent control to only DIY level.

Public Permissioned - Other

Canton Network

Network Access

The Canton Network is designed to be a public network, but is still undergoing testing in an invite only mode. Members of the general public can already deploy private Canton subsets (see Daml/Canton Private Networks) that will be able to connect into the wider network following a protocol upgrade.

Independent Security

The Canton protocol uses a unique per transaction proof of authority consensus model, which means exactly the stakeholders of a transaction - both application providers and users - participate in consensus. It therefore offers Full independent security to the application provider.

Privacy

Daml smart contracts on the Canton protocol provide subtransaction privacy. It is possible to execute an atomic DvP where the asset provider sees only the delivery leg and the payment provider only the payment leg. And it is possible to completely hide user holdings from each other. Canton Network therefore reaches sub-transaction privacy level.

Infrastructure Independence

Canton has a two-layer design. As in most decentralized blockchains, sophisticated users of Canton (which includes application providers) run their own full node, called a participant node. Participant nodes run their consensus algorithm through an infrastructure layer called a “synchronization domain” or just “synchronizer”. A participant node can connect to many such synchronizers and choose a suitable one on a transaction by transaction basis, reassigning contracts from one total order to another as needed.

This allows an application provider to run transactions pertaining only to their application through a synchronizer that they fully control. The infrastructure in play for a transaction includes the user’s participant node, the application provider's participant node (or nodes if the provider is a collective), the application provider's synchronizer, and the participant nodes of any additional validators participating in consensus for that transaction.

Only for transactions spanning multiple application providers is there some dependency on a shared synchronizer. This matches the definition of “Network of Networks”.

Independent Control

As already laid out in Independent Security above, stakeholders of a transaction participate in transactions. Therefore, Canton Network meets the consensus control level.

Composability

All Daml smart contracts on the whole network of networks are atomically composable with one another as long as there is a shared synchronizer that all consensus participants of the transaction can use to run their consensus.

Finality

Canton has instant deterministic finality.

Native Token Required

The Canton protocol does not require the use of a native token and does not have an inbuilt token. For full clarity, there are plans as part of Canton Network to run a decentralized synchronizer with an associated decentralized payment application which can be used to pay for additional bandwidth on the synchronizer. But use of that synchronizer as well as the payment utility when using the synchronizer are both optional. There is no staking, or pay-as-you-go gas. So in all ways, use of an inbuilt or native is not required to participate in the network in full.

Private Permissioned

EEA (Besu, Quorum) Single Group
Besu without privacy groups is one of the most widely used deployment models for private blockchain networks so will serve as the reference for Private Permissioned Networks.

It's technically possible to run a public Besu network as EEA (Enterpise Ethereum Alliance) as an extension of the standard Ethereum node specifications. But the deployment model under discussion here is a typical one where access is controlled at the network level (i.e. closed access to the devp2p layer), and consensus is Proof of Authority amongst a closed set of nodes and entities. This evaluation also assumes the typical scenario where the application provider is a single entity or small group of entities that fully operates the blockchain.

Network Access
By definition, network access is private.

Independent Security
The application provider fully runs or at least participates in the PoA consensus so this qualifies as Full independent security.

Privacy
Within a privacy group, the blockchain is fully transparent to all nodes and users. So a typical single group setup is fully transparent.

Infrastructure Independence
The application provider as the operator of the network has full infrastructure independence almost by definition.

Independent Control
The application provider as the operator of the network has full consensus control.

Composability
Within the private network there is full smart contract composability. But since the assumption is that such a network is provider-operated, this only qualifies for Single-Provider composability.

Finality
With the right PoA algorithm (e.g. QBFT), there is instant finality.

Native Token Required
It's possible to set free gas in which case there is no need to handle the inbuilt token to pay for gas.

EEA (Besu, Quorum) Group Per App
EEA Ethereum nodes allow for "Privacy Groups" through the private transaction processor Tessera. A typical use for that is to isolate users of different applications from each other to get some privacy.

Privacy
Each privacy group is fully transparent to all its users. So this setup reaches Sharded privacy level.

Composability
There are no transactions across groups. So composability disappears completely. The only capability is to read data from public Ethereum as part of a private transaction.

Fabric Single Channel
"Channel" is Fabric's term for EEA's privacy groups. This setup is almost equivalent to EEA Single Group.

Independent Control
Just a small note here that Fabric's consensus could be considered Interactive by virtue of chaincode allowing I/O, meaning non-deterministic validation. Each "Endorser" (Full Node) that's involved in consensus could therefore make their own decisions whether a transaction is valid or not. But this is a side-effect of the system's design, not an intended use so in this evaluation Fabric is listed as Consensus control.

Fabric Channel Per App
This setup is fully equivalent to EEA Group Per App.

Corda Validating Notary
Privacy
In this evaluation Corda is evaluated as having Transaction level privacy. In fact, it's somewhat weaker than that due to an important mechanism called "backchains", but that's a nuance beyond the scope of this document.
With a validating notary, the notary, which is part of the network infrastructure, sees all transactions. Since this evaluation assumes the network is operated by the application provider, that's not a major problem.

Independent Control
Corda is designed for Interactive control. Each user runs their own "Flows" which independently decide whether to sign a transaction or not.

Corda Non-Validating Notary
Privacy
Compared to a validating notary setup, the privacy caveat against the notary/app provider goes away. The tradeoff is a well known availability attack called "Denial of State" which forks the ledger. Thus this is only feasible in a network where all users are trusted.

CometBFT/Tendermint/Cosmos SDK Private Network
Without connecting to Cosmos Hub, a private network built on the Cosmos stack is equivalent to a single group EEA setup.

Daml/Canton Private Networks
A private Daml/Canton network is one with a single synchronizer operated by the application provider.

Privacy
Private Canton deployments offer the same subtransaction privacy as the Canton Network.